

©Copyright

Matthew S. Patterson

Algebro-geometric algorithms for integrable systems

Matthew S. Patterson

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2007

Program Authorized to Offer Degree: Applied Mathematics

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Matthew S. Patterson

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

Bernard Deconinck

Reading Committee:

Bernard Deconinck

Robert E. O'Malley

J. Nathan Kutz

Date:

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, or to the author.

Signature_____

Date_____

University of Washington

Abstract

Algebro-geometric algorithms for integrable systems

Matthew S. Patterson

Chair of the Supervisory Committee:
Professor Bernard Deconinck
Applied Mathematics

Integrable systems such as the nonlinear Schrödinger, Korteweg-deVries and Kadomtsev-Petviashvili equations are used to model diverse physical phenomena. These and all other integrable systems admit families of multi-phase solutions given in terms of Riemann θ -functions parameterized by compact, finite-genus Riemann surfaces. Calculating the initial phase information such solutions involves computing the Abel map and the vector of Riemann constants associated with a genus g Riemann surface Γ .

The Abel map is a vector with g components, each of which is an integral between places on a Riemann surface. This map is well defined up to integrals of closed paths.

The vector of Riemann constants of a Riemann surface Γ is the vector offset between the image, under the Abel map, of the poles and zeroes of any meromorphic differential and the zero set of the Riemann θ -function associated with Γ .

Here algorithms are presented to compute the Abel map and the vector of Riemann constants. For these algorithms Riemann surfaces are defined by irreducible polynomials in two complex variables. The algorithms have been implemented as procedures in the computed algebra system Maple, and these implementations are demonstrated.

TABLE OF CONTENTS

	Page
List of Figures	iv
List of Tables	vi
Chapter 1: Introduction	1
1.1 Introduction to the relevant physics	1
1.2 The nonlinear Schrödinger equation and deep water	4
1.3 The Korteweg-deVries equation and one-dimensional waves on shallow water	7
1.4 The Kadomtsev-Petviashvili equation and two-dimensional waves on shallow water	8
1.5 Remarks on the Maple implementations	11
Chapter 2: Preliminary examples of genus zero and genus one Riemann surfaces	16
2.1 Genus zero: the Riemann sphere	16
2.2 Genus one: the first non-trivial example	21
Chapter 3: Introduction to Riemann surfaces	24
3.1 Points and places	24
3.2 Newton's Theorem and convergence of Puiseux expansions	27
3.3 Branches	33
3.4 The homology of a Riemann surface	35
3.5 The Riemann surface cut for the purpose of integration	36
3.6 Meromorphic functions and differentials	42
3.7 The cohomology of a Riemann surface	43
3.8 Riemann matrices and the associated Jacobian	47
3.9 Definition and properties of divisors	50
3.10 The Riemann-Roch Theorem	56
3.11 Special divisors	61
3.12 The Abel map	61
3.13 The Riemann θ -function	64

3.14	The Jacobi inversion problem	67
3.15	The vector of Riemann constants	73
Chapter 4:	Multi-phase solution formulae	77
4.1	The nonlinear Schrödinger equation	77
4.2	The Korteweg-deVries equation	79
4.3	The Kadomtsev-Petviashvili equation	79
Chapter 5:	Computing the Abel map	81
5.1	Initial considerations	81
5.2	Computing discriminant points	81
5.3	Computing paths on the Riemann sphere	84
5.4	Lifting paths to the Riemann surface	95
5.5	Remarks on integration	98
Chapter 6:	Computing the vector of Riemann constants	99
6.1	Initial considerations	99
6.2	The vector of Riemann constants of the left-place	101
6.3	Computing a canonical divisor	104
6.4	Approximating the θ -function	111
6.5	Computing the correct half-lattice vector	115
6.6	Filtering candidate half-lattice vectors	122
Chapter 7:	The implementation of the Abel map: computation and verification . .	127
7.1	Syntax of the implementation	127
7.2	Demonstrating convergence	128
7.3	A hyperelliptic example	130
7.4	Computing the Abel map of divisors of meromorphic functions	133
7.5	Computing the Abel map of divisors of meromorphic differentials	135
Chapter 8:	The implementation of the vector of Riemann constants	140
8.1	Syntax of the vector of Riemann constants	140
8.2	Example computation of the half-lattice vector	140
Chapter 9:	Applications of the Abel map	155
9.1	Integration of algebraic functions	155
9.2	Integral-linear dependence of divisors	165

Bibliography	168
Appendix A: Proofs and details	171
A.1 Details of Proposition 3.14.2	171
A.2 Proof of Proposition 3.15.2	174
A.3 Dimension of Riemann-Roch spaces	178
Appendix B: Maple Syntax	181
B.1 Preliminary matters	181
B.2 The <code>algcurves</code> package	185

LIST OF FIGURES

Figure Number	Page
1.1 Diagram of free-surface water-wave problem	2
1.2 Envelope soliton experiment	6
1.3 John Scott Russel’s “Great Wave of Translation”	8
1.4 One-dimensional waves described by the Kadomtsev-Petviashvili equation	9
1.5 Waves interacting to form hexagons	10
1.6 Hexagonal Waves described by the Kadomtsev-Petviashvili equation	11
1.7 Asymetric hexagonal waves	12
1.8 Hexagonal wave after Hurricane Grace	13
2.1 Graph of $y^2 = x$	17
2.2 Cutting and pasting a genus zero surface: one	18
2.3 Cutting and pasting a genus zero surface: two	19
2.4 Cutting and pasting a genus zero surface: three	20
2.5 Cutting and pasting a genus one surface: one	21
2.6 Cutting and pasting a genus one surface: two	22
2.7 Cutting and pasting a genus one surface: three	23
2.8 Cutting and pasting a genus one surface: four	23
3.1 Idealization of Riemann surface	29
3.2 Quintic example	31
3.3 Homology of Riemann surface	36
3.4 The cut surface: one	37
3.5 The cut surface: two	38
3.6 The cut surface: three	39
3.7 The cut surface: four	40
3.8 The cut surface: five	40
3.9 The cut surface: six	41
3.10 The cut surface: seven	42
3.11 The cut surface: eight	43
3.12 Example lattice	44

3.13	Graph of the θ -function	65
5.1	Integration paths in the complex x -plane	86
5.2	Integration path to a point away from discriminant points	89
5.3	Integration path to a point close to a discriminant point	90
7.1	Convergence of the Abel map	129
7.2	Cauchy convergence of the Abel map	133

LIST OF TABLES

Table Number	Page
8.1 Using the theta divisor to verify correct choice of the half-lattice vector . . .	152

ACKNOWLEDGMENTS

I wish to thank Sharon, David and Keegan Patterson and Lara Kidoguchi for all their love, help and support. Without them this would not have been possible. Thank you all very much.

I also wish to thank and acknowledge my academic advisor and friend Bernard Deconinck. I could not have had a better advisor.

I acknowledge Mark van-Hoeij for a number of fruitful conversations, and some very good ideas.

The University of Washington Applied Mathematics Department has been like a second family to me for the past years. I thank my fellow students, the faculty and staff for the teaching, camaraderie, assistance, funding, and things too numerous to list.

Over the years this work has been funded by National Science Foundation VIGRE Grants and the National Science Foundation UW GK-12 Grant.

Chapter 1

INTRODUCTION

The algorithms presented here to compute the Abel map and the vector of Riemann constants are the next steps in an ongoing research program to make effective the calculus on Riemann surfaces represented by plane algebraic curves. Here “effective” means that algorithms are devised and implemented in the form of black-box programs so that different relevant quantities associated with Riemann surfaces may be computed symbolically or numerically in an efficient way. The long-term goal is the computation of the so-called finite-genus solutions of integrable differential (or difference) equations. Among the most famous of such systems are the Korteweg-deVries (KdV) equation and the nonlinear Schrödinger (NLS) equation. These have been widely used in the last forty years to describe various physical phenomena, ranging from water waves, nonlinear optics and plasma physics to biological processes and cellular automata [4, 1, 24].

1.1 Introduction to the relevant physics

The nonlinear Schrödinger and Korteweg-deVries equations arise in a number of applications, and the tools presented here are generic to the quasiperiodic solutions of these and other integrable systems. My primary motivation, however, is the study of nonlinear water waves. The starting point for this study is the reduction of the Navier-Stokes equations known as the free-surface water-wave problem. An outline of this problem follows but please see, for instance [30], for more details. Suppose

$$\phi = \phi(x, y, z, t)$$

is a velocity potential and

$$\eta = \eta(x, y, t)$$

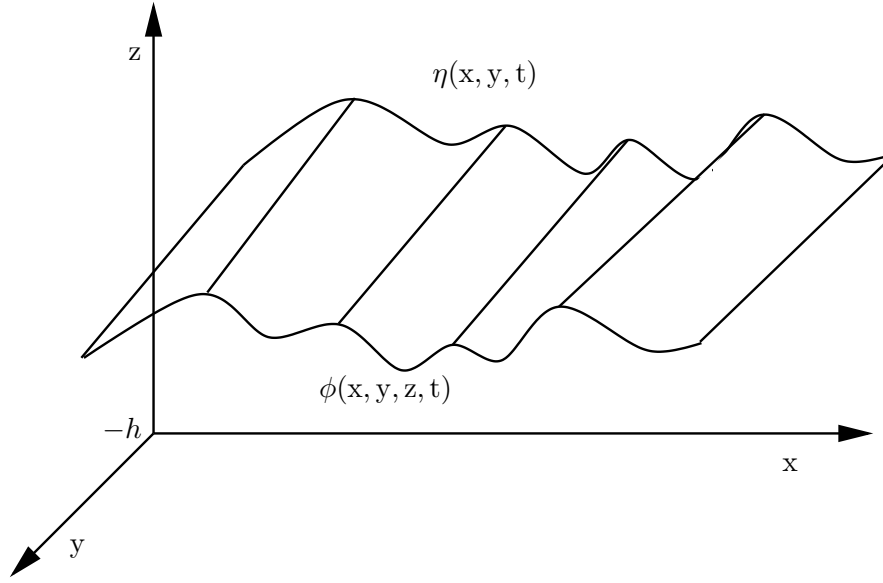


Figure 1.1: A diagram illustrating the coordinates and physical quantities in the free-surface water-wave problem.

is the elevation of the free surface. Then the evolution of ϕ and η is governed by the coupled system of partial differential equations (see Figure 1.1 for a schematic diagram of the coordinates and physical quantities):

$$\nabla^2 \phi = 0, \quad -h \leq z \leq \eta, \quad (1.1)$$

$$\phi_z = 0, \quad z = -h, \quad (1.2)$$

$$\eta_t + \phi_x \eta_x + \phi_y \eta_y - \phi_z = 0, \quad z = \eta, \quad (1.3)$$

$$\phi_t + g\eta + \frac{1}{2}(\phi_x^2 + \phi_y^2 + \phi_z^2) - T \frac{\eta_{xx} + \eta_{yy}}{(1 + \eta_x^2 + \eta_y^2)^{\frac{3}{2}}} = 0, \quad z = \eta. \quad (1.4)$$

The following notes hold for (1.1) through (1.4), collectively called the free-surface water-wave problem.

- Equation 1.1 states that the divergence of the vector velocity inside the fluid is zero. More physically this says that the fluid has constant density. That the velocity may be written as a potential already assumes that the fluid is inviscid and irrotational.

- Equation 1.2 is a boundary condition, and simply states that there is no flow of water through the bottom of the domain. Assuming h is a constant, as is done here, is equivalent to assuming the bottom of the domain is flat.
- Equation 1.3 is known as the *kinematic boundary condition*, and implies that a fluid parcel that starts on the surface stays on the surface.
- Equation 1.4 is known as the *dynamic boundary condition*, and describes the force balance at the free surface. Here g is acceleration due to gravity and T is the coefficient of surface tension. Waves where the restoring force is primarily gravity are referred to as gravity waves.
- Throughout the statement of the free-surface water-wave problem, subscripts denote partial differentiation and I have typeset the physical variables as

$$x, y, z, t$$

to distinguish them from the non-physical variables already introduced, as well as those to follow.

Remark: I use the terminology n -dimensional water-waves to refer to waves in which the variation of the surface elevation occurs in exactly n spatial dimensions. Thus $\eta(x, t)$ and $\eta(x, y, t)$ are the general form of the surface elevation of 1- and 2-dimensional waves respectively.

Further assumptions and restrictions lead from the free-surface water-wave problem to three integrable nonlinear equations. I introduce these equations in order to demonstrate the appearance of the Abel map and vector of Riemann constants in three different settings. The NLS equation arises for 1-dimensional gravity waves on deep water, the KdV equations arises for 1-dimensional gravity waves on shallow water. A generalization of the KdV equation to two dimensions, the Kadomtsev-Petviashvili (KP) equation, is the third and final system of gravity water-waves I will introduce. The following discussions are culled from [4] and [30]: please see these references for more detail.

1.2 The nonlinear Schrödinger equation and deep water

Suppose the free surface

$$\eta = \eta(x, y, t)$$

is 2-dimensional. If the wave amplitude is small compared to the water depth, then the evolution of slowly modulated, nearly 1-dimensional quasi-monochromatic waves is described by the complex wave function

$$\psi = \psi(x, y, t),$$

which in turn is approximately governed, for a certain choice of physical parameters, by the conditions imposed on the real and imaginary parts of the function ψ by the equation

$$i\psi_t = \psi_{xx} - \psi_{yy} + 2|\psi|^2\psi. \quad (1.5)$$

The solution for the scaled elevation η of the free surface around the mean is then

$$\eta(x, y, t) = \Re(\psi u) + \text{higher order terms} \quad (1.6)$$

$$u = a_0 e^{ik_0 x - i\sqrt{k_0} t}, \quad (1.7)$$

where \Re denotes real part, k_0 is the wave number of the assumed underlying carrier wave u and the amplitude a_0 is a constant. Note that the positive x -direction has been chosen as the direction of propagation. The higher order terms in (1.6) may be found, for instance, in the appendix of [29]. Assume the envelope varies in only one direction, although not necessarily the positive x -direction. Then the envelope ψ is given by

$$\psi = \psi(\xi, t),$$

where ξ is a traveling-wave coordinate specified by the direction vector $\boldsymbol{\kappa}_\psi$ through

$$\xi = \boldsymbol{\kappa}_\psi \begin{pmatrix} x \\ y \end{pmatrix},$$

and (1.5) reduces to the 1-dimensional NLS equation, which, as only the direction of $\boldsymbol{\kappa}_\psi$ is relevant, may be written as

$$i\psi_t = \psi_{\xi\xi} + 2|\psi|^2\psi. \quad (1.8)$$

If $\boldsymbol{\kappa} = c\mathbf{x}$ for some $c \in \mathbb{R}$, then the resulting free surface η varies in only one spatial direction, that is, the carrier wave and the envelope evolution are co-linear. If they are not co-linear then the resulting free surface is two-dimensional. That is, the carrier wave only varies in the x -dimension, but the evolution of the envelope has a component in the y -direction.

The assumptions made while obtaining the NLS equation for the evolution of the slowly modulated envelope of a carrier wave are given in a bit more detail in the itemized list below.

1. **Quasi-monochromatic:** The wave field is quasi-monochromatic if the power spectrum in the direction of propagation is sharply peaked about a central wavenumber say

$$k = k_x + k_y \quad (1.9)$$

2. **Small amplitude:** The water depth is considered large compared to the wave amplitude if

$$\epsilon = a\|k\| \ll 1,$$

where a is the characteristic amplitude of the wave field [4].

3. **Slowly modulated:** The envelope is considered to modulate slowly if

$$\frac{\delta k}{\|k\|} \ll 1,$$

where δk is the characteristic variation of the wavenumber k .

4. **Nearly 1-dimensional:** The wave field is nearly 1-dimensional if

$$\left\| \frac{k_y}{k} \right\| \ll 1$$

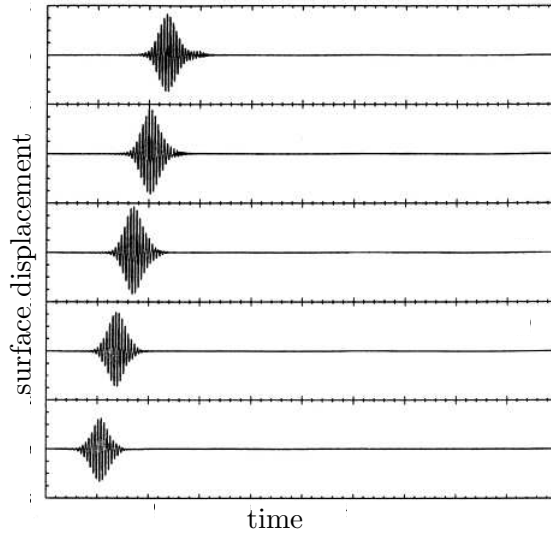


Figure 1.2: Experimental data generated using the NLS equation as a model for deep-water gravity waves. An envelope soliton is generated, and the surface displacement is measured at five distances down the tank. The fast oscillation is the carrier wave, and the overall “lump” shape is the slowly modulating envelope. The envelope is (approximately) a sech solution of the NLS equation. This plot from [2] was generated from data taken at the William G. Pritchard Fluid Mechanics Laboratory at The Pennsylvania State University.

5. **Balance of effects:** Items 1 through 4 balance:

$$\frac{\delta k}{\|k\|} = \mathcal{O}(\epsilon) = \left\| \frac{k_y}{k} \right\|$$

6. **Deep water:** Finally, the water is considered deep if

$$(k_x h)^2 \gg \epsilon.$$

While these assumptions may seem restrictive, much progress has been made using the NLS equation as a model [29, 3]. In particular the long-term evolution of so-called *envelope solitons* is well predicted by the NLS equation [4]. Figure 1.2 shows the experimental evolution of an envelope soliton. Envelope solitons are known to be stable with respect to perturbations in the direction of propagation[4, 15], although they are unstable to perturbations in the transverse direction[15].

1.3 *The Korteweg-deVries equation and one-dimensional waves on shallow water*

A different set of assumptions leads from the free-surface water-wave problem to the Korteweg-deVries equation. If the water depth is small compared to the wave amplitude, the wavelength is large compared to the wave amplitude, and the wavelength in the direction of propagation is small compared to the transverse scale length, then the scaled evolution of the free surface is approximately governed by the KdV equation

$$-4\eta_t + 6\eta\eta_x + \eta_{xxx} = 0, \quad (1.10)$$

where $\eta = \eta(x, t)$ is the elevation around the mean [4]. The assumptions and requirements for this limit are that,

1. the surface η is only a function of (x, t) .
2. The relevant length scale in the x -direction is small when compared to the nominal depth h of the fluid. Specifically, if k is the wavenumber in the x -direction

$$(kh)^2 \ll 1.$$

3. The wave amplitudes are small compared to the depth of the fluid:

$$\epsilon = \frac{|\eta|_{\max}}{h} \ll 1$$

4. Lastly, the effects in Items 2 and 3 approximately balance. That is

$$(kh)^2 = \mathcal{O}(\epsilon).$$

It is certainly true that the KdV equation continues to be useful in the study of waves, and it is assumed that even more headway will be made with effective methods to compute quasiperiodic solutions. However I will take this opportunity to cite a bit of history instead. Figure 1.3 shows a page from John Scott Russel's early work on what he called the "Great

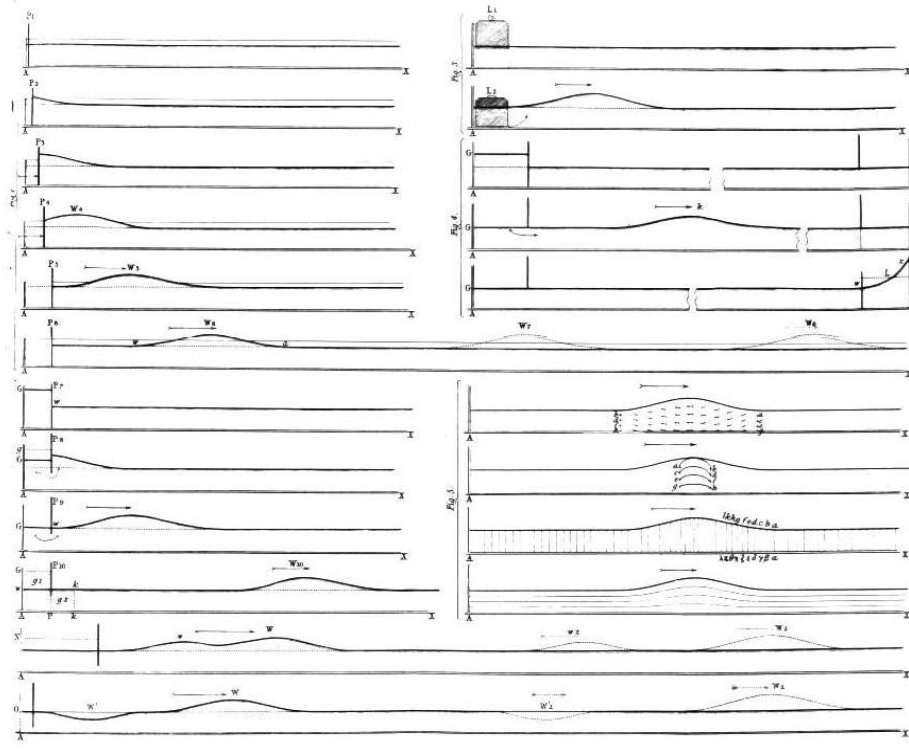


Figure 1.3: This page from Russel's work shows some of his investigations of the generation, evolution and interactions of solitary waves in shallow water.

Wave of Translation," or what we now call a solitary wave, in this case a soliton. In 1834 Russel witnessed a solitary wave in a canal in Scotland. His curiosity was piqued, and he began to experiment with waves in shallow water. I present this figure as Russel's work led, albeit circuitously, to the formulation of the Korteweg-deVries equation.

1.4 The Kadomtsev-Petviashvili equation and two-dimensional waves on shallow water

The Kadomtsev-Petviashvili (KP) equation generalizes the KdV equation to weakly 2-dimensional waves. Let the characteristic wavenumber in the transverse direction be k_y . Then wave fields $\eta(x, y, t)$ that satisfy the conditions required by KdV, and further

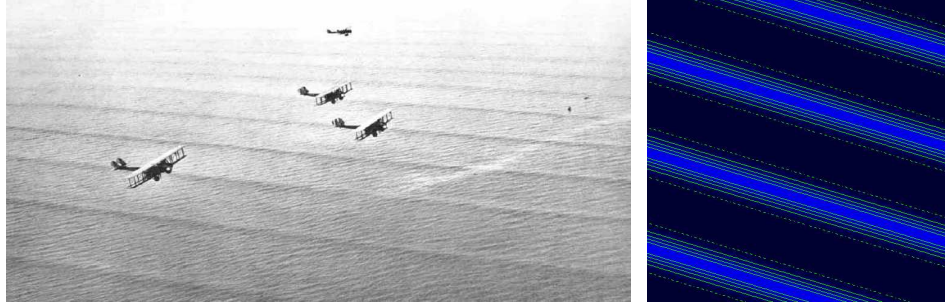


Figure 1.4: (Left) Waves off the coast of Pannama. The photo is taken from National Geographic 63 (1933). (Right) A color-contour plot of a computed genus one KP solution which also exhibits crests but no troughs. The computed solution is from Dubrovin, Flickinger & Segur [19].

satisfy

$$k_y^2 = \mathcal{O}(\epsilon)$$

are approximately governed by

$$(-4\eta_t + 6\eta\eta_x + \eta_{xxx})_x + 3\eta_{yy} = 0. \quad (1.11)$$

Equation 1.11 is also known as KP2 in the literature.

By “ g -phase KP solutions” I mean to indicate θ -function solutions of the KP equation where the surface exhibits g distinct interacting wave *phases*. A wave phase is a quasiperiodic one-dimensional variation of the surface propagating in the direction orthogonal to the variation. An example of a computed one-phase solution is shown on the right-hand side of Figure 1.4 [19]. This figure also shows shallow-water waves off the coast of Panama that exhibit the same qualitative behavior. The original caption of this picture reads:

As they near shallow water close to the coast of Panama, huge deep-sea waves, relics of a recent storm, are transformed into waves that have crests, but little or no troughs. A light breeze is blowing diagonally across the larger waves to produce a cross-chop.

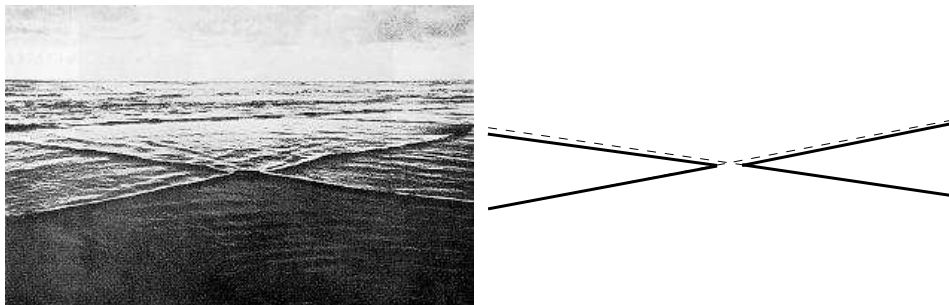


Figure 1.5: (Left) Waves off the coast of Oregon. Note that the left- and right-going waves do not simply add, but interact. The photo courtesy of T. Toedtemeier. (Left) The solid lines are traced from the left-hand photo. The interaction of the two waves is made more clear by tracing straight lines (dotted) colinear with the front crests. Notice that the crests behind the intersection of the waves is retarded, that is, the solid and dotted lines do not overlap.

Two-phase KP solutions are generically hexagons. The principle of superposition disallows such structures as the solution of linear equations. Figure 1.5 demonstrates how hexagons arise as two wave trains interact obliquely near shore. Each of the wave trains are essentially like those in Figure 1.4, but they are interacting, and not merely adding. Were they adding the crest lines traced on the right side of the figure would line up in both front and back. The left-hand side of figure 1.6 shows an aerial photograph of a Long Island (in the state of New York) beach where the crests form a repeating pattern of nearly regular hexagons propagating up the page. The right-hand side shows a computed two-phase KP solution which propagates, unchanged, up the page [19]. Neither water wave nor hexagonal KP solutions need be regular, as evidenced by Figure 1.7. Data from experiments similar to those shown in Figure 1.7 were compared in [28] with computed two-phase KP solutions (although *not* the one shown on the right from [19]). The comparisons showed very good agreement. It should be noted that [28] reported equivalent or better agreement with data when using two-phase solutions than when using a superposition of one-phase (known as *cnoidal*) solutions.

The final figure of this section shows waves generated by “The Halloween Storm of 1991,” Hurricane Grace. This already well-known storm was made famous by Sebastian

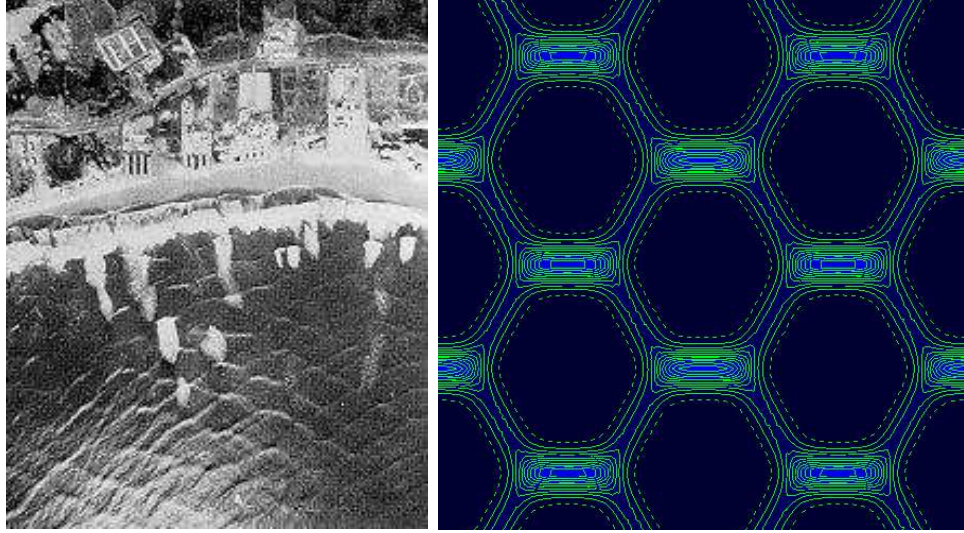


Figure 1.6: (Left) Aerial Photograph of waves off the southern coast of Long Island, west of Jones Inlet. Beyond the surf zone, the wave patterns are two-dimensional, and approximately periodic. They have flat troughs, sharp crests, and approximately hexagonal shape. The photo appears in [27]. (Right) A computed genus two solution of the KP equation that exhibits markedly similar features: also from [19].

Junger's book *The Perfect Storm*, and by the movie of the same name. The picture was taken from a small plane flying off the coast of Duck, North Carolina and shows a nearly periodic hexagonal structure of crests. The hexagon-like structure is best seen inside the dotted black ellipse. The structure contained therein is itself not a hexagon, but tiling with this shape would result in hexagons. The water in this figure is white, which is to say, as far as the Navier-Stokes equations and the free surface water wave problem are concerned, it is not water at all. The existence of hexagonal structures so far out of the putative range of the KP equation does not imply that such solutions are stable. It does however suggest that these solutions are at least robust to certain types of perturbations.

1.5 Remarks on the Maple implementations

In this thesis I present algorithms to compute the Abel map and the vector of Riemann constants. The main goal of this research program is to compute multi-phase (also called

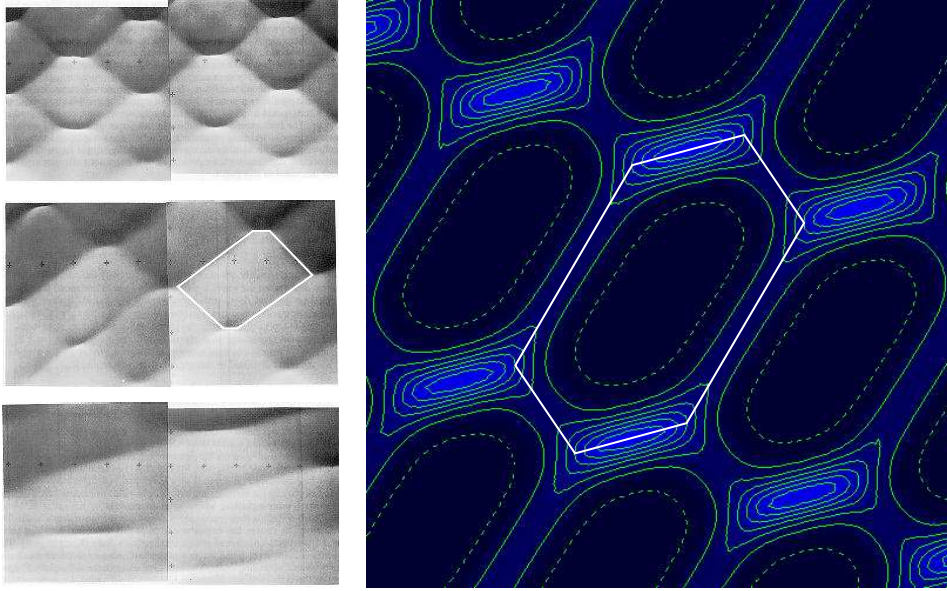


Figure 1.7: (Left) Three overhead mosaics of experiments performed in a large wave tank at the Coastal Engineering Research Center, US Army Engineer Waterways Experiment Station, in Vicksburg, MS [27]. Notice the progression in the three different experiments from quite regular to very irregular asymmetric hexagonal structures. (Right) A genus two KP solution exhibiting asymmetric hexagonal structure [19].

quasiperiodic, finite-genus and finite gap) solutions of integrable systems. In my case the application is to the study of water waves, but there are many other physical applications for such solutions. It is my hope that the algorithms created by Bernard Deconinck, Mark van Hoeij, Mike Nivala and myself will make it possible for researchers in many fields to work with these solutions without needing much background in algebraic geometry.

Toward this end I have implemented the algorithms for the Abel map and the vector of Riemann constants in the computer algebra system Maple. The implementation of the Abel map is available as part of the algebraic curves package (called `algcures` in Maple) included in every edition of Maple 11. The implementation of the vector of Riemann constants will be included as part of the algebraic curves package as of Maple 12. This package already contains procedures to compute many of the requisite objects in the formulae, given in Chapter 4, for the solutions of the NLS, KdV and KP equations. The implementations are



Figure 1.8: This picture due to Carl Miller was taken from a small plane flying off the coast of Duck, NC after Hurricane Grace. Note the nearly periodic, hexagonal structure.

discussed and demonstrated in Chapter 7. For now, let me just make some remarks about them.

- The implementations require that a Riemann surface Γ be entered as a polynomial in two complex variables, say x and y . This is no restriction for the applications I have proposed. The NLS, KdV and KP solutions I have written down require a compact, finite-genus Riemann surface as a parameter. Any such surface can be realized by polynomials in two complex variables using the methods described in Chapter 3.
- These algorithms and implementation do not presently handle non-compact or infinite-genus Riemann surfaces. Note also that floating-point coefficients are not allowed in the defining polynomial. This is due to the need to locate and define singularities precisely. Imprecisely defined branch points, for instance, can lead to the genus of a curve being incorrectly computed. This error is insurmountable, and very probable if coefficients are known only approximately. This is true because the genus of a curve

drops as roots coalesce. As an example, the genus of

$$y^2 - (x^3 + x^2 + \epsilon x)$$

is zero if

$$\epsilon \neq 0, \frac{1}{4}$$

and one otherwise. Thus if the coefficient ϵ *may* be zero, but is not known exactly, correct computation, of for example the genus, is not possible. In the applications for which these algorithms were intended, this is almost never a restriction as in these applications the polynomials are represented exactly.

- The algorithms and implementations *do* handle a compact, algebraic curve with arbitrary singularities however. The polynomial giving rise to the plane algebraic curve, and hence to the Riemann surface, is only required to be irreducible.

The general applicability of these algorithms bears repeating. Any irreducible plane algebraic curve (that is, the zero set of any irreducible polynomial in two complex variables) is allowed as input. There is *no* restriction to elliptic curves, hyperelliptic curves, Hurwitz curves *etc.*, as is common in some algorithms computing objects associated with algebraic curves.

- The generality emphasized in the previous item is naturally the bane of the implementation. That is to say that the insistence on dealing with general singularities has required much time, effort and, if I may be so bold, ingenuity. After throwing out unsuccessful attempts (that is, thousands of lines of code), the implementation now runs to nearly 3000 lines of Maple code.

Early on in this research, it was realized that the computation of specifically the Abel map, “well away” from the singularities, is relatively straightforward. Constructing, implementing and debugging algorithms to deal with the singularities correctly is not so. However, as will be noted later, the computation of the solutions given by the formulae (4.2), (4.3) and (4.4) requires that the Abel map be computed *up to* singularities.

A final Introductory remark is in order. Other approaches for the effective calculation with Riemann surfaces exist. For instance, the group at TU-Berlin is also interested in solutions of integrable systems, and they use a Schottky uniformization approach [9, 8]. This approach is equally general and has some advantages over the approach taken here, in which objects are computed, in most cases directly from their algebro-geometric definitions. The main advantage is that their algorithms are all geometric in nature, allowing for fast implementations using floating-point numbers. In contrast, although many of my algorithms are geometric (*and* use floating-point numbers), there may also be dependence (in the case of singular plane curves) on algebraic structure. Dealing with the algebraic structure requires exact arithmetic and is often computationally costly. An advantage of the approach taken here is that it allows (in the near future) for the use of the θ -function solutions as solutions of suitable initial-value problems. This is discussed for the KP equation in [16], and more generally in Section 4.6 of [6]. Such an approach is beneficial for the matching of the finite-genus solutions with experimental data.

Chapter 2

PRELIMINARY EXAMPLES OF GENUS ZERO AND GENUS ONE
RIEMANN SURFACES**2.1** *Genus zero: the Riemann sphere*

The soliton solutions of these equations have especially found a well-deserved niche in both theory and experiment. This popularity has not been shared with their periodic and quasiperiodic counterparts. I believe that the main reason for this is the inherent connection of these solutions with the theory of Riemann surfaces and Riemann θ -functions, and the attendant mathematical and computational difficulty associated with these topics. I will spend the remainder of this section attempting to justify this belief, giving along the way a very brief introduction to these theories. Much more detail, including citations, concerning these topics is offered in Chapter 3. Citations are kept to a minimum in this section, and many terms will remain undefined. My intention is to give the reader an introductory look at the mathematical machinery involved.

All applied mathematicians are familiar with the concept of Riemann surfaces: they are standard fare to most complex analysis textbooks. However, this familiarity in most cases is limited to the concepts of branch points and branch cuts used to accommodate the multi-valuedness of n -th roots. The canonical example, at least of a function with a finite number of values, is the complex multi-valued function

$$y = x^{\frac{1}{2}}, \quad x \in \mathbb{C} \cup \{\infty\} \tag{2.1}$$

which has branch points at $x = 0$ and $x = \infty$. Introducing a branch cut from 0 to ∞ by, for example, excluding the negative real axis from the domain, turns (2.1) into two single-valued functions y_{\pm} given by

$$y_{\pm} = \pm\sqrt{x}.$$

Analytically continuing y_{+} around either of the branch points (and necessarily across the

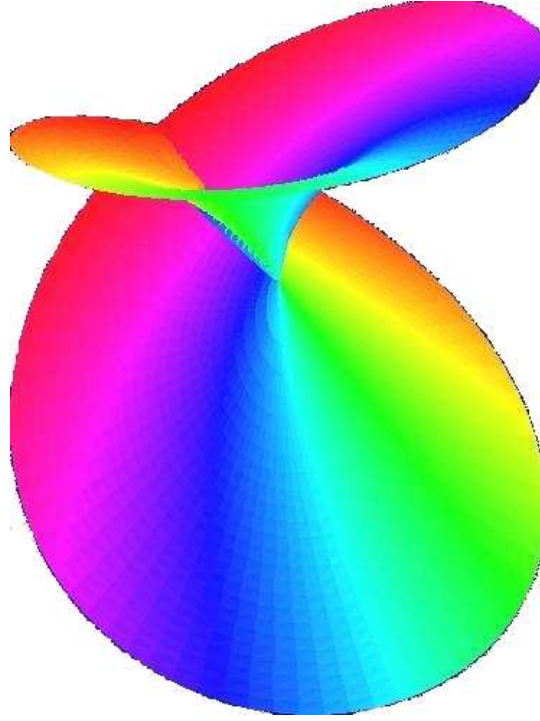


Figure 2.1: In this plot around the branch point $x = 0$, elevation represents $\|\pm\sqrt{x}\|$. Further, color represents argument. Thus by smoothly following color, one can tell how to analytically continue the complex values of $\pm\sqrt{y}$ around the branch point and “through the intersection.” The “intersection” is an artifact of plotting in 3-dimensions a surface that actually exists in 4-dimensions.

branch cut) carries it into the value of y_- . Thus the two surfaces defined by (2.1) are connected, where here I intend “connected” to be interpreted in the vernacular. It takes a bit more work to establish that when these two surfaces are joined, the result is a Riemann surface. In this case the surface is topologically the Riemann sphere

$$\mathbb{P}^1 := \mathbb{C} \cup \{\infty\}.$$

Equation (2.1) could also be written as

$$y^2 = x, \quad z \in \mathbb{P}^1,$$

The Riemann surface arising from this function could also be thought of the zero set of the

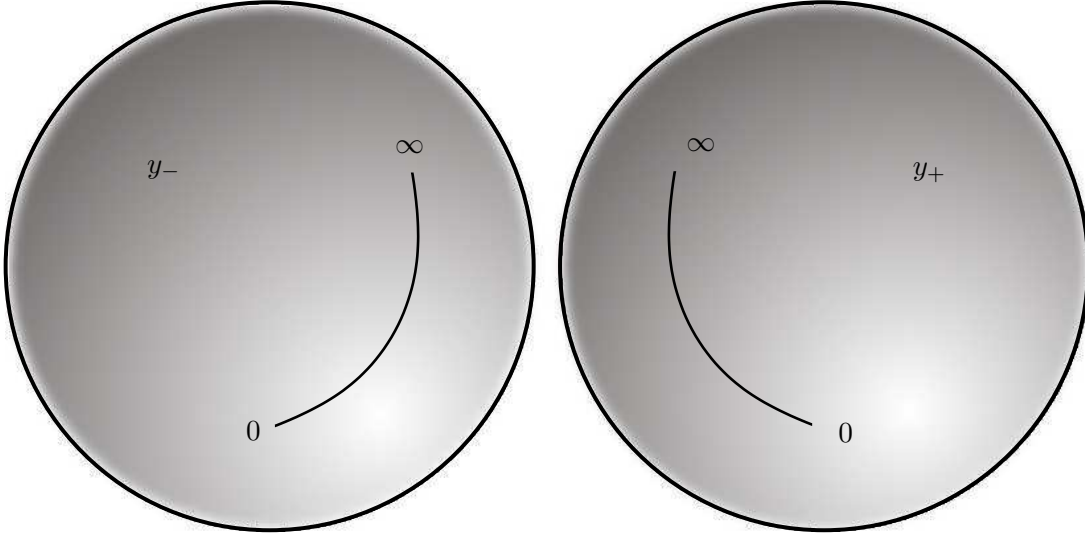


Figure 2.2: The two *sheets* of the multi-valued function $y = x^{1/2}$ are shown. Branch cuts are introduced on each sheet, creating two domains on which the functions y_- and y_+ may be analytically continued unambiguously. Note that the particular form of the branch cut is not stated, however assume that the branch cut has the same form on each sheet..

polynomial

$$F(x, y) = y^2 - x, \quad x, y \in \mathbb{P}^1,$$

where many details are being ignored for the moment. Setting

$$F(x, y) = y^2 - x = 0 \tag{2.2}$$

imposes two conditions on the 4-real-dimensional space \mathbb{C}^2 described by two complex variables. That is (2.2) imposes

$$\Re(F(x, y)) = \Re(y^2 - x) = 0$$

$$\Im(F(x, y)) = \Im(y^2 - x) = 0,$$

defining a 2-real-dimension (or 1-complex-dimensional) object in \mathbb{R}^4 (or \mathbb{C}^2). Recall from above that there are two *sheets* of this object. That is, for almost any α in the extended complex plane \mathbb{P}^1 , there are two y -roots of

$$F(\alpha, y) = 0$$

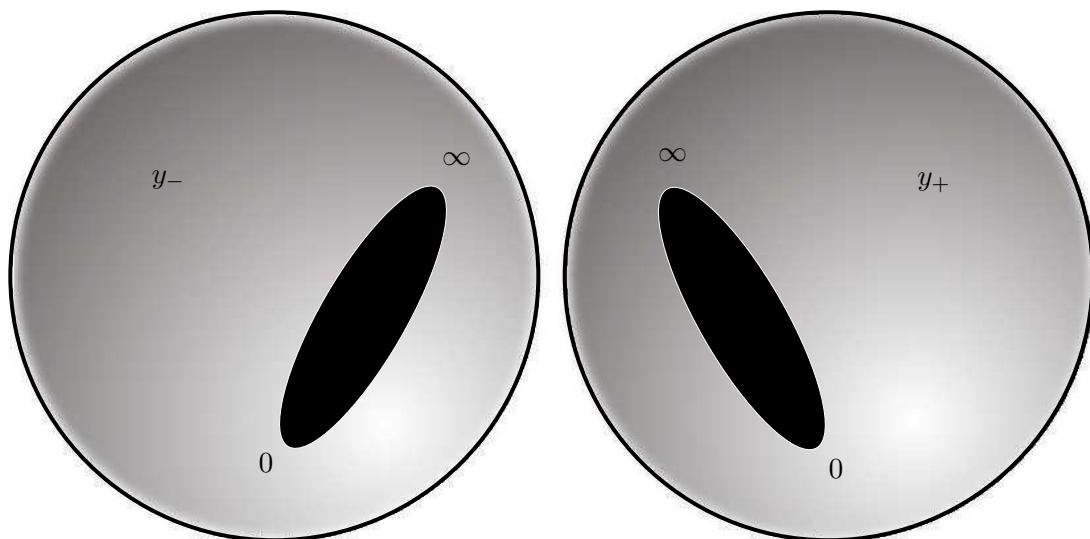


Figure 2.3: The two branch cuts from the previous figure are opened up.

represented by

$$y_+ = \sqrt{x}$$

and

$$y_- = -\sqrt{x}.$$

This situation is shown in Figure 2.2. A branch cut joining 0 to ∞ is shown. Note however I do not say *how* the branch cut is chosen. This is done purposefully, as for the algorithms presented, no choice of branch cuts is ever made and branch cuts are by nature arbitrary. The surface in Figure 2.2 has been “stretched” to easier display 0 and ∞ in the same figure. Such stretching is just a change of coordinates, and does not change the topology or affect the value of the integrals of holomorphic differentials.

The branch cuts indicate portions of the complex x -plane that are “off limits” for basic travel. That is, if a path does not cross the branch cut, then analytically continuing either y_+ or y_- results in a function well defined everywhere in the complex x -plane, except of course the cuts. Thus the cuts on both sheets are opened up in Figure 2.3.

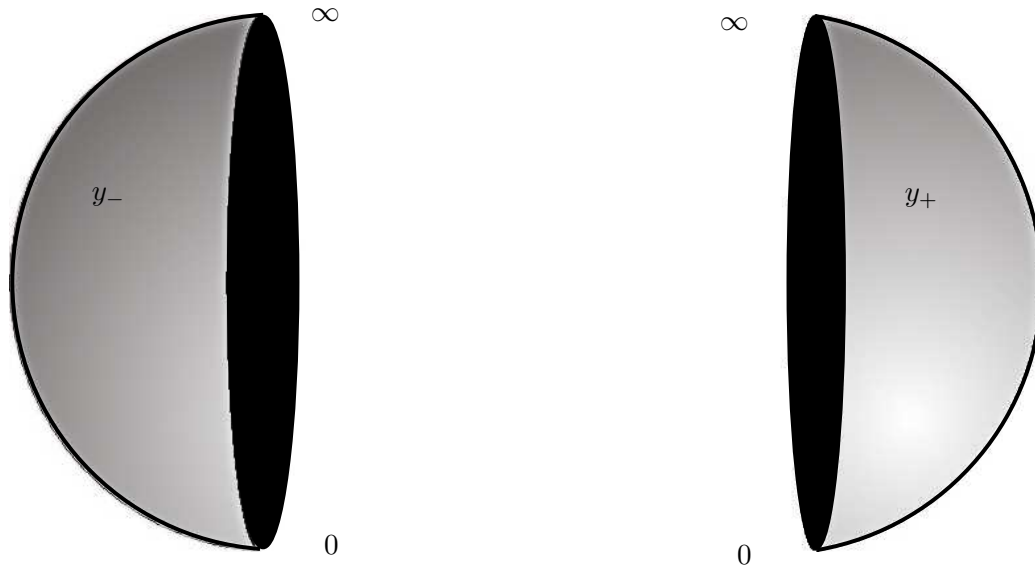


Figure 2.4: The cuts from the previous figure are further opened. The two sheets are also stretched again to resemble two halves of a sphere. This type of coordinate change leaves the integrals of holomorphic differentials unchanged. If care is taken to correctly match the orientation of the branch cuts on either sheet, the two halves may be joined to form a Riemann surface. The resulting surface is topologically equivalent to the Riemann sphere.

The cuts are opened further in Figure 2.4, and from this figure it should be clear how the two sheets are to be joined into one surface. The resulting surface is topologically the same as the Riemann sphere. The Riemann sphere \mathbb{P}^1 is the trivial example of a Riemann surface. It is a compact surface with no handles (or holes). More mathematically speaking, the genus is zero. Two things in particular are trivial about the genus zero case.

- By Stokes' Theorem, any closed path on the Riemann sphere is trivial for the purposes of integrating holomorphic differentials [25].
- Holomorphic differentials are integrands globally defined on a Riemann surface that do not have poles. A consequence of Stokes' Theorem and Liouville's Theorem is that there are no holomorphic differentials on the Riemann sphere [25].

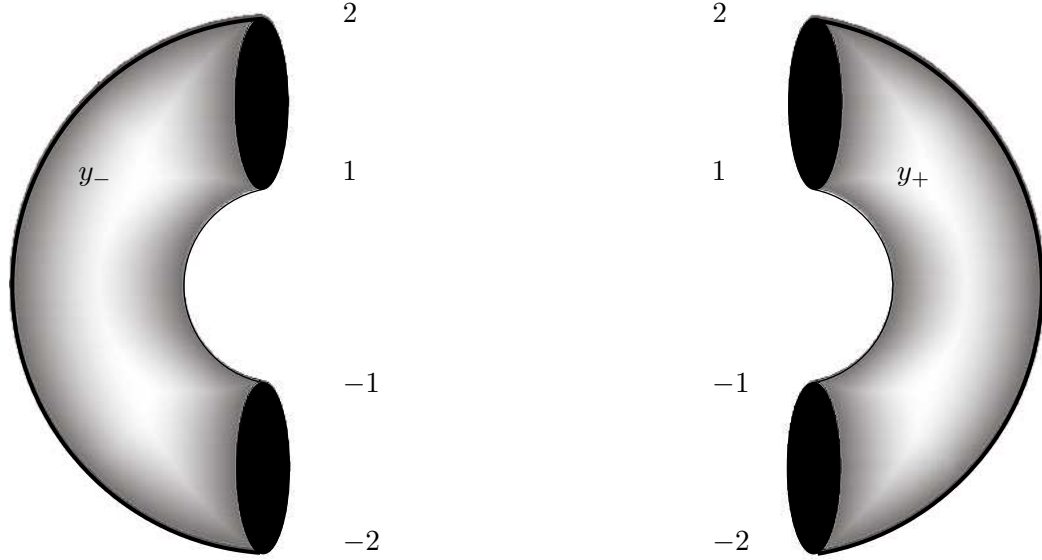


Figure 2.5: Analogous to the case of $F = y^2 - x = 0$, here branch cuts are introduced and opened on two copies of the Riemann sphere. In this case the Riemann surface comes from the polynomial $F = y^2 - (x + 2)(x + 1)(x - 1)(x - 2)$. The branch points are ± 1 and ± 2 . Here branch cuts are chosen joining -2 to -1 and 2 to 1 . Any other valid choice of branch cuts would yield an equivalent Riemann surface.

2.2 Genus one: the first non-trivial example

The polynomial giving rise to the Riemann surface does not need to get much more complicated in order for the above bullet-points to no longer hold. Take, for example the Riemann surface obtained from the zero set of the polynomial

$$F = y^2 - (x + 1)(x - 1)(x + 2)(x - 2).$$

Once again, for almost any $\alpha \in \mathbb{P}^1$ there are two y -roots of the equation

$$F(\alpha, y) = 0.$$

The points

$$\alpha = \pm 1, \pm 2$$

are the branch points of F .

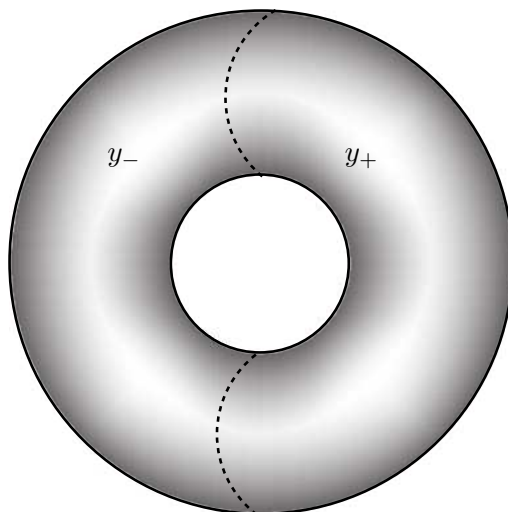


Figure 2.6: In this figure the two sheets are joined to form a compact surface with one handle (or hole). That is, a compact genus one Riemann surface.

Figure 2.5 is analogous to Figure 2.4. Branch cuts have been introduced on both the y_- and y_+ sheets. These cuts, here chosen as to join -2 to -1 and 2 to 1 have already been opened.

In Figure 2.6 the two sheets are joined to form a surface with one handle (or hole). The number of handles of a given surface is a topological invariant. That is, the number of holes in a surface can not be changed by stretching or twisting the surface. The surface must be ripped or punctured in order to change the genus.

In contrast to the Riemann sphere, a genus one Riemann surface supports non-trivial cycles. Figure 2.7 demonstrates two cycles that can not be deformed into each other. Clearly the red cycle can not be deformed into the blue cycle without leaving the surface in some way. As will be shown later, a basis of cycles can be constructed on a Riemann surface such that *any* closed path on the surface can be written as a sum of cycles from the basis. The dimension of this basis is twice the genus of the surface. Figure 2.8 shows how the cycles in Figure 2.7 appear as paths in the complex plane.

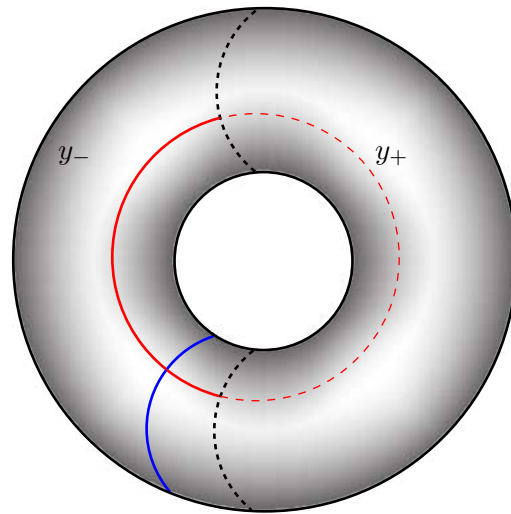


Figure 2.7: As opposed to the Riemann sphere, in this case there are closed paths that can not be contracted to points. These *cycles* are not trivial for the purpose of integrating holomorphic differentials. Note that the blue cycle goes around a handle of the surface, and here is shown to be completely on the y_- sheet. The red cycle, going around the hole, is partly on the y_- sheet and partly on the y_+ sheet. Note that the part of the red cycle on the y_+ sheet is indicated as a dotted line.

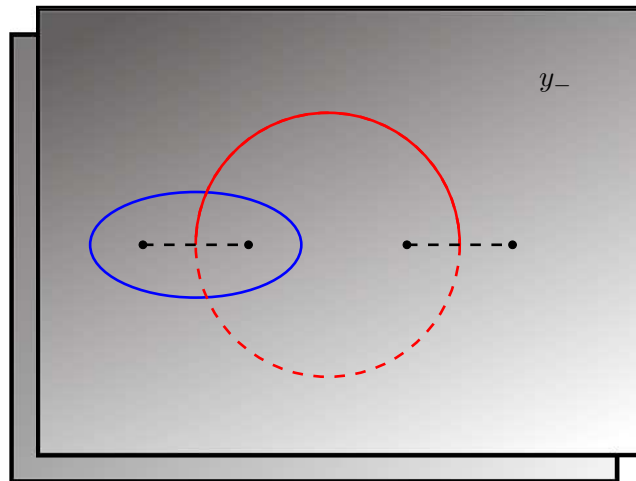


Figure 2.8: The cycles from the previous figure are shown here as they appear as paths in the complex x -plane. The plane on top is the sheet labeled y_- , whereas the plane mostly obscured is the y_+ sheet. Notice that the upper part of the red cycle is indicated as a solid line on the y_+ , just as it is in the previous figure. The portion of the red cycle displayed as a dotted is to indicate that that portion of the cycle is traversed on the bottom plane.

Chapter 3

INTRODUCTION TO THE THEORY OF RIEMANN SURFACES

A Riemann surface is a one-dimensional complex-analytic manifold [21]. The only topological invariant of a connected Riemann surface is its number of handles, or *genus*. A compact, connected surface Γ can be obtained from an irreducible plane algebraic curve C through a number of equivalent processes [36, 25]. In the algorithms I present here a Riemann surface is a branched algebraic y -cover of the complex x -sphere. I give a very brief introduction to the needed theory of Riemann surfaces from this perspective. More details can be found in the standard references [21, 36, 25, 26].

3.1 Points and places

Consider an irreducible polynomial in x and y

$$F(x, y) = f_0(x)y^n + f_1(x)y^{n-1} + \dots + f_n(x), \quad (3.1)$$

where $x, y \in \mathbb{C}$ and the $f_i(x)$ are polynomials in x with complex coefficients. The plane algebraic curve C is the locus of pairs (x, y) that satisfy

$$F(x, y) = 0.$$

I treat x as an independent variable, and by using different terms I intentionally distinguish between x -values and the (x, y) -pairs over the x -sphere.

- *Point* refers to a value of the independent variable

$$x \in \mathbb{P}^1 = \mathbb{C} \cup \{\infty\}.$$

- *Place* refers to a location on the Riemann surface Γ . Almost all places are specified by a pair (x_0, y_0) . However, for some places it is necessary to go beyond this by specifying

a pair of expansions for x and y , both in a local coordinate t . These expansions over a point $x \in \mathbb{P}^1$ solve (3.1), for the moment formally. Places are written as capital Roman letters, usually P or Q .

A point $\alpha \in \mathbb{C}$ is called a *regular point* if the equation

$$F(\alpha, y) = f_0(\alpha)y^n + f_1(\alpha)y^{n-1} + \dots + f_n(\alpha) = 0 \quad (3.2)$$

has n distinct finite roots

$$y(\alpha) = \{\beta \in \mathbb{C} : F(\alpha, \beta) = 0\}.$$

A point $\alpha \in \mathbb{C}$ is a *discriminant point* if it is not regular [25], which is to say, if there are less than n distinct y -roots of (3.2). These are exactly the points where

$$F(x, y) = 0 = F_y(x, y)$$

where F_y is the partial derivative of F with respect to y . As F_y is also a polynomial in x and y , there are only a finite number of discriminant points [25].

The y -roots of (3.1) at $x = \infty$ are defined as the roots of

$$G(X, y) = X^{\deg(F, x)} F(1/X, y) = 0,$$

where $\deg(F, x)$ is the degree of F in x [7]. The point at $x = \infty$ is a regular point if G has n distinct y -roots at $X = 0$, and a discriminant point otherwise.

Near a regular point $\alpha \in \mathbb{P}^1$ the y -roots of (3.1) are given by n power series in $x - \alpha$. See Example 1 for the calculation of such a series. Near a discriminant point, representing the y -roots may require a fractional power series, known as a Puiseux series. The need for fractional powers is evident in Example 2. Newton's Theorem will introduce a local coordinate and remove the need for fractional power series.

Example 1. *The algebraic curve*

$$F(x, y) = y^2 - (x^3 - 1) = 0$$

gives rise to a genus one Riemann surface. To find the y -roots near $x = 0$, solve for

$$y = (-1 + x^3)^{\frac{1}{2}},$$

then expand formally in x to get two power series:

$$y = \pm i \left(1 - \frac{x^3}{2} + \frac{x^6}{8} \dots \right).$$

To unambiguously specify the two places above $x = 0$, it suffices to take only the first term in each y -expansion. That is, the places are correctly and completely specified by the pairs $(0, i)$ and $(0, -i)$.

Example 2. *The only discriminant points of*

$$y^2 - (x^3 - 1) = 0$$

occur at the third roots of unity

$$\alpha_j = e^{2\pi i j/3}, \quad j = 0, 1, 2,$$

where the single y -root is $y = 0$. To find

$$y = y(x)$$

near α_j , let

$$x = \alpha_j + t.$$

For convenience I choose to work near

$$x = \alpha_0 = 1,$$

then the y -roots are given by

$$\begin{aligned} y &= \pm \left((1+t)^3 - 1 \right)^{\frac{1}{2}} \\ &= \pm \left(3t + 3t^2 + t^3 \right)^{\frac{1}{2}} \\ &= \pm \sqrt{3} t^{\frac{1}{2}} \left(1 + t + \frac{t^2}{3} \right)^{\frac{1}{2}} \end{aligned}$$

Expand in t , rearrange and back substitute to get a fractional power series in $x - 1$

$$y(x) = \pm\sqrt{3} \left((x-1)^{\frac{1}{2}} + \frac{(x-1)^{\frac{3}{2}}}{2} + \frac{(x-1)^{\frac{5}{2}}}{24} + \dots \right). \quad (3.3)$$

The presence of fractional powers in (3.3) defines α_0 to be a branch point. A similar conclusion is easily reached for α_1 and α_2 .

3.2 Newton's Theorem and convergence of Puiseux expansions

The following statement of Newton's Theorem is an abridged version of that in [7], and establishes the existence of the series found in these examples.

Newton's Theorem. *In a neighborhood of α , the n y -roots of (3.1) are determined by a finite number of pairs of expansions of the form*

$$P_j = \begin{cases} x &= \alpha + t^{r_j} \\ y &= \beta_j t^{s_j} + \beta'_j t^{s'_j} + \dots \end{cases}, \quad (3.4)$$

where $\alpha, r_j, \beta_j, \beta'_j, s_j$ and s'_j are as described below.

- **Finite places:** *A place is called finite if both the x - and y -components are finite for $t = 0$. For each finite place the following properties hold.*

- *The branching number r_j counts the number of y -roots that coalesce at the place P_j . If $r_j > 1$ for one of the P_j , then α is a branch point.*
- *For $|t| > 0$, a place P_j represents r_j distinct y -values. Further,*

$$\sum_j r_j = n.$$

- *The coefficients*

$$\beta_j, \beta'_j, \dots \in \mathbb{C}$$

are all non-zero.

- *The exponents*

$$r_j, s_j, s'_j, \dots \in \mathbb{Z}$$

are positive (possibly zero) and share no common factor.

- **Infinite places:** *Infinite places are those for which one or both components are infinite for $t = 0$. This occurs only when $\alpha = \infty$ or*

$$f_0(\alpha) = 0,$$

where f_0 is the leading polynomial coefficient of (3.1). Infinite places have the same properties as finite places except:

- *the y -components may have a finite number of negative exponents*

$$s, s',$$

and

- *for places over $x = \infty$, the x -components have*

$$x = 1/t^r$$

replacing

$$x = \alpha + t^r.$$

Two places as above are equivalent if the substitution

$$t \rightarrow \xi t,$$

with ξ being an r_j -th root of unity, identifies them. Away from $t = 0$, two equivalent places determine the same set of y -roots of

$$F(x, y) = 0.$$

Away from $t = 0$, non-equivalent places determine disjoint sets of y -roots.

In general, the x -component of a place can be written as

$$x = \alpha + \alpha' t^r \quad (\text{or } x = \alpha' / t^r),$$

but $\alpha' = 1$ is chosen for convenience. Any choice α' is valid, but the

$$\beta, \beta', \dots$$

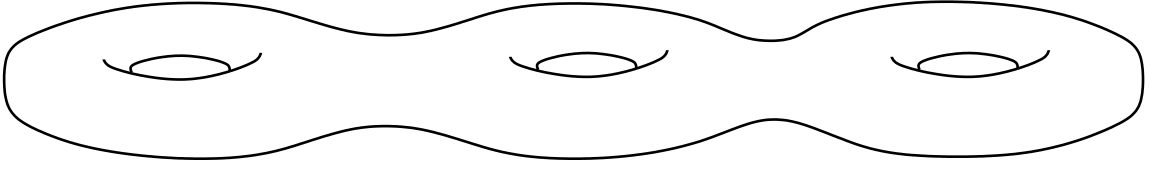


Figure 3.1: A cartoon of a genus three Riemann surface.

reflect this choice.

The y -component in Newton's Theorem converges at least over the punctured disc \mathcal{R} , where

$$\mathcal{R} = \{x : |x - \alpha| < |\alpha - \lambda|, x \neq \alpha\}. \quad (3.5)$$

and $\lambda \neq \alpha$ is the discriminant point closest to α . This restriction on the radius of \mathcal{R} arises as λ is the nearest point (distinct from α) where the derivative

$$dy/dt = -F_x/F_y$$

is not defined. Convergence of places over regular points is established in [7]. In this case the roots of

$$F(x, y) = 0$$

are represented as n convergent series in

$$t = x - \alpha$$

on the disc

$$\mathcal{R} \cup \{\alpha\}$$

that is, the unpunctured disc.

If α is a discriminant point, then convergence of the y -component is questionable at $x = \alpha$. For now consider α to be a non-branch discriminant point. The convergence of the

y -series over branch points is discussed in Section 3.3. Suppose P is given by

$$P = \begin{cases} x &= \alpha + t \quad (\text{or } 1/t \text{ over } \infty) \\ y &= \beta t^s + \beta' t^{s'} + \dots \end{cases}, \quad (3.6)$$

with α (or ∞) a non-branch discriminant point. The expansion

$$y = \beta(x - \alpha)^s + \beta'(x - \alpha)^{s'} + \dots \quad (3.7)$$

is unique and convergent at least on \mathcal{R} . Convergence of (3.6) over $x = \alpha$, that is for $t = 0$, depends on whether the x - and y -components of P are finite there.

- **Finite places** ($x, y \in \mathbb{C}$): For finite places, no one of the

$$s, s', \dots$$

are negative, and (3.7) is the unique power series expansion valid on

$$\mathcal{R} \cup \{\alpha\}.$$

- **Infinite places (i)** ($x \in \mathbb{C}, y = \infty$): Such places occur over points $x = \alpha$ for which the leading polynomial coefficient

$$f_0(\alpha) = 0,$$

where f_0 is as in (3.1). In this case, a finite number of the

$$s, s', \dots$$

will be negative and (3.7) is the unique Laurent series expansion in $x - \alpha$, convergent on \mathcal{R} .

- **Infinite places (ii)** ($x, y = \infty$): If $x = \infty$ is not a branch point, then each place over ∞ is of the form

$$P = \begin{cases} x &= 1/t \\ y &= \beta t^s + \beta' t^{s'} + \dots \end{cases},$$

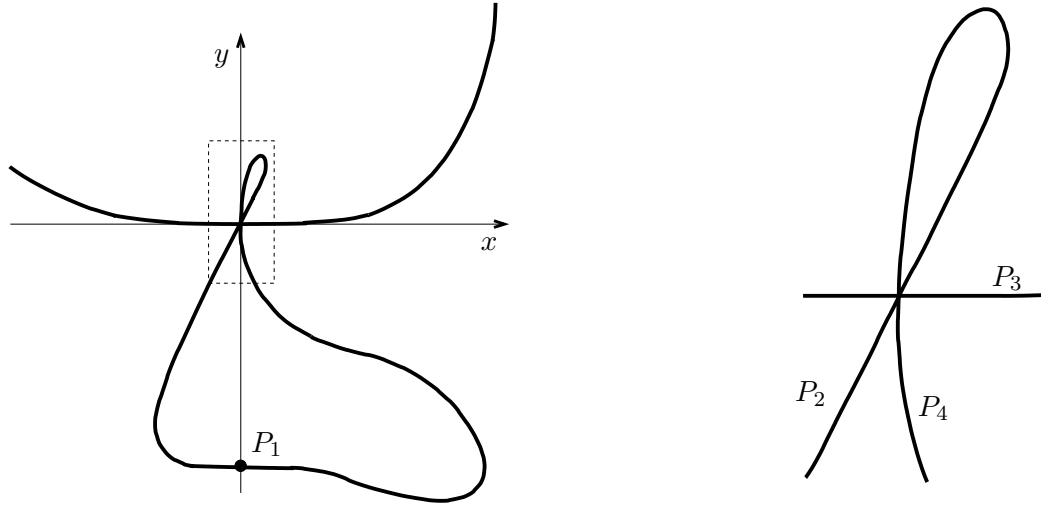


Figure 3.2: A plot of the real part of $F = -y^5 + (2x - 1)y^4 - (3x - x^2)y^3 - (x - 3x^2)y^2 + (x^3 - 2x^2)y + x^6$. The figure on the right is an enlarged view of the part of the graph on the left enclosed by the box.

where a finite number of

$$s, s', \dots$$

are negative. That is, $y(t)$ is a unique Laurent series expansion in t , where

$$t = \frac{1}{x}$$

is an analytic function for finite x . Thus $y(t)$ is convergent on

$$\mathcal{R}_\infty = \{x : |x| > \lambda_{\max}, x \neq \infty\}$$

where λ_{\max} is the finite discriminant point with largest absolute value.

Given an irreducible polynomial $F(x, y)$, the union of all places over all $x \in \mathbb{P}^1$ is a finite genus compact, connected Riemann surface [25]. Figure 3.3 shows an idealization of a genus three Riemann surface. The actual surface defined by an algebraic curve C is a two-real-dimensional manifold embedded in a four-dimensional space. The example that follows demonstrates the use of Puiseux expansions to determine the local structure of a plane algebraic curve.

Example 3. Over the point $x = 0$, the five y -roots of the polynomial

$$F = -y^5 + (2x - 1)y^4 + (3x - x^2)y^3 + (x - 3x^2)y^2 + (x^3 - 2x^2)y + x^6$$

are represented by the four places

$$\begin{aligned} P_1 &= \begin{cases} x &= t \\ y &= -1 - t^3 + \dots \end{cases}, & P_2 &= \begin{cases} x &= t \\ y &= 2t + t^2 + \dots \end{cases}, \\ P_3 &= \begin{cases} x &= t \\ y &= \frac{t^4}{2} + \frac{t^5}{4} + \dots \end{cases}, & P_4 &= \begin{cases} x &= t^2 \\ y &= t - \frac{t^3}{2} + \dots \end{cases}. \end{aligned}$$

Note that, in contrast to Example 1, here the first terms in the y -expansions do not suffice to unambiguously specify the places, but only distinguishes P_1 from the remaining three places. A graph of the real part of

$$F(x, y) = 0$$

near

$$(x, y) = (0, 0)$$

is seen in Fig. 3.2. I determine which part of the graph corresponds to which place in the example by calculating dy/dx for each P_j . Place P_1 is easily identified in the figure as it is the only place where $y \neq 0$ for $x = 0$.

For P_2 and P_3 , make the simple substitution $t = x$ in the series for y . At place P_2 , y and its derivative are

$$\begin{aligned} y &= 2x + x^2 + \dots \\ \frac{dy}{dx} &= 2 + 2x + \dots. \end{aligned}$$

The correct match for place P_2 is clear as

$$\left. \frac{dy}{dx} \right|_{x=0} = 2$$

and there is only one component of the graph which has positive finite slope near $x = 0$.

For place P_3

$$y = \frac{x^4}{2} + \frac{x^5}{4} + \cdots$$

$$\frac{dy}{dx} = 2x^3 + \frac{5x^4}{4} + \cdots .$$

As

$$\left. \frac{dy}{dx} \right|_{x=0} = 0,$$

this is also easy to match.

At place P_4 , parametrically differentiate

$$\frac{dy}{dx} = \frac{\partial y / \partial t}{\partial x / \partial t} = \frac{1 - \frac{3t^2}{2} + \cdots}{2t}$$

$$= \frac{1}{2t} - \frac{3}{4}t + \cdots .$$

Thus place P_4 matches the component of the graph for which dy/dx is undefined at $t = 0$.

Note $x = 0$ is a branch point since x is not linear in t for P_4 .

3.3 Branches

Over a regular point α there are n distinct places and the terms “place” and “branch” are equivalent. Over a branch point α , there are less than n places, and thus there are some number of places over α for which the branching number $r > 1$. Let

$$P = \begin{cases} x &= \alpha + t^r \\ y &= \beta t^s + \beta' t^{s'} + \cdots \end{cases}, \quad (3.8)$$

where $|r| > 1$, be such a place. From the place P , form the r branches

$$\tilde{P}_j = \begin{cases} x &= \alpha + (te^{2\pi ij/r})^r = \alpha + t^r \\ y_j &= \beta (te^{2\pi ij/r})^s + \beta' (te^{2\pi ij/r})^{s'} + \cdots \end{cases} \quad \arg(t) \in [0, 2\pi/r), \quad (3.9)$$

where “branch” signifies that the argument of t is restricted as indicated. As

$$r, s, s', \dots$$

are coprime, the y_j in (3.9) are all distinct. Further the y -series in (3.8) is related to the j -th series in (3.9) by

$$y_j(t e^{-2\pi i j/r}) = y(t).$$

The index j may be chosen such that

$$\exp(-2\pi i(j)/r) t$$

has argument restricted to $[0, 2\pi/r)$. Thus the r distinct roots represented by place P are also represented by the r branches

$$\tilde{P}_1, \dots, \tilde{P}_r,$$

with $\arg(t)$ restricted.

Suppose that $\lambda \neq \alpha$ is the nearest discriminant point to α , and let \mathcal{R} be given as in 3.5. The points $x \in \mathcal{R}$ are all represented exactly once by

$$x = \alpha + t^r$$

if the argument of t is restricted as in (3.9). Restricting t this way makes $t(x)$ the single valued function

$$t(x) = \sqrt[r]{x - \alpha},$$

where here and subsequently I denote by

$$\sqrt[r]{\xi}$$

the r -th root of ξ with argument on

$$[0, 2\pi/r).$$

Thus defined, $t(x)$ is analytic on \mathcal{R} . Each of the y -series is analytic in t , as noted in Section 3.1, and so the n roots of (3.2) are represented as n series which are convergent up to the nearest distinct discriminant point.

3.4 The homology of a Riemann surface

A *path* γ on Γ is a parameterized curve

$$\begin{aligned}\gamma : [0, 1] &\mapsto \Gamma \\ z &\mapsto (x(z), y(z)), \quad F(x(z), y(z)) = 0,\end{aligned}$$

where

$$F(x, y) = 0$$

is the equation that gives rise to the surface Γ . The parameter z is chosen on $[0, 1]$ only for convenience. A path is called closed if

$$\gamma(0) = \gamma(1).$$

A closed path that bounds a part of Γ can be contracted to a place, *i.e.* it is homologous to zero, and is therefore trivial for the purpose of integrating analytic functions.

A *cycle* on Γ is a closed path that is not homologous to zero. Cycles may be added. Traversing the sum, $c_j + c_k$, of the cycles c_j and c_k specifically means traversing the cycle c_j followed by the cycle c_k . The negation, $-c_j$, is the cycle c_j traversed in the opposite sense.

Let $c_j \circ c_k$ denote the number of oriented intersections of the cycles c_j and c_k . If the cycles c_j and c_k do not intersect, then

$$c_j \circ c_k = 0.$$

If c_j and c_k intersect once, then

$$c_j \circ c_k = \pm 1,$$

where the sign of the intersection index is the orientation of the basis

$$(\mathbf{t}_j, \mathbf{t}_k).$$

Here $\mathbf{t}_j, \mathbf{t}_k$ are the unit tangent vectors of the cycles c_j and c_k respectively at the point of intersection. For cycles with multiple intersections, $c_j \circ c_k$ is the sum over all intersections, each computed as a single intersection.

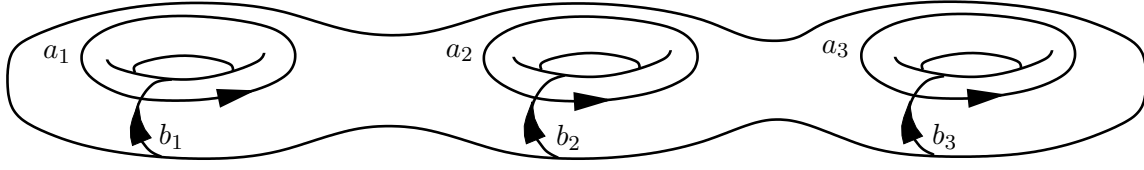


Figure 3.3: The paths indicated in this figure are the a - and b -cycles that form a basis of cycles on Γ . The a -cycles go around the holes, and the b -cycles encircle the handles. The a - and b -cycles shown indeed have the intersection indices required.

On a Riemann surface of genus g there exists a basis for its first homology group given by

$$\mathcal{H} = \{a_1, \dots, a_g, b_1, \dots, b_g\}, \quad (3.10)$$

- such that no cycle in \mathcal{H} is continuously deformable into any other,
- all possible cycles on Γ are sums of cycles in \mathcal{H} , and
- the intersections indices of the cycles are such that

$$a_j \circ a_k = 0, \quad b_j \circ b_k = 0, \quad a_j \circ b_k = \delta_{jk}, \quad (3.11)$$

where δ_{jk} denotes the Kronecker delta [17].

As in Figure 3.3, the a cycles may be thought of as encircling the holes, and the b cycles as going around the handles, of Γ .

Note: an algorithm to explicitly compute a representation of the homology of a Riemann surface associated with a plane algebraic curve was developed in [17]. The algorithms presented in this work make use of that previously developed algorithm.

3.5 The Riemann surface cut for the purpose of integration

Some of the results that follow rely on residue calculus on the cut surface $\tilde{\Gamma}$. For instance, the vector of Riemann constants arises by calculating a logarithmic integral, once as a sum of residues, and once as a line integral “around” Γ . The integrands of concern in these

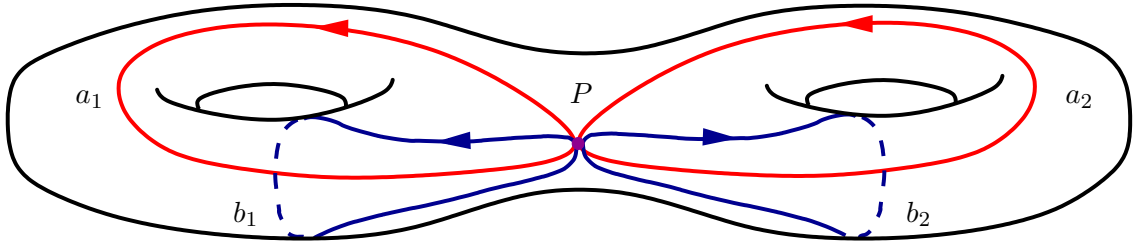


Figure 3.4: The paths indicated in this figure are the a - and b -cycles that form a basis of cycles on Γ . The a - and b -cycles are drawn in red and blue respectively. The place P indicated by the purple dot is on all the cycles.

cases are Abelian, and therefore invariant under any change of local coordinate. Thus the Riemann surface may be “stretched” by smooth changes of local coordinate without changing the value of the resulting integrals. A surface $\tilde{\Gamma}$ with a boundary $\partial\tilde{\Gamma}$ is obtained by constructing and cutting along a certain closed path on Γ . Figures 3.4 through 3.11 illustrate the construction of a cut surface $\tilde{\Gamma}$ from a Riemann surface Γ . The list below outlines this process, and extends the information given in the captions of the figures.

- **Figure 3.4, Page 37:** Suppose place P is on all of the cycles in \mathcal{H} . In the figure, the purple dot indicates such a place. The goal is to construct a path that bounds the surface Γ constructed completely from the a - and b -cycles indicated. That such a path exists is not immediately apparent. The bounding path $\partial\tilde{\Gamma}$ will be shown to be

$$\partial\tilde{\Gamma} = a_1 - b_1 - a_1 + b_1 - b_2 + a_2 + b_2 - a_2. \quad (3.12)$$

In what follows this path is deformed to make obvious the fact that it bounds the surface. Note that the blue and red arrows in the figure indicate the orientation of the cycles dictated by the intersection indices in (3.11).

- **Figure 3.5, Page 38:** In the interest of easier visualization, slide the cycles back into positions more like those in Figure 3.3. The paths in the figure indicated in purple may be thought of as part of either the a - or b -cycles. For example, consider the path a_1 to be

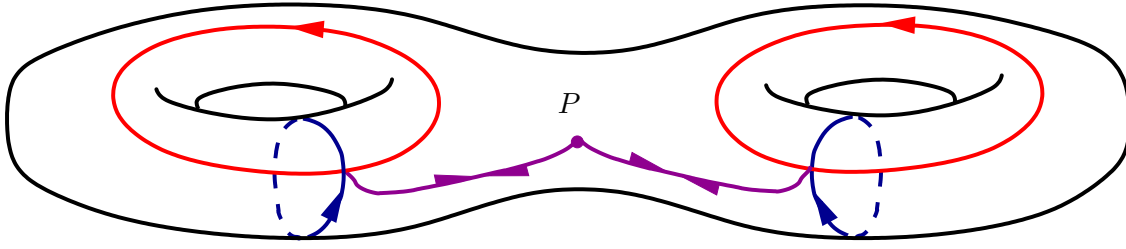


Figure 3.5: For ease of visualization, the cycles are deformed as shown.

1. starting at place P indicated by the purple dot,
2. traversing left along the purple path to the intersection of cycles a_1 and b_1
3. encircling the left hole of Γ along the red path in the sense indicated by the arrow, and finally
4. traversing the purple path from left to right back to place P .

The paths indicated in purple are of no consequence for the purposes of integration as they are each traversed once forwards and once backwards and

$$\int_P^Q = - \int_Q^P.$$

- **Figure 3.6, Page 39:** As the paths indicated in purple in the previous figure do not contribute to the integral, leave them out for the moment. Two small purple dots are indicated at the intersections of the a - and b -cycles as a reminder of these paths.

The surface is cut along the two b -cycles. Note that half-arrows are left on each side of the cut cycles indicating orientation. The cross-hatching in the figure denotes the inside of the surface.

- **Figure 3.7, Page 40:** No magic here, the surface has simply been bent around to make the next few steps easier to see.

Here and in the previous figure there are purple dots on only one end of each of the a -cycles, denoted P_l and P_r for left and right respectively. In order to represent the

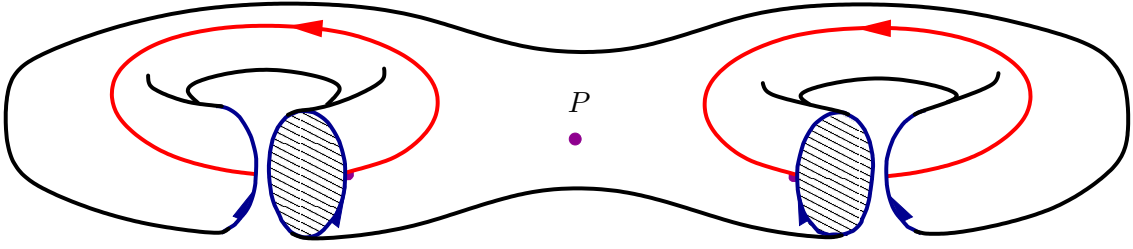


Figure 3.6: The surface is cut open along the b -cycles. The purple paths from the previous figure are omitted for the moment.

path *exactly* as in the previous figure, it would be necessary to also have purple dots indicating paths to opposite ends of the a -cycles. These paths are left out as they do not contribute to the integral and they are unnecessary to the construction of the closed path bounding the surface. To see this, note that the sum of the following sub-paths is closed.

1. the path from P to P_l
2. cycle a_1 traversed from P_l to the intersection with \tilde{b}_1 in the same direction as the cycle is oriented
3. the cycle \tilde{b}_1
4. cycle a_1 traversed back to P_l
5. the path from P_l to P_r
6. cycle a_2 traversed from P_r to the intersection with \tilde{b}_2 in the opposite direction as the cycle is oriented
7. the cycle \tilde{b}_2
8. cycle a_2 traversed back to P_r
9. the path from P_r to P

- **Figure 3.8, Page 40:** Once again there is little change. The a -cycles are deformed from the tops to the insides of the holes in order to make the next step easier to see.

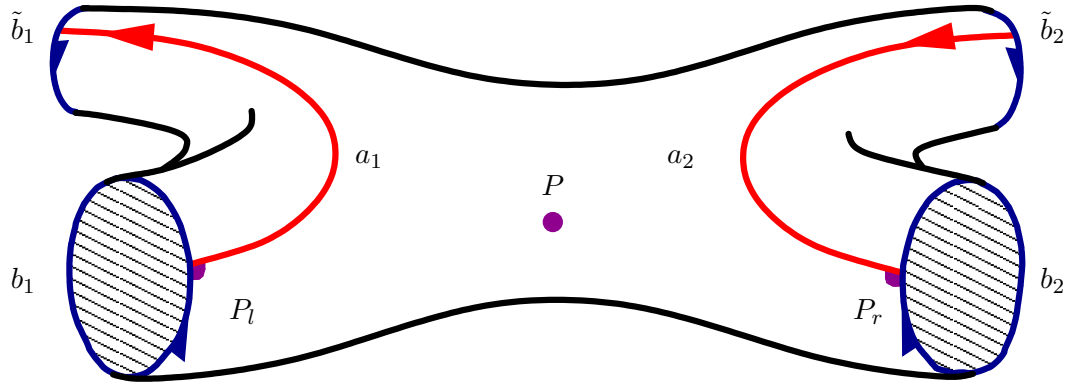


Figure 3.7: The shape of the surface is changed.

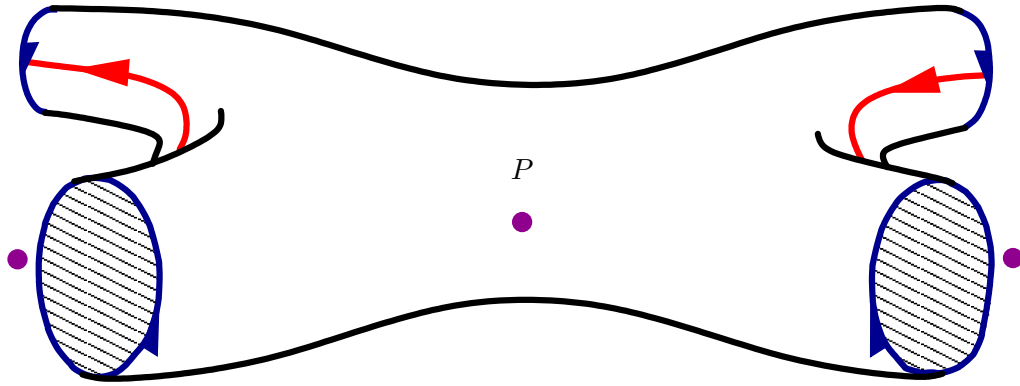


Figure 3.8: The a -cycles are slid from the tops to the insides of the holes.

Two purple dots are shown off the surface to indicate the unseen places P_l and P_r .

- **Figure 3.9, Page 41:** The surface is now cut along the two a -cycles. Again half-arrows are left on the surface to indicate orientation of these cycles. Note that the a -cycles on the upper side of the cuts are in a darker red. This is done for ease of visualization.
- **Figure 3.10, Page 42:** The surface is stretched and bent into the surface of an open-ended rectangular prism. The purple path from P_l to P_r is explicitly represented once

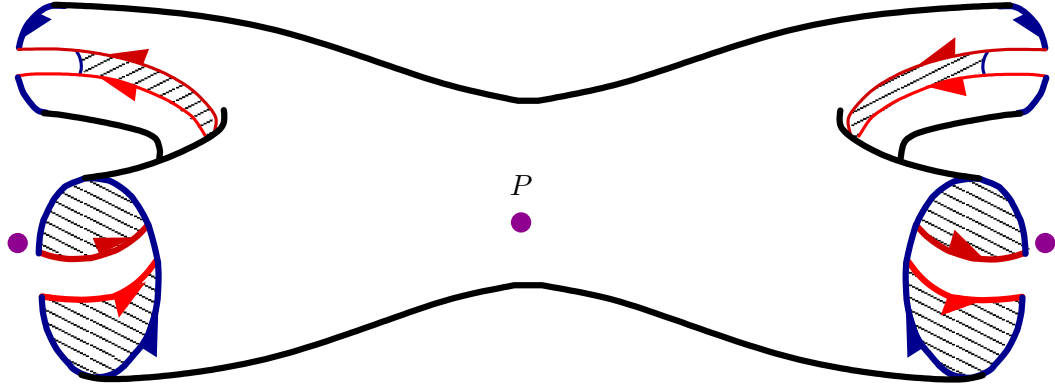


Figure 3.9: The surface is cut open along the a -cycles.

again.

- **Figure 3.11, Page 43:** The surface is cut along the path from P_l to P_r and unfolded. The result is a bounded surface. Finally the (now two) paths from P_l to P_r are shrunk to points, without affecting the integrals of Abelian differentials, resulting in an octagon. This is the bounded surface $\tilde{\Gamma}$, the boundary of which is denoted $\partial\tilde{\Gamma}$. In the right-hand figure the segments of $\partial\tilde{\Gamma}$ are labeled based on which cycle is traversed and in what sense. That is, suppose $\partial\tilde{\Gamma}$ is traversed clockwise starting at

$$P_l = P = P_r$$

indicated by the purple dot on the lower right corner of the octagon. Then the first segment is cycle a_1 traversed in the direction it is oriented. The second segment is cycle b_1 , but traversed against the direction in which it is oriented. Each segment traversed against its orientation is denoted as negative.

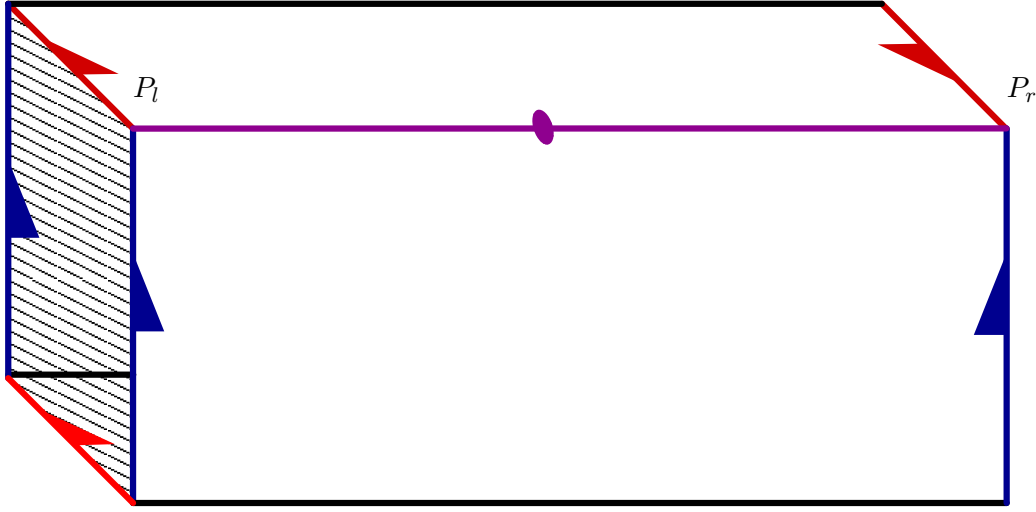


Figure 3.10: The surface is stretched and bent into the surface of an open-ended rectangular prism, and the purple path from P_l to P_r is explicitly represented once again.

3.6 Meromorphic functions and differentials

A function, u , meromorphic on Γ is one that may be locally represented everywhere on Γ as

$$u = \sum_{j=J}^{\infty} c_j t^j, \quad c_j \in \mathbb{C}$$

where t is a local coordinate assured to exist by Newton's Theorem. The field of functions meromorphic on Γ is denoted by Ω_{Γ}^0 , where the superscript zero is meant to denote the field of 0-forms.

A differential, ν , meromorphic on Γ is one that may be locally represented everywhere on Γ as

$$\nu = \sum_{j=J}^{\infty} c_j t^j dt, \quad c_j \in \mathbb{C}$$

The field of differentials meromorphic on Γ is denoted by Ω_{Γ}^1 , where the superscript one is meant to denote the field of 1-forms. The usual definitions of addition and multiplication apply to the fields Ω_{Γ}^0 and Ω_{Γ}^1 .

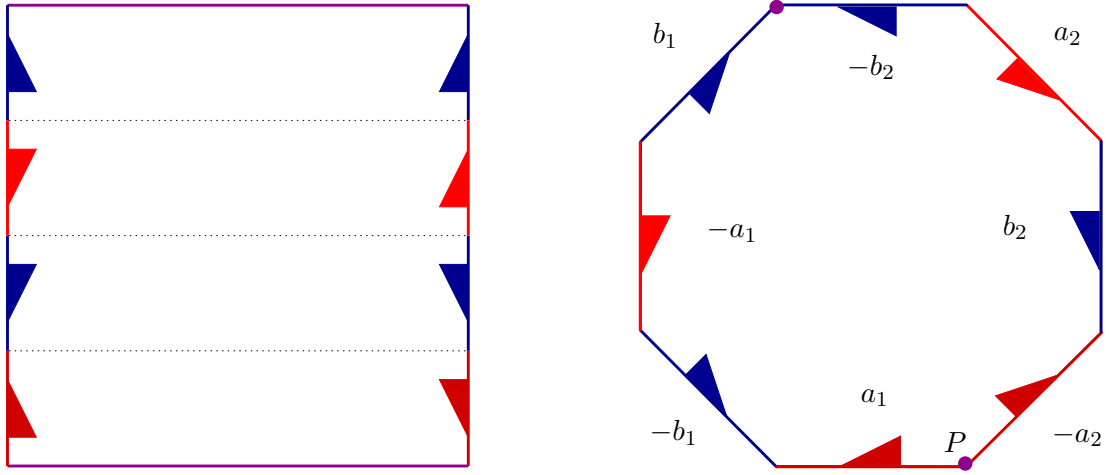


Figure 3.11: The process of constructing the cut surface $\tilde{\Gamma}$ is concluded. On the left the rectangular prism is cut and unfolded. On the right the paths indicated in purple are shrunk to the points P and Q . The labeling of the cycles is discussed in the text.

All meromorphic differentials may be represented, although not uniquely, as sums of three kinds of *Abelian differentials*. Abelian differentials are distinguished by their pole structure.

- Abelian differentials of the first kind are holomorphic differentials, that is, differentials with no poles.
- Abelian differentials of the second kind have a single pole of order greater than one.
- Abelian differentials of the third kind have two simple poles with residues $+1$ and -1 .

3.7 The cohomology of a Riemann surface

A *holomorphic differential* is an Abelian differential of the first kind. That is, a holomorphic differential is an integrand, or 1-form, ω that has no poles on Γ , transforms via the chain

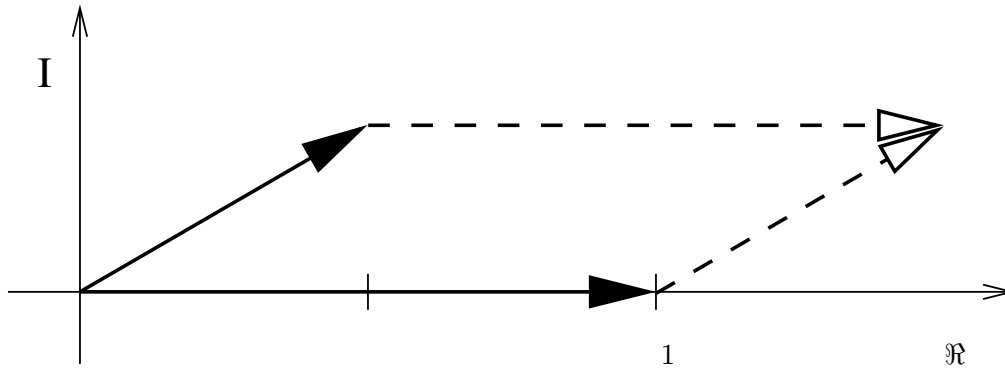


Figure 3.12: The fundamental parallelogram of the approximate lattice associated with the Riemann surface arising from the polynomial $y^3 - yx^2 + 1$. The two solid vectors are approximately (1) and $(0.500 + .289i)$. Any one-component vector, that is any complex number, is a unique linear combination of these two.

rule and is given by

$$\tilde{\omega} = \frac{w(x, y)}{F_y} dx \quad (3.13)$$

$$w(x, y) = \sum_{k+m \leq d-3} c_{km} x^k y^m, \quad (3.14)$$

where d is the degree in x and y of the polynomial giving rise to Γ . Imposing the regularity of $\tilde{\omega}$ at the discriminant points of F constrains $w(x, y)$ such that on a genus g Riemann surface it is possible to construct a g -dimensional basis linearly independent holomorphic differentials

$$(\tilde{\omega}_1, \dots, \tilde{\omega}_g) \quad (3.15)$$

in the form (3.13) [17, 25]. The algorithm presented in [17], and implemented in Maple as `algorithms[differentials]` is heretofore referred to as the Deconinck-van Hoeij differentials (DvHD) algorithm. The algorithm presented here uses the Maple implementation of the DvHD algorithm unchanged.

Example 4. Let Γ be the Riemann surface arising from the polynomial equation

$$\begin{aligned} F(x, y) &= y^2 - 4(x + 1 + i)(x + 1 - i)(x - 2) \\ &= y^2 - 4x^3 + 8x + 16 \\ &= 0. \end{aligned}$$

As

$$F_y = 2y,$$

and the degree of F in x and y is 3, the only candidate for a holomorphic differential is

$$\tilde{\omega} = \frac{1}{2y} dx.$$

At first glance, this differential does not appear holomorphic for $y = 0$. To see that it is, examine the behavior at the branch points

$$x = -1 - i, -1 + i, 2;$$

the only points for which $y = 0$. The place over $-1 - i$ is given by

$$P^{(-1-i)} = \begin{cases} x &= (-1 - i) + (-8 + 24i)t^2 \\ y &= (-8 + 24i)t + (192 - 96i)t^3 + \dots \end{cases}.$$

With this representation, the differential dx is given by

$$dx = 2(-8 + 24i)t dt,$$

and

$$\begin{aligned} \tilde{\omega} \Big|_{P^{(-1-i)}} &= \frac{1}{2} \frac{2(-8 + 24i)t}{(-8 + 24i)t + (192 - 96i)t^3 + \dots} dt \\ &= \frac{1}{1 + (6 + 6i)t^2 + \dots} dt \\ &= (1 - (6 + 6i)t^2 + \mathcal{O}(t^4)) dt. \end{aligned}$$

Thus $\tilde{\omega}$ is regular at

$$x = -1 - i.$$

Similar calculations at $x = -1 + i$ and $x = 2$ show that $\tilde{\omega}$ is regular at places over these points also. To be thorough, also check the behavior at $x = \infty$. Over $x = \infty$, the surface Γ is represented by

$$P^\infty = \begin{cases} x &= \frac{1}{4t^2} \\ y &= \frac{1}{4t^3} - 4t + \dots \end{cases}.$$

Here the differential dx is given by

$$dx = -\frac{1}{2t^3} dt,$$

and $\tilde{\omega}$ has the behavior

$$\begin{aligned} \tilde{\omega}\Big|_{P^\infty} &= -\frac{\frac{1}{2t^3}}{\frac{1}{4t^3} - 4t + \dots} dt \\ &= -\frac{2}{1 - 16t^4 + \dots} dt \\ &= (-2 - 32t^4 + \mathcal{O}(t^8)) dt. \end{aligned}$$

Thus the differential $\tilde{\omega}$ is also regular over $x = \infty$. There are no affine x -points for which $y = \infty$, so there are no other places to check. Thus

$$\left(\frac{dx}{2y}\right)$$

forms a basis for the non-normalized cohomology.

By taking linear combinations of differentials from (3.15), one may choose a g -dimensional basis of *normalized holomorphic differentials* given by

$$\{\omega_1, \dots, \omega_g\} \tag{3.16}$$

$$\oint_{a_k} \omega_j = \delta_{jk}, \tag{3.17}$$

where the a_j are from (3.11). It is common to refer to (3.16) as a basis for the *cohomology* of Γ .

3.8 Riemann matrices and the associated Jacobian

Integration of normalized differentials in (3.16) around the b cycles defines the *Riemann matrix*

$$\mathbf{B} = (\mathbf{B}_{j,k})_{j,k=1}^g \quad (3.18)$$

$$\mathbf{B}_{j,k} = \oint_{b_k} \omega_j. \quad (3.19)$$

The Riemann bilinear relations assure that the Riemann matrix is invertible, symmetric, and has positive definite imaginary part [17, 18].

The $(g \times 2g)$ matrix $(\mathbf{1} \ \mathbf{B})$, where $\mathbf{1}$ is the $(g \times g)$ identity matrix, contains all the information about integrating holomorphic differentials around closed paths and is called the *normalized period matrix* of Γ . The columns of the period matrix are linearly independent [18] and describe a lattice Λ inside the vector space \mathbb{C}^g given by

$$\Lambda = \{\mathbf{V} : \mathbf{V} = \mathbf{1}\mathbf{M} + \mathbf{B}\mathbf{N}; \ \mathbf{M}, \mathbf{N} \in \mathbb{Z}^g\}. \quad (3.20)$$

Before continuing, define the following operations on a vector

$$\mathbf{V} \in \mathbb{R}^g, \quad \mathbf{V} = (V_1, \dots, V_g)$$

that are used to make explicit the definition of the Jacobian of a Riemann surface.

- **Floor:** The *component-wise floor* of \mathbf{V} , denoted by $\lfloor \mathbf{V} \rfloor$ is given by

$$\lfloor \mathbf{V} \rfloor = (\lfloor V_1 \rfloor, \dots, \lfloor V_g \rfloor),$$

where the floor $\lfloor V_j \rfloor$ of $V_j \in \mathbb{R}$ is the greatest integer less than or equal to V_j .

- **Fractional part:** The *component-wise fractional part* of $\mathbf{V} \in \mathbb{R}^g$, denoted by $\llbracket \mathbf{V} \rrbracket$ is given by

$$\llbracket \mathbf{V} \rrbracket = (\llbracket V_1 \rrbracket, \dots, \llbracket V_g \rrbracket),$$

where

$$\llbracket V_j \rrbracket = V_j - \lfloor V_j \rfloor.$$

Note that each component

$$\llbracket V_j \rrbracket \in [0, 1)$$

by construction.

Any finite dimensional vector \mathbf{V} with real components has the unique decomposition

$$\mathbf{V} = \lfloor \mathbf{V} \rfloor + \llbracket \mathbf{V} \rrbracket.$$

As the matrix \mathbf{B} is non-singular, any vector $\mathbf{V} \in \mathbb{C}^g$ has a unique decomposition $(\mathbf{V}', \mathbf{V}'')^t$, where superscript t denotes transposition, into the columns of $(\mathbf{1} \ \mathbf{B})$. That is, there exists a $(\mathbf{V}', \mathbf{V}'')^t$ specified by the equation

$$\begin{pmatrix} \Re(\mathbf{V}) \\ \Im(\mathbf{V}) \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \Re(\mathbf{B}) \\ \mathbf{0} & \Im(\mathbf{B}) \end{pmatrix} \begin{pmatrix} \mathbf{V}' \\ \mathbf{V}'' \end{pmatrix}$$

where $\mathbf{0}$ is the $(g \times g)$ matrix of zeros. The $2g$ component vector $(\mathbf{V}', \mathbf{V}'')^t$ can clearly be written as

$$\begin{pmatrix} \mathbf{V}' \\ \mathbf{V}'' \end{pmatrix} = \begin{pmatrix} \lfloor \mathbf{V}' \rfloor \\ \lfloor \mathbf{V}'' \rfloor \end{pmatrix} + \begin{pmatrix} \llbracket \mathbf{V}' \rrbracket \\ \llbracket \mathbf{V}'' \rrbracket \end{pmatrix}.$$

Thus, given a Riemann matrix \mathbf{B} , any vector $\mathbf{V} \in \mathbb{C}^g$ is given by

$$\begin{pmatrix} \Re(\mathbf{V}) \\ \Im(\mathbf{V}) \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \Re(\mathbf{B}) \\ \mathbf{0} & \Im(\mathbf{B}) \end{pmatrix} \left(\begin{pmatrix} \lfloor \mathbf{V}' \rfloor \\ \lfloor \mathbf{V}'' \rfloor \end{pmatrix} + \begin{pmatrix} \llbracket \mathbf{V}' \rrbracket \\ \llbracket \mathbf{V}'' \rrbracket \end{pmatrix} \right).$$

Denote by $\lfloor \mathbf{V} \rfloor_\Lambda$ the vector given by

$$\begin{pmatrix} \Re(\lfloor \mathbf{V} \rfloor_\Lambda) \\ \Im(\lfloor \mathbf{V} \rfloor_\Lambda) \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \Re(\mathbf{B}) \\ \mathbf{0} & \Im(\mathbf{B}) \end{pmatrix} \begin{pmatrix} \lfloor \mathbf{V}' \rfloor \\ \lfloor \mathbf{V}'' \rfloor \end{pmatrix},$$

that is,

$$\begin{aligned} \lfloor \mathbf{V} \rfloor_\Lambda &= \lfloor \mathbf{V}' \rfloor + \Re(\mathbf{B}) \lfloor \mathbf{V}'' \rfloor + i \Im(\mathbf{B}) \lfloor \mathbf{V}'' \rfloor \\ &= \lfloor \mathbf{V}' \rfloor + \mathbf{B} \lfloor \mathbf{V}'' \rfloor. \end{aligned}$$

Then $\lfloor \mathbf{V} \rfloor_\Lambda$ is a vector in the lattice Λ . Further, denote by $\llbracket \mathbf{V} \rrbracket_\Lambda$ the vector given by

$$\begin{pmatrix} \Re(\llbracket \mathbf{V} \rrbracket_\Lambda) \\ \Im(\llbracket \mathbf{V} \rrbracket_\Lambda) \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \Re(\mathbf{B}) \\ \mathbf{0} & \Im(\mathbf{B}) \end{pmatrix} \begin{pmatrix} \llbracket \mathbf{V}' \rrbracket \\ \llbracket \mathbf{V}'' \rrbracket \end{pmatrix},$$

that is,

$$\begin{aligned} \llbracket \mathbf{V} \rrbracket_\Lambda &= \llbracket \mathbf{V}' \rrbracket + \Re(\mathbf{B})\llbracket \mathbf{V}'' \rrbracket + i\Im(\mathbf{B})\llbracket \mathbf{V}'' \rrbracket \\ &= \llbracket \mathbf{V}' \rrbracket + \mathbf{B}\llbracket \mathbf{V}'' \rrbracket. \end{aligned}$$

Summarizing, any $\mathbf{V} \in \mathbb{C}^g$ may be written as

$$\mathbf{V} = \lfloor \mathbf{V} \rfloor_\Lambda + \llbracket \mathbf{V} \rrbracket_\Lambda, \quad (3.21)$$

with $\lfloor \mathbf{V} \rfloor_\Lambda$ and $\llbracket \mathbf{V} \rrbracket_\Lambda$ as defined above. The vector $\lfloor \mathbf{V} \rfloor_\Lambda \in \Lambda$ and is the representative of \mathbf{V} in the fundamental parallelogram of Λ . The fundamental parallelogram is the quotient space

$$J(\Gamma) = \mathbb{C}^g / \Lambda$$

known as *Jacobian* of Γ . The Jacobian is topologically a $2g$ -dimensional torus [18, 25]. The notation

$$\mathbf{V} \equiv \mathbf{V}'$$

is consistently used to indicate that

$$\mathbf{V} - \mathbf{V}' \in \Lambda,$$

that is if the vectors \mathbf{V} and \mathbf{V}' have the same representative in the fundamental parallelogram.

Note that choosing a different basis of cycles \mathcal{H}' results in a different Riemann matrix \mathbf{B}' , however \mathbf{B} and \mathbf{B}' are related by a symplectic transformation, and the Jacobians induced by both are identical [18, 25].

Example 5. Suppose the genus one Riemann surface Γ arises from the polynomial

$$F = y^3 - yx^2 + 1.$$

Any holomorphic differential on Γ can be represented as

$$\tilde{\omega} = \frac{c}{-3y^2 + x^2} dx, \quad c \in \mathbb{C}.$$

The Maple package `algcurves` can be used to compute a homology basis $\{a, b\}$ for Γ . Computing numerically, again with Maple,

$$\int_a \frac{1}{-3y^2 + x^2} dx \approx -2.103 - 3.643i.$$

Thus a normalized differential is approximately given by

$$\omega \approx \frac{-0.119 + 0.206i}{-3y^2 + x^2} dx,$$

and the one-component Riemann matrix is given by

$$B_{1,1} = \int_b \frac{-0.119 + 0.206i}{-3y^2 + x^2} dx \approx 0.500 + 0.289i.$$

Figure 3.12 shows the fundamental parallelogram of the approximate lattice that arises from this example. Identifying the two sets of parallel sides gives rise to a torus.

Note that an algorithm to explicitly compute the normalized period matrix of a Riemann surface associated with a plane algebraic curve was developed in [17]. The algorithms presented in this work make use of that previously developed algorithm.

3.9 Definition and properties of divisors

A *divisor* on a Riemann surface Γ is a set of places with multiplicities. A divisor \mathcal{D} is written as

$$\mathcal{D} = \sum_j p_j P_j,$$

where $p_j P_j$ denotes that the place P_j has multiplicity p_j . Divisors may be negated,

$$-\mathcal{D} = \sum_j (-p_j) P_j,$$

and added

$$\begin{aligned}\mathcal{D}' &= \sum_j p'_j P'_j, \\ (\mathcal{D} + \mathcal{D}') &= \sum_j p_j P_j + \sum_j p'_j P'_j.\end{aligned}$$

The *degree* of a divisor is the sum of the multiplicities, denoted

$$\deg \mathcal{D} = \sum_j p_j.$$

A divisor is called *effective* if all the multiplicities are positive. That a divisor \mathcal{D} is effective is denoted by $\mathcal{D} \geq 0$.

A meromorphic function u on Γ that has zeros at the places

$$P_1, \dots, P_m$$

with multiplicities

$$p_1, \dots, p_m$$

and poles at the places

$$Q_1, \dots, Q_n$$

with multiplicities

$$q_1, \dots, q_n$$

defines a *valuation divisor*

$$(u)_{\text{val}} = \sum_j p_j P_j - \sum_j q_j Q_j.$$

Similarly, the divisor

$$(\nu)_{\text{val}}$$

of a differential ν that has zeros at the places

$$P_1, \dots, P_m$$

with multiplicities

$$p_1, \dots, p_m$$

and poles at the places

$$Q_1, \dots, Q_n$$

with multiplicities

$$q_1, \dots, q_n$$

is given by

$$(\nu)_{\text{val}} = \sum_{j=1}^m p_j P_j - \sum_{j=1}^n q_j Q_j.$$

Two divisors \mathcal{D} and \mathcal{D}' are *linearly equivalent*, denoted

$$\mathcal{D} \sim \mathcal{D}',$$

if

$$\mathcal{D} - \mathcal{D}'$$

is the divisor of a meromorphic function [18, 25]. Section 3.23 will show the reason for this particular equivalence. If ν and ν' are Abelian differentials, then

$$\begin{aligned} (\nu)_{\text{val}} - (\nu')_{\text{val}} &= \sum_j p_j P_j - \sum_j q_j Q_j - \left(\sum_j p'_j P'_j - \sum_j q'_j Q'_j \right) \\ &= \left(\frac{\nu}{\nu'} \right)_{\text{val}}, \end{aligned}$$

and ν/ν' is a meromorphic function on Γ as the differential dt cancels locally. Thus the divisors of any two meromorphic differentials differ by the divisor of a meromorphic function, and are therefore linearly equivalent. The equivalence class of divisors of meromorphic differentials is known as the *canonical class* [21].

The final definition of this section will be used when discussing the vector of Riemann constants. The set of all effective divisors with d places is known as the d -th *symmetric power* of Γ , is denoted by ST^d and is itself a complex manifold [25].

Example 6. Computing two divisors in the canonical class: *Let Γ be the Riemann surface arising from the polynomial equation*

$$F = y^2 - (x^2 - 1)(x^2 - 4)(x^2 - 9) = 0.$$

A basis for the holomorphic differentials on Γ is

$$(\omega_1, \omega_2) = \left(\frac{dx}{y}, \frac{x dx}{y} \right).$$

As these differentials are holomorphic, they have no poles. Thus any divisor

$$(c_1\omega_1 + c_2\omega_2)_{\text{val}} = \left(\frac{c_1 + c_2x}{y} dx \right)_{\text{val}},$$

where

$$c_1, c_2 \in \mathbb{C},$$

is effective. Over regular points

$$dx = dt,$$

and thus the only way for a regular place to yield zero is if

$$x = 0, c_2 = 0,$$

or

$$x = -\frac{c_1}{c_2}, c_2 \neq 0.$$

Over $x = 0$, the surface is given by

$$P_{\pm}^0 = \begin{cases} x & = x \\ y & = \pm 6i + \dots \end{cases}.$$

Substituting into the differential gives

$$\pm \frac{c_1 + c_2x}{6i} dx \pm \mathcal{O}(x^2) dx,$$

showing that if $c_1 = 0$, then both places over $x = 0$ appear in the valuation divisor.

Over regular points

$$x = \alpha := -\frac{c_1}{c_2},$$

y is given by

$$y = \pm \sqrt{(\alpha^2 - 1)(\alpha^2 - 4)(\alpha^2 - 9) + \dots}.$$

Over the branch points

$$x = \alpha := -\frac{c_1}{c_2} = \pm 1, \pm 2, \pm 3,$$

the surface is given by

$$P^\alpha = \begin{cases} x &= \alpha + t^2 \\ y &= t + \dots \end{cases},$$

and the differential dx is given by

$$dx = 2t \, dt.$$

For these places

$$\begin{aligned} c_1\omega_1 + c_2\omega_2 &= \frac{c_1 + c_2(\alpha + t^2)}{t + \dots} 2t \, dt \\ &= \frac{2c_2t^2}{1 + \dots} dt, \end{aligned}$$

showing that if

$$\alpha := -c_1/c_2 = \pm 1, \pm 2, \pm 3,$$

then P^α appears in the valuation divisor with multiplicity two.

Lastly, the behavior of the differentials at ∞ must be examined. There are two places over ∞ , given by the Puiseux representations

$$P_\pm^\infty = \begin{cases} x &= \frac{1}{t} \\ y &= \pm \frac{1}{t^3} + \dots \end{cases}.$$

The differential dt is, at both places,

$$dx = -\frac{1}{t^2} dt.$$

Substituting into $c_1\omega_1 + c_2\omega_2$ gives

$$c_1\omega_1 + c_2\omega_2 = \mp \frac{1}{2}(c_1t + c_2) dt \pm \mathcal{O}(t^2) dt,$$

so if $c_2 \neq 0$ then neither place over ∞ is a zero. However, if $c_2 = 0$ and $c_1 \neq 0$ then both places over ∞ appear in the valuation divisor, each with multiplicity one.

Summarizing, the valuation divisor

$$(c_1\omega_1 + c_2\omega_2)_{\text{val}}$$

depends on the constants c_1 and c_2 .

- If

$$\alpha := -\frac{c_1}{c_2} \neq \pm 1, \pm 2, \pm 3,$$

then

$$(c_1\omega_1 + c_2\omega_2)_{\text{val}} = P_+^\alpha + P_-^\alpha.$$

- If

$$\alpha := -\frac{c_1}{c_2} = \pm 1, \pm 2, \pm 3,$$

then

$$(c_1\omega_1 + c_2\omega_2)_{\text{val}} = 2P^\alpha.$$

Note that these divisors are of degree $2g-2 = 2$. More will be said about this in Section 3.10.

3.10 The Riemann-Roch Theorem

The Riemann-Roch Theorem is one of the main results in the theory of algebraic curves. This theorem, in essence about the nature of the space of functions with prescribed poles on a Riemann surface, is used in Section 3.15 to prove an essential property of the vector of Riemann constants. Some preliminary definitions are needed. In these definitions the divisor \mathcal{D} is given by

$$\mathcal{D} = \sum_j p_j P_j - \sum_j p'_j P'_j$$

where the integers

$$p_j, p'_j \in \mathbb{Z}$$

are all positive and non-zero, and the places P_j, P'_j are all on the Riemann surface Γ .

- **Riemann-Roch space $\mathcal{L}^0(\mathcal{D})$:** The space $\mathcal{L}^0(\mathcal{D})$ is given by

$$\mathcal{L}^0(\mathcal{D}) = \{u \in \Omega_\Gamma^0 : (u)_{\text{val}} + \mathcal{D} \geq 0\}.$$

Thus functions in $\mathcal{L}^0(\mathcal{D})$

- *may* have poles only at places P_j with at most multiplicity p_j , and
- *must* have zeroes at places P'_j with at least multiplicity p'_j .

more colloquially, $\mathcal{L}^0(\mathcal{D})$ can be thought of as the space of functions that make functions with valuation divisor \mathcal{D} well-behaved. That is, suppose that \mathcal{D} is the valuation divisor of the meromorphic function v , then for all functions

$$u \in \mathcal{L}^0(\mathcal{D}),$$

the function vu has no poles. The dimension of the linear space $\mathcal{L}^0(\mathcal{D})$ is denoted $l^0(\mathcal{D})$.

More colloquially still, if

$$\mathcal{D} = p_1 P_1 + p_2 P_2 + \cdots$$

is an effective divisor, then the space $\mathcal{L}^0(\mathcal{D})$ consists of all functions u satisfying the following.

- At any place

$$P \notin P_1, P_2, \dots,$$

u is given by

$$u = t^k h(t), \quad h(t) \neq 0,$$

where $h(t)$ is holomorphic, but k may be greater than one.

- At any place

$$P_j \in P_1, P_2, \dots,$$

u is given by

$$u = t^k h(t), \quad h(t) \neq 0,$$

where $h(t)$ is holomorphic, and $k \geq p_j$.

- **Differential space $\mathcal{L}^1(\mathcal{D})$:** The space $\mathcal{L}^1(\mathcal{D})$ is given by

$$\mathcal{L}^1(\mathcal{D}) = \{\omega \in \Omega_\Gamma^1 : (\omega)_{\text{val}} - \mathcal{D} \geq 0\}.$$

Thus differentials in $\mathcal{L}^1(\mathcal{D})$

- *must* have zeroes at places P_j with at least multiplicity p_j , and
- *may* have poles only at places P'_j with at most multiplicity p'_j .

Thus, $\mathcal{L}^1(\mathcal{D})$ can be thought of as the space of differentials that make differentials with valuation divisor $-\mathcal{D}$ well-behaved. That is, suppose that $-\mathcal{D}$ is the valuation divisor of the meromorphic differential ν , then for all differentials $\omega \in \mathcal{L}^1(\mathcal{D})$, the differential $\omega\nu$ has no poles. The dimension of the linear space $\mathcal{L}^1(\mathcal{D})$ is denoted $l^1(\mathcal{D})$.

As with the notation for meromorphic functions and differentials, the superscripts on \mathcal{L}^0 and \mathcal{L}^1 are meant to denote linear function spaces of 0-forms (functions) and 1-forms (differentials) respectively. The integers $l^0(\mathcal{D})$ and $l^1(\mathcal{D})$ depend only on the equivalence class of the divisor \mathcal{D} . For proof of this, see Appendix A.3.

Proof of the Riemann-Roch Theorem requires much machinery, and may be found in any standard text on Riemann surfaces or algebraic curves. See, for instance [36, 25, 21].

Riemann-Roch Theorem. *Suppose Γ is a genus g Riemann surface, and*

$$\mathcal{D} = \sum_j p_j P_j$$

is a divisor with all the places $P_j \in \Gamma$, and all the multiplicities $p_j \in \mathbb{Z}$. Then

$$l^0(\mathcal{D}) = \deg \mathcal{D} - g + l^1(\mathcal{D}) + 1.$$

As an example, I use the Riemann-Roch Theorem to calculate the degree of $(\nu)_{\text{val}}$ for a general meromorphic differential ν . This example follows the proof of Theorem 10-11 in [36].

Example 7. *In this example the Riemann-Roch Theorem is used to show that the degree of the divisor of a meromorphic function is $2g - 2$. Let Γ be a genus $g > 0$ Riemann surface,*

$$\omega_1, \dots, \omega_g$$

be a basis for the cohomology on Γ , and

$$\mathcal{D}_1 = (\omega_1)_{\text{val}}.$$

Riemann-Roch gives

$$l^0(\mathcal{D}_1) = \deg \mathcal{D}_1 - g + l^1(\mathcal{D}_1) + 1.$$

Suppose ν is any other Abelian differential in $\mathcal{L}^1(\mathcal{D}_1)$:

$$(\nu)_{\text{val}} - \mathcal{D}_1 \geq 0,$$

Then, by definition, the function $\omega_1\nu$ has no singularities and so by Liouville's Theorem must be a constant. Thus

$$\omega_1 = c\nu$$

for some $c \in \mathbb{C}$. Therefore, up to the choice of the constant c , ω_1 is the only Abelian differential with the correct zero structure, that is,

$$l^1(\mathcal{D}_1) = 1.$$

To calculate $l^0(\mathcal{D}_1)$, first note that the g functions

$$\frac{\omega_1}{\omega_1}, \dots, \frac{\omega_g}{\omega_1}$$

are linearly independent meromorphic functions. As ω_j is holomorphic it has no negative multiplicities, thus

$$(\omega_j)_{\text{val}} \geq 0.$$

Therefore

$$\begin{aligned} \left(\frac{\omega_j}{\omega_1}\right)_{\text{val}} &= (\omega_j)_{\text{val}} - (\omega_1)_{\text{val}} \geq -(\omega_1)_{\text{val}} \quad j = 1, \dots, g, \\ &= -(\mathcal{D}_1)_{\text{val}}. \end{aligned}$$

that is, each function ω_j/ω_1 is in the linear space $\mathcal{L}^0(D_1)$, and thus

$$l^0(\mathcal{D}_1) \geq g.$$

To see that these are the only functions in $\mathcal{L}^0(\mathcal{D}_1)$, and thus that

$$l^0(\mathcal{D}_1) = g,$$

first assume the contrary. Let \tilde{u} be a meromorphic function such that

$$\begin{aligned} \tilde{u} &\in \mathcal{L}^0(\mathcal{D}_1), \\ \tilde{u} &\notin \text{span} \left\{ \frac{\omega_1}{\omega_1}, \dots, \frac{\omega_g}{\omega_1} \right\}. \end{aligned} \tag{3.22}$$

As \tilde{u} is in

$$\mathcal{L}^0(\mathcal{D}_1) = \mathcal{L}^0((\omega_1)_{\text{val}}),$$

the differential $\tilde{u}\omega_1$ is holomorphic, and therefore

$$\tilde{u}\omega_1 = \sum_j c_j \omega_j,$$

thus as $\omega_1 \neq 0$

$$\tilde{u} = \sum_j c_j \frac{\omega_j}{\omega_1},$$

violating the assumption (3.22). By Riemann-Roch

$$g = \deg \mathcal{D}_1 - g + 1 + 1,$$

and rearranging gives

$$\deg \mathcal{D}_1 = 2g - 2.$$

All Abelian differentials are linearly equivalent and thus in the same equivalence class. The integers l^0 and l^1 are only dependent on equivalence class, thus

$$\deg(\nu)_{\text{val}} = 2g - 2$$

for any $\nu \in \Omega_{\Gamma}^1$.

Note that the result of Example 7 establishes that any holomorphic differential, the divisor of which is effective, has exactly $2g - 2$ zeros counting multiplicities.

The following proposition, which relies on the Riemann-Roch Theorem is used in Section 3.15. Proof can be found in [21].

Proposition 3.10.1. *Given an effective divisor \mathcal{D} , if $s - 1$ places in \mathcal{D} may be chosen arbitrarily then $l^0(\mathcal{D}) \geq s$.*

3.11 Special divisors

An effective divisor \mathcal{D} is called *special* if there exists an effective divisor \mathcal{D}' such that

$$\mathcal{D} + \mathcal{D}' \in \mathcal{C},$$

where \mathcal{C} is the canonical class of divisors [21]. Notice that this means that if

$$\mathcal{C} = P_1 + \cdots + P_{2g-2}$$

the divisor of a holomorphic differential then *any* sub-sum of the places in \mathcal{C} is a special divisor.

For any effective divisor, the following three properties hold [21].

1. \mathcal{D} is special if and only if

$$l^1(\mathcal{D}) > 0.$$

2. \mathcal{D} is special if

$$\deg \mathcal{D} \leq g - 1.$$

3. if $\deg \mathcal{D} = g$ then \mathcal{D} is special if and only if

$$l^0(\mathcal{D}) \geq 2,$$

that is, if and only if $\mathcal{L}^0(\mathcal{D})$ contains a non-constant function.

3.12 The Abel map

The Abel map of a place P on a genus g Riemann surface is defined by

$$\mathbf{A} = (A_1, \dots, A_g) : \tag{3.23}$$

$$A_j(P_0, P) = \int_{P_0}^P \omega_j, \tag{3.24}$$

where P_0 is a fixed place on the Riemann surface, the ω_j are normalized as in (3.16) and the path of integration from P_0 to P is the same for each index j . Equation (3.23) can be written in vector form as

$$\mathbf{A} : \Gamma \mapsto J(\Gamma)$$

$$P \mapsto \mathbf{A}(P_0, P) = \int_{P_0}^P \boldsymbol{\omega}$$

where

$$\boldsymbol{\omega} = (\omega_1, \dots, \omega_g),$$

is a normalized basis for the cohomology of Γ .

Given any two paths γ and η with the same initial place and final place, the difference between γ and η is a closed path, and therefore a linear combination of cycles in the homology basis \mathcal{H} . That is,

$$\gamma - \eta = \sum_{j=1}^g m_j a_j + \sum_{j=1}^g n_j b_j$$

where m_i, n_i are integers and

$$a_i, b_i \in \mathcal{H}.$$

Thus the Abel map of a place P is well defined in that its image is a unique element of $J(\Gamma)$.

The definition of the Abel map is extended to divisors \mathcal{D} . Written in vector notation:

$$\mathbf{A}(P_0, \mathcal{D}) = \sum_i p_i \mathbf{A}(P_0, P_i).$$

Note that the Abel map of a degree zero divisor is independent of the initial point P_0 . This is seen by writing \mathcal{D} as

$$\mathcal{D} = \sum_{j=1}^m (P_j - Q_j)$$

where places may appear multiple times. The Abel map is then

$$\begin{aligned} A(P_0, \mathcal{D}) &= \sum_{j=1}^m \left(\int_{P_0}^{P_j} \omega - \int_{P_0}^{Q_j} \omega \right) \\ &= \sum_{j=1}^m \int_{Q_j}^{P_j} \omega. \end{aligned}$$

By applying the Residue Theorem appropriate for Riemann surfaces to the differential

$$\frac{du}{u},$$

it can be shown that, for any $u \in \Omega_{\Gamma}^0$,

$$\deg(u)_{\text{val}} = 0.$$

This shows that the Abel map of the divisor of a meromorphic function is independent of initial place P_0 .

Abel's theorem sheds more light on what it means for divisors to be linearly equivalent. This phrasing of Abel's Theorem is from [25].

Abel's Theorem.

- *For any meromorphic function u on Γ with*

$$(u)_{\text{val}} = \mathcal{D},$$

we have

$$A(P_0, \mathcal{D}) \equiv 0.$$

- *Given a degree zero divisor \mathcal{D} such that*

$$A(P_0, \mathcal{D}) \equiv 0,$$

there exists a meromorphic function u such that $(u)_{\text{val}} = \mathcal{D}$.

A direct consequence of Abel's Theorem is that two divisors \mathcal{D} and \mathcal{D}' differ by the divisor of a meromorphic function if and only if [25]

$$\deg \mathcal{D} = \deg \mathcal{D}'$$

and

$$A(\mathcal{D}) \equiv A(\mathcal{D}').$$

Thus linearly equivalent divisors map to the same vector on $J(\Gamma)$ under the Abel map.

3.13 The Riemann θ -function

I now introduce more thoroughly the Riemann θ -function mentioned in the introduction. This function is the basic building block of quasiperiodic solutions of integrable systems [6], and is used in the algorithm I present to compute the vector of Riemann constants (VRC). Moreover, the θ -function is a standard tool for examining and understanding the structure of the Jacobian $J(\Gamma)$ of a Riemann surface Γ and the relationship between the two. More detail can be found in the standard references [5, 21, 32, 33, 22].

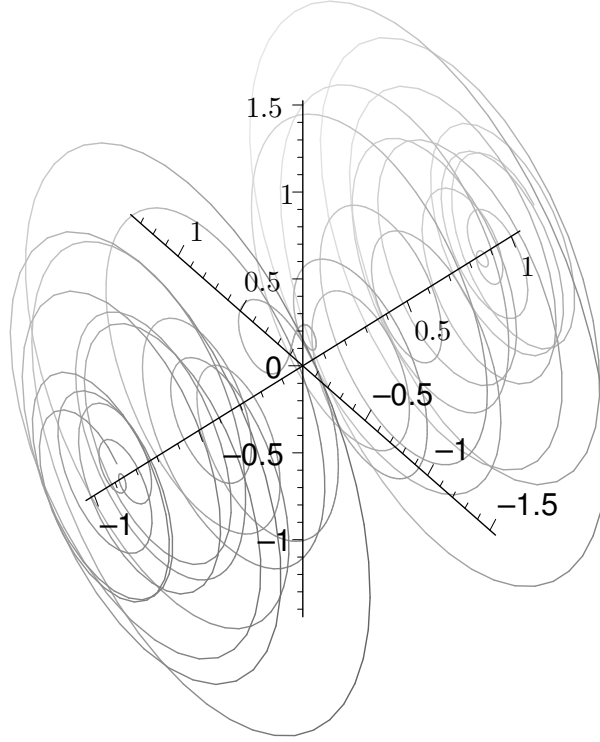


Figure 3.13: This graph shows a “slice” of the θ -function associated with the Riemann surface defined by $F = y^2 - (x^2 - 1)(x^2 - 2)(x^2 - 3)(x^2 - 4)(x^2 - 5) = 0$. In this figure the x -axis goes from the lower left to the upper right. The complex function $\theta(\mathbf{V}x|\mathbf{B})$, where \mathbf{V} is a 4 component vector and \mathbf{B} is the 4×4 Riemann matrix associated with F , is plotted in the plane parallel to the y, z -plane for each value of x from -1 to 1 .

The Riemann θ -function

$$\theta : \mathbb{C}^g \mapsto \mathbb{C} \quad (3.25)$$

$$z \mapsto \theta(z|\mathbf{B}) = \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i \left(\frac{1}{2} \mathbf{n} \cdot \mathbf{B} \mathbf{n} + z \cdot \mathbf{n} \right)}, \quad (3.26)$$

is parameterized by a $g \times g$ Riemann matrix \mathbf{B} [14]. Here

$$\mathbf{u} \cdot \mathbf{v} = \sum_j u_j v_j$$

is the scalar product of the vectors \mathbf{u} and \mathbf{v} . The summation in (3.26) is over all vectors with g integer components. As $\Im(\mathbf{B})$ is positive definite, terms of (3.26) for which $\|\mathbf{n}\|$ is

large are very small. Thus the θ -function is analytic in \mathbb{C}^g [14, 18]. When the matrix \mathbf{B} is fixed and understood from context, I will write the θ -function simply as $\theta(\mathbf{z})$.

Recall that a function $\psi : \mathbb{C} \mapsto \mathbb{C}$ is quasiperiodic¹ with period w if

$$\psi(z + w) = e^{(c+c'z)}\psi(z)$$

for a suitable choice of $c, c' \in \mathbb{C}$. In the special case that

$$c, c' = 0,$$

the function ψ is periodic. From (3.26) it is clear that the θ -function is periodic with integer periods. That is,

$$\theta(\mathbf{z} + \mathbf{e}_j | \mathbf{B}) = \theta(\mathbf{z} | \mathbf{B}), \quad (3.27)$$

where

$$(\mathbf{e}_j)_k = \delta_{j,k}.$$

Less clear is that

$$\theta(\mathbf{z} + \mathbf{B}\mathbf{e}_j | \mathbf{B}) = e^{2\pi i(-\frac{1}{2}\mathbf{B}_{jj}-\mathbf{z}_j)}\theta(\mathbf{z} | \mathbf{B}), \quad (3.28)$$

and thus that the θ -function is quasiperiodic in the columns of \mathbf{B} . Equation (3.28) can be seen by changing summation index to $\mathbf{m} = \mathbf{n} + \mathbf{e}_j$ as in [18].

Let the definition of (quasi)periodic functions be extended to functions of vector arguments as follows. Given a function $\psi : \mathbb{C}^g \mapsto \mathbb{C}$,

- $\psi(\mathbf{z})$ is periodic with period $\mathbf{w} \in \mathbb{C}^g$ if

$$\psi(\mathbf{z} + \mathbf{w}) = \psi(\mathbf{z}).$$

- $\psi(\mathbf{z})$ is quasiperiodic with quasiperiod $\mathbf{w} \in \mathbb{C}^g$ if, for suitable choices of $\mathbf{c}, \mathbf{c}' \in \mathbb{C}^g$,

$$\psi(\mathbf{z} + \mathbf{w}) = e^{(\mathbf{c} \cdot \mathbf{w} + \mathbf{c}' \cdot \mathbf{z})}\psi(\mathbf{z}).$$

¹Note this is not the *only* definition of a quasiperiodic function. A function ψ is also called quasiperiodic with period w if $\psi(z + w) = \psi(z) + c + c'z$.

The (quasi)periodicity of $\theta(z|\mathbf{B})$ is summarized by the transformation law

$$\theta(z + M + \mathbf{B}N|\mathbf{B}) = e^{2\pi i(-\frac{1}{2}N \cdot \mathbf{B}N - N \cdot z)} \theta(z|\mathbf{B}),$$

for $M, N \in \mathbb{Z}^g$.

3.14 The Jacobi inversion problem

The Abel map \mathbf{A} introduced in Section 3.12 maps a Riemann surface Γ into its Jacobian $J(\Gamma)$. The task of constructing a map

$$\mathbf{A}^{-1} : J(\Gamma) \mapsto S\Gamma^g$$

from the Jacobian back to a set of g places on the Riemann surface is known as the Jacobi inversion problem, and may be stated as follows. Given a vector $\zeta \in J(\Gamma)$, find a divisor $\mathcal{D} \in S\Gamma^g$ such that

$$\mathbf{A}(P_0, \mathcal{D}) \equiv \zeta,$$

where the initial place P_0 is considered a fixed place not on any of the cycles in \mathcal{H} for the remainder of the section. The notation will be adjusted accordingly for the sake of brevity.

The following results rely on integration on the cut surface $\tilde{\Gamma}$ and the boundary $\partial\tilde{\Gamma}$ of that surface. For the purposes of these integrations let \mathcal{A} be a version of the Abel map appropriate to $\tilde{\Gamma}$. That is, let \mathcal{A} be given by

$$\mathcal{A} : \tilde{\Gamma} \mapsto \mathbb{C}^g,$$

$$\mathcal{A}(P) = \int_{P_0}^P \omega$$

The function \mathcal{A} , which is simply the Abel map without the equivalence classes induced by Λ , is single valued and analytic on $\tilde{\Gamma}$.

Let

$$\mathbf{Z} = (Z_1, \dots, Z_g)$$

be a fixed vector on the Jacobian $J(\Gamma)$. For any \mathbf{Z} , the function

$$\phi : \tilde{\Gamma} \mapsto \mathbb{C}, \quad \phi(P) = \theta(\mathcal{A}(P) - \mathbf{Z}), \quad (3.29)$$

is analytic and single valued on the cut surface $\tilde{\Gamma}$. Further, ϕ is not identically zero on $\tilde{\Gamma}$ if $\theta(\mathbf{Z}) \neq 0$ [18].

Proposition 3.14.1. *Suppose the function $\phi(P)$ is not identically zero on the cut surface $\tilde{\Gamma}$, then $\phi(P)$ has exactly g zeroes,*

$$P_1, \dots, P_g,$$

counting multiplicities.

In [18], for example, Proposition 3.14.1 is proved by calculating the logarithmic residue

$$\frac{1}{2\pi i} \int_{\partial \tilde{\Gamma}} d \log \phi.$$

The places

$$P_1, \dots, P_g$$

turn out to solve the Jacobi inversion problem for a correctly chosen vector $\boldsymbol{\zeta}$, as established below.

Proposition 3.14.2. *Suppose the divisor \mathcal{D} is comprised as*

$$\mathcal{D} = P_1 + \dots + P_g$$

where each place P_j satisfies

$$\phi(P_j) = \theta(\mathcal{A}(P_j) - \mathbf{Z}) = 0, \quad j = 1, \dots, g.$$

Then $\mathbf{Z} \in J(\Gamma)$ is defined by

$$\mathbf{Z} \equiv \mathbf{A}(P_0, \mathcal{D}) + \mathbf{K}(P_0),$$

where $\mathbf{K}(P_0)$ is the vector of Riemann constants, with the same initial place P_0 as the Abel map, given by

$$\mathbf{K}(P_0) = (K_1(P_0), \dots, K_g(P_0)),$$

$$K_j(P_0) \equiv \frac{1 + B_{jj}}{2} - \sum_{k \neq j} \int_{a_k} \omega_k(P) \mathcal{A}_j(P). \quad (3.30)$$

Proposition 3.14.2 belongs to the standard theory of Riemann surfaces and is proved in [21] and [18], for instance. Nonetheless I provide a proof as this result is absolutely fundamental to the work at hand.

Proof. The proof relies on calculating the integral

$$\zeta = (\zeta_1, \dots, \zeta_g)$$

given by

$$\zeta = \frac{1}{2\pi i} \int_{\partial \tilde{\Gamma}} d \log \phi(P) \mathcal{A}(P) \quad (3.31)$$

in two different ways to follow.

- **Sum of residues:** At any place $P_j \in \mathcal{D}$ the analytic function ϕ has a local representation

$$\phi = t^k h(t), \quad k \in \mathbb{N} > 0, \quad h(0) \neq 0,$$

where h is holomorphic and k is assured to be greater than zero since

$$\phi(P_j) = 0.$$

Thus the logarithmic derivative is given locally as

$$d \log \phi = \frac{k}{t} dt + \tilde{h}(t) dt \quad (3.32)$$

where $\tilde{h}(t)$ is analytic and non-zero at P_j . If $k > 1$, that is if P_j is a zero of ϕ multiplicity higher than 1, then the place P_j appears in \mathcal{D} a total of k times. Thus (3.31) yields

$$\zeta = \mathcal{A}(P_1) + \cdots + \mathcal{A}(P_g),$$

by the definition of \mathcal{A} ,

$$\zeta \equiv \mathcal{A}(P_0, P_1) + \cdots + \mathcal{A}(P_0, P_g),$$

which can be restated as

$$\zeta \equiv \mathcal{A}(P_0, \mathcal{D}) \tag{3.33}$$

where a place P_j may appear more than once and the dependence on the initial place P_0 is not explicitly shown. That ζ has the form given in the right-hand side of (3.33) is the reason to examine this particular integral.

- **Line integral:** By calculating the line integral explicitly around the boundary $\partial\tilde{\Gamma}$ one finds (details are presented in Appendix A.1) that the j -th component of ζ is given by

$$\zeta_j = Z_j - \frac{B_{jj}}{2} - \mathcal{A}_j(Q_j) + \sum_k \int_{a_k} \omega_k(P) \mathcal{A}_j(P) + \lambda_j \tag{3.34}$$

where

$$\lambda = (\lambda_1, \dots, \lambda_g)$$

is a lattice vector and thus λ_j may be omitted, and Q_j is the initial place of cycle b_j on the cut surface $\tilde{\Gamma}$. The term $\mathcal{A}_j(Q_j)$ is dependent upon the choice of homology \mathcal{H} . To remove this unnecessary dependence examine the integral

$$\begin{aligned} \int_{a_j} \omega_j(P) \mathcal{A}_j(P) &= \int_{a_j} \frac{1}{2} d\mathcal{A}_j^2(P) \\ &= \frac{1}{2} (\mathcal{A}_j^2(Q_j) - \mathcal{A}_j^2(Q'_j)) \end{aligned}$$

where Q_j is the end of the j -th a -cycle (as well as the beginning of the j -th b -cycle), and Q'_j is the beginning of the j -th a -cycle. Note that

$$\mathcal{A}_j(Q'_j) = \mathcal{A}_j(Q_j)\omega_j - 1$$

by (3.16), and thus that

$$\begin{aligned} & \frac{1}{2} (\mathcal{A}_j^2(Q_j) - \mathcal{A}_j^2(Q'_j)) \\ &= \frac{1}{2} (\mathcal{A}_j^2(Q_j) - (\mathcal{A}_j(Q_j) - 1)^2) \\ &= \mathcal{A}_j(Q_j) - \frac{1}{2}. \end{aligned}$$

Therefore (3.34) gives

$$\zeta_j = Z_j - \frac{B_{jj} + 1}{2} - \int_{a_j} \omega_j(P) \mathcal{A}_j(P) + \sum_k \int_{a_k} \omega_k(P) \mathcal{A}_j(P),$$

leading finally to

$$\zeta_j = Z_j - \frac{B_{jj} + 1}{2} + \sum_{k \neq j} \int_{a_k} \omega_k(P) \mathcal{A}_j(P). \quad (3.35)$$

Combining Equations (3.33) and (3.35) establishes Proposition 3.14.2. \square

The following three theorems complete the presentation of the Jacobi inversion problem. Two of these theorems will be useful in the next section. The following statement is from [18], where it is presented but not proved. For proof, see [26].

Theorem 1. *The function*

$$\phi(P) = \theta(\mathbf{A}(P) - \mathbf{Z})$$

is identically zero on Γ if and only if the vector \mathbf{Z} may be written

$$\mathbf{Z} \equiv \mathbf{A}(\mathcal{D}) + \mathbf{K}$$

where the divisor \mathcal{D} is effective, of degree g and special.

Propositions 3.14.1 and 3.14.2 and Theorem 1 lead directly to the following, which establishes the existence of the map

$$\mathbf{A}^{-1} : J(\Gamma) \mapsto S\Gamma^g.$$

Theorem 2. *Suppose for the vector*

$$\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_g),$$

the function

$$\phi(P) = \theta(\mathbf{A}(P_0, P) - \boldsymbol{\zeta} - \mathbf{K}_{P_0})$$

is not identically zero on Γ . Then

- *the function ϕ has g zeros*

$$P_1, \dots, P_g.$$

Further, these zeroes give a solution to the Jacobi inversion problem for the vector $\boldsymbol{\zeta}$.

That is

$$\mathbf{A}(P_0, P_1) + \dots + \mathbf{A}(P_0, P_g) \equiv \boldsymbol{\zeta}. \quad (3.36)$$

- *The divisor*

$$\mathcal{D} = P_1 + \dots + P_g$$

of zeroes is non-special.

- *The divisor \mathcal{D} is uniquely determined by (3.36) up to permutations.*

The Jacobi Inversion Theorem follows, proof of which may be found throughout the standard literature. One may consult, for example [25].

Jacobi Inversion Theorem. *The Abel map restricted to effective divisors of degree g ,*

$$\mathbf{A} : S\Gamma^g \mapsto J(\Gamma)$$

is surjective, and generically² injective.

²A property holds *generically* on a manifold if the property is true *except* at points on a sub-manifold of strictly lower dimension.

3.15 The vector of Riemann constants

The vector of Riemann constants (VRC), \mathbf{K} , appeared in the previous section as part of the solution to the Jacobi inversion problem. In this section I will go into a bit more detail. In part I will elaborate on the connections between \mathbf{K} , Γ and $J(\Gamma)$.

The VRC is generally seen as a constant vector $\mathbf{K} \in J(\Gamma)$ with fixed initial place $P_0 \in \Gamma$. It is equally valid to see this object as a *function*

$$\mathbf{K}(P_0) : \Gamma \mapsto J(\Gamma).$$

The convention of seeing the VRC as a constant seems to arise from the fact that one generally begins calculating the aforementioned objects associated with Riemann surfaces by fixing an initial place once and for all. This point of view is perfectly sensible, however there is occasion to treat this object as a function, and I will use either notation as needed.

The algorithm I present to compute the VRC relies on the following proposition which holds for all $P_0 \in \Gamma$. Thus any mention thereof is suppressed in the interest of easier reading.

Proposition 3.15.1. *Let \mathcal{C} be an arbitrary divisor such that*

$$\deg \mathcal{C} = 2g - 2.$$

Then \mathcal{C} is the divisor of a meromorphic differential on Γ if and only if

$$2\mathbf{K} \equiv -A(\mathcal{C}). \tag{3.37}$$

Proof of this proposition may be found, for instance, in [21]. However, as Proposition 3.37 is fundamental to the algorithm to compute the VRC, I provide a sketch of the proof therein.

Proof. The first goal is to prove that $-2\mathbf{K}$ is the image, under the Abel map, of a holomorphic differential, and thus also of a meromorphic differential. To establish this, first believe Proposition 3.15.2, proof of which is provided in Appendix A.2 in one direction. For a full proof see, for example [21].

Proposition 3.15.2. *Suppose \mathbf{B} is the Riemann matrix associated with the Riemann surface Γ , and P_0 is any place on Γ . The vector $\mathbf{W} \in J(\Gamma)$ is such that*

$$\theta(\mathbf{W}|\mathbf{B}) = 0$$

if and only if \mathbf{W} is given by

$$\mathbf{W} \equiv \mathbf{A}(P_0, \mathcal{D}) + \mathbf{K}(P_0)$$

for some choice of

$$\mathcal{D} = P_1 + \cdots + P_{g-1}.$$

Note that Proposition 3.15.2 may be written in the shorthand

$$\Theta = \mathbf{A}(S\Gamma^{g-1}) + \mathbf{K}, \quad (3.38)$$

where the θ -divisor Θ is

$$\Theta = \{\mathbf{V} \in J(\Gamma) : \theta(\mathbf{V}) = 0\}.$$

Θ is a $2g - 2$ real-dimensional subvariety inside the Jacobian $J(\Gamma)$ [21].

Now to establish that $2\mathbf{K}$ is the image under the Abel map of a holomorphic differential, first recall that an effective divisor

$$\mathcal{E}' = \sum p_j P_j$$

is one for which all the multiplicities p_j are positive. Let \mathcal{E}' be of degree $g - 1$ and given by

$$\mathcal{E}' = P_1 + \cdots + P_{g-1}.$$

Then the vector

$$\mathbf{W} = \mathbf{A}(\mathcal{E}') + \mathbf{K}$$

is a zero of the θ -function by Proposition 3.15.2. Since the θ -function is even,

$$\theta(-\mathbf{W}) = 0.$$

Again using Proposition 3.15.2, there exists an effective divisor \mathcal{E}'' such that

$$-\mathbf{W} = \mathbf{A}(\mathcal{E}'') + \mathbf{K}.$$

Using the two previous equations,

$$A(\mathcal{E}') + K \equiv -A(\mathcal{E}'') - K,$$

so

$$A(\mathcal{E}' + \mathcal{E}'') \equiv -2K.$$

Thus if the divisor

$$\mathcal{E} = \mathcal{E}' + \mathcal{E}''$$

is the divisor of a meromorphic differential, then $-2K$ is indeed the image of the divisor of a meromorphic differential. That \mathcal{E} is the divisor of a meromorphic function is guaranteed by Proposition 3.10.1.

Noting that $g - 1$ of the places in \mathcal{E} are arbitrary,

$$l^0(\mathcal{E}) \geq g.$$

Then, by Riemann-Roch

$$\begin{aligned} g \leq l^0(\mathcal{E}) &= 2g - 2 - g + l^1(\mathcal{E}) + 1 \\ &= g - 1 + i(\mathcal{E}). \end{aligned}$$

Therefore $l^1(\mathcal{E}) \geq 1$, which assures the existence of a meromorphic differential ν with $(\nu)_{\text{val}} = \mathcal{E}$.

To show the converse, let \mathcal{D} be an arbitrary divisor with degree $2g - 2$ such that

$$A(\mathcal{D}) \equiv -2K.$$

It has just been shown that there is a differential ν such that

$$A((\nu)) \equiv -2K.$$

Therefore

$$A(\mathcal{D}) \equiv -2K \equiv A((\nu)_{\text{val}}),$$

and further

$$A(\mathcal{D}) \equiv A((\nu)_{\text{val}}).$$

Rearranging the previous equivalence yields

$$A(\mathcal{D}) - A((\nu)_{\text{val}}) \equiv 0,$$

which, by use of Abel's Theorem implies the existence of a function u such that

$$(u)_{\text{val}} = \mathcal{D} - (\nu)_{\text{val}}.$$

The arbitrary divisor \mathcal{D} is then given by

$$\begin{aligned} \mathcal{D} &= (u)_{\text{val}} + (\nu)_{\text{val}} \\ &= (u\nu)_{\text{val}}, \end{aligned}$$

and is thus the divisor of the differential $u\nu$, proving proposition 3.37. □

Chapter 4

MULTI-PHASE SOLUTION FORMULAE

4.1 The nonlinear Schrödinger equation

Polynomials $F(x, y)$ that can be written

$$F(x, y) = y^2 - f(x)$$

where $f(x)$ is a polynomial in x with no repeated roots are said to be in *Weierstrass form*. Any algebraic curve that can be converted to Weierstrass form by bi-rational transformations of the coordinates x and y is said to be *hyperelliptic*. Hyperelliptic curves are a generalization of elliptic curves, all of which may be written as

$$y^2 = 4x^3 + bx + c$$

for some choice of $b, c \in \mathbb{C}$. The quasiperiodic solutions of the nonlinear Schrödinger (NLS) and Korteweg-deVries (KdV) equations are parameterized only by hyperelliptic curves. This will turn out not to be the case for solutions of the Kadomtsev-Petviashvili (KP) equation [6]. The class of hyperelliptic curves is quite broad. Riemann surfaces of arbitrarily high finite genus may be realized as the desingularization and compactification of hyperelliptic curves [25]. The *only* restrictions on the polynomials parameterizing KP solutions is that they be irreducible.

Suppose Γ is a Riemann surface arising from a polynomial $F(x, y)$ given by

$$F(x, y) = y^2 - \prod_{j=1}^{2g+2} (x - \lambda_j), \quad (4.1)$$

where all the branch points $\lambda_j \in \mathbb{C}$ are distinct. For (4.1), $x = \infty$ is not a branch point: there are two distinct ways to satisfy

$$F(\infty, y) = 0.$$

Call these two ordered pairs $P^{+\infty}$ and $P^{-\infty}$. Let \mathbf{B} be the Riemann matrix associated with Γ , and \mathcal{D} be a divisor on Γ . Recall the NLS equation:

$$i\psi_t = \psi_{\xi\xi} + 2|\psi|^2\psi.$$

A solution ψ of the NLS equation, parameterized by Γ is given by [6]

$$\begin{aligned} \psi = e^{-ic_1\xi + ic_2t} \frac{2\theta(\mathbf{U}_0)}{\alpha\theta(\mathbf{U}_0 - \mathbf{A}^\infty)} \times \\ \times \frac{\theta(i\mathbf{U}_\xi\xi + i\mathbf{U}_t t - \mathbf{U}_0 + \mathbf{A}^\infty)}{\theta(i\mathbf{U}_\xi\xi + i\mathbf{U}_t t - \mathbf{U}_0)}. \end{aligned} \quad (4.2)$$

There is a similar formula for the complex conjugate of ψ . Recall also that the free surface η is then given by

$$\begin{aligned} \eta(x, y, t) &= \Re(\psi u) + \text{higher order terms} \\ u &= a_0 e^{ik_0 x - i\sqrt{k_0} t}. \end{aligned}$$

All the concepts needed to explain (4.2) have been introduced in the preceding chapters.

- Each of the four θ -functions are parameterized by \mathbf{B} .
- c_1 and c_2 are constants obtained as the second-highest order behavior at $P^{-\infty}$ of Abelian differentials of the third kind.
- The constant vector \mathbf{U}_0 is given by

$$\mathbf{U}_0 = \mathbf{A}(P^{-\infty}, \mathcal{D}) + \mathbf{K}(P^{-\infty}),$$

where \mathbf{A} is the Abel map and \mathbf{K} is the vector of Riemann constants.

- The Abel map \mathbf{A}^∞ is given by

$$\mathbf{A}^\infty = \mathbf{A}(P^{-\infty}, P^\infty).$$

- The constant vectors \mathbf{U}_ξ and \mathbf{U}_t are found by integrating Abelian differentials of the third kind around the b -cycles.

4.2 The Korteweg-deVries equation

To construct quasiperiodic solutions of the KdV equation, assume a Riemann surface Γ arising from a polynomial as given by (4.1). Further assuming a divisor \mathcal{D} on the surface, the quasiperiodic solution of the KdV equation with these parameters is given by

$$u = c + 2\partial_x^2 \ln \theta(\mathbf{U}_x x + \mathbf{U}_y y + \mathbf{U}_t t + \mathbf{A}(P^\infty, \mathcal{D}) - \mathbf{K}(P^\infty)). \quad (4.3)$$

where $\mathbf{U}_x, \mathbf{U}_y$, and \mathbf{U}_t are vectors obtained by integrating Abelian differentials of the third kind around the b -cycles. Note that the variables (x , and y) typeset Roman are physical variables, and have nothing whatsoever to do with the coordinates x and y in the defining polynomial.

4.3 The Kadomtsev-Petviashvili equation

The quasiperiodic solutions of (1.11) are of the form

$$\eta(x, y, t) = c + 2\partial_x^2 \ln \theta(\mathbf{U}_x x + \mathbf{U}_y y + \mathbf{U}_t t + \mathbf{A}(P^\infty, \mathcal{D}) - \mathbf{K}(P^\infty) | \mathbf{B}). \quad (4.4)$$

Here

1. c is a constant, obtained as the second order behavior of a Abelian differential of the third kind with a pole only at P^∞ .
2. $\mathbf{U}_x, \mathbf{U}_y$ and \mathbf{U}_t are constant vectors, obtained, as in the KdV case, as the integrals of Abelian differentials of the third kind around b -cycles.
3. \mathcal{D} is a degree g divisor;
4. $\mathbf{A}(P^\infty, \mathcal{D})$ is the Abel map of \mathcal{D} for the initial place P^∞ , and
5. $\mathbf{K}(P^\infty)$ is the vector of Riemann constants for P^∞ .

Many of the objects in (4.2), (4.4) and (4.3) have already been made effective. For instance the computation of the Riemann matrix was the main topic of [17]. Algorithms to

compute theta functions and their derivatives are found in [14]. Effective computation of the Abel map and the vector of Riemann constants is discussed in this work.

Computing the constant c , and the constant wavenumber vectors \mathbf{U}_x , \mathbf{U}_y and \mathbf{U}_t appearing in the KP solution (and the KdV solution) has yet to be made effective. Likewise, computing constants and constant vectors appearing in the NLS solutions must be made effective. The algorithms required in all three cases are quite similar: all involve computations with Abelian differentials of the third kind. Once these have been made effective, the quasiperiodic solutions of the NLS, KdV, and KP equations will be readily available for use.

Note that solutions as in (4.3) and (4.4) are not in general real solutions. Guaranteeing that the solutions are real, as is desired for water-waves, requires either more algebro-geometric work, or the approach of [13]. In [13], Deconinck describes a method for determining the surface Γ and the divisor \mathcal{D} from initial data, thus solving the KP equation as an initial value problem for finite-genus quasiperiodic initial data.

Chapter 5

COMPUTING THE ABEL MAP

5.1 Initial considerations

The input for computing $A(P', P)$ is

- a Riemann surface Γ , entered as a polynomial $F(x, y)$ in two complex variables x and y and
- two places P' and P on Γ , entered as Puiseux expansions as in Newton's theorem. Places are entered as truncated Puiseux expansions to avoid potential ambiguity at places over discriminant points, and at infinity. For places over regular points it suffices to enter an x, y pair.

Note that, because $A(P', P)$ may be written as

$$\int_{P'}^P \omega = \int_{P^\ell}^P \omega - \int_{P^\ell}^{P'} \omega,$$

it suffices to develop an algorithm to compute integrals from P^ℓ to any other place P .

5.2 Computing discriminant points

Computing the discriminant points of an algebraic curve C is just a matter of computing the roots of the discriminant $D(F, y)$ with respect to y of the polynomial F that gives rise to C . I briefly address the resultant and discriminant of polynomials. This elementary material is included in the interest of completeness. See, for instance [25] for more detail about what follows in this section.

The *resultant* $R(f, g)$ of two polynomials

$$f = f(x) \quad \text{and} \quad g = g(x)$$

is a determinant formed from the two polynomials. This determinant is zero if and only if the two polynomials have simultaneous solutions. The resultant with respect to one variable, say y , denoted $R(F, G, y)$ of the polynomials

$$F = F(x, y) \quad \text{and} \quad G = G(x, y)$$

is a polynomial in x , the roots of which are the only affine x -values for which F and G have simultaneous solutions.

The *discriminant* $D(f)$ of a polynomial

$$f = f(x)$$

is the resultant $R(f, f_x)$ of

$$f = f(x)$$

and

$$f_x(x) = \frac{df(x)}{dx}.$$

The discriminant

$$D(f) = 0$$

if and only if the polynomial f has a finite multiple root. The discriminant $D(F, y)$ with respect to y of the polynomial

$$F = F(x, y)$$

is a polynomial in x given by.

$$D(F, y) = R\left(F, \frac{\partial F(x, y)}{\partial x}, y\right).$$

The roots of

$$D(F, y) = 0$$

are exactly the affine x -values for which F has multiple y -roots. These roots are the discriminant points oft mentioned in Chapter 3.

Suppose F is as in (3.1), then F and F_y are given by

$$F = f_0 y^n + \cdots + f_{n-1} y + f_n$$

$$F_y = n f_0 y^{n-1} + \cdots + f_{n-1}$$

The discriminant $D(F, y)$ of F with respect to y is then the determinant

$$D(F, y) = \begin{vmatrix} f_0 & \cdots & \cdots & f_n \\ & \ddots & \cdots & \cdots & \ddots \\ & & f_0 & f_1 & \cdots & f_n \\ n f_0 & \cdots & f_{n-1} & & & \\ & n f_0 & \cdots & f_{n-1} & & \\ & & \ddots & \cdots & \ddots & \\ & & & n f_0 & \cdots & f_{n-1} \end{vmatrix}.$$

Explicitly calculating the roots of

$$D(F, y) = 0$$

is generally impossible because generally,

$$\deg D(F, y) > 5.$$

Implicitly representing these roots is no work at all. Suppose, again the general case, that $D(F, y)$ is given by

$$D(F, y) = g_0 x^N + \cdots + g_N, \quad N > 5$$

where

$$g_j \in \mathbb{C}, \quad j = 1, \dots, N.$$

The discriminant points may be represented implicitly as, using Maple notation for instance

$$\text{Rootof}(D(F, y), x) = \text{Rootof}(g_0 x^N + \cdots + g_N, x).$$

Note however, when explicitly computing with these discriminant points, or places over them, one must either work with this representation or approximate the roots numerically. Both approaches introduce difficulties which must be dealt with.

5.3 *Computing paths on the Riemann sphere*

Before proceeding, I need to introduce some notation and definitions.

- In what follows $C(\alpha, \rho)$ denotes the circle in the complex x -plane with center $x = \alpha$ and radius ρ .
- By $L(\alpha, \alpha')$ we indicate the straight line segment in the complex plane from α to α' .
- The “left point,” $\alpha^\ell \in \mathbb{C}$, is an algorithmically chosen point depending only on the polynomial F . If $\widehat{\lambda}$ is the set of finite discriminant points of

$$F(x, y) = 0,$$

then α^ℓ is chosen such that

$$\Re(\alpha^\ell) < \Re(\lambda), \quad \forall \lambda \in \widehat{\lambda},$$

that is, α^ℓ is to the left (in the complex plane) of all the finite discriminant points.

- Denote by

$$\mathbf{y}^\ell = (y_1^\ell, \dots, y_n^\ell)$$

the *ordered* roots of

$$F(\alpha^\ell, y) = 0.$$

It is in principle immaterial how the order is imposed. In practice the order is assigned by increasing absolute value. If two of the roots have the same absolute value, then the root with the smaller argument on $[0, 2\Pi)$ appears first.

- The “left place” is $P^\ell = (\alpha^\ell, y_1^\ell)$.

- The ordered n -tuple \mathbf{y}^ℓ can be analytically continued anywhere on

$$\mathbb{P}^1/\widehat{\lambda}.$$

The n -tuple of complex numbers $\mathbf{y}^\ell(\chi)$ is the result of analytically continuing each component of \mathbf{y}^ℓ along the path

$$\chi = x(z) \in \mathbb{P}^1/\widehat{\lambda}, \quad z \in [0, 1].$$

- If χ and χ' are two paths in $\mathbb{P}^1/\widehat{\lambda}$ with

$$\chi(0) = \chi'(0), \quad \text{and} \quad \chi(1) = \chi'(1),$$

it is not necessary that

$$\mathbf{y}^\ell(\chi) = \mathbf{y}^\ell(\chi').$$

If χ and χ' differ only by a circle around exactly one branch point then

$$\mathbf{y}^\ell(\chi') = \sigma(\mathbf{y}^\ell(\chi)),$$

where $\sigma(\mathbf{v})$ denotes a permutation.

- To order the places over a regular point α we impose the order of the y -values in $\mathbf{y}^\ell(L(\alpha^\ell, \alpha))$. Note that if the straight line segment $L(\alpha^\ell, \alpha)$ happens to pass through a discriminant point λ_k , then the path to α is adjusted to pass above λ_k . In this way, the *sheets* of the Riemann surface are defined. A place

$$P = (\alpha, \beta)$$

is said to lie on sheet j if

$$\left(\mathbf{y}^\ell(L(\alpha^\ell, \alpha)) \right)_j = \beta.$$

Let F be a polynomial as in (3.1). Further let

$$\widehat{\lambda} = \{\lambda_1, \dots, \lambda_m\}$$

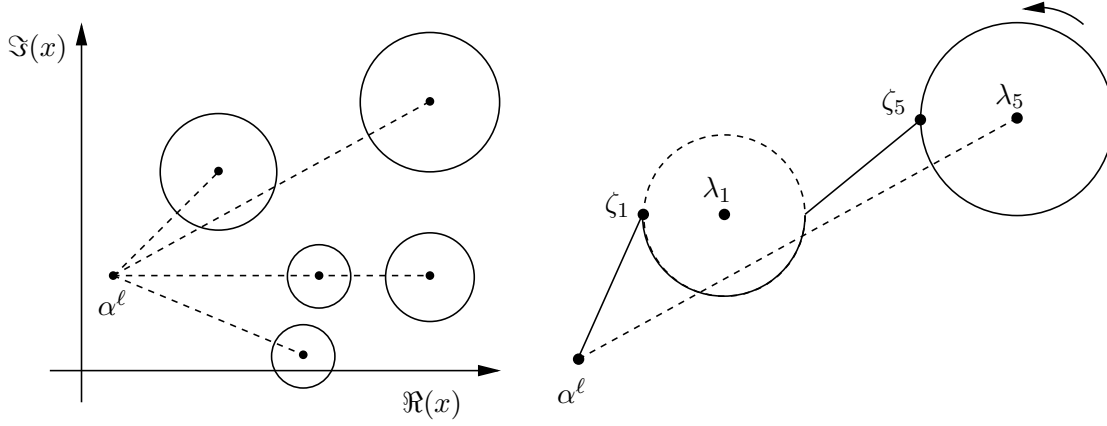


Figure 5.1: On the left are straight lines $L(\alpha^\ell, \lambda_j)$ (dashed) and circles C_j (solid). On the right I isolate α^ℓ , C_1 and C_5 in order to demonstrate how the path encircling only λ_5 is deformed through a region to stay well away from λ_1 . The solid line on the right is the path χ'_5 : the path to ζ_5 deformed once and homotopic to the line $L(\alpha^\ell, \zeta_5)$ (not shown).

be the set of finite discriminant points of

$$F(x, y) = 0.$$

I now construct paths on the Riemann sphere that will be lifted to the Riemann surface Γ arising from F in Section 5.4.

Paths avoiding discriminant points

Around each $\lambda_j \in \hat{\lambda}$ we compute a radius ρ_j that is two-fifths of the distance to the next nearest discriminant point:

$$\rho_j = \frac{2}{5} \min\{|\lambda_j - \lambda_k|\},$$

where $j \neq k$. Let C_j denote the circle $C(\lambda_j, \rho_j)$. The y -sheets are sufficiently well-separated, and numerical analytic continuation works well on all paths that lie completely outside all C_j 's. Let ζ_j denote the point $\lambda_j - \rho_j$. This is the point on C_j furthest to the left. A path χ_j to each ζ_j is created as follows.

1. If the line $L(\alpha^\ell, \zeta_j)$ does not intersect any other C_k then

$$\chi_j = L(\alpha^\ell, \zeta_j).$$

2. If the line $L(\alpha^\ell, \zeta_j)$ intersects circle C_k , then χ_j is deformed to χ'_j , which traverses half of C_k . Path χ'_j includes either the top or bottom half of C_k , depending on which path is homotopic to $L(\alpha^\ell, \zeta_j)$. The right side of Figure 5.1 shows an example of this deformation.
3. If the newly deformed path χ'_j infringes on circle C_m , it is deformed to χ''_j , which traverses half of C_m as in the previous step.
4. Steps 2 and 3 are iterated until a path $\chi_j^{\prime\prime\prime}$ is constructed that avoids all C_i , for $i \neq j$.

The path Π_j around discriminant point λ_j is then, dropping the primes, χ_j followed by the circle C_j traversed counter-clockwise and ending with χ_j traversed backwards. The solid line on the right of Figure 5.1 is such a path. The Π_j are used to calculate the monodromy group of F [17] and the χ_j serve as the basic building blocks of paths to any $\alpha \in \mathbb{P}^1$.

The monodromy group

If the point λ_j is a branch point, then the analytic continuation $\mathbf{y}^\ell(\Pi_j)$ does not match up with \mathbf{y}^ℓ itself. Instead

$$\mathbf{y}^\ell(\Pi_j) = \sigma_j(\mathbf{y}^\ell),$$

where $\sigma_j(\mathbf{v})$ is the permutation of the n -tuple \mathbf{v} induced by analytically continuing \mathbf{v} around branch point λ_j . The collection of all the permutations σ_j gives a representation of the monodromy group of the curve F [17]. A procedure to compute the monodromy of an arbitrary irreducible plane algebraic curve is available as the Maple command `algcures[monodromy]`. The implementation of the Abel map uses this command unaltered to compute the branch points and the permutations around those branch points.

As the format of the output of this algorithm is important in Section 5.3, I give it here. Given a polynomial $F(x, y)$ of y -degree n with discriminant points

$$\lambda_1, \lambda_2, \dots,$$

the output of the monodromy algorithm is:

1. the left place α^ℓ , and
2. the list \mathbf{y}^ℓ of ordered roots of $F(\alpha, y) = 0$, and
3. a table, indexed by the discriminant points, of permutation groups around said discriminant points. Suppose for example that $n = 6$. If the λ_2 entry of the table is

$$(\{1, 2, 3\}, \{4, 6\})$$

then around λ_2 , sheets 1, 2 and 3 cyclically permute, as do sheets 4 and 6. Sheet 5 remains unchanged.

Example 8. *The places over $\lambda_2 \in \mathbb{C}$ in the example in Item 3 would have Puiseux expansions of the form.*

$$P_1 = \begin{cases} x &= \alpha + t^3 \\ y &= \beta_1 t^{s_1} + \dots \end{cases},$$

$$P_2 = \begin{cases} x &= \alpha + t^2 \\ y &= \beta_2 t^{s_2} + \dots \end{cases}$$

and

$$P_3 = \begin{cases} x &= \alpha + t \\ y &= \beta_3 t^{s_3} + \dots \end{cases}.$$

The first place, P_1 , has branching number 3 thus three sheets coalesce around this place. This could account for the permutation group $\{1, 2, 3\}$. Place P_2 has branching number 2, and could account for permutation group $\{4, 6\}$. Place P_3 would then be on sheet five, which does not appear in a permutation group.

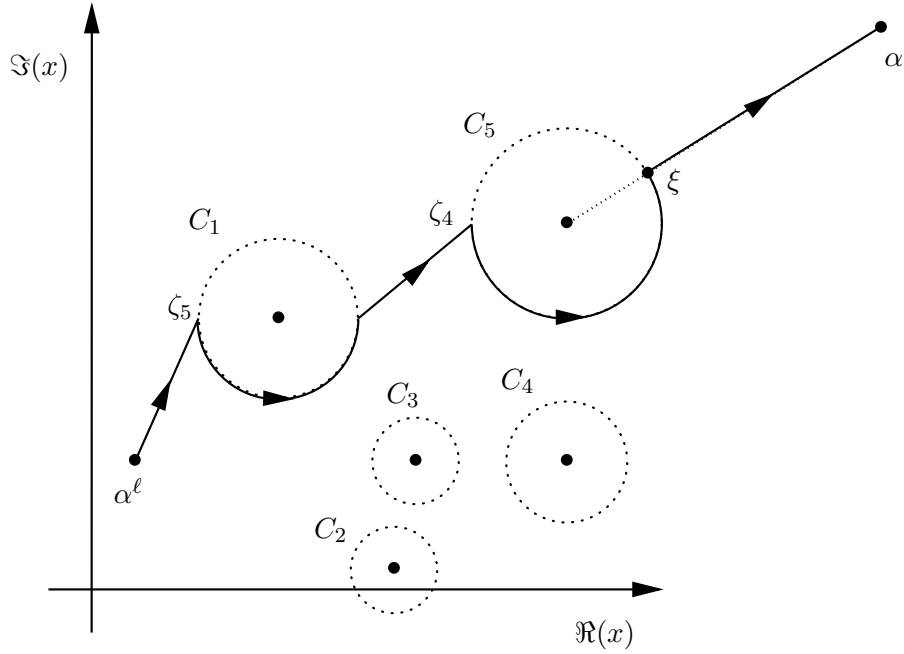


Figure 5.2: The solid lines and arcs indicate a path χ in the complex x -plane to the point α . These paths are lifted to the Riemann surface via numerical analytic continuation. Note that this path is *not* homotopic to $L(\alpha^\ell, \alpha)$. It need not be. The n images of this path on Γ are computed, and the correct image is chosen.

Path to a finite point not near a discriminant point

If α does not lie in one of the C_k , then a path χ from α^ℓ to α is constructed as follows. Suppose λ_j is the closest discriminant point to α . The line $L(\alpha, \lambda_j)$ intersects C_j at the point ξ . The path χ is then

1. the previously discussed path χ_j to ζ_j followed by
2. the arc on C_j from ζ_j to ξ , traversed counter-clockwise, and finally
3. the line $L(\xi, \alpha)$.

This path avoids all C_k for $j \neq k$ by virtue of the construction of the previously discussed paths χ_j . Figure 5.2 demonstrates a path to a regular point α that is closer to the discriminant point λ_5 than any other.

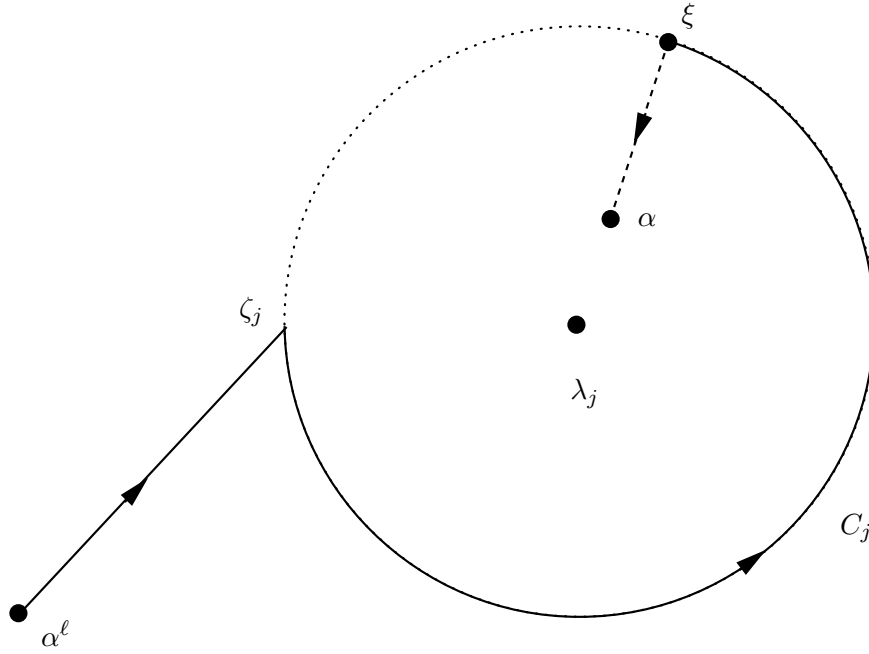


Figure 5.3: The solid and dashed lines together form a path from α^ℓ to α . The solid line is lifted to the Riemann surface Γ by numerical analytic continuation. The dashed line is lifted by means of Puiseux expansions. The dotted line again indicates the circle around the discriminant point λ_j with radius ρ_j . As in Figure 5.2, the path shown is not, and need not be, homotopic to $L(\alpha^\ell, \alpha)$.

Path to a point near a discriminant point

If the final point α is inside C_j , then the path χ is constructed as follows. Denote by ξ the point on C_j such that $L(\lambda_j, \xi)$ contains α . Then the path χ is, as in Figure 5.3,

1. the path χ_j to ζ_j ,
2. the arc on C_j from ζ_j to ξ , traversed counter-clockwise, and finally
3. the line segment from ξ inward towards λ_j to α .

Note that, as far as the creation of a path is concerned, it does not matter if $\alpha = \lambda_j$. That is, the path to the discriminant point λ_j is constructed in the same manner as one to a point anywhere within C_j .

Path to $x = \infty$

If $\alpha = \infty$, then a point $\xi \in \mathbb{R}$ is picked such that

$$|\xi| = \frac{5}{2} \max |\lambda|, \lambda \in \widehat{\lambda}.$$

The path χ to $\alpha = \infty$ is then $L(\alpha^\ell, \xi)$ followed by $L(\xi, \infty)$, where the line “to infinity” is traversed by the transformation

$$x = \frac{1}{t}$$

as $t \mapsto 0$.

Navigating between sheets

The previous sections have shown how a path $\chi \in \mathbb{P}^1$ is constructed from the algorithmically determined point α^ℓ to a generic point with x -component α . Note that α is given by the user-specified place P which is either over regular point, in which case P is given by

$$P = \begin{cases} x &= \alpha \\ y &= \beta \end{cases},$$

or over a discriminant place, in which case P is given by

$$P = \begin{cases} x &= \alpha + t^r \\ y &= \beta t^s + \beta' t^{s'} + \dots \end{cases}.$$

The path χ has n images on Γ . From these possible images, $\gamma \in \Gamma$ is chosen such that the y -component of $\gamma(1)$ is β . Let k be the index of the place $\gamma(0)$ over α^ℓ . If $k = 1$, then the lift of χ is exactly a path on Γ from the left place P^ℓ to the user-specified place P . If not, the monodromy group of F is used to create a path that will permute sheet 1 into sheet k .

From the output of the Maple command `algcurves[monodromy]`, a table \mathcal{M} is built that has as its entry \mathcal{M}_{ij} a list of branch points

$$\lambda_1, \lambda_2, \dots$$

to encircle in order to permute sheet m to sheet m' [17]. For example, if the 3,5 entry

$$\mathcal{M}_{3,5} = ((\lambda_1, \text{true}), (\lambda_7, \text{false}))$$

then the “shortest” way to navigate from sheet 3 into sheet 5 is to first encircle branch point λ_1 clockwise, and then branch point λ_7 anti-clockwise. The path is shortest in a graph-theoretic sense: it includes the fewest possible discriminant points as found by a depth-first search, as demonstrated in Algorithm 1 on Page 93. Since branch points are discriminant points, the list $\mathcal{M}_{1,k}$ defines a path

$$\Pi_{1,k} = \Pi_j, \Pi_{j'}, \dots$$

where the paths

$$\Pi_j, \Pi_{j'}, \dots,$$

were computed in Section 5.3.

Algorithm 1 on the following page is used to compute a connection table for a Riemann surface. The algorithm is of course not new. It is an instance of a depth-first search. I include it here for completeness, and because I wish to introduce this format, which I will use for displaying algorithms, with a simple example. The symbols \Rightarrow and \Leftarrow are used to denote the input and output of the algorithm respectively. Algorithms are typeset in a sans-serif font to set them apart from the regular text. Commands that effect the control flow of the algorithm, *e.g.* **if**, **for** *etc.*, are typeset in bold. The steps in the algorithms are labeled, and notes that directly follow the algorithms give more detail for some of the labeled steps.

Algorithm 1. Connection table:

\Rightarrow monodromy representation of Riemann surface Γ

\Leftarrow connection table \mathcal{M}

1. fill the table \mathcal{M} with all direct connections
2. **for** all empty fields \mathcal{M}_{jk} of \mathcal{M} **do**
 - A. Initialize the temporary nested list
 - B. **while** k not in \mathcal{S} **do**
 - i. search the deepest level of nesting in \mathcal{S} and amend as needed
 - end do** (ends 2.B)
 - C. use the information in \mathcal{S} to complete field \mathcal{M}_{jk}
- end do** (ends 2)

Notes for Algorithm 1:

\Rightarrow The algorithm presented in [17] for computing a monodromy representation of a Riemann surface is discussed in Section 5.3.

- 1 For each pair of sheets j and k , search the monodromy for a branch point, the permutation of which contains both j and k . For example, if the monodromy representation \mathcal{T} contains the entries

$$\mathcal{T}_{\lambda_3} = (\dots, \{4, k, j, 1, 7\}, \dots)$$

and

$$\mathcal{T}_{\lambda_7} = (\dots, \{\dots, j, 9, 4, k, \dots\}, \dots)$$

then both λ_3 and λ_7 directly connect sheets j and k . The j, k -th entry of \mathcal{M} would then be

$$\mathcal{M}_{jk} = (\lambda_3, \text{false})$$

where λ_3 is chosen because it is closer to α^ℓ by construction. The second element is “false” because going backwards through the permutation from j to k is shorter than going forwards from j to 1 to *etc.* on to 4 and finally to k .

2.A The integers

$$j', j'', \dots$$

represent the sheets to which sheet j is directly connected where these connections are via analytic continuation around the branch points

$$\lambda', \lambda'', \dots$$

respectively.

2.B.i Suppose that \mathcal{S} is currently given by

$$\mathcal{S} = \dots, \left((3, 2 + i), \left((1, -7i), (5, 2 - i) \right) \right), \dots,$$

which means that sheets 1 and 5 are directly connected to sheet 3 via branch points $-7i$ and $2 - i$ respectively. Further suppose that sheet 7, not yet in \mathcal{S} , connects directly to sheet 5 via branch point $3 - 3i$. In this case \mathcal{S} is updated to

$$\mathcal{S} := \dots, \left((3, 2 + i), \left((1, -7i), \left((5, 2 - i), (7, 3 - 3i) \right) \right) \right), \dots$$

If $k = 7$ then control would pass to Step 2.C, otherwise this step is repeated until k is found.

2.C This step is just a matter of reformatting the information in \mathcal{S} .

5.4 Lifting paths to the Riemann surface

Computing $\mathbf{A}(P^\ell, P)$ requires the computation of integrals

$$\int_{\gamma} \omega,$$

where ω is a normalized holomorphic differential, and γ is a parameterized path on Γ such that

$$\gamma(0) = P^\ell, \quad \text{and} \quad \gamma(1) = P.$$

A parameterized path $\gamma \in \Gamma$ is uniquely determined by $\chi \in \mathbb{P}^1$ and an initial y -component $y = \beta$ that solves

$$F(\alpha^\ell, \beta) = 0.$$

Assume the path χ from α^ℓ to α avoids all discriminant points unless α is itself a discriminant point. This path has n distinct images on Γ , each of which is uniquely determined by analytically continuing one of the elements of \mathbf{y}^ℓ along χ . The path γ is the image of $\chi \in \Gamma$ for which the starting place

$$\gamma(0) = (\alpha^\ell, \beta).$$

For the purposes of integrating holomorphic differentials this determination is unique up to elements of \mathcal{H} .

The algorithms used for the computation of the Abel map perform analytic continuation along a path χ using two methods, namely numerical analytic continuation and Puiseux expansions.

- **Numerical analytic continuation:** If a path $\chi \in \mathbb{C}$ begins and ends respectively at regular points $x = \alpha_0$ and $x = \alpha_1$, then the values of $\mathbf{y}(\alpha_0)$ follow paths on the Riemann surface from simple roots of

$$F(\alpha_0, y) = 0$$

to simple roots of

$$F(\alpha_1, y) = 0.$$

If α_1 is close to α_0 , and χ is nearly a straight line path avoiding discriminant points, then $\mathbf{y}(\alpha_1)$ is well approximated by

$$\mathbf{y}(\alpha_1) = \mathbf{y}(\alpha_0) + \mathbf{y}_x(\alpha_0)(\alpha_1 - \alpha_0) + \mathcal{O}(|\alpha_1 - \alpha_0|^2),$$

and the last term is small in comparison with the first two. Here \mathbf{y}_x is the n -tuple of derivatives of the algebraic function \mathbf{y} at the point α_0 . By implicit differentiation

$$\mathbf{y}_x(\alpha_0) = - \left(\frac{F_x(\alpha_0, y_1(\alpha_0))}{F_y(\alpha_0, y_1(\alpha_0))}, \dots, \frac{F_x(\alpha_0, y_n(\alpha_0))}{F_y(\alpha_0, y_n(\alpha_0))} \right),$$

where the subscripts x, y denote partial differentiation. Comparing the unordered $y(\alpha_1)$ with the ordered approximation given by

$$\mathbf{y}(\alpha_0) + \mathbf{y}_x(x_1)(\alpha_1 - \alpha_0)$$

results in an ordered n -tuple $\mathbf{y}(\alpha_1)$. This method may not resolve the order properly if $|\alpha_1 - \alpha_0|$ is too large or the path from α_0 to α_1 deviates significantly from a straight line. In this case the path is broken up into sub-intervals and the analytic continuation is performed adaptively, over the smaller segments until the order is correctly assigned. More detail is found in [17].

- **Puiseux expansion method:** Analytic continuation near discriminant points is problematic. At a discriminant point $x = \lambda$ there are less than n y -values, so the above approach is not possible. Further, in some neighborhood around λ , the y -sheets are close together, and distinguishing them numerically is difficult. Puiseux expansions resolve these issues.

The path χ constructed above joining

$$\chi(0) = \alpha^\ell \quad \text{to} \quad \chi(1) = \alpha$$

avoids almost all discriminant points. The path will avoid all discriminant points if α is not inside C_j for some j . If any part of χ is inside C_j the Puiseux expansions over λ_j , the center of C_j , is used to represent the sheets of the Riemann surface within C_j . The rest of χ is lifted to Γ via analytic continuation.

Let \tilde{z} be the value of the path parameter such that $\chi(\tilde{z})$ is the point where χ and C_j intersect. By the construction of χ there is only one such point. The y -sheets are ordered over $\chi(\tilde{z})$ with respect to α^ℓ by numerical analytic continuation. Denote the ordered y -values by $\mathbf{y}^\ell(\chi(\tilde{z}))$. Let

$$\hat{P} = (P_1, \dots, P_n)$$

be the n branches over the discriminant point λ_j . The intersection point $\chi(\tilde{z})$ is closer to λ_j than any other discriminant point, thus the branches representing the y -values converge over $\chi(\tilde{z})$ (see Section 3.2). Therefore the n branches can be ordered with respect to α^ℓ by matching P_j to one of the elements of the ordered n -tuple $\mathbf{y}^\ell(\chi(\tilde{z}))$. Suppose that

$$P_j = \begin{cases} x &= \lambda_j + t^{r_j} \\ y &= \beta_j t^{s_j} + \beta'_j t^{s'_j} + \dots \end{cases}, \quad t \in [0, 2\pi/r_j).$$

The y -value over the point $\chi(\tilde{z})$ represented by this branch is

$$\beta_j \tilde{t}^{s_j} + \beta'_j \tilde{t}^{s'_j} + \dots,$$

where

$$\tilde{t} = \sqrt[r_j]{\chi(\tilde{z}) - \lambda_j},$$

and as this is a convergent series it necessarily matches one of the elements of $\mathbf{y}^\ell(\chi(\tilde{z}))$. All the branches are assigned to one of the sheets in this way, and these branches define the local structure of all the places on Γ over C_j . If $\alpha = \lambda_j$, that is if the user specifies a place P over a discriminant point, then we match only this branch to $\mathbf{y}^\ell(\chi(\tilde{z}))$.

Note that only a finite number of terms of the Puiseux expansion $y(t)$ are computed. Thus the above matching is generally done by adaptively refining the branches until the assignments are made correctly over both $\chi(\tilde{z})$ and the end of the path $\chi(1)$.

5.5 *Remarks on integration*

Once the paths have been lifted to the Riemann surface, computing the Abel map is a matter of integrating holomorphic differentials. The integral

$$\begin{aligned}\int_{\gamma} \omega_j &= \int_{\gamma} g_j(x, y) \, dx \\ &= \int_{\gamma} g_j(x(z), y(z)) \frac{dx}{dz} \, dz\end{aligned}$$

is computable piecewise once the path γ is fixed. In practice a bit more needs to be said. Specifically it should be noted that the method of integration is a matter of what form of analytic continuation is used over any particular sub-path. A sub-path lifted to the Riemann surface by means of numerical analytic continuation is integrated using internal Maple routines (adaptive numerical integration). Paths lifted by Puiseux expansion method are integrated by computing series approximations of the holomorphic differentials. These convergent series of ever higher order are integrated and evaluated term-by-term symbolically.

Chapter 6

COMPUTING THE VECTOR OF RIEMANN CONSTANTS

6.1 Initial considerations

I first remind the reader of the form of the vector of Riemann constants (VRC). Given a Riemann surface Γ and a fixed initial place $P_0 \in \Gamma$, the VRC

$$\mathbf{K}(P_0) \in J(\Gamma).$$

is given by

$$\mathbf{K}(P_0) = (K_1(P_0), \dots, K_g(P_0)),$$

$$K_j(P_0) = \frac{1 + B_{jj}}{2} - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j(P_0, P), \quad (6.1)$$

where B_{jj} is the j, j -th component of the Riemann matrix \mathbf{B} associated with Γ . While the VRC is dependent upon the choice of homology basis \mathcal{H} and the fixed initial place P_0 , the properties of the VRC used here are dependent upon neither. Specifically, the formulas (3.37) and (3.38):

$$2\mathbf{K}(P_0) \equiv -\mathbf{A}(P_0, \mathcal{C})$$

where \mathcal{C} is in the canonical class:

$$\mathcal{C} \sim (\nu)_{\text{val}}, \quad \nu \in \Omega_{\Gamma}^1,$$

and

$$\Theta = \mathbf{A}(P_0, S\Gamma^g) - \mathbf{K}(P_0)$$

hold for any choice of homology and for all $P_0 \in \Gamma$. Computing the double line integral in (6.1) proved difficult. Specifically, algorithms using this definition to compute \mathbf{K} were

abandoned, after much time and effort, as too slow and inaccurate. The method currently employed hinges on the fact that the VRC may be written as

$$\mathbf{K}(P_0) = \mathbf{K}(P^\ell) - (g-1)\mathbf{A}\left(P_0, P^\ell\right), \quad (6.2)$$

where $\mathbf{K}(P^\ell) \in \Gamma$ is a vector that only depends on the choices made for P^ℓ and \mathcal{H} . To obtain (6.2), begin by recalling that \mathbf{A} is defined as an integral computed between its arguments, and any arbitrary place P' on Γ may be inserted:

$$\mathbf{A}(P_0, P) \equiv \mathbf{A}(P_0, P') + \mathbf{A}(P', P), \quad \forall P' \in \Gamma$$

and writing (6.1) as

$$K_j(P_0) = \frac{1 + \mathbf{B}_{jj}}{2} - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) \left(A_j\left(P_0, P^\ell\right) + A_j\left(P^\ell, P\right) \right),$$

where P^ℓ is the left place from Section 5.1. Rearranging the above gives

$$K_j(P_0) = \frac{1 + \mathbf{B}_{jj}}{2} - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j\left(P_0, P^\ell\right) - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j\left(P^\ell, P\right).$$

Note that the j -th component of the Abel map $A_j(P_0, P^\ell)$ in the first sum has no dependence on the variable of integration P . Thus

$$K_j(P_0) = \frac{1 + \mathbf{B}_{jj}}{2} - \sum_{k \neq j}^g A_j\left(P_0, P^\ell\right) \int_{a_k} \omega_k(P) - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j\left(P^\ell, P\right),$$

and using

$$\int_{a_k} \omega_k = 1,$$

yields

$$\begin{aligned} K_j(P_0) &= \frac{1 + \mathbf{B}_{jj}}{2} - \sum_{k \neq j}^g A_j\left(P_0, P^\ell\right) - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j\left(P^\ell, P\right), \\ K_j(P_0) &= \frac{1 + \mathbf{B}_{jj}}{2} - (g-1)A_j\left(P_0, P^\ell\right) - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j\left(P^\ell, P\right). \end{aligned}$$

Therefore letting

$$\begin{aligned} \mathbf{K}^\ell &:= (K_1(P^\ell), \dots, K_g(P^\ell)) \\ K_j(P^\ell) &= \frac{1 + \mathbf{B}_{jj}}{2} - \sum_{k \neq j}^g \int_{a_k} \omega_k(P) A_j(P^\ell, P). \end{aligned} \quad (6.3)$$

establishes (6.2). Two notable properties of the system specified by (6.2) and (6.3) are that

- the vector \mathbf{K}^ℓ is only dependent on the homology \mathcal{H} and the left place P^ℓ . Changing the order of the cohomology will induce identical permutations on the indices of \mathbf{K} , \mathbf{K}^ℓ , \mathbf{A} and \mathbf{B} . This is a trivial dependence as it is only a reordering of the coordinates of \mathbb{C}^g , or alternatively, of $J(\Gamma)$.
- If Γ is a Riemann surface represented by

$$F(x, y) = 0$$

for a polynomial F , then once \mathcal{H} has been computed for a surface Γ , and \mathbf{K}^ℓ has been computed with this choice, the VRC associated with any place $P_0 \in \Gamma$ can be computed using only the Abel map.

An algorithm has already been presented to compute the Abel map of an arbitrary place P on a Riemann surface represented as a polynomial in two complex variables. Thus it is sufficient to have an algorithm to compute the VRC of an arbitrary place to develop an algorithm to compute \mathbf{K}^ℓ as defined in (6.3).

6.2 The vector of Riemann constants of the left-place

Introducing a bit of notation simplifies the following discussion. Denote by $\Lambda_{\frac{1}{2}}$ the set of 2^{2g} half-lattice vectors in the Jacobian. That is, let

$$\Lambda_{\frac{1}{2}} = \{\mathbf{h} : \mathbf{h} = \mathbf{mod}(\mathbf{M}, 2) + \mathbf{B} \mathbf{mod}(\mathbf{N}, 2); \mathbf{M}, \mathbf{N} \in \mathbb{Z}^g\}, \quad (6.4)$$

where $\mathbf{mod}(\mathbf{M}, 2)$ and $\mathbf{mod}(\mathbf{N}, 2)$ are interpreted component-wise. That is, given the integer vector

$$\mathbf{M} = (M_1, \dots, M_g) \in \mathbb{Z}^g$$

the half-integer vector $\mathbf{mod}(M, 2)$ is

$$\mathbf{mod}(M, 2) = (\text{mod}(M_1, 2), \dots, \text{mod}(M_g, 2)).$$

It is clear that $\Lambda_{\frac{1}{2}}$ has 2^{2g} unique elements.

Given (3.37), one would like to say that \mathbf{K}^ℓ is given by

$$\mathbf{K}^\ell \equiv -\frac{1}{2}\mathbf{A}(P^\ell, \mathcal{C}).$$

However, division is not unique in this equivalence class. Recall Equivalence (3.37):

$$2\mathbf{K}^\ell \equiv -\mathbf{A}(P^\ell, \mathcal{C}), \quad (3.37)$$

which, using the definition of $\llbracket \cdot \rrbracket$ from Section 3.8, may be rewritten as the identity

$$\llbracket 2\mathbf{K}^\ell \rrbracket = \llbracket -\mathbf{A}(P^\ell, \mathcal{C}) \rrbracket. \quad (6.5)$$

Allow the notational convenience

$$\mathbf{A}_\mathcal{C}^\ell := \mathbf{A}(P^\ell, \mathcal{C}).$$

By the definition of $\llbracket \cdot \rrbracket$, (6.5) means that in the vector space \mathbb{C}^g ,

$$2\mathbf{K}^\ell - (-\mathbf{A}_\mathcal{C}^\ell) = 2\mathbf{K}^\ell + \mathbf{A}_\mathcal{C}^\ell = \boldsymbol{\lambda}$$

where $\boldsymbol{\lambda} \in \Lambda$ is an unknown vector. Dividing by 2 is legitimate in \mathbb{C}^g and gives

$$\mathbf{K}^\ell + \frac{1}{2}\mathbf{A}_\mathcal{C}^\ell = \frac{1}{2}\boldsymbol{\lambda}. \quad (6.6)$$

Setting $\mathbf{h} = \boldsymbol{\lambda}/2$ yields

$$\mathbf{K}^\ell = \mathbf{h} - \frac{1}{2}\mathbf{A}_\mathcal{C}^\ell.$$

which is once again an identity on \mathbb{C}^g . The corresponding identity on the Jacobian is

$$\llbracket \mathbf{K}^\ell \rrbracket = \llbracket \mathbf{h} - \frac{1}{2}\mathbf{A}_\mathcal{C}^\ell, \rrbracket \quad (6.7)$$

As $\mathbf{K}^\ell \in J(\Gamma)$,

$$\llbracket \mathbf{K}^\ell \rrbracket = \mathbf{K}^\ell,$$

and Equation (6.7) can be written as

$$\mathbf{K}^\ell = \left\llbracket \mathbf{h} - \frac{1}{2} \mathbf{A}_c^\ell \right\rrbracket \quad (6.8)$$

where, as the lattice vector $\boldsymbol{\lambda}$ is unknown, there are 2^{2g} possibilities for the half-lattice vector $\mathbf{h} \in \Lambda_{1/2}$. These possibilities, labeled

$$\mathbf{h}_1, \dots, \mathbf{h}_{2^{2g}},$$

lead to the 2^{2g} inequivalent vectors

$$\left\llbracket \mathbf{h}_j - \frac{1}{2} \mathbf{A}_c^\ell \right\rrbracket.$$

The form of the theta-divisor Θ is employed in order to determine which of the 2^{2g} possibilities for \mathbf{K}^ℓ given in (6.8) is correct.

Recall that Proposition 3.38 assures that for any $P_0 \in \Gamma$,

$$\theta(\mathbf{V}) = 0$$

if and only if

$$\mathbf{V} \equiv \mathbf{A}(P_0, \mathcal{D}) + \mathbf{K}(P_0),$$

where \mathcal{D} is an effective degree $g - 1$ divisor. Choosing \mathcal{D} to be

$$\mathcal{D} = (g - 1)P_0$$

shows that

$$\begin{aligned} \theta(\mathbf{A}(P_0, \mathcal{D}) + \mathbf{K}(P_0)) &= \theta(\mathbf{A}(P_0, (g - 1)P_0) + \mathbf{K}(P_0)) \\ &= \theta(\mathbf{0} + \mathbf{K}(P_0)), \quad \mathbf{0} = (0, \dots, 0) \\ &= \theta(\mathbf{K}(P_0)) = 0, \quad \forall P_0 \in \Gamma. \end{aligned}$$

Specifically

$$\theta(\mathbf{K}^\ell) = 0.$$

Therefore it must be that

$$\theta \left(\left[\left[\mathbf{h}_j - \frac{1}{2} \mathbf{A}_{\mathcal{C}}^{\ell} \right] \right] \right) = 0 \quad (6.9)$$

for *at least* one of the 2^{2g} possibilities for $\mathbf{h}_j \in \Lambda_{\frac{1}{2}}$. To date I have been unable to find conditions on Γ and P^{ℓ} that will insure that (6.9) holds for *only* one $\mathbf{h}_j \in \Lambda_{\frac{1}{2}}$, although I suspect such conditions should be attainable. In practice thus far, it has always been the case that *only* one of the 2^{2g} possibilities for \mathbf{K}^{ℓ} yields

$$\theta(\mathbf{K}^{\ell}) = 0.$$

6.3 Computing a canonical divisor

Section 6.2 proposed a way to compute the VRC by computing the possible vectors indicated by

$$\mathbf{K}^{\ell} = \left[\left[\mathbf{h}_j - \frac{1}{2} \mathbf{A}(P^{\ell}, \mathcal{C}) \right] \right] \quad (6.10)$$

where \mathcal{C} is a canonical divisor and $\mathbf{h}_j \in \Lambda_{\frac{1}{2}}$ is unknown. In this section I describe a method for calculating canonical divisors. Note that Example 6 demonstrates an explicit calculation of a canonical divisor.

According to Section 3.7, the differentials in a basis for the cohomology are all of the form

$$\tilde{\omega}_j = \frac{w_j}{F_y} dx \quad (6.11)$$

where the numerator w_j is given by

$$\begin{aligned} w_j &= w_j(x, y) \\ &= \sum_{k+m \leq d-3} c_{jkl} x^k y^l, \end{aligned}$$

and F_y is the y -derivative of the polynomial F giving rise to Γ . The differential $\tilde{\omega}$ is over-tilded to denote that it is not normalized. The valuation divisor of any meromorphic differential, and specifically any *holomorphic* differential, is a canonical divisor [25].

In theory the Abel map of any

$$\nu \in \Omega_{\Gamma}^1$$

is appropriate for calculating (6.10), but to keep the computation time reasonable it is useful to use the differentials resulting from the DvHD algorithm discussed in Section 3.7. This is so for three reasons, two of which lead to the same consequence, namely minimizing distinct places in the divisor. Keeping the number of distinct places in a divisor low saves computational effort as fewer Puiseux expansions, as well as fewer Abel maps to the places those expansions represent, must be computed.

1. The valuation divisors of holomorphic differentials in general are comprised of fewer places than those of meromorphic differentials. The degree of the divisor of a meromorphic differential is $2g - 2$. Thus there must always be $2g - 2$ more zeroes (counting multiplicities) than poles. So to minimize places, minimize poles and choose holomorphic differentials.
2. The DvHD algorithm first computes an integral basis of the functions that are holomorphic on the affine portion of Γ . This basis is “reduced” in a certain sense, and, for non-singular curves, returns differentials where the numerators are monomial. That is, a holomorphic differential on a non-singular curve is of the form

$$\tilde{\omega}_j = \frac{x^k y^l}{F_y} dx, \quad k, l \in \mathbb{N}. \quad (6.12)$$

For such differentials it is easy to see that the only possible zeroes occur at places where $x = 0, \infty$, $y = 0, \infty$ or at places where $dx = 0$. Thus the places to check are restricted to the following.

- **Places over $y = 0$:** The x -points for which $y = 0$ are the x -roots of $F(x, 0) = 0$.
- **Places over $y = \infty$:** If the leading polynomial coefficient $f_0(x)$ of $F(x, y)$ is identically unity, then there are no affine x -points for which $y = \infty$. Otherwise, there are places over the affine roots of $f_0(x) = 0$ for which $y = \infty$.

- **Places over affine discriminant points $x = 0, \infty$:** Computing discriminant points was discussed in Section 5.2.

Once the set of discriminant points with respect to x is computed, finding the places over these points, as well as the places over $x = 0, \infty$ and $y = 0, \infty$ is done using the algorithm to compute Puiseux expansions over x -points.

3. The DvHD algorithm is extremely fast. Further, it has been implemented as a Maple procedure and is already available for use.

If differential $\tilde{\omega}_j$ does not have a monomial numerator, then the situation is slightly more complicated. Say $\tilde{\omega}_j$ is given by (6.11) for a polynomial w_j with at least two terms. In order to find the zeroes of $\tilde{\omega}_j$, the resultant $R(F, w_j)$ of the polynomials F and w is computed with respect to y . As stated in Section 5.2, computing resultants is computationally expensive and often to be avoided, hence the preference for differentials with monomial numerators. The computation of $R(F, w_j)$ yields the set of points in the complex x -plane over which it is possible for

$$F(x, y) = 0$$

and

$$w_j = 0$$

to have simultaneous solutions. The places over each x -point in this set must be computed along with the places over the discriminant points and $x = \infty$.

Whatever the form of the differential $\tilde{\omega}_j$, denote by \mathcal{X}_j the set

$$\mathcal{X}_j = \left\{ x = \alpha \in \mathbb{C} : \exists Q_l^\alpha \in \Gamma : \tilde{\omega} \Big|_{Q_l^\alpha} = 0 \right\} \quad (6.13)$$

of points $x = \alpha$ over which there are places Q_l^α which *may* satisfy

$$\tilde{\omega}_j \Big|_{Q_l^\alpha} = 0. \quad (6.14)$$

Condition (6.14) may be stated, as it appears in the algorithm, in terms of valuations:

$$\begin{aligned}\mathrm{val}(\tilde{\omega}_j, Q_l^\alpha) &= \mathrm{val}\left(\frac{w_j}{F_y} dx, Q_l^\alpha\right) \\ &= \mathrm{val}(w_j, Q_l^\alpha) + \mathrm{val}(dx, Q_l^\alpha) - \mathrm{val}(F_y, Q_l^\alpha) > 0.\end{aligned}$$

Remarks:

- The sets \mathcal{X}_i are perfectly adequate from the point of developing an algorithm. However, when implementing the algorithm I noticed a significant gain in efficiency if \mathcal{X}_i is ordered such that the places over

$$x, y = \infty \quad \text{and} \quad x, y = 0$$

appear first. The efficiency this induces will be pointed out in the notes directly following Algorithm 2.

- Computing the sets \mathcal{X}_j as in (6.13) gets arbitrarily complex for singular curves. Recall from Section 3.7 that imposing regularity at discriminant points imposes conditions on candidate differentials. These conditions get satisfied by taking linear combinations of the candidate differentials. As these linear combinations involve more terms, the resultants computed with them also have more terms, and thus require more time to compute, and have more solutions.
- In principle only one divisor is needed in order to compute the VRC, however, the implementation of the VCR offers an option to validate the computation using divisors arising from all g basis elements of the cohomology.

Algorithm 2. Canonical divisor(s):

\Rightarrow : $\aleph = \text{false}$ (true).

\Leftarrow : a canonical divisor \mathcal{C} (g canonical divisors $\mathcal{C}_1, \dots, \mathcal{C}_g$).

1. initialize $\mathcal{D}_j := 0$; compute $\mathcal{W} := \{\tilde{\omega}_1, \dots, \tilde{\omega}_g\}$ and the sets \mathcal{X}_j

2. **for** $\tilde{\omega}_j \in \mathcal{W}$ **do**

A. for α_k **in** \mathcal{X}_j **do**

 i. $\mathcal{Q} := \{Q_1^{\alpha_k}, Q_2^{\alpha_k}, \dots\}$.

ii. for $Q_l^{\alpha_k}$ **in** \mathcal{Q} **do**

 a. $q_{jkl} := \text{val}(\tilde{\omega}_j, Q_l^{\alpha_k})$.

b. if $q_{jkl} = 0$ **then** increment k and **goto** Step 2.A

else if $q_{jkl} > 0$ **then**

 1. $\mathcal{D}_j := \mathcal{D}_j + q_{jkl}Q_l^{\alpha_k}$

 2. **if** $\deg \mathcal{D}_j = 2g - 2$ **then**

α **if** $\aleph = \text{false}$ **then** return $\mathcal{C} := \mathcal{D}_j$

else remove $\tilde{\omega}_j$ from \mathcal{W} and store $\mathcal{C}_j := \mathcal{D}_j$.

end if

end if (ends II.A.2. b. 2)

 3. **if** $\deg \mathcal{D}_j > 2g - 2$ **then** error **end if**

end if (ends 2.A.ii.b)

end do(ends 2.A.ii)

end do (ends 2.A)

end do (ends 2)

3. **if** $\aleph = \text{false}$ **then** error **end if**

4. **if** for all j it is true that $\deg \mathcal{D}_j = 2g - 2$ **then**

A. return $\{\mathcal{C}_1, \dots, \mathcal{C}_g\} = \{\mathcal{D}_1, \dots, \mathcal{D}_g\}$

else error (continues 4)

end if (ends 4)

Notes on Algorithm 2:

\Rightarrow : The variable \aleph is a flag: if \aleph is **false**, then one canonical divisor is computed. Otherwise Algorithm 2 attempts to compute all g canonical divisors arising from the basis holomorphic differentials resulting from the DvHD algorithm.

1 The variables \mathcal{D}_j are initialized to zero for

$$j = 1, \dots, g.$$

This variable will hold the divisors as they are constructed. Computing holomorphic differentials is discussed in Section 3.7.

Computing the sets \mathcal{X}_j as in (6.13) is valid, but a time savings is realized if a particular order is imposed. Further savings are realized if the discriminant points and zeroes of the numerators of the differentials are only computed as needed. That is, the first time through step 2.A, $\alpha_1 := \infty$ and the places over ∞ are computed. If there is a basis holomorphic differential that has $2g - 2$ zeroes (including multiplicities) at places over $x = \infty$ and only one divisor is requested, then the algorithm terminates in Step 2.A.ii.b. 2. α . If a second time through Step 2 is needed the places over $y = \infty$ are checked. Thereafter the discriminant points, and points for which it may be that $w_j = 0$, are checked in turn. While it is certainly possible to construct a curve such that this amendment will make the algorithm perform *worse*, it has been my experience that for curves with a “reasonable” number of singularities, a savings is realized.

- 2** The differential $\tilde{\omega}_j$ is removed from the set \mathcal{W} at Step 2.A.ii.b. **2.** α when $2g - 2$ zeroes of $\tilde{\omega}_j$ have been found. If $\aleph = \text{false}$ the algorithm terminates, and if not it continues until all the differentials have been removed. Once all the differentials have been removed control passes to Step 3.

2.A Please see the notes for Step 1.

- 2.A.ii.a** Computing q_{jkl} , the valuation of the j -th differential at the l -th place over the k -th point $\alpha_k \in \mathcal{X}_j$, is done in pieces. Recall that q_{jkl} is given by

$$\begin{aligned} q_{jkl} &:= \text{val}(\tilde{\omega}_j, Q_l^{\alpha_k}) \\ &= \text{val}\left(\frac{w_j}{F_y} dx, Q_l^{\alpha_k}\right) \\ &= \text{val}(w_j, Q_l^{\alpha_k}) + \text{val}(dx, Q_l^{\alpha_k}) - \text{val}(F_y, Q_l^{\alpha_k}). \end{aligned}$$

This also increases efficiency as the valuations of the differential dx and the y -derivative of F must be computed only once for each place.

- 2.A.ii.b** A **goto** statement is not actually used, but the control flow is the same as if one had been used.

- 2.A.ii.b. 1** Update the j -th temporary valuation divisor to contain place $Q_l^{\alpha_k}$ q_{jkl} times.

- 2.A.ii.b. 1. α** If $\aleph = \text{false}$ then only one divisor is required. The algorithm terminates after returning

$$\mathcal{C} := \mathcal{D}_j.$$

If $\aleph = \text{true}$ the algorithm continues until $2g - 2$ zeroes are found for each of the g basis differentials.

- 2.A.ii.b. 3** If more than $2g - 2$ places are found then an error has certainly occurred as the differentials in \mathcal{W} are holomorphic. In an ideal world this check is unnecessary, but is nonetheless preformed as it is computationally inexpensive.

- 3 If only one divisor is desired, then this step is only reached if no differential is found with at least $2g - 2$ zeroes. Again, this check should not be needed, but it is performed as it is not computationally expensive.
- 4 If the conditional in this step is true, then g differentials have been found, each with $2g - 2$ zeroes. Each divisor has the appropriate degree, and all possible places for zeroes have been checked.

6.4 Approximating the θ -function

As seen in the previous section, computing the VRC relies on computing values of the Riemann θ -function. Specifically I wish to test (6.9) for the 2^{2g} possibilities for $\mathbf{h}_j \in \Lambda_{\frac{1}{2}}$. Calculating an exact value of a g -phase θ -function requires calculating a g -tuple infinite sum. This is generally not possible. Further, if as in this application the matrix \mathbf{B} is known only approximately, then exact calculation of $\theta(\mathbf{z})$ is not possible for general \mathbf{z} . Computing approximate values of θ -functions is the subject of [14], and the algorithm presented here to compute the VRC relies on the methods therein. In particular, let

$$\mathcal{K}(P_0) = \left\{ \boldsymbol{\kappa}_j : \boldsymbol{\kappa}_j = \left\lfloor \mathbf{h}_j - \frac{1}{2} \mathbf{A}(P_0, \mathcal{C}) \right\rfloor, \quad \mathbf{h}_j \in \Lambda_{\frac{1}{2}}, \quad j = 1, \dots, 2^{2g} \right\}$$

be the set of candidates for the VRC, $\mathbf{K}(P_0)$. If it can be established that

$$\theta(\boldsymbol{\kappa}_j) \approx 0$$

$$\theta(\boldsymbol{\kappa}_k) \not\approx 0$$

$$k = 1, \dots, \widehat{j}, \dots, 2^{2g},$$

where \widehat{j} denotes that this j is omitted from the list, then as

$$\mathbf{K} \in \Theta,$$

it must be that

$$\mathbf{K}(P_0) = \boldsymbol{\kappa}_j.$$

In [14] it is shown that, given an error bound ϵ , it is possible to construct a set

$$\mathcal{N}_\epsilon = \{\mathbf{n}_1, \mathbf{n}_2, \dots\} \quad (6.15)$$

of integer vectors

$$\mathbf{n}_1, \mathbf{n}_2, \dots$$

such that

$$\vartheta(\mathbf{z}|\mathbf{B}) \approx \sum_{\mathbf{n} \in \mathcal{N}_\epsilon} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{B}\mathbf{n} + \mathbf{z} \cdot \mathbf{n})}, \quad (6.16)$$

where $\vartheta(\mathbf{z}|\mathbf{B})$ is the *purely oscillatory* part of $\theta(\mathbf{z}|\mathbf{B})$ and the absolute error of the sum (6.16) is bounded above by ϵ . As will be shown shortly, the purely oscillatory part ϑ has the same zero set as the θ -function itself. That is

$$\vartheta(\mathbf{z}) = 0$$

implies

$$\mathbf{z} \in \Theta.$$

I will state without proof the relevant theorem from [14], however some preliminaries are needed. Recall that, given a Riemann matrix \mathbf{B} associated with the surface Γ , the definition of the θ -function is

$$\theta(\mathbf{z}|\mathbf{B}) = \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{B}\mathbf{n} + \mathbf{z} \cdot \mathbf{n})}, \quad (6.17)$$

where the summation is over integer vectors \mathbf{n} in \mathbb{R}^g . Following [14], let

$$\mathbf{z} = \mathbf{x} + i\mathbf{y}, \quad \text{and} \quad \mathbf{B} = \mathbf{X} + i\mathbf{Y}$$

be the decompositions into real and imaginary parts. Then, because $\Im(\mathbf{n}) = \mathbf{0}$, Equation (6.17) is

$$\begin{aligned} \theta(\mathbf{z}|\mathbf{B}) &= \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x}) - 2\pi(\frac{1}{2}\mathbf{n} \cdot \mathbf{Y}\mathbf{n} + \mathbf{n} \cdot \mathbf{y})} \\ &= \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-2\pi(\frac{1}{2}\mathbf{n} \cdot \mathbf{Y}\mathbf{n} + \mathbf{n} \cdot \mathbf{y})} \\ &= \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-\pi(\mathbf{n} \cdot \mathbf{Y}\mathbf{n} + \mathbf{n} \cdot \mathbf{y} + \mathbf{n} \cdot \mathbf{y} + \mathbf{y} \cdot \mathbf{Y}\mathbf{y} - \mathbf{y} \cdot \mathbf{Y}\mathbf{y})}, \end{aligned}$$

(recalling that $\mathbf{n} \cdot \mathbf{y} = \mathbf{y} \cdot \mathbf{n}$)

$$= \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-\pi(\mathbf{n} \cdot \mathbf{Y}\mathbf{n} + \mathbf{Y}^{-1}\mathbf{Y}\mathbf{y} \cdot \mathbf{n} + \mathbf{n} \cdot \mathbf{y} + \mathbf{y} \cdot \mathbf{Y}\mathbf{y} - \mathbf{y} \cdot \mathbf{Y}\mathbf{y})}.$$

As $\mathbf{Y}^t = \mathbf{Y}$

$$\begin{aligned} &= \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-\pi(\mathbf{n} \cdot \mathbf{Y}\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y} \cdot \mathbf{Y}\mathbf{n} + \mathbf{n} \cdot \mathbf{y} + \mathbf{y} \cdot \mathbf{Y}\mathbf{y} - \mathbf{y} \cdot \mathbf{Y}\mathbf{y})} \\ &= \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-\pi((\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y}) \cdot \mathbf{Y}(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y}) - \mathbf{y} \cdot \mathbf{Y}\mathbf{y})} \\ &= e^{\pi \mathbf{y} \cdot \mathbf{Y}\mathbf{y}} \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-\pi(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y}) \cdot \mathbf{Y}(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y})}. \end{aligned}$$

As \mathbf{Y} is strictly positive definite, \mathbf{Y}^{-1} exists and is strictly positive definite [14]. Thus the general term in the sum is a purely oscillatory factor:

$$e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})}$$

multiplied by a factor:

$$e^{-\pi(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y}) \cdot \mathbf{Y}(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y})}$$

that is exponentially damped because

$$\mathbf{v} \cdot \mathbf{Y}\mathbf{v} > 0$$

for real \mathbf{Y} and \mathbf{v} . The overall factor

$$e^{\pi \mathbf{y} \cdot \mathbf{Y}\mathbf{y}} \tag{6.18}$$

accounts for all the exponential growth of the θ -function. Define the purely oscillatory part ϑ as

$$\vartheta(z|\mathbf{B}) = \sum_{\mathbf{n} \in \mathbb{Z}^g} e^{2\pi i(\frac{1}{2}\mathbf{n} \cdot \mathbf{X}\mathbf{n} + \mathbf{n} \cdot \mathbf{x})} e^{-\pi(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y}) \cdot \mathbf{Y}(\mathbf{n} + \mathbf{Y}^{-1}\mathbf{y})} \tag{6.19}$$

$$= e^{-\pi \mathbf{y} \cdot \mathbf{Y}\mathbf{y}} \theta(z|\mathbf{B}). \tag{6.20}$$

Note that as the overall factor (6.18) is non-zero, the zero structure of θ and ϑ are identical.

Let $[\mathbf{w}]$ denote the vector nearest \mathbf{w} with integer components. Further, let

$$\mathbf{Y} = \mathbf{T}^t \mathbf{T}$$

be the Cholesky decomposition of \mathbf{Y} . It is now possible to state a slight rephrasing of the Uniform Approximation Theorem from [14].

Uniform Approximation Theorem. *The purely oscillatory part of the Riemann theta function can be uniformly approximated in $\mathbf{z} = \mathbf{x} + i\mathbf{y}$ by*

$$\vartheta(\mathbf{z}|\mathbf{B}) = \sum_{\mathbf{n} \in \mathcal{N}_\epsilon} e^{\pi i \mathbf{v}(\mathbf{n}) \cdot \mathbf{X} \mathbf{v}(\mathbf{n})} e^{2\pi i \mathbf{v}(\mathbf{n}) \cdot \mathbf{x}} e^{-\|\mathbf{v}'(\mathbf{n})\|^2}. \quad (6.21)$$

where

$$\begin{aligned} \mathbf{v}(\mathbf{n}) &= \mathbf{n} - [\mathbf{Y}^{-1} \mathbf{y}] \\ \mathbf{v}'(\mathbf{n}) &= \sqrt{\pi} \mathbf{T} (\mathbf{v}(\mathbf{n}) + \mathbf{Y}^{-1} \mathbf{y}), \end{aligned}$$

$$\mathcal{N}_\epsilon = \{\mathbf{n} \in \mathbb{Z}^g : \pi (\mathbf{n} - \mathbf{c}) \cdot \mathbf{Y} (\mathbf{n} - \mathbf{c}) < R^2(\epsilon)\},$$

and

$$\mathbf{c} = (c_1, \dots, c_g), \quad c_j \in \left(-\frac{1}{2}, \frac{1}{2}\right), \quad j = 1, \dots, g.$$

The radius $R(\epsilon)$ is the greater of

$$\frac{\sqrt{2g} + \rho}{2}$$

and the real positive solution of

$$\epsilon = \frac{g^{2g-1}}{\rho^g} \Gamma\left(\frac{g}{2}, \left(R - \frac{\rho}{2}\right)^2\right),$$

where Γ is the incomplete Gamma function. The value ρ is the distance of the shortest lattice vector in the lattice given by

$$\{\mathbf{v}'(\mathbf{n}) : \mathbf{n} \in \mathbb{Z}^g\}.$$

The absolute error in the sum in (6.21) is bounded by ϵ .

Thus it is possible to compute the Riemann θ -function with a controlled error. The algorithms presented in [14] are implemented in Maple and available in versions 9 and greater under the name `RiemannTheta`. Since Maple allows for arbitrary precision computation, this means that θ -functions can be computed with arbitrarily small error. It is clear that the number of terms in the finite sum (6.21) will tend to grow exponentially in the size g of the matrix \mathbf{B} , that is, in the genus of the underlying surface Γ . This leads to a practical limit for some combination of genus and accuracy. In the current implementation of the VRC, this limit is pushed back by compiling the procedure to compute the finite sum in (6.21).

6.5 Computing the correct half-lattice vector

I present an outline of the algorithm for computing the vector of Riemann constants $\mathbf{K}(P^\ell)$ of the left place P^ℓ . Recall that once \mathbf{K}^ℓ has been computed, $\mathbf{K}(P_0)$ follows as

$$\mathbf{K}(P_0) = \mathbf{K}^\ell - (g - 1)\mathbf{A}(P_0, P^\ell)$$

for any $P_0 \in \Gamma$. The function of the following algorithm is to determine, for a particular canonical divisor \mathcal{C} , the correct half-lattice vector \mathbf{h}_j such that

$$\mathbf{K}^\ell = \left\lfloor \mathbf{h}_j - \frac{1}{2}\mathbf{A}(P^\ell, \mathcal{C}) \right\rfloor$$

Note that this is an *exact* computation. That is, there is no error in \mathbf{h}_j , aside from that the computation of the matrix \mathbf{B} . Further, once \mathbf{h}_j is known, the algorithm to compute the VRC may be trivially stated, and is as accurate as the algorithm to compute the Abel map.

The basic idea of Algorithm 3 is to use three “filter passes” to eliminate incorrect half-lattice vectors. Many choices were made heuristically when developing this algorithm. For instance, the choice of using three passes, as well as the choice of filtering mechanism in each pass, was made after much trial-and-error. There are of course infinitely many ways to fine tune the filtration mechanism. The over-arching approach that I use is to attempt to keep any particular filter pass from taking an order of magnitude more effort than the others. This seems, at least in testing, to keep the computational effort reasonable.

Steps 3 through 5 in Algorithm 3 make up the first filter pass. Likewise, Steps 6 through 8 make up the second filter pass. The third filter pass is encompassed by Step 9.

Algorithm 3. The half-lattice vector:

\Rightarrow : a Riemann surface Γ

\Leftarrow : a canonical divisor \mathcal{C} and half-lattice vector \mathbf{h}

1. compute tolerances ϵ_1, ϵ_2 , divisor \mathcal{C} and vector

$$\mathbf{Z} = \frac{1}{2}\mathbf{A}(P^\ell, \mathcal{C})$$

2. $\mathcal{J} = (1, \dots, 2^{2g})$

3. compute list \mathcal{N}_{ϵ_1} and construct finite-sum ϑ , denoted by T_{ϵ_1}

4. **for** j **in** \mathcal{J} **do**

A. $\kappa_j := \llbracket \mathbf{h}_j - \mathbf{Z} \rrbracket$

B. **if** $\|T_{\epsilon_1}(\kappa_j)\| > \epsilon_1$ **then** remove j from \mathcal{J} **end if**

end do (ends 4)

5. **if** \mathcal{J} empty **then** error

else if \mathcal{J} has exactly one element j^* **then** return \mathcal{C} and \mathbf{h}_{j^*}

end if (ends 5)

6. compute \mathcal{N}_{ϵ_2} , divisor \mathcal{D}_0 and construct T_{ϵ_2}

7. **for** j **in** \mathcal{J} **do**

A. $\tilde{\kappa}_j := \llbracket \mathbf{h}_j - \mathbf{Z} + \mathbf{A}(P^\ell, \mathcal{D}_0) \rrbracket$

B. **if** $\|T_{\epsilon_2}(\tilde{\kappa}_j)\| > \epsilon_2$ **then** remove j from \mathcal{J} **end if**

end do (ends 7)

```

8. if  $\mathcal{J}$  empty then error

    else if  $\mathcal{J}$  has exactly one element  $j^*$  then return  $\mathcal{C}$  and  $\mathbf{h}_{j^*}$ 

    end if (ends 8)

9. while  $\mathcal{J}$  is not empty do

    A. compute  $\mathcal{D}_k$ 

    B. for  $j$  in  $\mathcal{J}$  do

        i.  $\tilde{\boldsymbol{\kappa}}_j := \llbracket \mathbf{h}_j - \mathbf{Z} + \mathbf{A}(P^\ell, \mathcal{D}_k) \rrbracket$ 

        ii. if  $\|T_{\epsilon_2}(\tilde{\boldsymbol{\kappa}}_j)\| > \epsilon_2$  then remove  $j$  from  $\mathcal{J}$  end if

        iii. if  $\mathcal{J}$  empty then error

            else if  $\mathcal{J}$  has exactly one element  $j^*$  then return  $\mathcal{C}$  and  $\mathbf{h}_{j^*}$ 

            end if (ends 9.B.iii)

    end do (ends 9)

10. error

```

Notes for Algorithm 3:

\Rightarrow : The surface Γ is the only argument for the algorithm. The surface is given as a polynomial in two variables.

\Leftarrow : If the method is successful and exactly one half-lattice \mathbf{h} is found then it and the divisor \mathcal{C} are returned. Otherwise an error is returned.

1 Computing the tolerances ϵ_1 and ϵ_2 is discussed in Section 6.6. For now let it suffice to say that these values are computed so as to balance the amount of computational effort done by the first and second filter passes. The goal is to compute ϵ_1 in order to first evaluate ϑ somewhat inaccurately, but cheaply. The inexpensive evaluations filter out candidates that are definitely not zero.

Algorithm 2 is used to compute a canonical divisor.

Computing the Abel map is the subject of Chapter 5. The Abel map is computed with error tolerance ϵ_2 .

2 The list

$$\mathcal{J} = (1, \dots, 2^{2g})$$

is used to keep track of which \mathbf{h}_j are still candidates. The half-lattice vector \mathbf{h}_j is constructed based on the $2g$ digit binary representation of the number j . For example, if $g = 4$, there are

$$2^8 = 256$$

possible half-lattice vectors. Suppose $j = 123$, the binary representation of which is

$$1111011.$$

As this has only 7 digits, pad it with a zero at the front to form the 8 element vector \mathbf{j}_{bin} as

$$\mathbf{j}_{\text{bin}} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}^t.$$

Then \mathbf{h}_{123} is given by

$$\mathbf{h}_{123} = \frac{1}{2} \begin{pmatrix} \mathbf{1} & \mathbf{B} \end{pmatrix} \mathbf{j}_{\text{bin}}.$$

Elements of \mathcal{J} are removed in Steps 4.B, 7.B and 9.B.ii if

$$T(\boldsymbol{\kappa}_j) \approx 0$$

for the induced candidate $\boldsymbol{\kappa}_j$.

- 3** The list \mathcal{N}_ϵ as in (6.15) is computed using the algorithm discussed in [14]. The length of \mathcal{N}_ϵ grows exponentially in the genus of the underlying curve due purely to the increase in the dimension of \mathbb{C}^g . As ϵ is decreased, the radius $R(\epsilon)$ increases. The

volume enclosed by the bounding ellipsoid increases like $R^{(2g)}$, so of course increasingly more vectors are added to \mathcal{N}_ϵ when asking for higher accuracy.

Denote by T_ϵ the finite-sum version of the purely oscillatory part of the θ -function, given explicitly as

$$T_\epsilon(\mathbf{z}|\mathbf{B}) = \sum_{\mathbf{n} \in \mathcal{N}_\epsilon} e^{\pi i \mathbf{v}(\mathbf{n}) \cdot \mathbf{X} \mathbf{v}(\mathbf{n})} e^{2\pi i \mathbf{v}(\mathbf{n}) \cdot \mathbf{x}} e^{-\|\mathbf{v}'(\mathbf{n})\|^2}.$$

The computational time for this algorithm is essentially all spent in the evaluation of T_ϵ . For a fixed genus, the evaluation time increases significantly with decreasing ϵ due to the additional vectors in \mathcal{N}_ϵ . For this reason the algorithm recursively “filters” the candidates κ_j . By judiciously choosing ϵ_1 on the first pass, many candidates are eliminated with relatively little effort.

In the Maple implementation of this algorithm I use the command `RiemannTheta` to compute \mathcal{N}_ϵ . I developed, following [14], Maple code to construct and compile T_ϵ . There is a (highly) reduced set of functions that Maple can compile on-the-fly. One may write Maple functions using this reduced set, and then have Maple compile the function. Note that T_{ϵ_1} and T_{ϵ_2} are not the same function, in either the compiled or non-compiled form. The compiling time is not trivial, but the compiled code evaluates significantly faster. The down side to using compiled code, aside from the overhead incurred by compiling, is that accuracy is limited to double precision floating point numbers. Further, writing Maple code to compile functions is *extremely* unpleasant.

5 If \mathcal{J} is empty, then no vector κ_j was found such that

$$\vartheta(\kappa_j) \approx 0.$$

The error message returned reflects this shortcoming. If \mathcal{J} has exactly one element, say \hat{j} , then $\mathbf{h}_{\hat{j}}$, as well as the canonical divisor \mathcal{C} that was used to find it, are returned. Note that the half-lattice vector $\mathbf{h}_{\hat{j}}$ is an exact result.

6 The list \mathcal{N}_{ϵ_2} is computed in direct analogy with Step 3.

The effective, degree $g - 1$ divisor \mathcal{D}_0 is chosen as

$$\mathcal{D}_0 = P_1 + \cdots + P_{g-1}, \quad P_k \neq P_l, \quad k \neq l,$$

where all the places P_k are regular places. More detail is given in Section 6.6, and an example computation is given in Section 8.2.

7.A Recall that Proposition 3.15.2 says that

$$\theta(\mathbf{W}) = 0$$

if and only if

$$\mathbf{K}(P^\ell) + \mathbf{A}(P^\ell, \mathcal{D}_k),$$

where $\mathbf{K}(P^\ell)$ is the VRC and $\mathcal{D}_k \in S^{g-1}\Gamma$. Thus, if the candidate $\boldsymbol{\kappa}_j$ actually *is* the VRC, then the perturbed candidate $\tilde{\boldsymbol{\kappa}}_j$ given by

$$\tilde{\boldsymbol{\kappa}}_j := \left\llbracket \mathbf{h}_j - \mathbf{Z} + \mathbf{A}(P^\ell, \mathcal{D}_0) \right\rrbracket \quad (6.22)$$

must also be in Θ . Thus it *must* be so that

$$\theta(\tilde{\boldsymbol{\kappa}}_{j^*}) < \epsilon_2$$

for the correct half-lattice vector \mathbf{h}_{j^*} . On the other hand, it is only *possible* this is so for other \mathbf{h}_j . Thus perturbing the candidates makes it very unlikely that a candidate vector that is not the VRC will evaluate (under T_ϵ) close to zero on both the first and second passes. This is demonstrated explicitly in Section 8.2.

9.A When computing $\mathbf{A}(P^\ell, \mathcal{D}_0)$ in Step 7.A, the Abel map $\mathbf{A}(P_0, P_j)$ of each place P_j was computed. Thus the Abel map of any divisor consisting of only the places

$$P_1, \dots, P_{g-1} \in \mathcal{D}_0$$

may be computed as a simple vector sum. Specifically, any divisor \mathcal{D}_k given by

$$\begin{aligned} \mathcal{D}_k &= m_1 P_1 + \cdots + m_{g-1} P_{g-1} \\ m_1, \dots, m_{g-1} &\in \mathbb{N} \end{aligned} \quad (6.23)$$

$$m_1 + \cdots + m_{g-1} = g - 1$$

may be easily computed.

9.B.ii It would seem sensible to compute a separate tolerance ϵ_3 for this filter pass, but experience has shown this not to be needed. Approximate absolute values are computed to accuracy ϵ_2 again in this step.

10 If this step is reached, then more than one candidate half-lattice vectors remain. Return an error to this effect.

The algorithm to compute the VRC is quite simple to outline, and for the sake of completeness I do so below.

Algorithm 4. The vector of Riemann constants:

\Rightarrow : a Riemann surface Γ , place $P_0 \in \Gamma$ and requested accuracy ϵ

\Leftarrow : vector of Riemann constants $\mathbf{K}(P_0)$

1. compute Riemann matrix \mathbf{B}
2. compute canonical divisor \mathcal{C} and half-lattice vector \mathbf{h} .
3. compute $\mathbf{A}(P_0, P^\ell)$ to accuracy ϵ
4. return $\mathbf{K}(P_0)$ given by the system

$$K_j(P_0) = \frac{1 + B_{jj}}{2} - (g - 1)A(P_0, P^\ell) - \left(\left\| \mathbf{h} - \mathbf{A}(P^\ell, \mathcal{C}) \right\| \right)_j.$$

Note that any error messages that may occur do not appear in the outline of Algorithm 4. Should errors occur, the user is informed by either Algorithm 2 or 3.

6.6 Filtering candidate half-lattice vectors

Algorithm 3 makes use of three filter passes to eliminate incorrect candidate half-lattice vectors. The three passes each use two filtering mechanisms.

- The first filter mechanism is the approximated ϑ function.
- The second filter mechanism is the zero set, Θ , of the ϑ -function.

As stated earlier, the decision to use three filter passes is heuristic, and perhaps there are better choices. So too with the way in which the two different mechanisms are balanced within each of the three passes. In the remainder of this section I will outline how the first of the filter mechanisms is currently implemented, and present some justification for the choices made.

Recall that ϑ denotes the purely oscillatory part of the Riemann θ -function as explained in Section 6.4. Further recall the notation T_ϵ from Algorithm 3 to denote the finite sum approximation of the ϑ -function which is accurate to within ϵ .

The goal of computing ϵ_1 and ϵ_2 effectively is to minimize the time needed to compute the half-lattice vector. On the first pass of Algorithm 6.4 *all* 2^{2g} vectors must be tested, but by choosing a relatively high value for ϵ_1 , the evaluations of T_{ϵ_1} are kept relatively inexpensive. The relatively high value of ϵ_1 can still be used to eliminate vectors for which T_{ϵ_1} definitely does not evaluate to zero. The value of ϵ_2 is chosen to make it *almost certain* that no vector for which $\|T_{\epsilon_2}\| \neq 0$ is passed through the filter.

Inspection of (6.21) shows that the number of operations for each ϑ_Σ evaluation is governed by the evaluation of the quadratic form

$$\mathbf{v}(\mathbf{n}) \cdot \mathbf{X}\mathbf{v}(\mathbf{n})$$

in the exponential, and the number of vectors in the \mathcal{N}_ϵ . The computational cost for the evaluation of the quadratic form is only a function of the genus g , and scales as g^4 . The number of vectors in \mathcal{N}_ϵ increases essentially linearly with the volume of the bounding ellipsoid with “radius” $R(\epsilon)$. Thus the number of operations per evaluation of T_ϵ scales as $g^4 R^g(\epsilon)$.

Denote by

$$\text{Op}_1 \quad \text{and} \quad \text{Op}_2$$

the approximate number of floating-point operations performed on the first and second filter passes in Algorithm 3 respectively. The number Op_1 and Op_2 scale with the genus g and with the absolute errors ϵ_1 and ϵ_2 . As the genus is fixed for any particular computation, the scaling with ϵ_j , $j = 1, 2$ is the primary concern. Ignoring the operations to compute the quadratic form and the exponential, which are the same for each pass, the number of floating-point operations for the first pass scales like

$$\text{Op}_1 \sim \underbrace{2^{2g}}_{\#\mathcal{N}_{\epsilon_1}} \underbrace{R^g(\epsilon_1)}_{\#\mathcal{J}}, \quad (6.24)$$

where \mathcal{J} is the set of indices of the candidate vectors and $R^g() = (R())^g$. On the second pass the number of operations scales like

$$\text{Op}_2 \sim \frac{2^{2g}}{\sigma(\epsilon_1)} R^g(\epsilon_2), \quad (6.25)$$

where $\sigma(\epsilon_1)$ is the factor by which the number of vectors in \mathcal{N}_ϵ is reduced by the first pass.

As the absolute error in the evaluation of T_{ϵ_1} is ϵ_1 , if

$$\left\| T_{\epsilon_1} \left(\left\lfloor \mathbf{h}_j - \frac{1}{2} \mathbf{A}_C \right\rfloor \right) \right\| > \epsilon_1,$$

then

$$\vartheta \left(\left\lfloor \mathbf{h}_j - \frac{1}{2} \mathbf{A}_C \right\rfloor \right) \neq 0.$$

The factor $\sigma(\epsilon_1)$ depends on the probability

$$\text{prob} \left(\left\| T_\epsilon \left(\left\lfloor \mathbf{h}_j - \frac{1}{2} \mathbf{A}_C \right\rfloor \right) \right\| \leq \epsilon \mid \mathbf{h}_j \in \Lambda_{\frac{1}{2}} \right). \quad (6.26)$$

That is, the probability that the absolute value of $T_\epsilon(\boldsymbol{\kappa}_j)$ is less than ϵ for the candidate VRC induced by the half-lattice vector \mathbf{h}_j . I make the assumption that (6.26) can be approximated by ϵ for all \mathbf{h}_j *except* the correct one \mathbf{h}_{j^*} , for which

$$\left\lfloor \mathbf{h}_{j^*} - \frac{1}{2} \mathbf{A}_C \right\rfloor = \mathbf{K}(P^\ell).$$

Thus I assume

$$\text{prob} \left(\left\| T_\epsilon \left(\left\lfloor \mathbf{h}_j - \frac{1}{2} \mathbf{A}_C \right\rfloor \right) \right\| \leq \epsilon \mid \mathbf{h}_j \in \Lambda_{\frac{1}{2}} \right) = \epsilon. \quad (6.27)$$

To make this approximation I assume the following.

- Γ and $P^\ell \in \Gamma$ are *not* such that

$$\vartheta(\mathbf{K}(P^\ell) + \mathbf{h}_j) = 0, \quad \mathbf{0} \neq \mathbf{h}_j \in \Lambda_{\frac{1}{2}}. \quad (6.28)$$

Said another way: although the VRC

$$\mathbf{K}(P^\ell) \in \Theta$$

by Proposition 3.15.2, I assume that $\mathbf{K}(P^\ell)$ is not exactly a non-zero half-lattice vector away from any other vector in Θ . If it does happen that (6.28) is satisfied for some half-lattice vector, then Algorithm 3 will return an error message, and not an incorrect result.

- The absolute value

$$\left\| T_{\epsilon_1} \left(\left\| \mathbf{h}_j - \frac{1}{2} \mathbf{A}_c \right\| \right) \right\|$$

is a random variable approximately uniformly distributed on $[0, 1]$ as a function of the index j . The absolute value of ϑ is an order 1 quantity [14], although it is *not* restricted to the unit interval. While this assumption is heuristic, it leads to an underestimate of the factor $\sigma\epsilon_1$ in most cases.

The argument used to minimize the computational effort expended by the first and second filter passes is this. Both passes 1 and 2 are expensive: Pass 1 because of the 2^{2g} candidate vectors in $\Lambda_{\frac{1}{2}}$, and Pass 2 due to the large unknown number of vectors in \mathcal{N}_{ϵ_2} needed for accurate computation. Choosing ϵ_1 and ϵ_2 such that

$$\text{Op}_1 \sim \text{Op}_2$$

means that the effort is essentially balanced, and thus that neither pass is acting as a bottleneck to the computation. I am aware that this is not actually an optimal condition in most cases, but experience has shown this choice to work. Thus, ϵ_2 is chosen such that it is very unlikely that

$$\|T_{\epsilon_2}(\boldsymbol{\kappa}_j)\| < \epsilon_2$$

for any incorrect candidate $\boldsymbol{\kappa}_j$. Assuming the statistics of $T_{\epsilon_2}(\boldsymbol{\kappa}_j)$ given by (6.27) leads to the choice

$$\epsilon_2 = \frac{1}{2^{2g}}$$

Then ϵ_1 is chosen such that

$$\begin{aligned}\text{Op}_1 &\sim \text{Op}_2 \\ 2^{2g} g^4 R^g(\epsilon_1) &\sim \frac{2^{2g}}{\sigma(\epsilon_1)} g^4 R^g(\epsilon_2) \\ R^g(\epsilon_1) &\sim \frac{R^g(\epsilon_2)}{\sigma(\epsilon_1)}\end{aligned}$$

Again using the statistics assumed in (6.27), the factor $\sigma(\epsilon_1)$ can be approximated by ϵ_1 .

Thus ϵ_1 is chosen so

$$\epsilon_1 R^g(\epsilon_1) \sim R^g(\epsilon_2) \quad (6.29)$$

Recall that $R(\epsilon)$ is given by the maximum of

$$\frac{\sqrt{2g} + \rho}{2}$$

and the real positive solution of

$$\epsilon = \frac{g2^{g-1}}{\rho^g} \Gamma\left(\frac{g}{2}, \left(R - \frac{\rho}{2}\right)^2\right),$$

with Γ the incomplete Gamma-function, and ρ a known quantity based on the Riemann matrix. Thus $R^g(\epsilon_2)$, the right-hand side of (6.29), can be computed and the equations

$$\epsilon_1 \frac{\sqrt{2g} + \rho}{2} = R^g(\epsilon_2) \quad (6.30)$$

and

$$\epsilon_1 \frac{g2^{g-1}}{\rho^g} \Gamma\left(\frac{g}{2}, \left(R - \frac{\rho}{2}\right)^2\right) = R^g(\epsilon_2) \quad (6.31)$$

solved for ϵ_1 numerically (using Maple, for instance). The tolerance ϵ_1 is then chosen to be the root of (6.30) if, for the resulting value of ϵ_1 ,

$$\frac{\sqrt{2g} + \rho}{2} > \frac{g2^{g-1}}{\rho^g} \Gamma\left(\frac{g}{2}, \left(R - \frac{\rho}{2}\right)^2\right) = R^g(\epsilon_2),$$

and the root of (6.31) otherwise. As a final remark, if the result is that $\epsilon_1 > 0.1$ a value of $\epsilon_1 = 0.1$ is instead chosen. This is an *ad hoc* cut-off, which may be removed in later versions of the implementation.

Chapter 7

THE IMPLEMENTATION OF THE ABEL MAP: COMPUTATION
AND VERIFICATION**7.1 Syntax of the implementation**

In this chapter I provide examples of the algorithm to compute the Abel map by means of a Maple implementation. This implementation of the Abel map is included in the regular distribution of Maple 11 (release: Summer 2007). Updated versions may be obtained at <http://www.amath.washington.edu/~bernard/papers.html>. The examples also serve to verify that these algorithms and implementations are performing as desired. The Maple implementation is called in Maple with the syntax

```
>AbelMap(F, x, y, P1, P2, digits);
```

- **F** is a polynomial in only the variables **x** and **y**. The coefficients of **F** may not be floating point numbers, but may be irrational if entered as radicals or using the Maple `RootOf` notation.
- **P1** and **P2** are places on the Riemann surface obtained from **F**. Places are entered in three ways.

– **Regular places:** Places over regular points are entered as

$$(x = \alpha, y = \beta)$$

pairs. For example, on the curve arising from

$$F = y^4 + x^4 - 1,$$

the point $x = 2$ is regular, so the place(s) over 2 may be entered in Maple notation (see Appendix) as

```
>P1 := [x = 2, y = RootOf(_Z^4 + 15)];.
```

- **Discriminant places and ∞ :** Places over discriminant points and infinity must be entered as truncated Puiseux expansions in the variable \mathbf{t} . Further, the y -series must contain enough terms to distinguish the intended place from all other places over the same point. The curve in Example 3 illustrates this point. There are four places on this curve over $x = 0$, and for three of these $y = 0$ also. Further, for two of these places, the behavior of the x -component is identical, thus the behavior of the y -component must be entered explicitly to distinguish the places. For instance, P_2 would be entered as

```
>P2 := [x = t, y = 2*t];
```

It is possible that an arbitrary finite number of terms of the y -components must be specified to distinguish between places. It is highly recommended that users of the Maple implementation use the command `algcures[puisseux]` to compute places over discriminant points and ∞ .

- **The algorithmically chosen left place:** The flag ‘ZERO’ signifies that the algorithmically chosen place P^ℓ is to be used. This option is useful when computing the Abel map of a divisor independent of the place P' .

- **digits** is an integer declaring the desired number of correct digits. The implementation returns a vector of g complex numbers. Each component of the numerically computed vector should share **digits** significant figures, in both real and imaginary parts, with the symbolically calculated component.

For those unfamiliar with Maple, Appendix B explains the syntax and commands needed to understand the examples below.

7.2 Demonstrating convergence

The genus four Hurwitz curve given by

$$F = y^4 + x^4 - 1$$

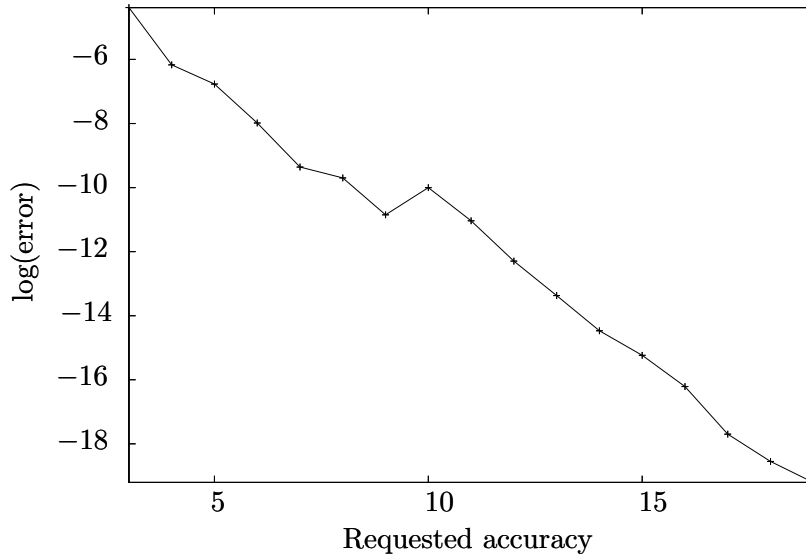


Figure 7.1: A plot of the convergence of the Maple implementation. The j -th data point shows the logarithm of the difference between an approximation with j digits and $j + 1$ digits of requested accuracy. The “hiccup” in the graph is due to the algorithm obtaining better than expected accuracy for `digits` = 9 and only expected accuracy for `digits` ≥ 10 .

is not hyperelliptic, and has branch points

$$1, i, -1, -i.$$

In Figure 7.1, I demonstrate that the Maple implementation of the Abel map produces a Cauchy sequence as the required accuracy is increased. This figure is created by first computing a sequence

$$\mathbf{A}^{(2)}(P^\ell, i), \dots, \mathbf{A}^{(20)}(P^\ell, P)$$

where $P = (i, 0)$ and

$$\mathbf{A}^{(j)}(P^\ell, P)$$

is the Abel map computed with `digits` = j , and then plotting the base-ten logarithmic difference of the infinity norm

$$\|\mathbf{A}^{(j)} - \mathbf{A}^{(j+1)}\|_\infty.$$

7.3 A hyperelliptic example

All curves that may be written in Weierstrass form

$$F = y^2 - (x - \lambda_1) \cdots (x - \lambda_{2g+m}), \quad m = 1 \text{ or } 2,$$

where all the λ_j are distinct, are hyperelliptic curves of genus g . If $m = 1$ the branch points are the $2g + 1$ points λ_j and the point at ∞ . If $m = 2$ the $2g + 2$ points λ_j are the only branch points.

The Abel map between any two branch points on a hyperelliptic curve is two-torsion [32]. That is, if P', P are branch points then

$$2A(P', P) = 2 \int_{P'}^P \omega \equiv \mathbf{0}.$$

In what follows the command `ModPeriodlattice(V, B)` returns $\llbracket V \rrbracket$: the vector V reduced modulo the period lattice. Thus the unreduced vector is

$$V = \lfloor V \rfloor + \llbracket V \rrbracket, \quad \lfloor V \rfloor \in \Lambda.$$

With the option ‘`fraction`’ the procedure will return the length $2g$ vector $\langle V \rangle$ such that each component

$$\langle V \rangle_j \in [0, 1)$$

and

$$\llbracket V \rrbracket = [\mathbf{1} \ \mathbf{B}] \langle V \rangle.$$

This option simplifies the detection of vectors that are torsion. Start by reading in the necessary packages and entering a genus 3 plane algebraic curve and computing its associated Riemann period matrix \mathbf{B} .

```
># read in the packages
>with(algcurves): read("AbelMap.mpl"):
># define a hyperelliptic curve
>F:=y^2 - (x^2 - 1)*(x^2 - 4)*(x^2 - 9)*(x - 4);
```

```
># calculate the Riemann period matrix
>B := periodmatrix(F, x, y, 'Riemann');
```

$$F := y^2 - (x^2 - 1)(x^2 - 4)(x^2 - 9)(x - 4)$$

Next compute the places over a sample set of the branch points of our curve.

```
># the branch points of F
>X := [-3, -2, -1, 1, 2, 3, 4, infinity];
># compute the Puiseux expansions over the branch points of F
>puis := 'algcures/puiseux':
>P := seq(op(puis(F, x = X[i], y, 0, t)), i = 1..8):
>P[6]; P[8];    # display two of the places
```

$$\alpha := [-3, -2, -1, 1, 2, 3, 4, \infty]$$

$$P_6 := [x = 3 - 240t^2, y = -240t]$$

$$P_8 := \left[x = \frac{1}{t^2}, y = \frac{1}{t^7} \right]$$

Next define a function `IndexedAbel` such that `IndexedAbel(j,k,digits)` is the Abel map computed from P_j to P_k with `digits` significant digits. Compute $V = \text{IndexedAbel}(6,8,10)$, the Abel map from the place over $x = 3$ to the place over $x = \infty$ with 10 significant digits.

```
>IndexedAbel := (j, k, digits) -> AbelMap(F, x, y, P[j], P[k],
  t, digits):
>V := A(6, 8, 10);
```

$$[.5362947073 - .2163229247 * I, -.5700652852 + .3104163172 * I, \\ .4395355374 - .4252823968 * I]$$

Verify that the vector \mathbf{V} is two-torsion by verifying that $2\langle \mathbf{V} \rangle$ is “nearly integral,” *i. e.*

$$\|2\langle \mathbf{V} \rangle - \mathbf{M}\|_{\infty}$$

is small, with

$$\mathbf{M} \in \mathbb{Z}^{2g}$$

the vector closest to $\langle \mathbf{V} \rangle$.

># $\langle \mathbf{V} \rangle$ is denoted by ‘BraVKet’

>BraVKet := AbelMap:-ModPeriodLattice(V, B, ‘fraction’);

$$\begin{aligned} BraVKet := & [.3732 \cdot 10^{-13}, .999999999999965, .5000000000000197620, \\ & .999999999999978, .126604458540559303 \cdot 10^{-12}, .5000000000000074] \end{aligned}$$

># InfNorm(z) computes the infinity norm of the vector z

># minus the closest integer vector

>InfNorm := z -> max(op(map(w -> abs(round(w) - w), z))):

>DeltaInteger := InfNorm(2*BraVKet); # compute infinity norm

$$DeltaInteger := .397 \cdot 10^{-12}$$

Figure 7.2 demonstrates that the Maple implementation of the Abel map is converging. It is constructed as follows. Compute 8×8 matrices $V(\text{digits})$ such that $V_{j,k}(\text{digits})$ is the Abel map computed from P_j to P_k with digits significant digits. Each component of $V(\text{digits})$ is a vector in \mathbb{C}^3 . Reduce each of these vectors modulo the period lattice, and each of these reduced vectors $\langle \mathbf{V} \rangle$ is approximately a half-lattice vector in $J(\Gamma)$. Then compute the 8×8 matrices $DeltaInteger(\text{digits})$ such that

$$DeltaInteger_{j,k}(\text{digits}) = \|2\langle V_{j,k}(\text{digits}) \rangle - \mathbf{M}\|_{\infty}$$

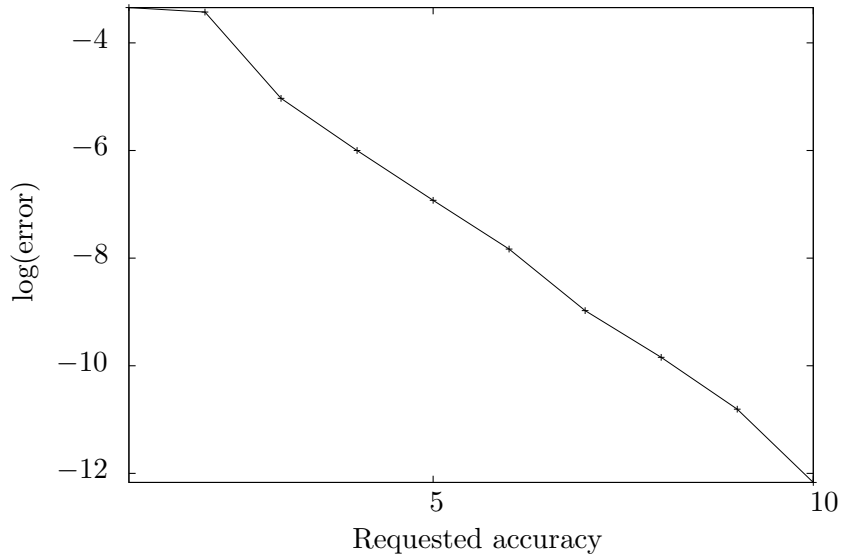


Figure 7.2: A plot of the convergence of the Maple implementation. The y -axis is the logarithm of the maximum difference from a lattice vector over all the entries of the matrix A . The x -axis is the accuracy requested.

where

$$M \in \mathbb{Z}^6$$

is the integer vector closest to

$$2\langle V_{j,k}(\text{digits}) \rangle.$$

In Figure 7.2 the base-ten logarithm of the largest component of `DeltaInteger(digits)` is graphed for values of `digits` from one to ten.

7.4 Computing the Abel map of divisors of meromorphic functions

Many algorithms that compute objects associated with Riemann surfaces are restricted in their use. For example it is common for an algorithm to only work in the case of an elliptic, or hyperelliptic curve (please see [25] for definitions of *elliptic* and *hyperelliptic*). To emphasize the general nature of the Abel map algorithm, in the next two examples I work with non-hyperelliptic curves. In this example I use that, by Abel's Theorem, the

Abel map of the divisor arising from a meromorphic function is zero modulo the lattice Λ . Specifically I compute the Abel map of the divisor arising from the function y on the genus 8 surface Γ defined by

$$F = y^8 + xy^5 + x^4 - x^6 = 0.$$

```
># define curve, compute genus, establish as non-hyperelliptic
>F := y^8 + x*y^5 + x^4 - x^6: genus(F, x, y);
```

8

```
is_hyperelliptic(F, x, y);
```

false

```
># enter the divisor of the meromorphic function y
>P[1] := [x = -1/t^4, y = -1/t^3 + t^2/8]:
>P[2] := [x = 1/t^4, y = 1/t^3 + t^2/8]:
>P[3] := [x = -1 + t^5/2, y = t]:
>P[4] := [x = 1 + t^5/2, y = t]:
>P[5] := [x = -t^3, y = t]: P[6] := [x = t^5, y = -t^3]:
>Dvsr := [ [1, P[3]], [1, P[4]], [1, P[5]], [3, P[6]], [-3, P[1]
], [-3, P[2]]];
```

$$Dvsr := \left[\left[1, \left[-1 + \frac{t^5}{2}, y = t \right] \right], \left[1, \left[1 + \frac{t^5}{2}, y = t \right] \right], [1, [x = -t^3, y = t]], \right. \\ \left. [3, [x = t^5, y = -t^3]] \left[-3, \left[x = \frac{-1}{t^4}, y = \frac{-1}{t^3} + \frac{t^2}{8} \right] \right], \right. \\ \left. \left[-3, \left[x = \frac{1}{t^4}, y = \frac{1}{t^3} + \frac{t^2}{8} \right] \right] \right]$$


```
># compute Abel map of Dvsr asking for 10 digits of accuracy
>A := AbelMap:-DivisorAbelMap(F, x, y, t, 'ZERO', Dvsr, 10);
```

$$A := [-1.701888413 - 1.660865446 I, - .4032208712 + .2717377626 I, \\ 1.645822432 - 1.590163370 I, \quad .9110383714 + .8278818528 I, \\ - .249748051 + 1.152988939 I, \quad .2595511419 - 1.437472396 I, \\ -1.222656877 + .8174634287 I, \quad .3898925624 + .4483560160 I]$$

```
># compute the Riemann matrix, used to reduce A modulo lattice
>B := periodmatrix(F, x, y, 'Riemann'):
># reduce the vector A
>V := AbelMap:-ModPeriodLattice(A, B, 'fraction');
```

$$V := [.2 \cdot 10^{-9}, .9999999948, .104151222 \cdot 10^{-8}, .999999999, .9999999952, \\ .9999999991, 0., .9999999981, .9999999982, .9999999977, .9999999950, \\ .9999999950, .9999999932, .9999999924, .9999999925, .9999999944]$$

The vector V above is approximately the vector

$$(0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1),$$

and thus the vector A is approximately a lattice vector, that is, approximately zero modulo the lattice Λ .

7.5 Computing the Abel map of divisors of meromorphic differentials

On a genus g Riemann surface, all divisors

$$D_j = (\omega_j), \quad k = 1, \dots, g \tag{7.1}$$

arising from holomorphic differentials

$$\omega, \dots, \omega_g$$

are linearly equivalent and of degree $2g + 2$. For these linearly equivalent divisors

$$A(P, D_1) \equiv \cdots \equiv A(P, D_g).$$

I use this fact to further demonstrate and validate the Maple implementation. Here I slightly change the curve from the previous example to demonstrate the use of algebraic numbers as coefficients.

```
># define an algebraic number
>beta := RootOf(_Z^3 + 7, index = 1):
># define a curve and establish that it is not hyperelliptic
>F := y^8 + x*y^5 + x^4 - x^6*beta; is_hyperelliptic(F, x, y);
```

$$F := y^8 + xy^5 + x^4 - x^6\beta$$

false

Input the places on the Riemann surface Γ defined by

$$F(x, y) = 0$$

that constitute the eight divisors

$$(\hat{\omega}_1), \dots, (\hat{\omega}_8)$$

where $\hat{\omega}_j$ is the j -th differential from **differentials**. Choose P^ℓ as the fixed place of the Abel map by using the flag ‘ZERO’ as an argument of the Abel map. The places computed next are used to construct the eight canonical divisors.

The places P_1 and P_2 are both over the point $x = 0$, and both are branched.

```
>P[1] := [x = t^5, y = -t^3]; P[2] := [x = -t^3, y = t];
```

$$P_1 := [x = t^5, y = -t^3]$$

$$P_2 := [x = -t^3, y = t]$$

The places P_3 and P_4 are regular places over the two roots of

$$x = \sqrt{-\beta^2/7}.$$

```
>P[3] := [x = 1/2*beta*t^5 + RootOf(beta^2 + 7*_Z^2, index = 1)
, y = t]:
>P[4] := [x = 1/2*beta*t^5 + RootOf(beta^2 + 7*_Z^2, index = 2)
, y = t]:
>P[3];P[4];
```

$$P_3 := \left[x = \frac{1}{2}\beta t^5 + \text{RootOf}(\beta^2 + 7_Z^2, index = 1), y = t \right]$$

$$P_4 := \left[x = \frac{1}{2}\beta t^5 + \text{RootOf}(\beta^2 + 7_Z^2, index = 2), y = t \right]$$

There are two branch places over $x = \infty$, these are P_5 and P_6 .

```
>P[5] := [x = -7/(t^4*RootOf(beta^2 + 7*_Z^2, index = 1)^2*RootOf(-RootOf(beta^2 + 7*_Z^2, index = 1) + _Z^2, index = 1)),
y = -7/(t^3*RootOf(beta^2 + 7*_Z^2, index = 1)^2*RootOf(-RootOf(beta^2 + 7*_Z^2, index = 1) + _Z^2, index = 1))]:
>P[6] := [x = -7/(t^4*RootOf(beta^2 + 7*_Z^2, index = 1)^2*RootOf(-RootOf(beta^2 + 7*_Z^2, index = 1) + _Z^2, index = 2)),
y = -7/(t^3*RootOf(beta^2 + 7*_Z^2, index = 1)^2*RootOf(-RootOf(beta^2 + 7*_Z^2, index = 1) + _Z^2, index = 2))]:
>P[5];P[6];
```

$$P_5 := \left[x = \frac{-7}{t^4 \beta^2 \text{RootOf}(-\beta + _Z^2, index = 1)}, y = \frac{-7}{t^3 \beta^2 \text{RootOf}(-\beta + _Z^2, index = 1)} \right]$$

$$P_6 := \left[x = \frac{-7}{t^4 \beta^2 \text{RootOf}(-\beta + _Z^2, index = 2)}, y = \frac{-7}{t^3 \beta^2 \text{RootOf}(-\beta + _Z^2, index = 2)} \right]$$

Form divisors (CD) using integer combinations of these points, compute the Abel maps (A) of these divisors and reduce the resulting vectors modulo the lattice to obtain vectors (V) on $J(\Gamma)$.

```
>CD[1] := [[2, P[1]], [5, P[3]], [5, P[4]], [1, P[6]], [1, P[7]]:
]:
>CD[2] := [[4, P[1]], [2, P[2]], [4, P[3]], [4, P[4]]]:
>CD[3] := [[1, P[1]], [1, P[2]], [3, P[3]], [3, P[4]], [3, P[6]]
], [3, P[7]]]:
>CD[4] := [[3, P[1]], [3, P[2]], [2, P[3]], [2, P[4]], [2, P[6]]
], [2, P[7]]]:
>CD[5] := [[5, P[1]], [5, P[2]], [1, P[3]], [1, P[4]], [1, P[6]]
], [1, P[7]]]:
>CD[6] := [[7, P[1]], [7, P[2]]]:
>CD[7] := [[2, P[2]], [1, P[3]], [1, P[4]], [5, P[6]], [5, P[7]]:
]:
>CD[8] := [[2, P[1]], [4, P[2]], [4, P[6]], [4, P[7]]]:
>A := i -> AbelMap:-DivisorAbelMap(F, x, y, 'ZERO', CD[i], 8):
>BraVKet := i -> AbelMap:-ModPeriodLattice(A(i), B, 'fraction'):
```

Again use the function `InfNorm`. Construct the $g \times g$ matrix `DeltaInteger` where the j, k -th component is the infinity norm of the difference

$$\text{BraVKet}(j) - \text{BraVKet}(k),$$

where here `BraVKet(index)` is a function of the index *index*. As these two vectors are supposed to be identical on $J(\Gamma)$, this difference should be nearly integral, and the components of `DeltaInteger` are small if the Maple implementation is computing correctly.

```
>DeltaInteger := Matrix(8, (j, k)-> InfNorm(BraVKet(j) - BraVKet(k))):
```

$$\begin{aligned}
& \Delta Integer := \\
& 10^{-7} * \begin{bmatrix}
0.0 & 2.143 & 1.273 & 1.877 & 3.969 & 6.095 & 2.499 & 2.929 \\
2.143 & 0.0 & 2.401 & 1.254 & 1.826 & 3.952 & 3.235 & 2.474 \\
1.273 & 2.401 & 0.0 & 2.135 & 4.227 & 6.353 & 1.226 & 1.854 \\
1.877 & 1.254 & 2.135 & 0.0 & 2.092 & 4.218 & 2.416 & 1.220 \\
3.969 & 1.826 & 4.227 & 2.092 & 0.0 & 2.126 & 4.508 & 2.373 \\
6.095 & 3.952 & 6.353 & 4.218 & 2.126 & 0.0 & 6.634 & 4.499 \\
2.499 & 3.235 & 1.226 & 2.416 & 4.508 & 6.634 & 0.0 & 2.135 \\
2.929 & 2.474 & 1.854 & 1.220 & 2.373 & 4.499 & 2.135 & 0.0
\end{bmatrix},
\end{aligned}$$

which confirms the quality of the algorithm and its implementation.

Chapter 8

THE IMPLEMENTATION OF THE VECTOR OF RIEMANN
CONSTANTS**8.1 Syntax of the vector of Riemann constants**

In this section I provide examples of the algorithm to compute the vector of Riemann constants (VRC) by means of a Maple implementation. This implementation is to be included in the regular distribution of Maple 12 (release date as yet to be announced). Updated versions may be obtained at <http://www.amath.washington.edu/~bernard/papers.html>. The examples also serve to verify that these algorithms and implementations are performing as desired. The Maple implementation is called in Maple with the syntax

```
>RiemannConstants(F, x, y, digits);
```

- **F** is a polynomial in only the variables **x** and **y**. The coefficients of **F** may not be floating point numbers, but may be irrational if entered as radicals or using the Maple **RootOf** notation.
- **digits** is an integer declaring the desired number of correct digits. The implementation returns a vector of g complex numbers. Each component of the numerically computed vector should share **digits** significant figures, in both real and imaginary parts, with the symbolically calculated component.

The Maple syntax needed to understand the examples below is explained in Appendix B.

8.2 Example computation of the half-lattice vector

In this section I present an example of computing the correct half-lattice vector

$$\mathbf{h} \in \Lambda_{\frac{1}{2}}$$

such that the VRC of the left-place P^ℓ is given by

$$\mathbf{K}^\ell = \left[\left[\mathbf{h}_j - \frac{1}{2} \mathbf{A}(P^\ell, \mathcal{C}) \right] \right], \quad (8.1)$$

where \mathcal{C} is the valuation divisor of a holomorphic differential.

Consider the hyperelliptic curve given in Weierstrass form by the polynomial

$$F = y^2 - (x^2 - 1)(x^2 - 4)(x^2 - 9)(x^2 - 16)(x^2 - 25)(x^2 - 36).$$

A basis for the cohomology of the Riemann surface Γ defined by

$$F(x, y) = 0$$

is

$$\left(\frac{1}{y} dx, \quad \frac{x}{y} dx, \quad \frac{x^2}{y} dx, \quad \frac{x^3}{y} dx, \quad \frac{x^4}{y} dx \right).$$

Note that there are five differentials, agreeing with the fact that F has

$$2g + 2 = 2(5) + 2 = 12$$

branch points. The algorithm described in Section 6.3 can be used to compute a canonical divisor. The following Maple input and output demonstrates this. Alternatively, one may use the Maple command `AbelMap:-CanonicalDivisor`, described in Appendix B.

```
># Read in the needed packages and files
>with(algcurves):
>read("AbelMap.mpl"): read("RiemannConstants.mpl"):
># Define a hyperelliptic curve
>F:=y^2 - (x^2 - 1)*(x^2 - 4)*(x^2 - 9)*(x^2 - 16)*(x^2 - 25)*
(x^2 - 36):
```

$$F := y^2 - (x^2 - 1)(x^2 - 4)(x^2 - 9)(x^2 - 16)(x^2 - 25)(x^2 - 36)$$

```
># Compute the differentials
>omega := differentials(F, x, y);
```

$$\omega := \left[\frac{dx}{y}, \frac{x dx}{y}, \frac{x^2 dx}{y}, \frac{x^3 dx}{y}, \frac{x^4 dx}{y} \right]$$

Again, as per Section 6.3, a divisor composed of only the places over $x = \infty$ is first sought, and in this case this is successful. For this particular choice of F , infinity is not a branch point, and thus there are two places over $x = \infty$. The places over infinity are computed using the Maple command **AbelMap:-AllBranches**. This output of this command is essentially the same as the output of the command **puiseux**. The output from **puiseux** is reformatted, and the valuations of the curve and the y -derivative of the curve are also output for use by subsequent routines.

```
># Compute the places over x=infinity
>Pinf := AbelMap:-Allbranches(F, x, infinity, y, 0, t);
```

$$Pinf := \left[\left[x = \frac{1}{t}, y = -\frac{1}{t^6} + \frac{91}{2t^4} - \frac{3731}{8t^2} + \frac{16263}{16}, valuation = -6, \right. \right. \\ \left. dfdy_degree = -6, parameter = t \right], \\ \left[x = \frac{1}{t}, y = \frac{1}{t^6} - \frac{91}{2t^4} + \frac{3731}{8t^2} - \frac{16263}{16}, valuation = -6, \right. \\ \left. dfdy_degree = -6, parameter = t \right]$$

The parameter *dfdy_degree* in the Maple output above is the valuation of F_y at the places specified. Notice that, for both places over $x = \infty$, *dfdy_degree* is given by

$$dfdy_degree = -6.$$

As $x = 1/t$ for both of the places over $x = \infty$, the differential dx is given by

$$dx = -\frac{1}{t^2} dt.$$

The valuation of any power of x is given by

$$\text{val}(x^j, Pinf[l]) = -j, \quad l = 1, 2.$$

Since the five holomorphic differentials are all of the form

$$\omega_k = \frac{x^{k-1}}{y} dx,$$

the valuations of the five differentials at places over $x = \infty$ are given by

$$\begin{aligned} \text{val}(\omega_k, \text{Pinf}[l]) &= \overbrace{-(k-1)}^{x^{k-1}} + \overbrace{(-2)}^{dx} \underbrace{-}_{\div} \overbrace{(-6)}^y, \quad l = 1, 2 \\ &= 5 - k, \end{aligned}$$

where the over- and under-braces indicate the part of ω_k giving rise to each summand in the valuation. Collecting all this information about the valuations of basis holomorphic differentials at the two infinite places $\text{Pinf}[1]$ and $\text{Pinf}[2]$:

$$\begin{aligned} \text{val}(\omega_1, \text{Pinf}[1]) &= 4 = \text{val}(\omega_1, \text{Pinf}[2]), \\ \text{val}(\omega_2, \text{Pinf}[1]) &= 3 = \text{val}(\omega_2, \text{Pinf}[2]), \\ \text{val}(\omega_3, \text{Pinf}[1]) &= 2 = \text{val}(\omega_3, \text{Pinf}[2]), \\ \text{val}(\omega_4, \text{Pinf}[1]) &= 1 = \text{val}(\omega_4, \text{Pinf}[2]), \\ \text{val}(\omega_5, \text{Pinf}[1]) &= 0 = \text{val}(\omega_5, \text{Pinf}[2]). \end{aligned}$$

Recall that any holomorphic differential has, as shown in Example 7, $2g - 2$ zeros. As Γ is a genus five Riemann surface, the differentials

$$\omega_1, \dots, \omega_5$$

each have eight zeros and no poles. Consider the valuation divisor $(\omega_1)_{\text{val}}$ of the holomorphic differential ω_1 . The results above show that $(\omega_1)_{\text{val}}$ consists of at least the places $\text{Pinf}[1]$ and $\text{Pinf}[2]$, each with multiplicity positive four. As ω_1 has no poles, there are no places with negative multiplicities, which also guarantees that $\text{Pinf}[1]$ and $\text{Pinf}[2]$ are the only places where $\omega_1 = 0$. Thus the divisor $(\omega_1)_{\text{val}}$ is

$$(\omega_1)_{\text{val}} = 4 \text{Pinf}[1] + 4 \text{Pinf}[2].$$

The divisor $(\omega_1)_{\text{val}}$ is canonical as ω_1 is an Abelian differential. The work above to compute a canonical divisor can be circumvented using the implementation of Algorithm 2, available as the Maple command `AbelMap:-CanonicalDivisor`.

The Maple implementation is used to compute the Abel map of the divisor $(\omega_1)_{\text{val}}$. Only four significant figures are displayed although the calculation is performed with 10 digits of requested accuracy.

```
># Input the divisor of omega[1]
>CD := [[4, Pinf[1]], [4, Pinf[2]]]:
># Compute the Abel map
>A := AbelMap:-DivisorAbelMap(F, x, y, 'ZERO', CD, 10);
```

$$A := [0.05273 + 1.3532i, -0.07934 - .19752i, -0.02425 + 0.15744i, \\ 0.34410 - 0.06792i, -0.026611 + 0.02784i]$$

The vector V computed next is the same as vector

$$\frac{\mathbf{A}_c}{2}$$

from Section 6.2. That is V is the Abel map A computed above reduced modulo the period lattice and divided by negative two. Note that the Riemann matrix (B) in the Maple input below was computed and stored while computing the Abel map.

```
># Reduce mod the lattice, negate and halve
>V := -AbelMap:-ModPeriodLattice(A, B)/2;
```

$$V := [0.53493 + 0.92887i, -0.55256 - 0.62048i, 0.37357 + 0.59393i, \\ -0.36684 - 0.31246i, 0.48237 + 0.28591i]$$

Using the methods of Section 6.6 the error tolerances ϵ_1 and ϵ_2 for the first and second passes respectively through Algorithm 3 are calculated. For convenience I will just define them instead of showing the details of computing them with Maple.

```
># Define error tolerances
>epsilon[1] := 0.1; epsilon[2] := 0.0004883;
```

$$\epsilon_1 := 0.1$$

$$\epsilon_2 := 0.0004883$$

The first filter pass

A compiled procedure¹ of the finite sum in (6.21) with error tolerance ϵ_1 is constructed.

```
># Construct a compiled approximation of the purely oscillatory
># part of the theta function with max error epsilon[1]
>FiniteSum, SumArgs := CompTheta(V, epsilon[1], B):
```

The output of the procedure `CompTheta`, which is not displayed, is a compiled procedure and a list of arguments required by the procedure. Compiling the procedure requires computational time, but experience has shown this overhead is made up for by the faster evaluation times when compared to the interpreted procedure `RiemannTheta`. The procedure `FiniteSum` called as

```
>FiniteSum(j, SumArgs):
```

computes the absolute value of the candidate VRC, K_j^ℓ , formed with the j -th half-lattice vector \mathbf{h}_j as in (8.1). Recall that Γ is a genus five surface, and thus there are

$$2^{2 \times 5} = 1024$$

candidates for the half-lattice vector in (8.1). For each choice of the 1024 choices of \mathbf{h}_j , the absolute value of the approximated ϑ -function is computed. The variable `ti` and the procedure `time()` in what follows are used to get a measure of the central processing unit (CPU) time used. See Appendix B for details on the Maple procedures `seq`, `time()`, `nops` and `'if'`.

```
># Initialize CPU timing variable
>ti:= time():
># Compute the absolute value of the purely oscillatory part
>ABS_K := seq(FiniteSum(j, SumArgs), j = 1..2^(2*g)):
# Display the CPU 'time' used
>time() - ti;
```

¹The arguments shown in the Maple input are simplified for ease of display.

16.34

```

># Construct a list of the candidate indices that pass through
># the first filter
>J1 := seq('if'(ABS_K[j] < epsilon[1], j, NULL), j = 1..2^(2*g
  )):
># Count how many candidates are left
>nops(J1);

```

172

All elements of ABS_K larger than ϵ_1 indicate half-lattice vectors that are no longer candidates after the first pass of Algorithm 3. In the example at hand, there are 172 candidates for which

$$\text{ABS_K}_j < \epsilon_1.$$

The sequence $J1$ consists of the indices of these remaining candidates. Note that

$$\frac{172}{1024} \approx 0.168,$$

and thus the assumption, from Section 6.6, that the fraction of ϑ evaluations with absolute value less than ϵ_j is approximately ϵ_j seems not to be drastically violated, at least at this point.

The second filter pass

At this stage in Algorithm 3 a generic, effective divisor and its Abel map would be computed. To show the savings gained by doing so, I will perform the second pass in two different ways. First, *without* perturbing the candidates by the Abel map of a generic, effective divisor. Second, with such a perturbation added to each candidate vector. Note that these perturbations are *not* meant to be small in any sense.

1. Construct the compiled procedure to compute the approximate ϑ -function, and use this procedure to compute the absolute value of the remaining candidate vectors.

```

># Construct a compiled approximate vartheta
>FiniteSum, SumArgs := CompTheta(V, epsilon[2], B):
>ti := time():
# Use compiled procedure to approximate absolute values
>ABS_K := seq([j, FiniteSum(j, SumArgs)], j = 1..nops(J)):
# Display CPU 'time'
>time() - ti;

```

7.267

I do not display the next steps explicitly as, while simple, they are difficult to read when written as a Maple input/output stream. The list $J2$ consists of the indices and absolute values of the 20 remaining candidates for the VRC.

```

>J2;

```

```

[16, .261 10-4], [32, .210 10-3], [512, .431 10-4], [536, .682 10-5],
[568, .128 10-3], [635, .445 10-3], [788, .321 10-5], [792, .420 10-4],
[820, .564 10-4], [855, .439 10-3], [887, .170 10-3], [914, .102 10-5],
[916, .350 10-4], [946, .158 10-4], [977, .185 10-6], [978, .241 10-4],
[1008, .376 10-4], [1009, .328 10-7], [1010, .484 10-3], [1024, .109 10-4]

```

Note that the absolute value of the 1009-th candidate has an absolute value more than an order of magnitude smaller than any other. In practice, the error in the approximation of ϑ is (as is the error in Maple implementation of the approximation) far smaller than the error bound imposed by choosing ϵ . Thus at this point it seems likely that candidate 1009 *is* the VRC. Of course as this greater accuracy is not guaranteed further work must be done. Also note that

$$\frac{20}{1024} \approx 0.0195$$

which quite a bit greater than

$$\epsilon_2 = 0.0004883,$$

and thus that the assumption regarding the statistics of the absolute value of ϑ is definitely violated. However, the desire to balance the computational effort of the first and second passes (see Section 6.6) is approximately satisfied. This is seen by noting that the first pass used about 16 units of CPU time, and the second about 7 units, these values are on the same order of magnitude, as desired.

2. To perform the second pass in the manner outlined in Algorithm 3, a generic, effective degree $g - 1$ divisor is needed. This divisor,

$$\mathcal{D}_0 = P_1 + \cdots + P_{g-1},$$

is chosen to meet the following three criteria.

- A. The Abel map of \mathcal{D}_0 must be a significant perturbation. That is, \mathcal{D}_0 should be chosen such that

$$\|\mathbf{A}(P^\ell, \mathcal{D}_0)\| \gg 0.$$

Otherwise, for a given candidate half-lattice vector \mathbf{h}_j , it may be that

$$\|\vartheta(\mathbf{V} + \mathbf{h}_j + \mathbf{A}(P^\ell, \mathcal{D}_0))\| \approx \|\vartheta(\mathbf{V} + \mathbf{h}_j)\| \approx 0$$

simply due to the fact that the Abel map of \mathcal{D}_0 is small in absolute value, irrespective of $\mathbf{V} + \mathbf{h}_j$ being on Θ .

- B. Divisor \mathcal{D}_0 is chosen to consist of $g - 1$ distinct places. Thus, once

$$\mathbf{A}(P^\ell, \mathcal{D}_0) = \mathbf{A}(P^\ell, P_1) + \cdots + \mathbf{A}(P^\ell, P_{g-1})$$

has been computed, a great many other perturbations may be computed simply as linear combinations of the

$$\mathbf{A}(P^\ell, P_1), \dots, \mathbf{A}(P^\ell, P_{g-1}).$$

These perturbations may be needed in the third pass of Algorithm 3.

C. The computation of $\mathcal{A}(P^\ell, \mathcal{D}_0)$ should be inexpensive.

To fit these criteria, let

$$\mathcal{D}_0 = P_1 + \cdots + P_{g-1}$$

where P_j is a regular place given by

$$P_j = \begin{cases} x &= \alpha^\ell - j \\ y &= \mathbf{y}_1^\ell(\alpha^\ell - j) \end{cases},$$

where

- α^ℓ is the left-point (see Section 5.3),
- $(\alpha^\ell - j) \in \mathbb{C}$ is a point in the complex plane j real units to the left of α^ℓ and
- using the notation of Section 5.3, the y -value

$$\mathbf{y}_1^\ell(\alpha^\ell - j)$$

is the y -root of

$$F((\alpha^\ell - j, y) = 0$$

on sheet one, where the sheets are ordered with respect to the left-place.

The perturbation (\mathbf{dV}) displayed below is the Abel map of \mathcal{D}_0 . In analogy with Item 1, a compiled version of ϑ is constructed, however this time with $\mathbf{V} + \mathbf{dV}$ replacing \mathbf{V} . In this example computation \mathbf{dV} is given by

$>\mathbf{dV};$

$$\begin{aligned} &[-.485 \, 10^{-1} - .661 * I, .730 \, 10^{-1} + .184 * I, .104 \, 10^{-1} - .147 * I, \\ &\quad - .198 \, 10^{-1} + .731 \, 10^{-1} * I, .245 \, 10^{-1} - .363 \, 10^{-1} * I] \end{aligned}$$

```

># Construct a compiled approximate vartheta
>FiniteSum, SumArgs := CompTheta(V + dV, epsilon[2], B):
># Compute the absolute values
>ABS_K := seq([j, FiniteSum(j, SumArgs)], j = 1..nops(J)):

```

Leaving aside the details, as in Item 1, the list J2 of indices and absolute values is displayed below.

```
>J2;
```

$$[977, .476 \cdot 10^{-3}], [1009, .645 \cdot 10^{-6}],$$

Performing the second pass without a perturbation left 20 candidates remaining and with a perturbation left 2. Thus, in this example, perturbing the candidate vectors has significantly increased efficiency as 18 fewer candidates must be checked using the more accurate, and hence more expensive, approximate ϑ -function `FiniteSum`. Note that it may be the case that one could construct a curve such that perturbing candidate vectors induces no increase, or perhaps even a decrease, in efficiency. At this point I do not know if such a construction is possible, but I can not rule out the possibility. Every test thus far has shown that a large increase in efficiency is realized. Algorithm 3 is designed, as much as possible to perform efficiently in the generic case.

The third filter pass

The third pass in Algorithm 3 uses the fact that

$$\vartheta(\mathbf{K}(P_0) + \mathbf{A}(P_0, \mathcal{D})) = 0, \quad \forall \mathcal{D} \in S^{g-1}\Gamma$$

to eliminate the remaining candidates for the VRC. A sequence of divisors

$$\tilde{\mathcal{D}} = \mathcal{D}_1, \mathcal{D}_2, \dots$$

is constructed where each $\mathcal{D} \in \tilde{\mathcal{D}}$ is a degree $g - 1$ divisor which consists only of places in

$$\mathcal{D}_0 = P_1, \dots, P_{g-1}.$$

Thus, as

$$\begin{aligned} \mathbf{A}(P_0, \mathcal{D}_0) &= \mathbf{A}(P_0, P_1, \dots, P_{g-1}) \\ &= \mathbf{A}(P_0, P_1) + \dots + \mathbf{A}(P_0, P_{g-1}) \end{aligned}$$

was previously computed, computing a perturbation $\mathbf{A}(P_0, \mathcal{D})$ for any

$$\mathcal{D} \in \tilde{\mathcal{D}}$$

is a matter of taking positive, integer, linear combinations of known Abel maps.

The number of divisors that may be constructed this way is

$$\#\tilde{\mathcal{D}} = \frac{((g-1)^2 + (g-1) - 1)!}{((g-1)^2 - 1)!(g-1)!},$$

which grows exponentially in g for $g > 2$. Practice has shown that nowhere near so many perturbations are required. In fact, in no test case has it been necessary to compute with more than two perturbations. Moreover, for curves with higher genus approximating the ϑ -function becomes very expensive². With this in mind, only $g - 1$ of the divisors in $\tilde{\mathcal{D}}$ are considered when implementing Algorithm 3. Specifically, the divisors constructed are

$$\mathcal{D}_1, \dots, \mathcal{D}_{g-1},$$

where \mathcal{D}_k is given by

$$\mathcal{D}_k = (g-1)P_k.$$

If these $g - 1$ divisors are insufficient, the implementation returns an error. Admittedly, this choice is *ad hoc*.

Beginning with $\mathcal{D} = \mathcal{D}_1$, the absolute value

$$\|\vartheta(V + \mathbf{h}_j + \mathbf{A}(P_0, \mathcal{D}))\| \tag{8.2}$$

²I present the following as a benchmark. Approximating, to within 0.001, the ϑ -function on a of a non-hyperelliptic, sparse, genus eight curve using Maple 10 on a Dell laptop with a 2.00GHz Intel Pentium processor took about a minute for each candidate half-lattice vector

Table 8.1: This table shows the results of testing that the candidate VRC resulting from the half-lattice vector with index 1009 is in fact on Θ . The right-hand column displays the approximated absolute value of $\vartheta(\mathbf{A}_C/2 + \mathbf{h}_{1009} + \mathbf{A}^{(\ell}P), P_j)$, where P_j is shown in the left-hand column. The middle column shows the magnitude of the perturbation $\mathbf{A}^{(\ell}P, P_j)$.

P_j	$\ \mathbf{A}(P^\ell, (g-1)P_j)\ $	$\ \vartheta\ $
$\left(-\frac{52}{5}, -\frac{133056}{15625}\sqrt{8441017311}\right)$	0.919	$0.189 \cdot 10^{-5}$
$\left(-\frac{47}{5}, -\frac{133056}{15625}\sqrt{1936133706}\right)$	0.814	$0.115 \cdot 10^{-5}$
$\left(-\frac{42}{5}, -\frac{3456}{15625}\sqrt{5004490563001}\right)$	0.669	$0.218 \cdot 10^{-5}$
$\left(-\frac{37}{5}, -\frac{12096}{15625}\sqrt{4508923991}\right)$	0.448	$0.105 \cdot 10^{-5}$

is approximated to within ϵ_2 subsequently for each candidate remaining: that is, for each $j \in J2$. If the approximate value of (8.2) is greater than ϵ_2 , then j is removed from $J2$. This is done for each

$$\mathcal{D} = \mathcal{D}_1, \dots, \mathcal{D}_{g-1}$$

until only one candidate remains. As stated above, if this does not exclude all but one candidate, then an error is returned.

In the example computation under consideration, the Abel map

$$\mathbf{dV} = \mathbf{A}(P^\ell, (g-1)P_{g-1})$$

is used as the first perturbation of the third pass. The computation of the perturbation \mathbf{dV} is omitted, but the result is given by

$$>\mathbf{V} + \mathbf{dV};$$

$$[.486 + .268 * I, -.480 - .437 * I, .384 + .447 * I, \\ - .387 - .239 * I, .507 + .250 * I]$$

```
># Construct an approximate vartheta
>FiniteSum, SumArgs := CompTheta(V + dV, epsilon[2], B):
># Compute the absolute value for half-lattice vector 977
>FiniteSum(977, SumArgs);
```

$$.22910^{-2}$$

```
>FiniteSum(1009, SumArgs);
```

$$.18910^{-5}$$

```
> Recall the value of epsilon[2]
>epsilon[2];
```

$$0.0004883$$

As 0.00229 is larger than ϵ_2 and thus

$$0 \notin [0.00229 - \epsilon_2, 0.00229 + \epsilon_2]$$

it is necessarily true that

$$\|\vartheta(\mathbf{V} + \mathbf{dV} + \mathbf{h}_{977})\| = \left\| \vartheta \left(\frac{\mathbf{A}_C}{2} + (g-1)\mathbf{A}(P^\ell, P_{g-1}) + \mathbf{h}_{977} \right) \right\| \neq 0,$$

However, at least within ϵ_2

$$\|\vartheta(\mathbf{V} + \mathbf{dV} + \mathbf{h}_{1009})\| = \left\| \vartheta \left(\frac{\mathbf{A}_C}{2} + (g-1)\mathbf{A}(P^\ell, P_{g-1}) + \mathbf{h}_{1009} \right) \right\| \approx 0.$$

Figure 8.1 displays the results of further testing that the half-lattice vector \mathbf{h}_{1009} gives the correct VRC. Specifically the table shows that

$$\left\lceil \frac{\mathbf{A}_C}{2} + \mathbf{h}_{1009} + \mathbf{A}(P^\ell, \mathcal{D}_k) \right\rceil \in \Theta$$

for the divisors

$$\mathcal{D}_1, \dots, \mathcal{D}_g \in S^{g-1}\Gamma.$$

The left-hand column displays the regular place P_j as indicated by an ordered pair (x, y) satisfying

$$F(x, y) = 0.$$

The absolute values in the center column are displayed to demonstrate that the perturbations

$$\mathbf{dV}_k = \mathbf{A}(P^\ell, (g-1)P_k)$$

are not small. For comparison note that the first g lattice vectors in Λ have a magnitude of one. Finally, the right-hand column demonstrates that the perturbed candidate is such that

$$\left\| \vartheta \left(\frac{\mathbf{A}_c}{2} + \mathbf{h}_{1009} + \mathbf{dV}_k \right) \right\| \approx 0$$

to within $\epsilon_2 = 0.0004883$.

Chapter 9

APPLICATIONS OF THE ABEL MAP

9.1 Integration of algebraic functions

My primary motivation for constructing an algorithm to compute the Abel map is for use in the construction of multi-phase solutions of integrable equations. A second motivation for the numerical computation of the Abel map is that it can be used to compute the *torsion* of a divisor. This, in conjunction with a theorem of Risch, can be used to answer the question of whether certain algebraic functions can be integrated in finite terms. The full details of this concept are beyond the scope of this dissertation and for details the reader is referred to the literature [11, 34, 35, 23]. However, the mathematics used in this section can be gleaned from [25]. I will give an example in which the existence of an elementary anti-derivative is established by using the Abel map, and then construct that anti-derivative.

For my purposes a *torsion divisor* with torsion index $\tau \in \mathbb{N}$ is a divisor \mathcal{D} with the property

$$\mathbf{A}(P', \tau\mathcal{D}) = \tau\mathbf{A}(P', \mathcal{D}) \equiv 0, \quad (9.1)$$

where τ is the smallest positive integer for which (9.1) holds [11]. The first equality is an identity by the definition of the Abel map applied to a divisor. In order to discuss torsion divisors in relation to the integration of algebraic functions I first define valuations, valuation divisors (introduced previously), residues and residue divisors on Riemann surfaces. In these definitions

- Γ is a Riemann surface,
- Q is a place on that surface,
- t is a local coordinate at Q ,

- u is a meromorphic function on Γ and
- ν is a meromorphic differential on Γ .

Definitions:

- **Valuation:** Near Q the function u is given by

$$u = t^k h(t),$$

where $h(0) \neq 0$ and $k \in \mathbb{Z}$ may be negative or zero and $h(t)$ is a holomorphic function. The integer k is the *valuation* of u at Q , denoted $\text{val}(u, Q)$.

- **Residue:** If the differential ν is given near Q as

$$\nu = \sum_j \nu_j t^j dt,$$

then

$$\nu_{-1} = \text{res}(\nu, Q)$$

is called the *residue* of ν at place Q .

- **Valuation divisor:** The valuation divisor $(\nu)_{\text{val}}$ of a function u is

$$(u)_{\text{val}} = \sum \text{val}(\nu, Q) Q$$

where the sum is over places on Γ for which

$$\text{val}(u, Q) \neq 0.$$

- **Residue divisor:** The residue divisor $(\nu)_{\text{res}}$ of a differential ν is

$$(\nu)_{\text{res}} = \sum \text{res}_Q(\nu) Q$$

where the sum is over places on Γ for which

$$\text{res}(\nu, Q) \neq 0.$$

Note that, unlike the divisors of meromorphic functions, the coefficients of Q in this divisor definition are not to be interpreted as multiplicities.

Given a meromorphic function u , I construct the meromorphic differential

$$d \log u = \frac{du}{u}.$$

Near Q

$$d \log u = k \frac{dt}{t} + \frac{dh}{h}. \quad (9.2)$$

As $h(0) \neq 0$, (9.2) shows that

$$\text{res}(d \log u, Q) = k = \text{val}(u, Q),$$

and as Q is a general place on the surface,

$$(d \log u)_{\text{res}} = (u)_{\text{val}}.$$

Note that (9.2) shows that poles of $d \log u$ only occur at places Q where

$$\text{val}(u, Q) \neq 0,$$

and therefore $d \log u$ has no poles that do not appear in

$$(d \log u)_{\text{res}} = (u)_{\text{val}}.$$

The following proposition is essentially one direction of Risch's Theorem [34, 35].

Proposition 9.1.1. *Suppose ν is an Abelian differential of the third kind. Further suppose that*

$$(\nu)_{\text{res}} = \mathcal{D} = \sum_j (p_j P_j - q_j Q_j),$$

where p_j, q_j are positive integers. If \mathcal{D} is a torsion divisor, then there exists a function u such that

$$(u)_{\text{val}} = \mathcal{D},$$

and further,

$$\nu - d \log u$$

is holomorphic.

Proof. Since \mathcal{D} is a torsion divisor there exists a positive integer τ such that

$$\tau \mathbf{A}(\mathcal{D}) = \mathbf{A}(\tau \mathcal{D}) \equiv 0.$$

As the sum of the residues of a meromorphic differential is zero [25],

$$\deg \mathcal{D} = 0.$$

The degree of $\tau \mathcal{D}$ is also zero, so by Abel's Theorem there exists a function u such that

$$(u)_{\text{val}} = \mathcal{D}.$$

Above it was shown that

$$(u) = (\mathrm{d} \log u)_{\text{res}}$$

and that the poles in the residue divisor are the only poles of $\mathrm{d} \log u$. As ν was supposed to be an Abelian differential of the third kind the pole structure of ν and $\mathrm{d} \log u$ are identical. Thus the difference

$$\nu - \mathrm{d} \log u \tag{9.3}$$

is a holomorphic differential. \square

If the holomorphic difference 9.3 is zero, then the meromorphic differential ν can be integrated in closed form. That is, if

$$\nu - \mathrm{d} \log u = 0$$

then

$$\int \nu = \log u + c.$$

I use this proposition to integrate an algebraic function. In this example I use the Maple implementation of the Abel map to verify that the residue divisor of (9.4) is a torsion divisor. Other algorithms and implementations exist for establishing that a divisor is torsion [11, 12]. For instance, using such algorithms the computer algebra system Axiom is able to compute (9.4). The curve in this example comes from [20], and the methods used are used in [23]. The values of the coefficients of the numerator of (9.4) are essential for the existence of the anti-derivative.

Example 9. Evaluate the integral

$$\int \frac{39x^2 + 9x - 1}{\sqrt{x^6 + 4x^4 + 10x^3 + 4x^2 - 4x + 1}} dx. \quad (9.4)$$

Equation (9.4) can be seen as the integral of the meromorphic differential

$$\nu = \frac{39x^2 + 9x - 1}{y} dx$$

on the Riemann surface Γ arising from the polynomial

$$F = y^2 - (x^6 + 4x^4 + 10x^3 + 4x^2 - 4x + 1) = 0. \quad (9.5)$$

I calculate the residue divisor of ν and verify that it is an Abelian differential of the third kind. If the residue divisor is torsion a function u can be found such that

$$d \log u = \nu.$$

A basis of the holomorphic differentials on Γ is given by

$$\left(\frac{dx}{y}, \frac{x dx}{y} \right),$$

thus, as the term

$$\frac{9x}{y}$$

and the term

$$-\frac{1}{y}$$

are holomorphic, to calculate the poles of ν it suffices to calculate the poles of

$$\tilde{\nu} = \frac{39x^2}{y}.$$

These may occur where $y = 0$ or $x = \infty$.

- The points for which $y = 0$ are the six roots

$$\lambda = \{x \in \mathbb{C} : F(x, 0) = 0\},$$

that is, the branch points. Using Puiseux expansions, it is straightforward to show that none of these places are poles of ν .

- The places over $x = \infty$ are, labeled by the sign of the y -series,

$$P_{\pm}^{\infty} = \begin{cases} x &= \frac{1}{t} \\ y &= \pm \frac{1}{t^3} + \cdots \end{cases}.$$

Substituting this and

$$dx = -\frac{1}{t^2} dt$$

into $\tilde{\nu}$ yields

$$\tilde{\nu} = \mp \left(\frac{39}{t} + \cdots \right) dt,$$

and thus the residue divisor

$$(\nu)_{\text{res}} = (\tilde{\nu})_{\text{res}} = 39 P_{-}^{\infty} - 39 P_{+}^{\infty}.$$

Below I show that

$$P_{-}^{\infty} - P_{+}^{\infty} \tag{9.6}$$

is a torsion divisor with torsion index 39, thus

$$\mathbf{A}((\nu)_{\text{res}}) \equiv 0, \tag{9.7}$$

proving there exists a function u such that

$$\nu = d \log u$$

up to addition of holomorphic differentials. Assume the function

$$u = \kappa_1(x)y + \kappa_2(x)$$

where

$$\kappa_1(x) = \kappa_{1,0} + \kappa_{1,1}x + \kappa_{1,2}x^2 + \cdots$$

and

$$\kappa_2(x) = \kappa_{2,0} + \kappa_{2,1}x + \kappa_{2,2}x^2 + \cdots,$$

where $\kappa_{j,k} \in \mathbb{C}$ for all valid j and k . Note that this assumption is justified because $(1, y)$ is an integral basis for the functions with no poles in the affine part of Γ [37]. That is, any meromorphic function on Γ may be written as a product of a polynomial in x multiplied by either the function 1 or the function y .

In order for u to have a pole of order 39 at

$$P_-^\infty = \begin{cases} x &= \frac{1}{t} \\ y &= -\frac{1}{t^3} + \dots \end{cases}$$

and a zero of order 39 at

$$P_+^\infty = \begin{cases} x &= \frac{1}{t} \\ y &= \frac{1}{t^3} + \dots \end{cases},$$

assume

$$\deg(\kappa_1(x), x) = 36 \quad \text{and} \quad \deg(\kappa_2(x), x) = 39.$$

Then at P_+^∞ , the function u is

$$u = \left(\kappa_{1,0} + \frac{\kappa_{1,1}}{t} + \dots + \frac{\kappa_{1,36}}{x^{36}} \right) \left(\frac{1}{t^3} \right) + \left(\kappa_{2,0} + \frac{\kappa_{2,1}}{t} + \dots + \frac{\kappa_{2,39}}{x^{39}} \right).$$

At P_-^∞ , the function u is

$$u = \left(\kappa_{1,0} + \frac{\kappa_{1,1}}{t} + \dots + \frac{\kappa_{1,36}}{x^{36}} \right) \left(\frac{-1}{t^3} \right) + \left(\kappa_{2,0} + \frac{\kappa_{2,1}}{t} + \dots + \frac{\kappa_{2,39}}{x^{39}} \right).$$

If the leading coefficients of $\kappa_1(x)$ and $\kappa_2(x)$ are set to one, solving for all the others such that at P_+^∞ all terms with t degree less than 39 vanish gives the required $\kappa_1(x)$ and $\kappa_2(x)$.

$$\begin{aligned}
\kappa_1(x) = & x^{36} - 9x^{35} + 78x^{34} - 393x^{33} + 1875x^{32} - 6228x^{31} + 20000x^{30} - 44364x^{29} \\
& + 100878x^{28} - 123754x^{27} + 205056x^{26} + 93834x^{25} + 93210x^{24} \\
& + 1154952x^{23} + 1154628x^{22} + 2305452x^{21} + 7667343x^{20} + 7398477x^{19} \\
& + 18519986x^{18} + 31511637x^{17} + 38292357x^{16} + 68810088x^{15} + 94270020x^{14} \\
& + 106641612x^{13} + 154404118x^{12} + 169319118x^{11} + 167242404x^{10} \\
& + 191230470x^9 + 167241762x^8 + 129775188x^7 + 117823332x^6 + 74411808x^5 \\
& + 40244337x^4 + 28237903x^3 + 10352370x^2 + 2380755x + 1856467, \\
\kappa_2(x) = & + x^{39} - 9x^{38} + 80x^{37} - 406x^{36} + 1986x^{35} - 6636x^{34} + 21881x^{33} \\
& - 48249x^{32} + 113386x^{31} - 129138x^{30} + 233880x^{29} + 201028x^{28} \\
& + 158320x^{27} + 1761084x^{26} + 2213274x^{25} + 4131222x^{24} + 13853079x^{23} \\
& + 16147329x^{22} + 37199924x^{21} + 69411762x^{20} + 92551930x^{19} \\
& + 166484056x^{18} + 250442895x^{17} + 308665185x^{16} + 459151006x^{15} \\
& + 563317218x^{14} + 607070900x^{13} + 732820904x^{12} + 734205504x^{11} \\
& + 637517772x^{10} + 621760542x^9 + 479086914x^8 + 303058869x^7 \\
& + 231510907x^6 + 123527844x^5 + 39445718x^4 + 27825546x^3 + 7163724x^2 \\
& - 1856467x + 1332179.
\end{aligned}$$

Note that since u is a meromorphic function the sum of the multiplicities of the zeroes must equal that of the poles. Then, because there are no other poles except those at P_-^∞ by construction, there are no other zeroes besides those at P_+^∞ . Thus

$$(u)_{\text{val}} = 39 P_-^\infty - 39 P_+^\infty.$$

Calculating $d \log u$ and simplifying shows

$$d \log u - \nu = 0.$$

Thus the anti-derivative of ν is given by

$$\int \nu = \log(\kappa_1(x)y + \kappa_2(x)) + c,$$

where

$$y = \sqrt{x^6 + 4x^4 + 10x^3 + 4x^2 - 4x + 1}.$$

The Maple implementation of the Abel map is used to compute the torsion of divisors on Γ defined by (9.5). Specifically I establish that (9.6) has torsion index 39. The points $-1, 0, 1$ and ∞ are not discriminant points of $F = 0$, thus there are $n = 2$ places on the Riemann surface over each of these points. From any two places in

$$Q_{1,1}, Q_{1,2}, Q_{2,2}, \dots, Q_{4,1}, Q_{4,2}, \quad (9.8)$$

form a degree zero divisor by taking one place with multiplicity 1 and one with multiplicity -1 . For the particular basis of the cycles on Γ that were chosen by the Maple command `homology`, the divisors thus formed have torsion indices of 3, 13 and 39. The command `AllBranches(F, x, α , y, 0, t)` below computes Puiseux expansions over the point $x = \alpha$ with the syntax needed by `AbelMap`.

```
># a hyperelliptic algebraic curve.
>F := y^2-(x^6+4*x^4+10*x^3+4*x^2-4*x+1):
># compute the Riemann matrix
>B := periodmatrix(F, x, y, 'Riemann'):

># compute the places used to form divisors
>Q[1] := AllBranches(F, x, -1, y, 0, t):
>Q[2] := AllBranches(F, x, 0, y, 0, t):
>Q[3] := AllBranches(F, x, 1, y, 0, t):
>Q[4] := AllBranches(F, x, infinity, y, 0, t):
>Q[1];Q[2];Q[3];Q[4];
```

$$\begin{aligned}
Q_1 &:= [[x = t - 1, y = 2 - t], [x = t - 1, y = -2 + t]] \\
Q_2 &:= [[x = t, y = 1 - 2t], [x = t, y = -1 + 2t]] \\
Q_3 &:= [[x = t + 1, y = 4 + 7t], [x = t + 1, y = -4 - 7t]] \\
Q_4 &:= [[x = 1/t, y = 1/t^3 +], [x = 1/t, y = -1/t^3 +]]
\end{aligned}$$

The function `IndexedAbelMap(j, k)` computes the Abel map $\mathbf{A}(P^\ell, \mathbf{CD}_{j,k})$ of the divisor

$$\mathbf{CD}_{j,k} = \begin{cases} Q_{j,1} - Q_{k,1} & \text{if } j \neq k \\ Q_{j,1} - Q_{k,2} & \text{if } j = k, \end{cases}$$

and construct a matrix `AbelMatrix` using the function `IndexedAbelMap`. As

$$\mathbf{A}(P', P) = -\mathbf{A}(P, P'),$$

the matrix `AbelMatrix` thus constructed represents the Abel map from any place in (9.8) to any other place in (9.8). The function `ReduceModLattice` below reduces vectors modulo the period lattice, and is used component-wise on the matrix `AbelMatrix`. As floating point Abel maps have been displayed previously, here only the results of computing the torsion of divisors $\mathbf{CD}_{j,k}$ are shown.

Note that the extra digits of accuracy (`Digits + 3`) in the call to `DivisorAbelMap` are requested in order to have enough significant digits to correctly assign the torsion index.

```

># define the function 'IndexedAbelMap'
>IndexedAbelMap:=(i, j) -> DivisorAbelMap(F, x, y, t, 'ZERO',
  'if'(i<> j, [[1, Q[i][1]], [-1, Q[j][1]]], [[1, Q[i][1]], [-1
    , Q[j][2]]]), Digits + 3):
># construct matrix 'A' using 'IndexedAbelMap'
>AbelMatrix := Matrix(4, IndexedAbelMap):
># construct function to reduce vectors in matrix 'AbelMatrix'
>ReduceModLattice := V -> ModPeriodLattice(V, B):
># reduce vectors in 'AbelMatrix' modulo the period lattice
>ReducedAbelMat := map(z -> ReduceModLattice(z), AbelMatrix):

```

The procedure¹ `torsion(V, ε)` computes the first integer τ such that

$$\|\tau \mathbf{V} - \lfloor \mathbf{V} \rfloor\| < \epsilon$$

where

$$\lfloor \mathbf{V} \rfloor \in \Lambda.$$

When `torsion` is mapped component-wise to the matrix (`ReducedAbelMat`) of vectors reduced modulo the period lattice, the resulting matrix summarizes the torsion indices.

```
>epsilon := 10^(-8): map(torsion, ReducedAbelMat, epsilon);
```

$$\begin{bmatrix} 39 & 39 & 39 & 13 \\ 39 & 39 & 13 & 3 \\ 39 & 13 & 39 & 39 \\ 13 & 3 & 39 & 39 \end{bmatrix}$$

The (4,4) element of this matrix is the torsion of

$$Q_{4,1} - Q_{4,2} = P_-^\infty - P_+^\infty,$$

which is indeed 39.

9.2 Integral-linear dependence of divisors

The Abel map may be used in conjunction with the LLL algorithm [31] to determine if, given a list of m divisors

$$\mathcal{D}, \dots, \mathcal{D}_m \in \Gamma,$$

there exists a linear combination

$$\tilde{\mathcal{D}} = \sum_j n_j \mathcal{D}_j, \quad n_j \in \mathbb{Z}, \quad j = 1, \dots, m$$

¹This procedure is written specifically for this example, and has not been released. The code is on the compact disc included with this dissertation.

such that \tilde{D} is the divisor of a meromorphic function. This problem arises, in for instance [10], when determining if certain algebraic functions are algebraically dependent. In what follows the Maple implementation of the Abel map is used to determine an integral-linear, or \mathbb{Z} -linear, relation between two divisors.

```
># define an algebraic curve
>F := x^4 + (y^2 - 1)*(y^2 - 4);
```

$$F := x^4 + (y^2 - 1)(y^2 - 4)$$

```
># compute the places over (x = 0)
>algcures[puiseux](F, x = 0, y, 0, t);
```

$$\{[x = t, y = 2], [x = t, y = -2], [x = t, y = 1], [x = t, y = -1]\}$$

```
># construct two divisors of the places over (x = 0)
>Dvsr[1] := [[1, [x = t, y = 1]], [-1, [x = t, y = -1]]]:
>Dvsr[2] := [[1, [x = t, y = 2]], [-1, [x = t, y = -1]]]:
>Dvsr[1]; Dvsr[2];
```

$$Dvsr[1] := [[1, [x = t, y = 1]], [-1, [x = t, y = -1]]]:$$

$$Dvsr[2] := [[1, [x = t, y = 2]], [-1, [x = t, y = -1]]]:$$

```
># compute the Abel maps of Dvsr[1] and Dvsr[2]
>A[1] := AbelMap:-DivisorAbelMap(F, x, y, t, 'ZERO', Dvsr[1],
  10):
>A[2] := AbelMap:-DivisorAbelMap(F, x, y, t, 'ZERO', Dvsr[2],
  10):
>A[1]; A[2];
```


$$\begin{aligned}
& [-.304946701650801 + .195053298320151 * I, \\
& \quad .304946701634244 - 1.19505329837223 * I, \\
& \quad -.250000000003239 + .445053298352439 * I] \\
& [\quad .195053298349199 - .1982887 \cdot 10^{-10} * I, \\
& \quad .304946701634245 - .500000000031671 * I, \\
& \quad .975266491670425 \cdot 10^{-1} + .975266491821570 \cdot 10^{-1} * I]
\end{aligned}$$

```

># compute the integer dependency of A[1], A[2]
># (and thus of Dvsr[1] and Dvsr[2])
>ZDependence([A], B, 10);

```

$$[4., -4.]$$

Thus

$$4\mathbf{Dvsr}_1 - 4\mathbf{Dvsr}_2 \in \Lambda,$$

ans it is the divisor of a meromorphic function on Γ . The Maple code of the procedure `ZDependence` is included on the compact disc accompanying this dissertation.

BIBLIOGRAPHY

- [1] M. J. Ablowitz and P. A. Clarkson. *Solitons, nonlinear evolution equations and inverse scattering*. Cambridge University Press, Cambridge, 1991.
- [2] M. J. Ablowitz, J. Hammack, D. Henderson, and C. M. Schober. Long-time dynamics of the modulational instability of deep water waves. *Phys. D*, 152/153:416–433, 2001.
- [3] M. J. Ablowitz, B. M. Herbst, and C. M. Schober. The nonlinear Schrödinger equation: asymmetric perturbations, traveling waves and chaotic structures. *Math. Comput. Simulation*, 43(1):3–12, 1997.
- [4] M. J. Ablowitz and H. Segur. *Solitons and the inverse scattering transform*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1981.
- [5] H. F. Baker. *Abelian functions*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1995.
- [6] E. D. Belokolos, A. I. Bobenko, V. Z. Enol'skii, A. R. Its, and V. B. Matveev. *Algebro-geometric approach to nonlinear integrable problems*. Springer Series in Nonlinear Dynamics. Springer-Verlag, Berlin, 1994.
- [7] G. A. Bliss. *Algebraic functions*. Dover Publications Inc., New York, NY, 1966.
- [8] A. I. Bobenko. All constant mean curvature tori in \mathbf{R}^3 , S^3 , H^3 in terms of theta-functions. *Math. Ann.*, 290:209–245, 1991.
- [9] A. I. Bobenko and L. A. Bordag. Periodic multiphase solutions of the Kadomtsev-Petviashvili equation. *J. Phys. A*, 22:1259–1274, 1989.
- [10] E. Compoint and M. F. Singer. Computing Galois groups of completely reducible differential equations. *J. Symbolic Comput.*, 28:473–494, 1999.
- [11] J. H. Davenport. *On the integration of algebraic functions*, volume 102 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1981.
- [12] J. H. Davenport and B. M. Trager. On the parallel Risch algorithm. II. *ACM Trans. Math. Software*, 11(4):356–362, 1985.
- [13] B. Deconinck. The initial-value problem for multiphase solutions of the Kadomtsev-Petviashvili equation. *University of Colorado Ph.D thesis*, 1998.

- [14] B. Deconinck, M. Heil, A. Bobenko, M. van Hoeij, and M. Schmies. Computing Riemann theta functions. *Mathematics of Computation*, 73:1417–1442, 2004.
- [15] B. Deconinck, D. E. Pelinovsky, and J. D. Carter. Transverse instabilities of deep-water solitary waves. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 462(2071):2039–2061, 2006.
- [16] B. Deconinck and H. Segur. The KP equation with quasiperiodic initial conditions. *Physica D*, 123:123–152, 1998.
- [17] B. Deconinck and M. van Hoeij. Computing Riemann matrices of algebraic curves. *Phys. D*, 152/153:28–46, 2001.
- [18] B. A. Dubrovin. Theta functions and nonlinear equations. *Russian Math. Surveys*, 36(2):11–80, 1981.
- [19] B. A. Dubrovin, R. Flickinger, and H. Segur. Three-phase solutions of the Kadomtsev-Petviashvili equation. *Stud. Appl. Math.*, 99(2):137–203, 1997.
- [20] N. D. Elkies. http://www.math.harvard.edu/~elkies/g2_tors.html
- [21] H. M. Farkas and I. Kra. *Riemann surfaces*. Springer-Verlag, New York, NY, second edition, 1992.
- [22] John D. Fay. *Theta functions on Riemann surfaces*. Springer-Verlag, Berlin, 1973.
- [23] E. V. Flynn. Large rational torsion on abelian varieties. *J. Number Theory*, 36(3):257–265, 1990.
- [24] A. P. Fordy, editor. *Soliton theory: a survey of results*. Nonlinear Science: Theory and Applications. Manchester University Press, Manchester, 1990.
- [25] P. Griffiths. *Introduction to algebraic curves*. American Mathematical Society, Providence, RI, 1989.
- [26] P. Griffiths and J. Harris. *Principles of algebraic geometry*. John Wiley & Sons Inc., New York, NY, 1994.
- [27] J. Hammack, D. McCallister, N. Scheffner, and H. Segur. Two-dimensional periodic waves in shallow water. II. Asymmetric waves. *J. Fluid Mech.*, 285:95–122, 1995.
- [28] J. Hammack, N. Scheffner, and H. Segur. Two-dimensional periodic waves in shallow water. *J. Fluid Mech.*, 209:567–589, 1989.

- [29] J. L. Hammack, D. M. Henderson, and H. Segur. Progressive waves with persistent two-dimensional surface patterns in deep water. *J. Fluid Mech.*, 532:1–52, 2005.
- [30] R. S. Johnson. *A modern introduction to the mathematical theory of water waves*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 1997.
- [31] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.
- [32] D. Mumford. *Tata lectures on theta. I*. Progress in Mathematics. Birkhäuser Boston Inc., Boston, MA, 1983.
- [33] D. Mumford. *Tata lectures on theta. II*, volume 43 of *Progress in Mathematics*. Birkhäuser Boston Inc., Boston, MA, 1984.
- [34] R. H. Risch. The problem of integration in finite terms. *Trans. Amer. Math. Soc.*, 139:167–189, 1969.
- [35] R. H. Risch. The solution of the problem of integration in finite terms. *Bull. Amer. Math. Soc.*, 76:605–608, 1970.
- [36] G. Springer. *Introduction to Riemann surfaces*. Addison-Wesley Publishing Company, Inc., Reading, MA., 1957.
- [37] M. van Hoeij. An algorithm for computing an integral basis in an algebraic function field. *J. Symb. Comput.*, 18:353–363, 1994.

Appendix A

PROOFS AND DETAILS

A.1 Details of Proposition 3.14.2

Details of the proof of Proposition 3.14.2 are provided mostly for completeness. Recall the starting point of this proposition is the integral

$$\zeta = \frac{1}{2\pi i} \int_{\partial \tilde{\Gamma}} d \log \phi(P) \mathcal{A}(P) \quad (\text{A.1})$$

where the integration is done around the boundary $\partial \tilde{\Gamma}$ of the cut surface $\tilde{\Gamma}$. Further recall that given a vector $\mathbf{Z} \in J(\Gamma)$, the function $\phi(P)$ given by

$$\phi(P) = \theta(\mathcal{A}(P) - \mathbf{Z}) \quad (\text{A.2})$$

is analytic on the cut surface. Some notation and facts will be useful to calculate the line integral (A.1) by exploiting information about the periods of normalized holomorphic differentials.

1. A useful shorthand for integrating to places on $\partial \tilde{\Gamma}$ is to denote by $\mathcal{A}^+(P)$ the integral from initial place P_0 to place P on a_j or b_j ; the positively oriented segments of $\partial \tilde{\Gamma}$.
2. In analogy with Item 1, denote by $\mathcal{A}^-(P)$ the integral from initial place P_0 to place P on cycle a_j^{-1} or b_j^{-1} ; the negatively oriented segments of $\partial \tilde{\Gamma}$.
3. On any a -cycle, the j -th component of the unreduced Abel map on the j -th negatively oriented cycle in terms of j -th component on the k -th positively oriented cycle is given by

$$\mathcal{A}_j^-(P) = \mathcal{A}_j^+(P) + B_{jk}.$$

This is by (3.18) and the nature of the cut surface. In vector notation,

$$\mathcal{A}^-(P) = \mathcal{A}^+(P) + \mathbf{B}e_k.$$

4. Similarly, if the place P is on cycle b_k

$$\mathcal{A}_j^-(P) = \mathcal{A}_j^+(P) - \delta_{jk}$$

by (3.16) and the nature of the cut surface. In the vector notation:

$$\mathcal{A}^-(P) = \mathcal{A}^+(P) - \mathbf{e}_k.$$

5. In analogy with Items 1 and 2, denote by ϕ^\pm the function ϕ evaluated on the positively and negatively oriented segments of $\partial\tilde{\Gamma}$:

$$\phi^\pm(P) = \theta(\mathcal{A}^\pm(P) - \mathbf{Z}). \quad (\text{A.3})$$

6. We use the quasiperiodicity properties of the θ -function and Item 3 to obtain the logarithm of ϕ^- in terms of quantities on the cycles traversed in a positive sense. On cycle a_k , $\log \phi^-(P)$ is given by

$$\log \phi^-(P) = -2\pi i \left(\frac{B_{kk}}{2} - \mathcal{A}_k^+(P) + Z_k \right) + \log \phi^+(P).$$

7. By Item 4, $\log \phi^-(P)$ on cycle b_k is given by

$$\log \phi^-(P) = \log \phi^+(P).$$

8. Items 6 and 7, along with the fact that

$$d\mathcal{A}_k = \omega_k,$$

lead directly to the relations between logarithmic derivatives evaluated at places on positively and negatively oriented cycles. For instance, on cycle a_k

$$d \log \phi^-(P) = d \log \phi^+(P) - \omega_k(P),$$

and on cycle b_k

$$d \log \phi^-(P) = d \log \phi^+(P).$$

Switching to only integrals calculated over positively oriented cycles, the equation for ζ_j becomes

$$\zeta_j = \frac{1}{2\pi i} \sum_{k=1}^g \left(\oint_{a_k} + \oint_{b_k} \right) \left(d \log \phi^+(P) \mathcal{A}_j^+(P) - d \log \phi^-(P) \mathcal{A}_j^-(P) \right). \quad (\text{A.4})$$

$$(\text{A.5})$$

Using the notation from Items 3, 4, 6 and 6, and dropping the integration place variable P for clarity I can rewrite (A.1) as

$$\begin{aligned} \zeta_j &= \frac{1}{2\pi i} \sum_{k=1}^g \oint_{a_k} \left[(d \log \phi^+) (\mathcal{A}_j^+) - (d \log \phi^+ - 2\pi i \omega_k) (\mathcal{A}_j^+ - B_{jk}) \right] + \\ &+ \frac{1}{2\pi i} \sum_{k=1}^g \oint_{b_k} \left[(d \log \phi^+) (\mathcal{A}_j^+) - (d \log \phi^+) (\mathcal{A}_j^+ - \delta_{jk}) \right]. \end{aligned}$$

Expanding and canceling terms gives

$$\begin{aligned} \zeta_j &= \frac{1}{2\pi i} \sum_{k=1}^g 2\pi i \oint_{a_k} \omega_k \mathcal{A}_j^+ - B_{jk} \oint_{a_k} d \log \phi^+ + 2\pi i B_{jk} \oint_{a_k} \omega_k + \\ &+ \frac{1}{2\pi i} \sum_{k=1}^g \delta_{jk} \int_{b_j} d \log \phi^+, \end{aligned}$$

which simplifies to

$$\begin{aligned} \zeta_j &= \frac{1}{2\pi i} \sum_{k=1}^g 2\pi i \oint_{a_k} \omega_k \mathcal{A}_j^+ - B_{jk} \oint_{a_k} d \log \phi^+ + 2\pi i B_{jk} + \\ &+ \frac{1}{2\pi i} \int_{b_j} d \log \phi^+. \end{aligned}$$

As ϕ takes on the same value at both the start and the end of cycle a_k , the only contribution to the integral

$$\oint_{a_k} d \log \phi^+$$

come from the multi-valued nature of the complex logarithm. Thus

$$\oint_{a_k} d \log \phi^+ = 2\pi i m_k$$

for some integer m_k . Therefore, the expression for ζ_j simplifies to

$$\zeta_j = \frac{1}{2\pi i} \int_{b_j} d \log \phi^+ + \sum_{k=1}^g \oint_{a_k} \omega_k \mathcal{A}_j^+ + \underbrace{\sum_{k=1}^g B_{jk}(1 - m_k)},$$

where, as m_k is not dependent on the index j , the underbraced terms yield the j -th component of a lattice vector. This simplification is not so clean on a b -cycle. Suppose Q_j and Q'_j are the initial and final places on cycle b_k . Then

$$\oint_{b_k} d \log \phi = \log \phi(Q'_j) - \log \phi(Q_j) + 2\pi i m'_k$$

with $m'_k \in \mathbb{Z}$. Using the definition of ϕ and the quasiperiodicity of the θ -function,

$$\begin{aligned} \oint_{b_j} d \log \phi^+ &= \log \theta(\mathcal{A}^+(Q'_j) - \mathbf{Z}) - \log \theta(\mathcal{A}^+(Q_j) - \mathbf{Z}) + 2\pi i m'_k \\ &= \log \theta(\mathcal{A}^+(Q_j) + \mathbf{B}e_j - \mathbf{Z}) - \log \theta(\mathcal{A}^+(Q_j) - \mathbf{Z}) + 2\pi i m'_k \\ &= \log \left(e^{2\pi i(-\mathcal{A}_j^+(Q_j) + \frac{B_{jj}}{2} - Z_j)} \theta(\mathcal{A}^+(Q_j) + \mathbf{B}e_j - \mathbf{Z}) \right) - \\ &\quad - \log \theta(\mathcal{A}^+(Q_j) - \mathbf{Z}) + 2\pi i m'_k \\ &= 2\pi i \left(-\mathcal{A}_j^+(Q_j) - \frac{1}{2}B_{jj} + Z_j + m'_k \right). \end{aligned}$$

The equation for ζ_j can be rewritten using the preceding results as

$$\zeta_j = -\mathcal{A}_j^+(Q_j) - \frac{1}{2}B_{jj} + Z_j + \frac{1}{2\pi i} \sum_{k=1}^g \oint_{a_k} \omega_k(P) \mathcal{A}_j^+(P) + \quad (\text{A.6})$$

$$+ \underbrace{m'_k + \sum_{k=1}^g B_{jk}(1 - m_k)}, \quad (\text{A.7})$$

where the underbraced terms give rise to some lattice vector, and can be ignored when computing on the Jacobian. With the underbraced terms omitted, (A.6) is Equation (3.34).

A.2 Proof of Proposition 3.15.2

This proof of Proposition 3.15.2 follows directly that of [18]. First I remind the reader of the Proposition.

Proposition. *Suppose \mathbf{B} is the Riemann matrix associated with the Riemann surface Γ . The vector $\mathbf{W} \in J(\Gamma)$ is such that*

$$\theta(\mathbf{W}|\mathbf{B}) = 0$$

if and only if \mathbf{W} is given by

$$\mathbf{W} \equiv \mathbf{A}(\mathcal{D}) + \mathbf{K}$$

for some choice of

$$\mathcal{D} = P_1 + \cdots + P_{g-1}.$$

Proof. Suppose

$$\theta(\mathbf{W}) = 0,$$

and construct the function

$$\phi(P) = \theta(\mathbf{A}(P) - \mathbf{W}).$$

There are two cases to consider.

1. $\phi \not\equiv 0$: Let the vector \mathbf{V} be given by

$$\mathbf{V} \equiv \mathbf{W} - \mathbf{K}.$$

Then $\phi \not\equiv 0$ implies that

$$\theta(\mathbf{A}(P) - \mathbf{W}) = \theta(\mathbf{A}(P) - \mathbf{V} - \mathbf{K}) \not\equiv 0,$$

and thus that the hypothesis for Theorem 2 is satisfied for the vector \mathbf{V} . Thus \mathbf{V} is given by

$$\mathbf{V} \equiv \mathbf{A}(P_1) + \cdots + \mathbf{A}(P_g)$$

where the places

$$P_1, \dots, P_g$$

are the zeros of the function ϕ and are uniquely determined up to multiplicity. As

$$\phi(\mathbf{W}) = 0,$$

and the θ -function is even,

$$\theta(\mathbf{W}) = \theta(-\mathbf{W}) = \theta(-\mathbf{V} - \mathbf{K}) = 0.$$

Noting that

$$\mathbf{A}(P_0) = \mathbf{0},$$

it must be that

$$\theta(\mathbf{A}(P_0) - \mathbf{V} - \mathbf{K}) = 0,$$

and thus that P_0 is in the set of zeros

$$P_1, \dots, P_g.$$

With no loss, say $P_0 = P_g$. Then

$$\mathbf{A}(P_0) = \mathbf{A}(P_g) = \mathbf{0},$$

and

$$\mathbf{V} \equiv \mathbf{A}(P_1) + \dots + \mathbf{A}(P_{g-1}),$$

and further

$$\mathbf{W} \equiv \mathbf{A}(P_1) + \dots + \mathbf{A}(P_{g-1}) + \mathbf{K}.$$

Letting

$$\mathcal{D} = P_1, \dots, P_{g-1},$$

\mathbf{W} is given by

$$\mathbf{W} \equiv \mathbf{A}(\mathcal{D}) + \mathbf{K}.$$

2. $\phi \equiv 0$: In this case Theorem 1 assures that

$$\mathbf{V} \equiv \mathbf{A}(Q_1) + \cdots + \mathbf{A}(Q_g) + \mathbf{K} \quad (\text{A.8})$$

where the divisor

$$\mathcal{D} = Q_1 + \cdots + Q_g$$

is special. Thus that there is a function $u \in \Omega_\Gamma$ such that the poles of u are

$$Q_1, \dots, Q_g$$

and, by Riemann-Roch

$$u(P_0) = 0.$$

A meromorphic function has the same number of zeroes as poles [25], so the divisor of zeros $(u)_0$ of the function u may be written as

$$\mathcal{D}' = (u)_0 = P_0 + P_1 + \cdots + P_{g-1}.$$

The valuation divisor of u is then

$$\begin{aligned} (u)_{\text{val}} &= P_0 + P_1 + \cdots + P_{g-1} - Q_1 - \cdots - Q_g, \\ &= \mathcal{D}' - \mathcal{D}. \end{aligned}$$

As $u \in \Omega_\Gamma$, by Abel's Theorem

$$\mathbf{A}(\mathcal{D}' - \mathcal{D}) \equiv 0,$$

and thus

$$\mathbf{A}(\mathcal{D}') \equiv \mathbf{A}(\mathcal{D}),$$

$$\mathbf{A}(P_0) + \mathbf{A}(P_1) + \cdots + \mathbf{A}(P_{g-1}) \equiv \mathbf{A}(Q_1) + \cdots + \mathbf{A}(Q_g) \quad (\text{A.9})$$

Substituting (A.9) into (A.8) and again noting

$$\mathbf{A}(P_0) = \mathbf{0}$$

completes Item 2.

□

A.3 The integers l^0 and l^1 depend only on divisor class

Proposition A.3.1. *The integers $l^0(\mathcal{D})$ and $l^1(\mathcal{D})$ depends only on the equivalence class of the divisor \mathcal{D} .*

The proof presented here follows closely that presented in [36].

Proof. By definition, two divisors \mathcal{D}' and \mathcal{D}'' are linearly equivalent if and only if

$$\exists u \in \Omega_{\Gamma}^0 : \mathcal{D}' - \mathcal{D}'' = (u)_{\text{val}}. \quad (\text{A.10})$$

That is \mathcal{D}' and \mathcal{D}'' are linearly equivalent if and only if they differ by the divisor of a meromorphic function. Suppose

$$\mathcal{D}' \sim \mathcal{D}'',$$

and further that the function u is such that

$$\mathcal{D}' - \mathcal{D}'' = (u)_{\text{val}}, \quad u \in \Omega_{\Gamma}^0.$$

Note that $u = 0$ if and only if

$$\mathcal{D}' = \mathcal{D}'',$$

in which case there is no work to be done. Therefore assume

$$\mathcal{D}' \neq \mathcal{D}''.$$

1. $\mathcal{D}' \sim \mathcal{D}'' \implies l^0(\mathcal{D}') = l^0(\mathcal{D}'')$: If $u'' \in \mathcal{L}^0(\mathcal{D}'')$ then by definition

$$\mathcal{D}'' \geq -(u'')_{\text{val}} \quad (\text{A.11})$$

Under the hypothesis (A.10)

$$\mathcal{D}' - (u)_{\text{val}} = \mathcal{D}'',$$

so using (A.11) yields

$$\mathcal{D}' - (u)_{\text{val}} \geq -(u'')_{\text{val}},$$

and rearranging gives

$$(u'')_{\text{val}} - (u)_{\text{val}} = \left(\frac{u''}{u} \right)_{\text{val}} \geq -\mathcal{D}'.$$

As $u \neq 0$ by assumption, the function

$$\frac{u''}{u} \in \mathcal{L}^0(\mathcal{D}').$$

is defined and non-zero for all $u'' \neq 0$. Thus the mapping

$$\mathcal{L}^0(\mathcal{D}'') \ni u'' \mapsto \frac{u''}{u} \in \mathcal{L}^0(\mathcal{D}')$$

is one-to-one and onto, and therefore

$$\mathcal{L}^0(\mathcal{D}') \cong \mathcal{L}^0(\mathcal{D}'')$$

and

$$l^0(\mathcal{D}') = l^0(\mathcal{D}'').$$

2. $\mathcal{D}' \sim \mathcal{D}'' \implies l^1(\mathcal{D}') = l^1(\mathcal{D}'')$: The argument for this implication is nearly identical to that of the previous case. If $\nu'' \in \mathcal{L}^1(\mathcal{D}'')$ then

$$(\nu'')_{\text{val}} \geq \mathcal{D}'' \tag{A.12}$$

Under the hypothesis (A.10) we have

$$\mathcal{D}' - (u)_{\text{val}} = \mathcal{D}'',$$

using (A.12) yields

$$\mathcal{D}' - (u)_{\text{val}} \geq (\nu'')_{\text{val}},$$

and rearranging gives

$$(u)_{\text{val}} + (\nu'')_{\text{val}} = (u\nu'')_{\text{val}} \geq \mathcal{D}'.$$

Thus the mapping

$$\mathcal{L}^1(\mathcal{D}'') \ni \nu'' \mapsto u\nu'' \in \mathcal{L}^1(\mathcal{D}')$$

is one-to-one and onto. Therefore

$$\mathcal{L}^1(\mathcal{D}') \cong \mathcal{L}^1(\mathcal{D}'')$$

and

$$l^1(\mathcal{D}') = l^1(\mathcal{D}'').$$

□

Appendix B

MAPLE SYNTAX

B.1 Preliminary matters

The following list outlines some standard Maple syntax and procedure. Everything in the following list is available to any Maple worksheet upon opening. More information on a procedure can always be obtained in a Maple worksheet by typing

```
>?command
```

where `command` is the procedure with which one wishes help.

- Maple use for arbitrary precision arithmetic. The variable `Digits` controls how many significant digits Maple uses while computing. The user can change the value of `Digits`, the default value is 10.
- Maple input punctuated with a semi-colon is displayed. Input ending with a colon is not.
- A colon-equal sign (`:=`) indicates an assignment. An equal sign (`=`) can have different uses. In this work it is used as a logical test and as an unevaluated assignment in a list of table.
- Square brackets (`[]`) are used as delimiters and as selection operators.

```
>a := [4, 5, 6];
```

$$a := [4, 5, 6]$$

```
>a[2];
```

5

- The procedure **op** removes the outermost set of delimiters from a Maple list or set. The procedure **nops** returns the number of entries in a list or set.

```
>op(a);
```

5

```
>nops(a);
```

3

- The procedure **seq** can be used to create a sequence. If the sequence is delimited by square brackets, it becomes a list; if delimited with braces ($\{\}$) it becomes a set. The variable, in the example below this is j , is local to the call to the procedure. The syntax $(..)$ in the example creates a range.

```
>a := [seq(j, j = 1..9)];
```

$$a := [1, 2, 3, 4, 5, 6, 7, 8, 9]$$


```
>j; # note j remains unassigned after the call
```

$$j$$

- The syntax arrow (\rightarrow) syntax creates function inline (as opposed to in a file exterior to the worksheet.)

```
>f := z -> z^2;
```

$$f := z \mapsto z^2$$

```
>f(9);
```

81

- The procedure `map` can be used to apply a function component-wise to many Maple structures. In this work `map` is used on lists and matrices.

```
>b := map(f, a);
```

$$b := 1, 4, 9, 16, 25, 36, 49, 64, 81$$

- The procedure `'if'` is an inline logical switch. Note the single quotes (`' '`) are required.

```
>f := z -> 'if'(z, Yes, No):
>f(true); f(false);
```

Yes

No

- The Maple input

```
>Matrix(N, (j, k) -> g(j, k)):
```

forms the $N \times N$ matrix where the j, k component is the function g evaluated with arguments j, k .

- The Maple procedures **with** and **read** are used to make routines (or variables) available in the active worksheet. **with** is used if the routines are stored as a Maple package, as is the case with **algcures**. **with** is used if the routines are stored in a file.

- The Maple input

```
>mypack:-myfunc(myargs):
```

calls the function **myfunc** from the package **mypack** with the arguments **myargs**.

- Maple represents general algebraic numbers by using the procedure **RootOf**. For example,

```
>c := RootOf(_Z^r - 1):
```

defines c as the r -th roots of unity. More generally,

```
>c := RootOf(Poly, index = j):
```

defines c as the j -th root of the polynomial $Poly$.

- A call to the Maple procedure `time()` returns the current amount of central processing unit (CPU) time used by Maple. The time is measured from the the Maple kernel was started. Computational cost can be measured using this function, but it is a very rough measure. CPU time may not reflect actual time for a number of reasons.

B.2 The `algcures` package

The following procedures are found in the package `algcures`. This package is available in Maple, versions 8 and greater. The procedures in this package can be made available in a Maple worksheet by the invocation

```
>with(algcures):
```

- The Maple input

```
>differentials(F, x, y):
```

computes a g -dimensional basis of the differentials that are holomorphic on the Riemann surface arising from the polynomial F .

- The Maple input

```
>periodmatrix(F, x, y, 'Riemann'):
```

computes the $g \times g$ Riemann matrix associated with the Riemann surface obtained from the polynomial F .

- The Maple input

```
>puiseux(F, x = alpha, y, 0, t):
```

computes the places over $x = \alpha$ in local coordinate t [17]. That the fourth argument is zero signifies that the y -series are computed with exactly enough terms to distinguish the n branches over α away from $t = 0$.

- The Maple input

```
>is_hyperelliptic(F, x, y):
```

returns *true* if there is a bi-rational transformation that converts F to Weierstrass form or *false* otherwise.

VITA

Matthew Patterson received his Bachelors of Science degree in Applied Computational and Mathematical Sciences from the University of Washington in 2002. He received his Masters of Science degree in Applied Mathematics from the same institution in 2006. Matthew was honored with the Chowla Research Post-Doctoral Fellowship at The Pennsylvania State University for 2007-2010.