Symbolic Integration

BJÖRN TERELIUS

KTH Computer Science
and Communication

Master of Science Thesis
Stockholm, Sweden 2009

Symbolic Integration

B J Ö R N T E R E L I U S

Master's Thesis in Computer Science (30 ECTS credits)
at the School of Engineering Physics
Royal Institute of Technology year 2009
Supervisor at CSC was Inge Frick
Examiner was Johan Håstad

TRITA-CSC-E 2009:095
ISRN-KTH/CSC/E--09/095--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.csc.kth.se

Abstract

Symbolic integration is the problem of expressing an indefinite integral $\int f$ of a given function f as a finite combination g of elementary functions, or more generally, to determine whether a certain class of functions contains an element g such that $g' = f$.

In the first part of this thesis, we compare different algorithms for symbolic integration. Specifically, we review the integration rules taught in calculus courses and how they can be used systematically to create a reasonable, but somewhat limited, integration method. Then we present the differential algebra required to prove the transcendental cases of Risch's algorithm. Risch's algorithm decides if the integral of an elementary function is elementary and if so computes it. The presentation is mostly self-contained and, we hope, simpler than previous descriptions of the algorithm. Finally, we describe Risch-Norman's algorithm which, although it is not a decision procedure, works well in practice and is considerably simpler than the full Risch algorithm.

In the second part of this thesis, we briefly discuss an implementation of a computer algebra system and some of the experiences it has given us. We also demonstrate an implementation of the rule-based approach and how it can be used, not only to compute integrals, but also to generate readable derivations of the results.

Sammanfattning

Symbolisk integration

Symbolisk integration är problemet att uttrycka en obestämd integral $\int f$ av en given funktion f som en ändlig kombination g av elementära funktioner, eller mera allmänt, att avgöra huruvida en viss klass av funktioner innehåller ett element g sådant att $g' = f$.

I den första delen av det här arbetet jämför vi olika algoritmer för symbolisk integration. Mer specifikt påminner vi om de integrationsregler som lärs ut i kurser i integralkalkyl och hur de kan användas för att skapa en rimlig, om än något begränsad, integrationsmetod. Därefter presenterar vi en den differentialalgebra som behövs för att bevisa de transcendenta fallen i Risch's algoritm. Risch's algoritm avgör om integralen av en elementär funktion är elementär och beräknar i så fall denna. Presentationen är i stort sett fristående och förhoppningsvis enklare än tidigare beskrivningar. Slutligen beskriver vi Risch-Norman's algoritm som, trots att den inte kan avgöra om en integral är elementär, ofta fungerar i praktiken. Den är också väsentligt enklare än Risch's algoritm.

I den andra delen av rapporten diskuterar vi en implementation av ett datoralgebrasystem samt några av de erfarenheter det givit oss. Vi demonstrerar också en implementation av metoden med integrationsregler samt hur den kan användas, inte bara för att beräkna integraler, utan också för att generera läsbara härledningar av resultaten.

Contents

Acknowledgements

I would like to thank Dr. Inge Frick for supervising this project, which has taken much more time than I originally anticipated. As Inge Frick himself wrote one of the first computer algebra systems for tensor manipulations, I should have payed more attention to his advice concerning the time an implementation would take. Nevertheless, implementing a computer algebra system has been very instructive and I am grateful for having had the freedom to explore some interesting areas of computer algebra.

I am also very grateful to Stiftelsen Frans von Sydows Hjälpfond for providing me with generous grants during much of my studies at the Royal Institute of Technology. Last, but certainly not least, I would like to thank my family for the support they have given me while I was working on this thesis.

Chapter 1

Introduction

Symbolic integration is the problem of finding a formula $g(x)$ for the indefinite integral of a given function $f(x)$. That is, to find $g(x)$ such that

$$g(x) = \int f(x)dx$$

or equivalently $g(x)' = f(x)$. From a mathematical perspective, we could just *define* $g(x)$ as

$$g(x) = \int_a^x f(t)dt$$

for some arbitrary a. Clearly this is not very useful, as we have only replaced an indefinite integral whose properties we do not know with a function $g(x)$ whose properties we also do not know. Furthermore, determining the properties of $g(x)$ is just as difficult as determining the properties of the integral itself. What we want to do is to express the integral using only a prescribed class of "well-known" functions.

Many people immediately think of Taylor- or Fourier series as a suitable class of functions in connection with algorithmic integration, and indeed the integration problem becomes easy with this representation. Using a series solution, however, causes other problems that one would likely avoid if one represented the integral differently. For example, the series will fail to converge outside its radius of convergence and even when it converges it may converge too slowly to be practical even for numerical evaluation. It is also very difficult to see whether the series can be expressed as a product or composition of previously studied functions or series.

In symbolic integration one seeks a finite expression for the integral. To distinguish from series solutions, the name *integration in finite terms* is sometimes used instead of symbolic integration.

Definition 1.1 *Symbolic integration is the problem of expressing an indefinite integral $\int f$ of a given function f as a finite combination g of elementary functions, or more generally, to determine whether a certain class of functions contains an element g such that $g' = f$.*

In the remainder of this thesis we will study the problem of integrating elementary transcendental functions whose integrals are also elementary. We will also see examples of elementary functions whose integrals are not elementary and thus cannot be integrated in this sense.

1.1 History

1.1.1 The mathematical history

The problems of symbolically computing derivatives and indefinite integrals has been studied ever since Newton and Leibniz invented calculus and discovered the fundamental theorem in the late 17th century, thereby relating the two concepts. The symbolic differentiation problem is easy to solve thanks to the product rule and chain rule. The lack of any corresponding rules relating the integral of a product to the integrals of the factors, or the integral of a composition to the integrals of its parts, makes the integration problem much more difficult. The conventional use of integration rules and special tricks does not explain why some functions cannot be integrated in finite terms while similar integrands can be integrated easily, even by elementary methods.

The systematic study of when an integral can be expressed in finite terms began in the early 19th century. In 1820, Laplace remarked that the integral of a function cannot contain other radicals than those in the function, or in his own words

> "l'intégrale d'une fonction différentielle ne peut contenir d'autres quantités radicaux que celle qui entrent dans cette fonction"

About a decade later, Liouville stated and proved a stronger and more precise theorem which roughly states that if the integral of an elementary function is elementary, then it can be expressed using only functions that appear in the integrand and a linear combination of logarithms of such functions. This theorem is now known as Liouville's theorem or Liouville's principle.

In 1845 the Russian mathematician Ostrogradsky discovered a method for computing the rational part of the integral of a rational function, but his discovery did not become widely known outside of Russia. In 1872 Hermite found a different and in some ways simpler method for computing the rational part of the integral.

Some of the first books on the general integration problem were written by Mordukhai-Boltovskoi in 1910 and 1913, and Hardy in 1916. Hardy [16] described methods for integrating certain types of functions, but he did not believe that there could be a decision procedure for the general case of integrating elementary functions or even the case of algebraic function, as indicated by the following statement.

> "But no method has been devised as yet by which we can always determine in a finite number of steps whether a *given* elliptic integral is

> pseudo-elliptic, and integrate it if it is, and there is reason to suppose that no such method can be given."[1]

Possibly because he did not consider the case of purely transcendental integrands, Hardy regarded integrating transcendental functions as a fundamentally more difficult problem than that of integrating algebraical functions.

> "The theory of integration of transcendental functions is naturally much less complete than that of the integration of rational or even of algebraical functions. It is obvious from the nature of the case that this must be so, ..."

This remark is interesting because modern texts on the subject take the opposite view, usually only outlining integration of algebraic functions if not omitting it entirely.

In the mid 20th century, mathematicians applied new techniques from abstract algebra to the problem of integration in finite terms. The integration problem was rephrased as a problem of differential algebra by Ritt in the 1940s, and Liouville's theorem was generalized to its modern form by Ostrowski. Recently there have been some extensions of Liouville's theorem for integration in terms of non-elementary functions [1, 25], but they are somewhat complicated and beyond the scope of this text.

1.1.2 The computational history

The idea of symbolic computation in general originated at least as early as in the 1840s when Lady Augusta Ada King, Countess of Lovelace, translated an article on Babbage's Analytical Engine. In her extensive annotations to the text, she wrote:

> "Many people not conversant with mathematical studies imagine that because the business of the engine is to give its results in numerical notation, the nature of its processes must consequently be arithmetical and numerical, rather than algebraical and analytical. This is an error. The engine can arrange and combine its numerical quantities exactly as if they were letters or any other general symbols; and in fact it might bring out its results in algebraical notation were provisions made accordingly."

Little more than this observation was done until computers became available in the 1940s and 1950s. The effort to automate not only numerical calculations but also symbolic ones, gave some of its first results in 1953 with the first symbolic differentiation programs written by Nolam and Kahrimanian. Since integration is much more difficult than differentiation, the first practical symbolic integrators did not

[1] An elliptic integral is an integral of the form $\int R(x, \sqrt{P(x)})dx$ where $R(x, y)$ is a rational function and $P(x)$ a polynomial of degree three or four. Hardy calls an elliptic integral pseudo-elliptic if it can be integrated in finite terms.

appear until the 1960s when Slagle and Moses wrote SAINT and SIN respectively. These programs proceeded along the same line of thought as humans do, essentially trying to rewrite the integrand using substitutions and other transformations until it reached a form with a known method of solution. Despite their heuristic approach they achieved rather good result, outperforming an average student in both speed and rate of success.

In 1969, Risch [24] used Liouville's theorem to outline an algorithm that finds an elementary expression for the integral of an elementary functions if one exists, or otherwise proves that the integral is non-elementary. Subsequent work has focused on improving the efficiency of the algorithm and to solve the subproblems left open by Risch.

At the 1976 SYMSAC conference, Risch and Norman presented another algorithm for integration in finite terms based on Liouville's theorem. Unlike Risch's original algorithm, it can fail to compute an elementary integral even when one exists but has the advantage of being easier to implement. In practice, it successfully computes many integrals and for that reason it is often used prior to the full Risch integration algorithm.

Trager and Rothstein, in 1976 and 1977 respectively, discovered an algorithm for computing the logarithmic part of the integral of a rational function using the minimal number of algebraic extensions. Lazard and Rioboo later discovered[2] an improvement of Rothstein-Trager's algorithm that entirely avoids using algebraic extensions in the intermediate computations.

In 1981, Davenport gave an algorithm for integrands containing a single algebraic extension only depending on x, but it turned out to be impractical. A simpler method was invented by Trager in 1984. The full problem of mixed algebraic and transcendental extensions was solved by Bronstein in 1990.

[2]The improvement had previously been discovered by Trager, but he did not publish his result.

Chapter 2

Elementary techniques

2.1 Polynomials and rational functions

Polynomials are trivial to integrate by using the linearity of the integral and the rule

$$\int x^n dx = \frac{x^{n+1}}{n+1}$$

Implementations mostly differ in low level details such as how they represent polynomials. One should notice, however, that some polynomials are better integrated in other ways. A typical example of such a polynomial would be $x(x^2+1)^{1000}$.

Rational functions are considerably more difficult to integrate. The method for calculating integrals of rational functions taught in most calculus courses is to factor the denominator completely and use the factorization to compute a partial fraction expansion of the integrand, cf. appendix A. The numerators in the partial fraction expansion will be constants and the denominators will be powers of linear factors. Once a complete partial fractions expansion is known, it is easy to evaluate the integral term by term using

$$\int \frac{1}{(x-\alpha_i)^j} dx = \begin{cases} \log(x-\alpha_i) & \text{if } j = 1, \\ \frac{1}{(1-j)(x-\alpha_i)^{j-1}} & \text{otherwise.} \end{cases}$$

More elaborate methods exist, of which some will be presented in chapter 3.

2.2 Description of a heuristic algorithm

We will now discuss a more systematic version of the integration rules used in calculus courses.

2.2.1 Linearity

The first step in the heuristic integration algorithm is to use the linearity of the integral to distribute the integration over sums, and to pull out any factors that are

free of the integration variable. Using the linearity early simplifies the later rules since we then only have to consider products and compositions of functions.

Although distributing the integral over sums might seem like a harmless simplification, doing so will in some cases replace an integral that can be expressed in finite terms with two or more integrals that cannot. For example, it is easy to see that

$$\int (2x+2)e^{x^2+2x+1}dx = e^{x^2+2x+1}$$

On the other hand

$$\begin{aligned}\int (2x+2)e^{x^2+2x+1}dx &= 2\int e^{x^2+2x+1}dx + 2\int xe^{x^2+2x+1}dx \\ &= 2\int e^{(x+1)^2}dx + 2\int xe^{(x+1)^2}dx\end{aligned}$$

where we know (and will also prove in section 4.3) that $\int e^{(x+1)^2}dx$ is not elementary. It follows that the other term, $\int xe^{x^2+2x+1}dx$, cannot be elementary either since

$$\int xe^{x^2+2x+1}dx = \frac{e^{x^2+2x+1}}{2} - \int e^{(x+1)^2}dx$$

2.2.2 Simple substitutions

After the use of linearity, we try different simple substitutions, also known as the derivative-divides method or the inverse chain rule. Here we try substituting each subexpression in turn for a new variable. In other words, we try the rule

$$\int f(g(x))dx = \int \frac{f(g(x))}{g'(x)}dg$$

for all possible choices of g appearing in the expression. If some $\frac{f(g(x))}{g'(x)}$ depend only on $g(x)$ and not directly on x, we consider it a successful substitution and proceed by integrating $\frac{f(g(x))}{g'(x)}$ with respect to g. Finally, we substitute $g = g(x)$ back into the integral.

The effectiveness of this method should not be underestimated as it greatly reduces the number of integrals that the system must know. In particular, this rule allows us to integrate $f(ax+b)$ whenever we know how to integrate $f(x)$.

2.2.3 Special forms

After the use of linearity, the integrand is matched to a list of special forms. This includes all elementary functions along with a number of composite types listed in tables 2.1-2.3, where $P(x)$ denotes a polynomial in x and $R(x)$ denotes a rational function in x.

Integrand	Method
$P(x)$	Integration of a polynomial is trivial
$R(x)$	Any method described in chapter 3
x^{-1}	$\int x^{-1}dx = \log\|x\|$
x^a	$\int x^a dx = x^{a+1}/(a+1)$ if $a \neq -1$
e^x	$\int e^x dx = e^x$
$\log x$	$\int \log x dx = x\log x - x$
$\sin x$	$\int \sin x dx = -\cos x$
$\cos x$	$\int \cos x dx = \sin x$

Table 2.1. Basic integrands

Integrand	Method
$R(e^x)$	Substitute $t = e^x$ to get a rational function
$P(\sin(x), \cos(x))$	Use formulas for $\int \sin^n(x)\cos^m(x)dx$
$R(\sin(x), \cos(x))$	Substitute $t = \tan(x/2)$ to get a rational function
$P(x)\exp(x)$	Integration by parts reduces the degree of $P(x)$
$P(x)\sin(x)$	Integration by parts reduces the degree of $P(x)$
$P(x)\cos(x)$	Integration by parts reduces the degree of $P(x)$
$P(x)\log(x)$	Integration by parts gives a rational integrand
$P(x)\arcsin(x)$	Integration by parts gives an algebraic integrand
$P(x)\arccos(x)$	Integration by parts gives an algebraic integrand
$P(x)\arctan(x)$	Integration by parts gives a rational integrand

Table 2.2. Transcendental integrands

Integrand	Method
$R(x, \sqrt[n]{\frac{ax+b}{cx+d}})$	Substitute $t = \sqrt[n]{\frac{ax+b}{cx+d}}$ to get a rational integrand
$R(x, \sqrt{a^2-x^2})$	Substitute $x = a\sin t$ to get a trigonometric integrand
$R(x, \sqrt{x^2+a^2})$	Substitute $x = a\tan t$ to get a trigonometric integrand
$R(x, \sqrt{x^2-a^2})$	Substitute $x = a\cosh t$ to get a hyperbolic integrand
$R(x, \sqrt{ax^2+bx+c})$	Complete the square by substituting $t = x + \frac{b}{2a}$

Table 2.3. Algebraic integrands

2.2.4 Other transformations

As a last resort, we expand the integral by distributing products over sums. We have delayed this both because distributing will often cause an expression swell, but also and more importantly, because we will be more likely to use the linearity to replace a relatively simple integral with two that can not be integrated in finite

terms as the example in section 2.2.1.

We can also apply other transformations and simplifications to the integrand to reduce the number of functions that appear in the integrand. For example, the following identities can be used to remove products of trigonometric functions.

$$\sin(mx)\sin(nx) = \frac{\cos((m-n)x) - \cos((m+n)x)}{2}$$
$$\sin(mx)\cos(nx) = \frac{\sin((m-n)x) + \sin((m+n)x)}{2}$$
$$\cos(mx)\cos(nx) = \frac{\cos((m-n)x) + \cos((m+n)x)}{2}$$

Similarly, one can convert products of trigonometric and exponential functions to complex exponentials with the following.

$$\sin(x) = \frac{1}{\csc(x)} = \frac{e^{ix} - e^{-ix}}{2i}$$
$$\cos(x) = \frac{1}{\sec(x)} = \frac{e^{ix} + e^{-ix}}{2}$$
$$\tan(x) = \frac{1}{\cot(x)} = \frac{e^{ix} - e^{-ix}}{ie^{ix} + ie^{-ix}}$$

If hyperbolic functions are not converted to exponentials as part of the automatic simplification, it may be useful to do so during integration. The formulas for converting hyperbolic functions are of course analogous to the ones above.

The tables of integrands and methods in the previous sections was chosen such that once a rule is applied, it will eventually lead to an elementary expression for the integral. Thus, there is no problem with non-terminating rewrite sequences or trouble caused by choosing the wrong rule when several apply. However, many of the substitutions can be helpful even when this cannot be guaranteed, if treated with some care. For example, the substitution $t = \sqrt[n]{\frac{ax+b}{cx+d}}$ transforms

$$\int f(x, \sqrt[n]{\frac{ax+b}{cx+d}})dx = \int f(\frac{dt^n - b}{a - ct^n}, t)\frac{dx}{dt}dt$$

Although it is not certain that the latter is an easier problem, it will often be the case, so it makes sense to try this transformation even when f is not a rational function.

It is also possible to let the users define their own functions and integration rules, in which case they could be applied last to avoid any interference with the built-in rules.

2.3 Uses for heuristic algorithms

There are several reasons for using a heuristic, rule-based method despite the existence of definitive decision procedures like Risch's algorithm.

1. Heuristic methods are often more efficient for simple problems. To quote Geddes et al. [12], "the heuristic methods solve a trivial problem in trivial time, a highly desirable feature". For this reason, heuristics are actually tried in computer algebra systems such as Maple prior to using an algorithmic approach.

2. The Risch algorithm will usually use only exponentials and logarithms to express the result, even when it could be expressed in a simpler way for example by using inverse trigonometric functions. It is possible to convert the complex exponentials and logarithms to trigonometric functions, or to extend Risch's algorithm to work directly with these functions but doing so would complicate already complicated code. Heuristic methods, on the other hand, will usually express the integral in a similar way to what a human would do.

3. Heuristics are considerably easier to understand and implement, requiring nothing beyond introductory calculus. Risch's algorithm on the other hand requires a great deal of abstract algebra and algebraic algorithms. As a consequence, it is simple to extend the heuristic by adding new rules. Extending Risch's algorithm to include new classes of functions requires significant developments of the underlying mathematics. Such extensions are at the front of current research.

There is one additional reason for using heuristic rules rather than Risch's algorithm, and that is the ability to generate an understandable derivation of the result. It is relatively simple to modify a rule-based integration procedure to print the rules it uses to compute the integral, and to include some additional explanations if necessary. By design, the heuristic will try the same rules as a human, so the proof will look similar to what a human would produce.

In theory it is also possible to modify Risch's algorithm to generate a proof, but the proof will not be comprehensible to most humans as it relies heavily on Liouville's theorem.

Chapter 3

Integration of rational functions

It turns out that integrating rational functions are of fundamental importance to any integration algorithm. Not only are the rational functions a common and interesting class of functions, but many other types of integrals can also be reduced to integrals of rational functions by applying suitable substitutions. This was the idea behind many of the rules in the previous chapter.

3.1 The naive method

We begin by showing that the naive method taught in calculus courses is correct, and at least in theory capable of integrating any rational function $\frac{p}{q}$.

By the fundamental theorem of algebra we know that the denominator q can be written as a product of linear factors $\prod_{i=1}^{k} q_i^{e_i}$. As appendix A shows, we can use partial fraction decomposition to express $\frac{p}{q}$ as $\sum_{i=1}^{k}\sum_{j=1}^{e_i} \frac{r_{ij}}{q_i^j}$, where all r_{ij} are constants. Without loss of generality, we can assume that the factors are monic. This allows us to express the integral as follows

$$\int \frac{r_{ij}}{q_i^j} = \begin{cases} r_{ij}\log(q_i) & \text{if } j = 1, \\ \frac{r_{ij}}{(1-j)q_i^{j-1}} & \text{otherwise.} \end{cases}$$

This proves the following simple theorem which can be interpreted as a special case of Liouville's theorem (4.16).

Theorem 3.1 *Let $f \in \mathbb{Q}(x)$. Then $\int f$ is elementary and*

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

where all $c_i \in \bar{\mathbb{Q}}$, and all $v_i \in \bar{\mathbb{Q}}(x)$, Here $\bar{\mathbb{Q}}$ denotes the algebraic closure of the rational numbers, i.e. the algebraic numbers.

From a computational point of view, this method is not satisfactory. First, although the factorization exists, we cannot in general represent the factors of a

5th or higher degree polynomial using nested radicals. (This is the well-known Abel-Ruffini theorem). Secondly, even when we can represent the factors, it is still difficult to actually compute the factorization. In other words, we should try to avoid factoring as long as possible. The following sections will discuss better methods for integrating rational functions.

3.2 Hermite's method for determining the rational part

If we are to integrate a rational function we can use the division algorithm to split the integrand into a polynomial and a proper fraction. The polynomial part is trivial to integrate so we will concentrate on the fraction.

After performing a square-free factorization (cf. appendix B) of the denominator followed by a partial fraction decomposition, the fractions will be of the form q_i/r_i^i where q_i and r_i are polynomials in x, $\deg(q_i) < \deg(r_i)$ and r_i is square-free.

We integrate each such fraction in turn. To increase readability, we omit the subscript in the remainder of the section, letting r denote one of the square-free r_i and q denote the corresponding q_i. The condition that r is square-free implies that $\gcd(r, r') = 1$, so the extended euclidean algorithm computes polynomials a and b, such that

$$ar + br' = 1$$

We can use this to reduce the degree of the denominator, as shown in the following computation

$$\begin{aligned}\int \frac{q}{r^i} &= \int \frac{q(ar+br')}{r^i} = \int \frac{qa}{r^{i-1}} + \int \frac{qbr'}{r^i} \\ &= \int \frac{qa}{r^{i-1}} - \frac{qb}{(i-1)r^{i-1}} + \int \frac{(qb)'}{(i-1)r^{i-1}} \\ &= -\frac{qb}{(i-1)r^{i-1}} + \int \frac{(i-1)qa + (qb)'}{(i-1)r^{i-1}}\end{aligned}$$

which holds for any $i > 1$. When $i = 1$ we use Rothstein-Trager's method described in the next section.

3.3 Rothstein - Trager's method for the logarithmic part

In the previous section we removed any repeated factors from the denominator, so here we assume that the integrand q/r is a proper fraction where r is square-free and monic. Let the factorization of r be

$$r = \prod_{i=1}^{n} (x - a_i)$$

where all a_i are different. As we saw in section 3.1, this integral is just a sum of logarithms of the factors of r, i.e.

$$\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(x - a_i)$$

Notice that in some cases it may not be necessary to factor the denominator completely. For example, $\int 2x/(x^2-2)dx = \log(x^2-2)$ can be computed and expressed without introducing the extraneous algebraic extension $\sqrt{2}$ by factoring $x^2-2 = (x+\sqrt{2})(x-\sqrt{2})$. The problem of expressing an integral using the minimal number of algebraic extensions was solved independently by Rothstein and Trager.

Let

$$\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(v_i)$$

be the expression of the integral using the fewest possible algebraic extensions. The v_i are square-free and we can also assume that they are relatively prime without introducing new algebraic extensions, since

$$c_1 \log pq + c_2 \log qr = c_1 \log p + (c_1 + c_2) \log q + c_2 \log r$$

Lemma 3.2 *Let $\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(v_i)$ where r is a monic polynomial and the v_i are monic, square-free and relatively prime polynomials. Then*

$$r = \prod_{j=1}^{n} v_j \quad \textit{and} \quad q = \sum_{i=1}^{n} \left(c_i v_i' \prod_{j \neq i} v_j \right)$$

Proof Differentiating and cross-multiplying the denominators gives

$$q \prod_{j=1}^{n} v_j = r \sum_{i=1}^{n} \left(c_i v_i' \prod_{j \neq i} v_j \right)$$

Notice that v_i has no factor in common with v_i' (since the v_i are square-free) and no factor in common with $\prod_{j \neq i} v_j$ (since they are relatively prime). Hence v_j divides every term in the sum on the right hand side except the one where $i = j$, so it cannot be a factor of the sum. Instead, v_i must be a factor of r for all i, so $\prod v_i \mid r$. Conversely, r is a factor of the right hand side but has no factor in common with q, so $r \mid \prod v_i$. Since r and all v_i are monic, we can conclude that $r = \prod v_i$ as desired. The expression for q then follows immediately. □

At this point, it might be a good idea to recall that for all a, b, n

$$\gcd(a + nb, b) = \gcd(a, b)$$

This theorem is used repeatedly in the following computation.

$$\begin{aligned}
\gcd(q - c_l r', v_k) &= \gcd(\sum_{i=1}^{n} \left(c_i v_i' \prod_{j \neq i} v_j \right) - c_l \sum_{i=1}^{n} \left(v_i' \prod_{j \neq i} v_j \right), v_k) \\
&= \gcd(c_k v_k' \prod_{j \neq k} v_j - c_l v_k' \prod_{j \neq k} v_j, v_k) \\
&= \gcd((c_k - c_l)\, v_k' \prod_{j \neq k} v_j, v_k) \\
&= \begin{cases} \gcd(0, v_k) = v_k & \text{if } k = l \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

The last equality uses the fact that v_k has no factor in common with either v_k' or $\prod_{j \neq k} v_j$, and $c_k \neq c_l$ if $k \neq l$. Once we know the c_i, we can use the computations above to obtain the v_i by

$$\begin{aligned}
\gcd(q - c_i r', r) &= \gcd(q - c_i r', \prod_{j=1}^{n} v_j) \\
&= \prod_{j=1}^{n} \gcd(q - c_i r', v_j) \\
&= v_i
\end{aligned}$$

Notice that greatest common divisors and derivatives are computed using only rational operations, so they do not introduce any new algebraic extensions.

To obtain the c_i, we observe that they are precisely the numbers such that $\gcd(q - c_i r', r) \neq 1$, or equivalently, numbers such that $\deg(\gcd(q - c_i r', r)) > 0$. According to theorem C.7 in the appendix, $\operatorname{res}_x(q - c_i r', r) = 0$ if and only if $\deg(\gcd(q - c_i r', r)) > 0$, so it suffices to compute $\operatorname{res}_x(q - cr', r)$ and find the roots. The degree of $\operatorname{res}_x(q - cr', r)$ as a polynomial in c can not exceed the degree of r as a polynomial in x, but factoring the resultant may be easier since it can have repeated factors. Repeated factors can be found quickly using the square-free factorization described in appendix B. This proves the following theorem.

Theorem 3.3 *Let $p, q \in \mathbb{Q}[x]$ be relatively prime polynomials such that q is monic and square-free. Let S be the set of distinct zeros to $\operatorname{res}_x(q - cr', r)$. Then*

$$\int \frac{p}{q} = \sum_{c \in S} c \log(\gcd(q - cr', r))$$

is the expression for the integral which uses the fewest possible algebraic extensions of $\mathbb{Q}$.

We will finish the section with an example by Tobey of a rational function whose denominator is difficult to factor while the integral only requires a single algebraic extension of degree 2.

Example 3.4

Compute the integral $\displaystyle\int \frac{7x^{13}+10x^8+4x^7-7x^6-4x^3-4x^2+3x+3}{x^{14}-2x^8-2x^7-2x^4-4x^3-x^2+2x+1}dx$

In this example,

$$\begin{aligned} q &= 7x^{13}+10x^8+4x^7-7x^6-4x^3-4x^2+3x+3 \\ r &= x^{14}-2x^8-2x^7-2x^4-4x^3-x^2+2x+1 \end{aligned}$$

so

$$\begin{aligned} \mathrm{res}_x(q-cr',r) = &-237743967662453552c^{14}+166420777363717488864c^{13} \\ &-45765713775022309376c^{12}+582472720773011210 24c^{11} \\ &-23922986746034388992c^{10}-176822075948949831 68c^9 \\ &+15861980342479323136c^8+3417569535147769856c^7 \\ &-3965495085619830784c^6-1105137974680936448c^5 \\ &+37379666790678732 8c^4+227528406551957504c^3 \\ &+44693079858420224c^2+4063007259856384c+145107402137728 \\ = &-145107402137728\left(4c^2-4c-1\right)^7 \end{aligned}$$

The resultant has only two distinct roots $c_1 = (1+\sqrt{2})/2$ and $c_2 = (1-\sqrt{2})/2$. Computing the greatest common divisors gives

$$\begin{aligned} \gcd(q-c_1r',r) &= x^7-\sqrt{2}x^2-(1+\sqrt{2})x-1 \\ \gcd(q-c_2r',r) &= x^7+\sqrt{2}x^2-(1-\sqrt{2})x-1 \end{aligned}$$

and thus the integral

$$\begin{aligned} \int \frac{q}{r}dx = &\frac{(1+\sqrt{2})}{2}\log(x^7-\sqrt{2}x^2-(1+\sqrt{2})x-1) \\ &+\frac{(1-\sqrt{2})}{2}\log(x^7+\sqrt{2}x^2-(1-\sqrt{2})x-1) \end{aligned}$$

3.3.1 The Lazard - Rioboo - Trager improvement

Although the Rothstein-Trager algorithm can save us some factoring and does succeed in expressing the integral using a minimal number of algebraic extensions, the improvement comes at the cost of several gcd computations over algebraic number fields. As these gcd computations tend to be expensive (both in terms of running time and programming time), we would like some other way of evaluating $\gcd(q(x), p(x)-\alpha q'(x))$. Such a method was discovered and published in 1990 by Lazard and Rioboo [18] who also remarked that the method had been discovered independently (but not published) by Trager while implementing the Rothstein-Trager

algorithm in Axiom. Lazard and Rioboo's statement and proof is not entirely clear and, according to Mulders [22], the wrong interpretation was used by Geddes et al. [12] and also implemented in Axiom 2.0.

Theorem 3.5 *Let $q(x)$ and $r(x)$ be relatively prime polynomials with $\deg p(x) < \deg q(x)$ and $q(x)$ square-free as above. Let $S_i(x, y)$ be the remainder of degree i in the subresultant PRS of $q(x)$ and $p(x) - yq'(x)$ and α a zero of multiplicity n of $\mathrm{res}_x(q(x), p(x) - yq'(x))$. Then*

$$\gcd(q(x), p(x) - \alpha q'(x)) = \begin{cases} q(x) & \text{if } n = \deg(q_x) \\ \mathrm{pp}(S_n)(x, \alpha) & \text{if } n < \deg(q_x) \end{cases}$$

We will omit the proof because the technical difficulties would take us too far afield. The interested reader may consult for example Bronstein's book [4].

Chapter 4

Liouville's theorem

This chapter is concerned with proving a theorem of Liouville, which gives a precise form of the integral, if it is elementary. Before stating and proving the theorem, we need some concepts from differential algebra.

4.1 Differential algebra

Definition 4.1 *Let $\mathbb{F}$ be a field. A map $\partial : \mathbb{F} \to \mathbb{F}$ such that*

$$\begin{aligned} \partial(f+g) &= \partial f + \partial g \\ \partial(fg) &= g\partial f + f\partial g \end{aligned}$$

is called a derivation. The derivative ∂f is also written f'.

Definition 4.2 *A field $\mathbb{F}$ equipped with a derivation ∂ is called a differential field.*

Definition 4.3 *An element c of a differential field is said to be constant if $\partial c = 0$.*

Lemma 4.4 *Many of the common rules for derivatives hold in this algebraic setting. For example:*

$$\begin{aligned} \partial 0 = \partial 1 &= 0 \\ \partial(f/g) &= \frac{(\partial f)g - f\partial g}{g^2} \end{aligned}$$

Proof

$$\begin{aligned} \partial 0 = \partial(0+0) = \partial 0 + \partial 0 = 2\partial 0 \quad &\Longrightarrow \quad \partial 0 = 0 \\ \partial 1 = \partial(1\cdot 1) = 1\cdot\partial 1 + 1\cdot\partial 1 = 2\partial 1 \quad &\Longrightarrow \quad \partial 1 = 0 \end{aligned}$$

To prove the quotient rule, notice that

$$0 = \partial 1 = \partial(g\cdot g^{-1}) = g\partial(g^{-1}) + g^{-1}\partial g$$

which we can solve for ∂g^{-1} to get

$$\partial(g^{-1}) = -\frac{\partial g}{g^2}$$

By applying the product rule

$$\partial(f/g) = \partial(fg^{-1}) = \frac{\partial f}{g} - \frac{f\partial g}{g^2} = \frac{(\partial f)g - f\partial g}{g^2}$$

□

Corollary 4.5 *The set of all constants in $\mathbb{F}$ is a subfield of $\mathbb{F}$. If $\mathbb{F}$ has characteristic 0, then the constant subfield contains $\mathbb{Q}$.*

Proof The field structure follows immediately from the definition and the previous lemma. It is well known that any field of characteristic 0 must contain $\mathbb{Q}$. □

Definition 4.6 *A field $\mathbb{E}$ is an extension field of $\mathbb{F}$ if there exists an injective homomorphism $\phi : \mathbb{F} \to \mathbb{E}$. We also say that $\mathbb{F}$ is a subfield of $\mathbb{E}$.*

Definition 4.7 *Let $\mathbb{E}$ and $\mathbb{F}$ be differential fields. The field $\mathbb{E}$ is a differential extension field of $\mathbb{F}$ if it is an extension field and the homomorphism commutes with the derivation, i.e. if there exists an injective homomorphism $\phi : \mathbb{F} \to \mathbb{E}$ and $\phi(\partial f) = \partial\phi(f)$ for all $f \in \mathbb{F}$.*

The interpretation of definition 4.6 is that there exists a subset of $\mathbb{E}$ isomorphic to $\mathbb{F}$ and the interpretation of definition 4.7 is that the derivations coincide on this subset.

Definition 4.8 *Let $\mathbb{E}$ be a differential extension field of $\mathbb{F}$. An element $\theta \in \mathbb{E}$ is said to be algebraic over $\mathbb{F}$ if there exist a polynomial $p \in \mathbb{F}[x]$ such that $p(\theta) = 0$. If there is no such polynomial, then θ is transcendental over $\mathbb{F}$.*

Definition 4.9 *Let $\mathbb{E}$ be a differential extension field of $\mathbb{F}$. An element $\theta \in \mathbb{E}$ is said to be logarithmic over $\mathbb{F}$ if there exist a $u \in \mathbb{F}$ such that $\theta' = \frac{u'}{u}$. If this is the case, we write $\theta = \log(u)$.*

Definition 4.10 *Let $\mathbb{E}$ be a differential extension field of $\mathbb{F}$. An element $\theta \in \mathbb{E}$ is said to be exponential over $\mathbb{F}$ if there exist a $u \in \mathbb{F}$ such that $\theta' = u'\theta$. If this is the case, we write $\theta = \exp(u)$.*

The alert reader may have noticed that the definitions of logarithms and exponentials are given in the form of differential equations. As usual, the solution may not be unique, so in the remainder of this text $\exp(u)$ and $\log(u)$ should be interpreted as unspecified exponential and logarithmic elements with inner derivative u'. In the ordinary case of real, differentiable functions we can of course avoid the problem by specifying initial values, viz. $\exp(0) = 1$ and $\log(1) = 0$.

Theorem 4.11 *Using only the previous definitions of exponentials and logarithms in a differential field, we can deduce the following useful properties.*

1. $\log(u)$ *is unique up to an additive constant.*
2. $\exp(u)$ *is unique up to a multiplicative constant.*
3. $\log(\exp(u)) = u + c$ *where* c *is a constant.*
4. $\log(\exp(u)) = cu$ *where* c *is a constant.*
5. $\log(u) + \log(v) = \log(uv)$
6. $\exp(u)\exp(v) = \exp(u+v)$

Proof

1. Let $\log_\alpha(u)$ and $\log_\beta(u)$ both satisfy definition 4.9. Then
$$\Big(\log_\alpha(u) - \log_\beta(u)\Big)' = \frac{u'}{u} - \frac{u'}{u} = 0$$
Hence $\log_\alpha(u) - \log_\beta(u)$ is constant.

2. Let $\exp_\alpha(u)$ and $\exp_\beta(u)$ both satisfy definition 4.10. Then
$$\left(\frac{\exp_\alpha(u)}{\exp_\beta(u)}\right)' = \frac{u'\exp_\alpha(u)\exp_\beta(u) - u'\exp_\alpha(u)\exp_\beta(u)}{\exp_\beta(u)^2} = 0$$
Hence $\frac{\exp_\alpha(u)}{\exp_\beta(u)}$ is constant.

3. To prove that $\log(\exp(u)) = u + c$, it suffices to take the derivative of the left hand side, i.e.
$$(\log(\exp(u)))' = \frac{(\exp(u))'}{\exp(u)} = \frac{u'\exp(u)}{\exp(u)} = u'$$
Hence $\log(\exp(u)) = u + c$ for some constant c.

4. Similarly, one can prove that $\exp(\log(u)) = cu$ by differentiating the quotient
$$\left(\frac{\exp(\log(u))}{u}\right)' = \frac{u'\exp(\log(u)) - u'\exp(\log(u))}{u^2} = 0$$
Hence $\exp(\log(u)) = cu$ for some constant c.

5. Proving $\log(u) + \log(v) = \log(uv)$ is easy.
$$(\log(u) + \log(v))' = \frac{u'}{u} + \frac{v'}{v} = \frac{u'v + v'u}{uv} = (\log(uv))'$$

6. Proving $\exp(u)\exp(v) = \exp(u+v)$ may be less obvious but not very difficult.

$$\begin{aligned}(\exp(u)\exp(v))' &= u'\exp(u)\exp(v) + v'\exp(u)\exp(v) = \\ &= (u+v)'\exp(u)\exp(v)\end{aligned}$$

Recall that if $\theta' = f'\theta$, then $\theta = \exp(f)$ by definition. Thus $\exp(u)\exp(v) = \exp(u+v)$.

□

Following Geddes et al. [12], we will now investigate how some polynomials involving logarithms, exponentials or algebraic functions behave under differentiation.

Theorem 4.12 *Let $\mathbb{F}(\theta)$ be a differential extension field of $\mathbb{F}$ with the same field of constants, where θ is logarithmic over $\mathbb{F}$. Let $p(\theta) \in \mathbb{F}[\theta]$ be a polynomial of degree $n > 0$. Then $p(\theta)' \in \mathbb{F}[\theta]$ and*

$$\deg p(\theta)' = \begin{cases} n-1 & \text{if } c_n \text{ is constant,} \\ n & \text{otherwise.} \end{cases}$$

Proof

$$p(\theta)' = \left(\sum_{i=0}^{n} c_i\theta^i\right)' = \sum_{i=0}^{n}\left(c_i'\theta^i + ic_i\theta^{i-1}\theta'\right) = \sum_{i=0}^{n-1}\left(c_i' + (i+1)c_{i+1}\theta'\right)\theta^i + c_n'\theta^n$$

It is obvious that $\deg p(\theta)' = n$ if and only if c_n is non-constant. We must show that $c_{n-1}' + nc_n\theta' \neq 0$ if c_n is constant. Therefore suppose that c_n is constant and $c_{n-1}' + nc_n\theta' = 0$. Then $(nc_n\theta + c_{n-1})' = nc_n'\theta + nc_n\theta' + c_{n-1}' = 0$ contradicting the assumption that the extension had no new constants. □

Theorem 4.13 *Let $\mathbb{F}(\theta)$ be a differential extension field of $\mathbb{F}$ with the same field of constants, where θ is exponential over $\mathbb{F}$. Let $p(\theta) \in \mathbb{F}[\theta]$ be a polynomial of degree $n > 0$. Then $p(\theta)' \in \mathbb{F}[\theta]$ and $\deg p(\theta) = n$*

Proof

$$p(\theta)' = \left(\sum_{i=0}^{n} c_i\theta^i\right)' = \sum_{i=0}^{n}\left(c_i'\theta^i + ic_i\theta^{i-1}\theta'\right) = \sum_{i=0}^{n}\left(c_i' + ic_iu'\right)\theta^i$$

shows that $p(\theta)' \in \mathbb{F}[\theta]$. We must show that $c_n' + nc_nu' \neq 0$. But if $c_n' + nc_nu' = 0$, then $(c_n\theta^n)' = (c_n' + nc_nu')\,\theta^n = 0$, so $c_n\theta^n$ is a constant. This contradicts the assumption that the extension had no new constants. □

Definition 4.14 *A field $\mathbb{E}$ is an elementary extension of $\mathbb{F}$ if it is a differential field extension of $\mathbb{F}$ and there exists a finite tower of fields*

$$\mathbb{F} = \mathbb{E}_0 \subset \mathbb{E}_1 \subset \ldots \subset \mathbb{E}_{k-1} \subset \mathbb{E}_k = \mathbb{E}$$

such that each $\mathbb{E}_i = \mathbb{E}_{i-1}(\theta_i)$ where θ_i is algebraic, logarithmic or exponential over $\mathbb{E}_{i-1}$.

Definition 4.15 *Let $\mathbb{E}$ be a differential extension field of $\mathbb{F}$. An element $\theta \in \mathbb{E}$ is said to be elementary over $\mathbb{F}$ if $F(\theta)$ is an elementary extension field over $\mathbb{F}$.*

4.2 Liouville's theorem

We saw in the chapter on integration of rational functions that if $f \in \mathbb{Q}(x)$, then $\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$ where c_i are constants in $\bar{\mathbb{Q}}$ and $v_i \in \bar{\mathbb{Q}}(x)$. (As usual, $\bar{\mathbb{F}}$ is the algebraic closure of $\mathbb{F}$.) We shall now see how this generalizes to larger differential fields.

Theorem 4.16 *(Liouville)*
Let $\mathbb{F}$ be a differential field, and let $f \in \mathbb{F}$. If $\int f$ is in an elementary extension $\mathbb{E}$ of $\mathbb{F}$ with the same field of constants $\mathbb{K}$, then

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

where all $c_i \in \mathbb{K}$, and all $v_i \in \mathbb{F}$

Proof Let $\mathbb{E} = \mathbb{F}(\theta_1, \theta_2, \ldots, \theta_k)$ be a field containing the integral. We will use induction on the number of extensions to prove the theorem. The case of no extensions is obvious, since then $\int f = v_0$ where $v_0 \in \mathbb{F}$. We now want to prove that the theorem is true for $i+1$ extensions, given that it is true for i extensions.

Obviously, $f \in \mathbb{F}(\theta_1)$, so by the induction assumption

$$\int f = v_0(\theta_1) + \sum_{i=0}^{n} c_i \log v_i(\theta_1)$$

where the $v_0, v_1, \ldots, v_n$ are rational functions in θ_1. What remains is proving that the v_i are free of θ_1. To improve readability, we will omit the subscript and denote θ_1 just by θ. We now have three cases depending on whether θ is algebraic, a transcendental logarithm or a transcendental exponential.

4.2.1 Transcendental extensions

If θ is transcendental over the field $\mathbb{F}$, then $\mathbb{F}[\theta]$ is an euclidean domain and therefore also a unique factorization domain. This allows us to treat elements of $\mathbb{F}[\theta]$ and $\mathbb{F}(\theta)$

as polynomials or rational functions in θ, and to use well known algorithms such as the euclidean algorithm and partial fraction decomposition.

By using the logarithm rules

$$\log(fg) = \log f + \log g$$
$$\log(f/g) = \log f - \log g$$

we can assume that $v_1, \ldots, v_n$ are irreducible polynomials in $\mathbb{F}[\theta]$. Unless v_i is independent of θ, we can also factor out the leading coefficient, to make the polynomial monic. Furthermore, if two of the logarithms are equal, we can rewrite them as a single term by combining the coefficients. Hence we may safely assume that $v_1, \ldots, v_n$ are all distinct.

The element v_0 is a rational expression in $\mathbb{F}(\theta)$. We can use the euclidean algorithm to separate v_0 into a polynomial part and a proper fraction. After a partial fraction expansion,

$$v_0 = r_0(\theta) + \sum_{i=1}^{k} \sum_{j=1}^{e_i} \frac{r_{ij}(\theta)}{q_i(\theta)^j}$$

where $r_0, r_{ij}, q_i \in \mathbb{F}[\theta]$, $\deg(r_{ij}) < \deg(q_i)$ and q_i irreducible. Here we can also assume that the denominators are monic.

Since $\int f = v_0(\theta) + \sum_{i=1}^{n} c_i \log v_i(\theta)$, it follows that

$$f = r_0(\theta)' + \sum_{i=1}^{k} \sum_{j=1}^{e_i} \left(\frac{r_{ij}(\theta)'}{q_i(\theta)^j} - \frac{j r_{ij}(\theta) q_i(\theta)'}{q_i(\theta)^{j+1}} \right) + \sum_{i=0}^{n} c_i \frac{v_i(\theta)'}{v_i(\theta)}$$

The important point of this equation is that the left hand side is free of θ. After multiplying both sides of this equation by

$$d(\theta) = \operatorname{lcm}(q_1(\theta)^{e_1+1}, \ldots, q_k(\theta)^{e_k+1}, v_1(\theta), \ldots, v_n(\theta))$$

we obtain a polynomial equation in $\mathbb{F}[\theta]$

$$\begin{aligned} d(\theta) f = r_0(\theta)' d(\theta) &+ \sum_{i=1}^{k} \sum_{j=1}^{e_i} \left(\frac{r_{ij}(\theta)' d(\theta)}{q_i(\theta)^j} - \frac{j r_{ij}(\theta) q_i(\theta)' d(\theta)}{q_i(\theta)^{j+1}} \right) \\ &+ \sum_{i=0}^{n} c_i \frac{v_i(\theta)' d(\theta)}{v_i(\theta)} \end{aligned} \tag{4.1}$$

Logarithmic extensions

If θ is a transcendental logarithm, then by recalling that $q_i(\theta)$ is monic and using theorem 4.12 we see that $\deg(q_i(\theta)') < \deg(q_i(\theta))$. Since $q_i(\theta)$ is irreducible, this means that $q_i(\theta)'$ and $q_i(\theta)$ have no common factor.

For any q_i, it is easy to see that all terms in 4.1 except $\frac{e_k r_{ie_k}(\theta) q_i(\theta)' d(\theta)}{q_i(\theta)^{e_k+1}}$ are divisible by $q_i(\theta)$, and hence $q_i(\theta)$ must divide $\frac{e_k r_{ie_k}(\theta) q_i(\theta)' d(\theta)}{q_i(\theta)^{e_k+1}}$ too. But we have

established that $q_i(\theta)$ is relatively prime to $r_{ij}(\theta)$ and $q_i(\theta)'$, so $q_i(\theta)$ divides $\frac{d(\theta)}{q_i(\theta)^{e_k+1}}$. This is only possible if $q_i(\theta) \in \mathbb{F}$.

Let us consider the term $r_0(\theta)$ next. If $\deg(r_0(\theta)') > 0$ then the right hand side of equation 4.1 would have a higher degree than the left hand side. This is clearly a contradiction, so $\deg(r_0(\theta)') = 0$ and by theorem 4.12 either $\deg(r_0(\theta)) = 0$ or $\deg(r_0(\theta)) = 1$ with the coefficient of θ being a constant. In the latter case, we can move the θ-term into the sum $\sum_{i=0}^{n} c_i \log v_i$ since θ is itself a logarithm and the coefficient is constant. Without loss of generality, we assume that r_0 is free of θ.

If we insert what we have determined so far into equation 4.1, we arrive at

$$f \prod_{i=1}^{n} v_i(\theta) = v_0' \prod_{i=1}^{n} v_i(\theta) + \sum_{i=1}^{n} c_i \frac{v_i(\theta)' \prod_{i=1}^{n} v_i(\theta)}{v_i(\theta)}$$

where $v_i(\theta)$ divides all terms except $\frac{v_i(\theta)' \prod_{i=1}^{n} v_i(\theta)}{v_i(\theta)}$. To divide this term, $v_i(\theta)$ must divide $v_i(\theta)'$, which is only possible (recall theorem 4.12) if v_i is free of θ.

This proves that $v_0, v_1, \ldots, v_n \in \mathbb{F}$, as desired.

Exponential extensions

Let θ be a transcendental exponential. As before, we obtain the polynomial equation 4.1.

Just like the logarithmic case, if $\deg(r_0(\theta)') > 0$ then the right hand side of this equation would have a higher degree than the left hand side. We deduce that $\deg(r_0(\theta)') = 0$ and by theorem 4.13 $\deg(r_0(\theta)) = 0$.

Unlike the logarithmic case, it is not easy to see whether $q_i(\theta)$ is a factor of $q_i(\theta)'$. For this we need an additional lemma:

Lemma 4.17 *For any $p(\theta) \in \mathbb{F}[\theta]$, $p(\theta) \mid p(\theta)'$ if and only if p is of the form $f\theta^n$ with $f \in \mathbb{F}$.*

Proof ($\Leftarrow$) It is clear that if p is of the form above, $f\theta^n \mid (f' + nfu')\,\theta^n = (f\theta^n)'$. ($\Rightarrow$) On the other hand, if $p(\theta) \mid p(\theta)'$, then $p(\theta) = d(\theta)(p(\theta)')$. Comparing the degrees (using theorem 4.13), we see that $\deg d(\theta) = 0$. If p is not a monomial, it has at least two non-zero terms $a_n\theta^n$ and $a_m\theta^m$. These terms satisfy

$$a_n\theta^n d = (a_n' + na_nu')\theta^n$$
$$a_m\theta^m d = (a_m' + ma_mu')\theta^m$$

Eliminating d gives

$$\frac{a_m'}{a_m} - \frac{a_n'}{a_n} = (n-m)u'$$

This can be used to show that

$$\begin{aligned}\left(\frac{a_n}{a_m}\theta^{n-m}\right)' &= \left(\frac{a_n'}{a_m} - \frac{a_m'a_n}{a_m^2}\right)\theta^{n-m} + (n-m)u'\frac{a_n}{a_m}\theta^{n-m} \\ &= \left(\frac{a_m'}{a_m} - \frac{a_n'}{a_n} - (n-m)u'\right)\frac{a_n}{a_m}\theta^{n-m} = 0\end{aligned}$$

contradicting the assumption that θ is a transcendental non-constant. $\square$

Continuing with the exponential case of Liouville's theorem, we see that $q_i(\theta)$ divides all terms in 4.1 except $\frac{e_k r_{ie_k}(\theta) q_i(\theta)' d(\theta)}{q_i(\theta)^{e_k+1}}$. Since $r_i(\theta)$ is relatively prime to $q_i(\theta)$, and $d(\theta)$ contain no power of $q_i(\theta)$ greater than $q_i(\theta)^{e_k+1}$, $q_i(\theta)$ must divide $q_i(\theta)'$ to divide this term. By the lemma above, $q_i(\theta)$ must be a monomial to divide $q_i(\theta)'$. The assumption that $q_i(\theta)$ is monic and irreducible then implies that $q_i(\theta) = \theta$ unless $q_i(\theta) \in \mathbb{F}$.

Observe that when θ is exponential, we can assume that $v_i \neq \theta$ because $\log\theta = \log(\exp u) = u + c \in \mathbb{F}$. If we insert what we have determined so far into equation 4.1, we obtain

$$\begin{aligned} f d(\theta) &= r_0' d(\theta) + \sum_{j=1}^{e} \left(\frac{r_j' d(\theta)}{\theta^j} - \frac{j r_j \theta' d(\theta)}{\theta^{j+1}} \right) + \sum_{i=0}^{n} c_i \frac{v_i(\theta)' d(\theta)}{v_i(\theta)} \\ &= r_0' d(\theta) + \sum_{j=1}^{e} \frac{(r_j' - j r_j u') d(\theta)}{\theta^j} + \sum_{i=0}^{n} c_i \frac{v_i(\theta)' d(\theta)}{v_i(\theta)} \end{aligned}$$

where

$$d(\theta) = \operatorname{lcm}(\theta^e, v_1(\theta), \ldots, v_n(\theta)) = \theta^e \prod_{i=1}^{n} v_i$$

Notice that θ divides the left hand side and all terms on the right hand side except $\frac{(r_e' - e r_e u') d(\theta)}{\theta^e}$. This is a contradiction, so no θ can appear in the rational part. Similarly, all terms except $c_i \frac{v_i(\theta)' d(\theta)}{v_i(\theta)}$ are divisible by $v_i(\theta)$. Again, this is a contradiction, so all v_i are free of θ.

This proves that $v_0, v_1, \ldots, v_n \in \mathbb{F}$, as desired.

4.2.2 Algebraic extensions

Finally, suppose that θ is algebraic over $\mathbb{F}$, so there exists a polynomial $p \in \mathbb{F}[x]$ such that $p(\theta) = 0$. Now, with θ algebraic, $\mathbb{F}[\theta]$ is no longer isomorphic to the ordinary polynomial ring $\mathbb{F}[t]$ in the new variable t. It should come as no surprise that the proof of the algebraic case of Liouville's theorem is fundamentally different from the transcendental cases above.

Definition 4.18 *Let θ be algebraic over a field $\mathbb{F}$. The monic polynomial of least degree such that $p(\theta) = 0$ is called the minimal polynomial of θ over $\mathbb{F}$.*

It is not difficult to see that the minimal polynomial is unique. If there were two monic polynomials of least degree such that $p(\theta) = q(\theta) = 0$, then $p(t) - q(t)$ would be a a polynomial of lower degree but still satisfy $p(\theta) - q(\theta) = 0$.

Lemma 4.19 *Let θ be algebraic over a field $\mathbb{F}$, and let $p(t)$ be the minimal polynomial of θ. Then $\mathbb{F}(\theta)$ is isomorphic to $\mathbb{F}[t]/\langle p(t)\rangle$, where $\langle p(t)\rangle$ is the ideal generated by $p(t)$.*

Proof Define a map $\phi : \mathbb{F}[t]/\langle p(t)\rangle \to \mathbb{F}(\theta)$, by $\phi([q(t)]) = q(\theta)$ where $[q(t)]$ is the equivalence class of $q(t)$ mod $p(t)$. It is obviously a field homomorphism. It is surjective because $\operatorname{im}\phi$ is a field containing both $\mathbb{F}$ and θ, and $\mathbb{F}(\theta)$ is defined as the smallest such field. It is injective because $q(\theta) = r(\theta) \implies q(\theta) - r(\theta) = 0$ so θ is a zero of $q(t) - r(t)$. Then $p(t) \mid q(t) - r(t)$ since $p(t)$ is the minimal polynomial of θ, so $[q(t)] = [r(t)]$. □

Definition 4.20 *Let p be the minimal polynomial of θ. The roots of p (in the algebraic closure $\overline{\mathbb{F}}$) are called the conjugates of θ.*

Definition 4.21 *Let $\mathbb{F}(\theta)$ be an algebraic extension of $\mathbb{F}$ and let the conjugates of θ be $\{\theta_0, \theta_1, \ldots, \theta_k\}$. We define the norm $N : \mathbb{F}(\theta) \to \mathbb{F}$ and trace $Tr : \mathbb{F}(\theta) \to \mathbb{F}$ of an element $v(\theta)$ in $\mathbb{F}(\theta)$ by*

$$N(v(\theta)) = \prod_{i=0}^{k} v(\theta_i)$$

$$Tr(v(\theta)) = \sum_{i=0}^{k} v(\theta_i)$$

We are now ready to continue with the proof of Liouville's theorem. By the induction hypothesis

$$\int f = v_0(\theta) + \sum_{i=0}^{n} c_i \log v_i(\theta)$$

so

$$f = v_0(\theta)' + \sum_{i=0}^{n} c_i \frac{v_i(\theta)'}{v_i(\theta)}$$

where it remains to prove that the v_i are free of θ.

Let $\{\theta_0, \theta_1, \ldots, \theta_k\}$ be the set of conjugates of θ. Since all $\mathbb{F}(\theta_j)$ are isomorphic to $\mathbb{F}[t]/\langle p(t)\rangle$, it follows that

$$f = v_0(\theta_j)' + \sum_{i=0}^{n} c_i \frac{v_i(\theta_j)'}{v_i(\theta_j)}$$

for all j. Summing the equations over all conjugates gives

$$\begin{aligned}(k+1)f &= \sum_{j=0}^{k} \left(v_0(\theta_j)' + \sum_{i=0}^{n} c_i \frac{v_i(\theta_j)'}{v_i(\theta_j)} \right) = \\ &= \sum_{j=0}^{k} v_0(\theta_j)' + \sum_{i=0}^{n} c_i \frac{v_i(\theta_j)' \prod_{j\neq i} v_i(\theta_j)}{\prod_{j=0}^{k} v_i(\theta_j)} = \\ &= Tr(v_0(\theta))' + \sum_{i=0}^{n} c_i \frac{N(v_i(\theta))'}{N(v_i(\theta))}\end{aligned}$$

so

$$\int f = \frac{Tr(v_0(\theta))}{k+1} + \sum_{i=0}^{n} \frac{c_i}{k+1} \log N(v_i(\theta))$$

is another expression for the integral which does not use any algebraic extensions of $\mathbb{F}$. This concludes the proof of Liouville's theorem.

4.2.3 Strong form of Liouville's theorem

The proof of Liouville's theorem in the previous section depended on the assumption that the constant subfield of $\mathbb{E}$ containing the integral was equal to the constant subfield of $\mathbb{F}$. It is however possible to remove this restriction on the constants to obtain the following theorem.

Theorem 4.22 *Let $\mathbb{F}$ be a differential field containing the integrand f and let $\mathbb{K}$ be the subfield of constants in $\mathbb{F}$. If $\int f$ is in an elementary extension $\mathbb{E}$ of $\mathbb{F}$, then*

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

for some $v_0 \in \mathbb{F}, c_i \in \bar{\mathbb{K}}$, and $v_i \in \mathbb{F}(c_1, c_2, \ldots, c_n)$.

For a proof, see for example [4].

4.3 Examples

We will now give an example of how Liouville's theorem can be used to prove that a function defined as an integral is non-elementary.

Example 4.23 *The function $\int e^{x^2} dx$ is not elementary.*

Proof The integrand $\theta = e^{x^2}$ is in the field $\mathbb{Q}(x, \theta)$. Suppose there is an elementary expression for $\int \theta$. Then according to Liouville's theorem

$$\int \theta = \frac{p}{q} + \sum_{i=1}^{n} c_i \log(v_i)$$

where p, q and all v_i are polynomials in $\mathbb{Q}(x)[\theta]$. Differentiating and cross-multiplying the denominators gives

$$q^2\theta \prod_j v_j = (p'q - pq') \prod_j v_j + q^2 \sum_{i=1}^{n} c_i v_i' \prod_{j \neq i} v_j$$

where we can assume without loss of generality that all v_i are distinct and relatively prime. Since $\log \theta = x^2 \in \mathbb{Q}(x)$, we can also assume that no v_i is divisible by θ. Observe that for any k, all terms above except $c_k v_k' \prod_{j \neq k} v_j$ are divisible by v_k. To

divide this last term we would require that v_k divides v_k' which according to lemma 4.17 is only possible if $v_k = f\theta^m$. By the assumption, this is not the case, so all v_i are free of θ.

Next, observe that q^2 must divide $(p'q - pq')\prod_j v_j$ which implies that q^2 divides $(p'q - pq')$. As before, $q \mid q'$ implies that $q = f\theta^m$ for some $f \in \mathbb{Q}(x), k \in \mathbb{N}$. Hence, the quotient p/q can be written as a linear combination of (positive and negative) powers of θ, viz.

$$\tilde{p} = \frac{p}{q} = \sum_{i=0}^{\deg p} \frac{p_i}{f}\theta^{i-\deg q} = \sum_{i=-\deg q}^{\deg p - \deg q} \tilde{p}_i\theta^i$$

$$\tilde{p}' = \left(\sum_{i=-\deg q}^{\deg p-\deg q} \tilde{p}_i\theta^i\right)' = \sum_{i=-\deg q}^{\deg p-\deg q} (\tilde{p}_i' + 2ix\tilde{p}_i)\theta^i$$

Replacing p/q by $\tilde{p}$ in the expression for the integral gives

$$\theta = \tilde{p}' + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

from which one can immediately see that $\tilde{p}'$ and thus $\tilde{p}$ must be a polynomial in $\mathbb{Q}(x)[\theta]$ of degree one. To satisfy the equation

$$\theta = \tilde{p}_0' + (\tilde{p}_1' + 2x\tilde{p}_1)\theta + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

$\tilde{p}_1$ must be a solution to the differential equation $\tilde{p}_1' + 2x\tilde{p}_1 = 1$ in $\mathbb{Q}(x)$ and the other terms must cancel.

Let $s(x)/t(x)$ be a solution to $y' + 2xy = 1$ with $s, t \in \mathbb{Q}[x]$. After cross-multiplying the denominators to get $s(x)'t(x) - s(x)t(x)' + 2xs(x)t(x) = t(x)^2$, we see that $t(x)$ must divide $t(x)'$. This is impossible unless t is a constant, in which case we should be looking for solutions in $\mathbb{Q}[x]$ to $y' + 2xy = 1$. It is, however, clear that the equation can not have polynomial solutions since $\deg(y' + 2xy) = 1 + \deg(y) > \deg(1)$. □

Chapter 5

Risch's algorithm

This chapter will describe the proof of the following theorem:

Theorem 5.1 *(Risch, 1969)*
Let f be a function in $\mathbb{F} = \mathbb{K}(x, \theta_1, \ldots, \theta_n)$ where $\mathbb{K}$ is the field of constants in $\mathbb{F}$ and each θ_i is a transcendental logarithm or exponential over $\mathbb{K}(x, \theta_1, \ldots, \theta_{i-1})$. Then there exists an algorithm which either computes $\int f$ as an elementary function over $\mathbb{F}$ if it exists, or proves that $\int f$ is not elementary over $\mathbb{F}$

The proof is by induction on n, the number of transcendental extensions. The base case of the induction, $n = 0$, is integration of rational functions discussed in chapter 3. Assuming that the theorem holds for the field $\mathbb{K}(x, \theta_1, \ldots, \theta_{i-1})$, we must prove that it holds for $\mathbb{K}(x, \theta_1, \ldots, \theta_i)$ too. For brevity, we will drop the subscript and denote θ_i by just θ. We now have two cases depending on whether θ is logarithmic or exponential.

Although the proofs are more complicated, the integration methods will ultimately turn out to be similar to the ones described in chapter 3, with the exception that integration of a polynomial in θ is non-trivial.

5.1 Logarithmic extensions

Let θ be a logarithm and $f \in \mathbb{F}(\theta)$ We can express f as a $p + \frac{q}{r}$ where $p, q, r \in \mathbb{F}[\theta]$, and $\deg(q) < \deg(r)$.

We begin with a decomposition lemma from Davenport et al. [10].

Lemma 5.2 *If $\int f$ is elementary, then $\int p$ and $\int \frac{q}{r}$ are elementary too, so we can integrate the polynomial part p, and rational part $\frac{q}{r}$ separately.*

Proof By Liouville's principle, if $\int f$ is elementary then

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

so

$$f = p + \frac{q}{r} = v_0' + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i} = \left(\tilde{p} + \frac{\tilde{q}}{\tilde{r}}\right)' + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

Some of the v_i are polynomials dependent on θ and some are independent of θ. We'll assume that the ones independent of θ are $v_1, v_2, \ldots, v_k$. Using lemma 4.12, we see that the derivative of the polynomial $\tilde{p}$ is a polynomial, and the derivative of the proper fraction $\tilde{q}/\tilde{r}$ is a proper fraction. Recall that the decomposition into a polynomial part and a proper fraction is unique and apply this to the equation above.

$$p = \tilde{p}' + \sum_{i=1}^{k} c_i \frac{v_i'}{v_i}$$
$$\frac{q}{r} = \left(\frac{\tilde{q}}{\tilde{r}}\right)' + \sum_{i=k+1}^{n} c_i \frac{v_i'}{v_i}$$

Integrating these equations gives

$$\int p = \tilde{p} + \sum_{i=1}^{k} c_i \log v_i$$
$$\int \frac{q}{r} = \frac{\tilde{q}}{\tilde{r}} + \sum_{i=k+1}^{n} c_i \log v_i$$

so both integrals are elementary as well. □

5.1.1 Polynomial part

Let $p = \sum a_i \theta^i$ and $\tilde{p} = \sum b_i \theta^i$. The decomposition lemma above implies that

$$\sum_{i=0}^{\deg(p)} a_i \theta^i = \sum_{i=0}^{\deg(\tilde{p})} \left(b_i' + (i+1) b_{i+1} \theta'\right) \theta^i + \sum_{i=k+1}^{n} c_i \frac{v_i'}{v_i}$$

Comparing the degrees, we see that $\deg(\tilde{p}) = \deg(p)$ unless $\deg(\tilde{p}) = \deg(p) + 1$ and the leading coefficient in $\deg(\tilde{p})$ is a constant. Equating the coefficients gives $a_i = b_i' + (i+1) b_{i+1} \theta'$ for all $i > 0$, so

$$b_i = \int \left(a_i - (i+1) b_{i+1} \theta'\right) + d_i$$

where d_i is a constant. The value of the constant d_{i+1} is determined by the condition that

$$\begin{aligned} b_i &= \int \left(a_i - (i+1) b_{i+1} \theta'\right) + d_i = \\ &= \int \left(a_i - (i+1)(b_{i+1} - d_{i+1} + d_{i+1}) \theta'\right) + d_i = \\ &= \int \left(a_i - (i+1)(b_{i+1} - d_{i+1}) \theta'\right) - (i+1) d_{i+1} \theta + d_i \end{aligned}$$

should be free of θ. Notice that $b_{i+1} - d_{i+1}$ is precisely the integral appearing in the equation for b_{i+1}. We can thus compute the b_i starting with the leading coefficient and working our way down to b_1. When we equate the coefficients of terms of degree zero, we get

$$a_0 = b_0' + b_1\theta' + \sum_{i=k}^{n} c_i \frac{v_i'}{v_i}$$

so

$$b_0 + \sum_{i=k}^{n} c_i \log v_i = \int (a_0 - b_1\theta') + d_0$$

where the last constant d_0, which is not determined by any condition, is the constant of integration.

If some integral in the computation of the b_i involves extensions of $\mathbb{F}$ other than θ, then the integral of the polynomial part cannot be elementary.

5.1.2 Rational part

The previous section treated the polynomial part of the integrand, so in this section we assume that the integrand is a proper fraction q/r where q and r are polynomials in θ. After performing a square-free factorization of the denominator, followed by a partial fraction decomposition, the fractions will be of the form q_i/r_i^i where q_i and r_i are polynomials, $\deg(q_i) < \deg(r_i)$ and the r_i are square-free. We integrate each such fraction in turn.

Lemma 5.3 *Let $r \in \mathbb{F}[\theta]$ be a monic square-free polynomial. Then* $\gcd(r, \frac{d}{dx}r) = 1$.

Proof Let r have the factorization

$$r = \prod_{i=0}^{n} (\theta - a_i)$$

in $\bar{\mathbb{F}}[\theta]$ where all a_i are distinct. Then the derivative of r with respect to x is

$$r' = \sum_{i=0}^{n} (\theta' - a_i') \prod_{i \neq j} (\theta - a_j)$$

All terms on the right hand side except one are divisible by $\theta - a_j$, but as the last term is not divisible by $\theta - a_j$, the left hand side of the equation cannot be either. Thus the only possible factors of r does not divide r', and we can deduce that $\gcd(r, r') = 1$. □

Corollary 5.4 *Let $r \in \mathbb{F}[\theta]$ be a monic square-free polynomial, i.e. under the same conditions as the lemma above, then there exist polynomials $a, b \in \mathbb{F}[\theta]$ such that $ar + br' = 1$.*

Proof $\mathbb{F}[\theta]$ is an euclidean domain, so the extended euclidean algorithm gives a and b as desired. □

Returning to the problem of finding the rational part, we get

$$\begin{aligned}\int \frac{q}{r^k} &= \int \frac{q(ar+br')}{r^k} = \int \frac{qa}{r^{k-1}} + \int \frac{qbr'}{r^k} \\ &= \int \frac{qa}{r^{k-1}} - \frac{qb}{(k-1)r^{k-1}} + \int \frac{(qb)'}{(k-1)r^{k-1}} \\ &= -\frac{qb}{(k-1)r^{k-1}} + \int \frac{(k-1)qa + (qb)'}{(k-1)r^{k-1}}\end{aligned}$$

This reduces the degree of the denominator, so we can repeat this step until $k = 1$. As we shall see, we have fully determined the rational part of the integral when the denominator no longer has any repeated factors.

5.1.3 Logarithmic part

In the previous section we removed any repeated factors from the denominator, so here we assume that the integrand is q/r where r is square-free and monic.

Lemma 5.5 *Let $\frac{q}{r}$ be a proper fraction in $\mathbb{F}(\theta)$ such that the denominator is square-free. Without loss of generality we can assume that r is monic, so it has the factorization $r = \prod_{i=1}^{n}(\theta - a_i)$ (in $\bar{\mathbb{F}}[\theta]$) where all a_i are different. Then if $\int \frac{q}{r}$ is elementary,*

$$\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(\theta - a_i)$$

Proof From Liouville's theorem we know that

$$\int \frac{q}{r} = \frac{\tilde{q}}{\tilde{r}} + \sum_{i=1}^{n} c_i \log(v_i)$$

or equivalently

$$\frac{q}{r} = \frac{\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q}}{\tilde{r}^2} + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

As we did in the proof of Liouville's theorem, we can assume that the v_i are distinct, irreducible, monic polynomials, so after cross-multiplication we get the polynomial equation

$$q\tilde{r}^2 \prod_j v_j = (\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q})r \prod_j v_j + \sum_{i=1}^{n} \left(c_i r \tilde{r}^2 v_i' \prod_{j \neq i} v_j \right)$$

As we can see, $\tilde{r}^2$ divides all terms except $(\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q})r \prod_j v_j$. The polynomial $\tilde{r}$ does not divide $\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q}$ since it does not divide $\tilde{r}'$ and is relatively prime to $\tilde{q}$. Furthermore, it can only divide $\prod_j v_j$ once, as all v_i are irreducible and distinct.

Since r is square-free by assumption, $\tilde{r}$ can divide r at most once, so it must be the case that $\tilde{r} \mid \prod_j v_j$ and $\tilde{r} \mid r$. But this means that $\tilde{r}^3$ divides all terms except $(\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q})r\prod_j v_j$ and since $\tilde{r}$ can divide $\prod_j v_j$ only once, $\tilde{r}^2$ must divide r. This is a contradiction, so there cannot be any proper rational part. Thus

$$\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(v_i)$$
$$\frac{q}{r} = \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

and

$$q\prod_j v_j = r\sum_{i=1}^{n}\left(c_i v_i' \prod_{j\neq i} v_j\right)$$

Since every factor of r divides the right hand side, it must divide some v_i on the left hand side too. The v_i are irreducible, so every factor of r must in fact be equal to some v_i. Conversely, every v_i divides all terms in the sum except $c_i v_i' \prod_{j\neq i} v_j$, so it must divide r instead. We can conclude that the v_i are precisely the factors of r. □

If we cancel $r = \prod_j v_j$ from the last equation in the proof, we obtain

$$q = \sum_{i=1}^{n}\left(c_i(\theta' - a_i')\prod_{j\neq i}(\theta - a_j)\right)$$

which we can use to identify the c_i with appropriate coefficients in q. When determining the c_i, we should remember that they must be constants if the integral is elementary.

The problem with this approach is that it requires a complete factorization of r. Just like the case with rational functions in chapter 3, we can use the Rothstein-Trager method or the Lazard-Rioboo-Trager method to compute the integral using the minimal number of algebraic extensions.

Example 5.6 *The logarithmic integral* $\int \frac{1}{\log x}dx$ *is not elementary.*

Proof Let θ denote $\log x$ and notice that the integrand $\frac{1}{\theta}$ is a proper fraction with square-free denominator. It follows from the previous lemma that if the integral is elementary, it must be of the form $c\log\theta$. However, $(c\log\theta)' = c\frac{1}{x\theta} \neq \frac{1}{\theta}$ for every constant c, so the integral can not be elementary. □

5.2 Exponential extensions

If θ is exponential rather than logarithmic, the only major difference is that the polynomial part is more complicated to integrate. For reasons that will become

clear in the proofs, we express the integrand $f \in \mathbb{F}(\theta)$ as $p + \frac{q}{r}$ where $p \in \mathbb{F}[\theta, \theta^{-1}]$, $q, r \in \mathbb{F}[\theta]$, $\deg(q) < \deg(r)$ and θ does not divide r. The effect of this representation is that we will handle negative powers of θ in the polynomial part. We will again use a decomposition lemma, similar to the one from Davenport et al. [10], to integrate the p and $\frac{q}{r}$ separately.

Lemma 5.7 *If $\int f$ is elementary, then $\int p$ and $\int \frac{q}{r}$ are elementary too, so we can integrate the polynomial part p and rational part $\frac{q}{r}$ separately.*

Proof By Liouville's principle, if $\int f$ is elementary then

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

so

$$f = p + \frac{q}{r} = v_0' + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i} = \left(\tilde{p} + \frac{\tilde{q}}{\tilde{r}}\right)' + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

where $\tilde{p} + \frac{\tilde{q}}{\tilde{r}}$ is an expression for v_0 such that $\tilde{p} \in \mathbb{F}[\theta, \theta^{-1}]$, $\tilde{q}, \tilde{r} \in \mathbb{F}[\theta]$, $\deg(\tilde{q}) < \deg(\tilde{r})$ and θ does not divide $\tilde{r}$.

Some of the v_i in the logarithms are polynomials that depend on θ and some are independent of θ. Assume without loss of generality that $v_1, v_2, \ldots, v_k$ are independent of θ, and that $v_{k+1}, v_{k+2}, \ldots, v_n$ are monic and not divisible by θ. Unlike the logarithmic case, $\deg(v_i') = \deg(v_i)$, so $\frac{v_i'}{v_i}$ is not a proper fraction. To remedy the situation, notice that

$$\frac{v_i'}{v_i} - n_i u' = \frac{v_i' - n_i u' v_i}{v_i}$$

is a proper fraction if n_i is the degree of v_i and u' is the inner derivative of θ. Thus

$$p + \frac{q}{r} = \left(\tilde{p} + \frac{\tilde{q}}{\tilde{r}}\right)' + \sum_{i=1}^{k} c_i \frac{v_i'}{v_i} + \sum_{i=k+1}^{n} c_i \frac{v_i' - n_i u' v_i}{v_i} + \sum_{i=k+1}^{n} c_i n_i u'$$

Using lemma 4.13, one can see that the derivative of $\tilde{p}$ is a polynomial in $\mathbb{F}[\theta, \theta^{-1}]$, and the derivative of $\tilde{q}/\tilde{r}$ is a proper fraction such that θ does not divide the denominator. Identifying polynomial parts (in $\mathbb{F}[\theta, \theta^{-1}]$) and proper fractions with denominators not divisible by θ gives

$$p = \tilde{p}' + \sum_{i=1}^{k} c_i \frac{v_i'}{v_i} + \sum_{i=k+1}^{n} c_i n_i u'$$

$$\frac{q}{r} = \left(\frac{\tilde{q}}{\tilde{r}}\right)' + \sum_{i=k+1}^{n} c_i \frac{v_i' - n_i u' v_i}{v_i}$$

Integrating these equations gives

$$\int p = \tilde{p} + \sum_{i=1}^{k} c_i \log v_i + \sum_{i=k+1}^{n} c_i n_i u$$
$$\int \frac{q}{r} = \frac{\tilde{q}}{\tilde{r}} + \sum_{i=k+1}^{n} c_i (\log v_i - n_i u)$$

so both these integrals are elementary as well. □

5.2.1 Polynomial part

Let $p = \sum a_i \theta^i$ and $\tilde{p} = \sum b_i \theta^i$ where the index i may take on both positive and negative values. Equating terms of the same degree in the decomposition lemma above gives

$$a_0 = b_0' + \sum_{i=1}^{k} c_i \frac{v_i'}{v_i} + \sum_{i=k+1}^{n} c_i n_i u'$$
$$a_i \theta^i = \left(b_i \theta^i\right)' = (b_i' + i b_i u')\theta^i \qquad \text{if } i \neq 0$$

It is easy to integrate the first equation to get

$$b_0 = \int a_0 - \sum_{i=1}^{k} c_i \log v_i - \sum_{i=k+1}^{n} c_i n_i u$$

where the v_i and c_i (for $1 \leq i \leq k$) are to be chosen in such a way as to cancel any logarithmic extension in $\int a_0$.

To compute the coefficients b_i, we have to solve differential equations of the form

$$b_i' + i u' b_i = a_i$$

known as Risch's differential equation. The problem of solving Risch's differential equation will be discussed further in chapter 6.

5.2.2 Rational part

We now turn to the problem of integrating a proper fraction q/r where q and r are polynomials in θ with $\deg(q) < \deg(r)$. Like before, the idea is to use Hermite's reduction to simplify the integrand, but to do so we need the denominator to satisfy $\gcd(r, r') = 1$. When the denominator was a polynomial in either x or a logarithm θ, it was sufficient to make the denominator square-free. Unfortunately, it is not as simple when θ is an exponential. For example, this fails for the simple polynomial θ, as $\gcd(\theta, \theta') = \gcd(\theta, u'\theta) = \theta$. The way to avoid this problem is to not only do a square-free factorization, but also factor out the largest power of θ appearing in

the denominator. After this factorization and a partial fraction decomposition, the fractions will be of the form q_i/r_i^i where q_i and r_i are polynomials, $\deg(q_i) < \deg(r_i)$ and r_i is monic, square-free and either equal to θ, or not divisible by θ.

Lemma 5.8 *Let $r \in \mathbb{F}[\theta]$ be a monic, square-free polynomial of positive degree, such that $\theta \nmid r$. Then*

$$\gcd(r, r') = 1$$

Proof Let the factorization of r in $\bar{\mathbb{F}}[\theta]$ be

$$r = \prod_{i=0}^{n} (\theta - a_i)$$

where all a_i are distinct and non-zero. Then the derivative of r with respect to x is

$$r' = \sum_{i=0}^{n} (\theta' - a_i') \prod_{i \neq j} (\theta - a_j)$$

All terms on the right hand side, except the one where $i = j$, are divisible by $\theta - a_j$. The only way this last term term can be divisible by $\theta - a_j$ is if $\theta - a_j \mid \theta' - a_j'$ According to lemma 4.17, this can only happen if a_j is zero which contradicts the assumption. Since none of the factors of r divide r', we can deduce that $\gcd(r, r') = 1$. □

Corollary 5.9 *Let $r \in \mathbb{F}[\theta]$ be a monic square-free polynomial not divisible by θ, i.e. under the same conditions as the lemma above, then there exist polynomials $a, b \in \mathbb{F}[\theta]$ such that $ar + br' = 1$.*

Proof $\mathbb{F}[\theta]$ is an euclidean domain, so the extended euclidean algorithm gives a and b as desired. □

Returning to the problem of finding the rational part, we get

$$\begin{aligned}
\int \frac{q}{r^k} &= \int \frac{q(ar + br')}{r^k} = \int \frac{qa}{r^{k-1}} + \int \frac{qbr'}{r^k} \\
&= \int \frac{qa}{r^{k-1}} - \frac{qb}{(k-1)r^{k-1}} + \int \frac{(qb)'}{(k-1)r^{k-1}} \\
&= -\frac{qb}{(k-1)r^{k-1}} + \int \frac{(k-1)qa + (qb)'}{(k-1)r^{k-1}}
\end{aligned}$$

This reduces the degree of the denominator, so we can repeat this step until $k = 1$. As we shall see, we have fully determined the rational part of the integral when the denominator no longer has any repeated factors.

5.2.3 Logarithmic part

In the previous section we removed any repeated factors from the denominator, so here we assume that the integrand is q/r where r is square-free and monic.

Lemma 5.10 *Let $\frac{q}{r}$ be a proper fraction in $\mathbb{F}(\theta)$ such that the denominator is square-free, and not divisible by θ. Without loss of generality we can assume that r is monic, so it has the factorization $r = \prod_{i=1}^{n}(\theta - a_i)$ (in $\bar{\mathbb{F}}[\theta]$) where all a_i are different. Then if $\int \frac{q}{r}$ is elementary,*

$$\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(\theta - a_i)$$

Proof From Liouville's theorem we know that

$$\int \frac{q}{r} = \frac{\tilde{q}}{\tilde{r}} + \sum_{i=1}^{n} c_i \log(v_i)$$

or equivalently

$$\frac{q}{r} = \frac{\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q}}{\tilde{r}^2} + \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

As we did in the proof of Liouville's theorem, we can assume that the v_i are distinct, irreducible, monic polynomials, so after cross-multiplication we get the polynomial equation

$$q\tilde{r}^2 \prod_j v_j = (\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q})r \prod_j v_j + \sum_{i=1}^{n} \left(c_i r \tilde{r}^2 v_i' \prod_{j \neq i} v_j \right)$$

As we can see, $\tilde{r}^2$ divides all terms except $(\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q})r \prod_j v_j$. The polynomial $\tilde{r}$ does not divide $\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q}$ since it does not divide either of $\tilde{r}'$ and $\tilde{q}$ (recall lemma 4.17). Furthermore, it can only divide $\prod_j v_j$ once, as all v_i are irreducible and distinct. Since r is square-free by assumption, $\tilde{r}$ can divide r at most once, so it must be the case that $\tilde{r} \mid \prod_j v_j$ and $\tilde{r} \mid r$. But this means that $\tilde{r}^3$ divides all terms except $(\tilde{q}'\tilde{r} - \tilde{r}'\tilde{q})r \prod_j v_j$ and since $\tilde{r}$ can divide $\prod_j v_j$ only once, $\tilde{r}^2$ must divide r. This contradicts the assumption that r is square-free, so there cannot be any proper rational part. Thus

$$\int \frac{q}{r} = \sum_{i=1}^{n} c_i \log(v_i)$$

$$\frac{q}{r} = \sum_{i=1}^{n} c_i \frac{v_i'}{v_i}$$

and

$$q \prod_j v_j = r \sum_{i=1}^{n} \left(c_i v_i' \prod_{j \neq i} v_j \right)$$

Since every factor of r divides the right hand side, it must divide some v_i on the left hand side too. The v_i are irreducible, so every factor of r must in fact be equal to some v_i. Conversely, every v_i divides all terms in the sum except $c_i v_i' \prod_{j \neq i} v_j$, so it must divide r instead. We can conclude that the v_i are precisely the factors of r. □

Just as for the logarithmic extensions, we can obtain the coefficients c_i by solving the linear system

$$q = \sum_{i=1}^{n} \left(c_i(\theta' - a_i') \prod_{j \neq i} (\theta - a_j) \right)$$

but using this approach requires the full factorization of r. As before, we can use the Rothstein-Trager method or the Lazard-Rioboo-Trager method to avoid this problem.

Chapter 6

The Risch differential equation

The previous chapter reduced the problem of integrating a polynomial of an exponential to solving a certain differential equation known as the Risch differential equation. The goal of this chapter will be to solve that equation, i.e. find a solution $y \in \mathbb{F}(\theta)$ to

$$y' + fy = g$$

if one exists. The description given here is the one of Bronstein [3, 4], who gave a direct formula for the denominator of y.

6.1 Canonical representation

Definition 6.1 *Let $\mathbb{F}$ be a differential field and θ transcendental over $\mathbb{F}$ such that $\theta' \in \mathbb{F}[\theta]$. A polynomial $p \in \mathbb{F}[\theta]$ is called normal if $\gcd(p, p') = 1$, and special if $\gcd(p, p') = p$.*

Theorem 6.2 *Let θ be transcendental over $\mathbb{F}$ and $p \in \mathbb{F}[\theta]$.*

1. *If θ is logarithmic, then p normal $\iff$ p square-free.*
2. *If θ is exponential, then p normal $\iff$ p square-free and $\theta \nmid p$.*

Proof This is an immediate consequence of lemma 4.12 and 4.13. □

Definition 6.3 *Let $\mathbb{F}\langle\theta\rangle$ be the set of elements in $\mathbb{F}(\theta)$ whose denominators are special, i.e. $\mathbb{F}[\theta]$ if θ is a logarithm, and $\mathbb{F}[\theta, \theta^{-1}]$ if θ is exponential.*

Definition 6.4 *We define the canonical representation of f as a quotient p/q with $p \in \mathbb{F}\langle\theta\rangle$, $q \in \mathbb{F}[\theta]$, such that*

1. *p and q are relatively prime*
2. *q is monic*

3. all irreducible factors of q are normal

The first two conditions in definition 6.4 are the usual conditions on a canonical representation of a fraction. The third condition means precisely that $\theta \nmid q$ if θ is exponential.

Definition 6.5 *An element $f \in \mathbb{F}(\theta)$ is weakly normalized with respect to θ if*

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

for some $v_0 \in \mathbb{F}(\theta)$, $v_1 \dots v_n \in \mathbb{F}[\theta]$ and constants $c_i \notin \mathbb{Z}^+$.

Lemma 6.6 *Let $f, g \in \mathbb{F}(\theta)$. If $\int f$ is elementary over $\mathbb{F}(\theta)$, there exists $\tilde{f}, \tilde{g} \in \mathbb{F}(\theta)$ with $\tilde{f}$ weakly normalized, such that*

$$y' + fy = g \Longleftrightarrow z' + \tilde{f}z = \tilde{g}$$

where $z = py$ for some $p \in \mathbb{F}[\theta]$.

Proof We are done if f already is weakly normalized. According to Liouville's theorem,

$$\int f = v_0 + \sum_{i=1}^{n} c_i \log v_i$$

where $v_0 \in \mathbb{F}(\theta)$, $v_1 \dots v_n \in \mathbb{F}[\theta]$. The only way f can fail to be weakly normalized is when some of the constants are positive integers. Suppose that $c_1 \dots c_k$ are the positive integer coefficients and let

$$\begin{aligned} p &= \prod_{i=1}^{k} v_i^{c_i} \\ \tilde{f} &= f - \frac{p'}{p} \\ \tilde{g} &= pg \end{aligned}$$

Then

$$z' + \tilde{f}z = p'y + py' + (f - \frac{p'}{p})py = p(y' + fy) = \tilde{g}$$

where $\tilde{f}$ is weakly normalized since

$$\int \tilde{f} = \int f - \log p = v_0 + \sum_{i=1}^{n} c_i \log v_i - \sum_{i=1}^{k} c_i \log v_i = v_0 + \sum_{i=k+1}^{n} c_i \log v_i$$

and none of the coefficients $c_{k+1} \dots c_n$ is a positive integer. □

6.2 The denominator

This section will give a formula for the denominator of y and a new differential equation $aq' + bq = c$ for the numerator of y.

Definition 6.7 *Let $\mathbb{F}$ be a field and $p \in \mathbb{F}[\theta]$ be irreducible. Any element $f \in \mathbb{F}(\theta)\backslash\{0\}$ can be written uniquely as*

$$f = p^n \frac{q}{r}$$

where $n \in \mathbb{Z}$, $p \nmid q$, $p \nmid r$, $\gcd(q, r) = 1$ and r monic. We define a p-adic valuation ν_p of f by

$$\nu_p(f) = n$$

Lemma 6.8 *Let p be irreducible in $\mathbb{F}[\theta]$, $a, b \in \mathbb{F}[\theta]$ and $f, g \in \mathbb{F}(\theta)$.*

1. *$\nu_p(fg) = \nu_p(f) + \nu_p(g)$*
2. *$\nu_p(f + g) \geq \min(\nu_p(f), \nu_p(g))$ with equality if $\nu_p(f) \neq \nu_p(g)$*
3. *$\nu_p(\gcd(f, g)) = \min(\nu_p(f), \nu_p(g))$*
4. *If $\mathbb{F}$ has characteristic 0 and $\nu_p(f) \neq 0$, then $\nu_p(\frac{df}{d\theta}) = \nu_p(f) - 1$*

Proof The proofs are straightforward.

1. Let f and g have the canonical representations

$$f = p^{n_f} \frac{q_f}{r_f} \quad \text{and} \quad p^{n_g} \frac{q_g}{r_g}$$

$$fg = p^{n_f} \frac{q_f}{r_f} p^{n_g} \frac{q_g}{r_g} = p^{n_f + n_g} \frac{q_f q_g}{r_f r_g}$$

   where $q_f q_g$ and $r_f r_g$ are relatively prime to p, so $\nu_p(fg) = \nu_p(f) + \nu_p(g)$.

2. Without loss of generality we can assume that $\min(\nu_p(f), \nu_p(g)) = \nu_p(f)$. Then

$$f + g = p^{n_f} \frac{q_f}{r_f} + p^{n_g} \frac{q_g}{r_g} = p^{n_f} \frac{q_f r_g + r_f q_g p^{n_g - n_f}}{r_f r_g}$$

   and $r_f r_g$ are relatively prime to p. If $n_g \neq n_f$, then $q_f r_g + r_f q_g p^{n_g - n_f}$ is also relatively prime to p, but in general it could be divisible by p. Hence $\nu_p(f + g) \geq \min(\nu_p(f), \nu_p(g))$.

3. If f and g are polynomials, $\gcd(f, g) = \gcd(p^{n_f} q_f, p^{n_g} q_g) = p^{\min(n_f, n_g)} \gcd(q_f, q_g)$ so $\nu_p(\gcd(f, g)) = \min(\nu_p(f), \nu_p(g))$.

4. Let ∂_θ denote differentiation with respect to θ. If $f = p^n q/r$ and $n \neq 0$, then

$$\frac{df}{d\theta} = np^{n-1}(\partial_\theta p)\frac{q}{r} + p^n\frac{(\partial_\theta q)r - q\partial_\theta r}{r^2} = p^{n-1}\frac{n(\partial_\theta p)qr + p(\partial_\theta q)r - pq\partial_\theta r}{r^2}$$

where both numerator and denominator are relatively prime to p. Hence $\nu_p(\frac{df}{d\theta}) = \nu_p(f) - 1$.

□

Lemma 6.9 *Let p be a normal irreducible polynomial in $\mathbb{F}[\theta]$. If f is weakly normalized with respect to θ and $\nu_p(y) < 0$ then*

$$\nu_p(y' + fy) = \nu_p(y) + \min(-1, \nu_p(f))$$

Proof Express y as $p^n q/r$ where p does not divide q or r. The derivative y' is

$$y' = p^{n-1}\frac{np'qr + pq'r + pqr'}{r^2}$$

where p does not divide the denominator r^2. Furthermore, p cannot divide the numerator $np'qr + pq'r + pqr'$ since $n \neq 0$, $p \nmid qr$ and we assume that p is normal, so $p \nmid p'$. We can conclude that $\nu_p(y') = \nu_p(y) - 1$.

If $\nu_p(f) \neq -1$, then $\nu_p(y') \neq \nu_p(fy)$ so

$$\begin{aligned}\nu_p(y' + fy) &= \min(\nu_p(y'), \nu_p(fy)) = \min(\nu_p(y) - 1, \nu_p(f) + \nu_p(y)) = \\ &= \nu_p(y) + \min(-1, \nu_p(f))\end{aligned}$$

as desired.

On the other hand, if $\nu_p(f) = -1$ we let $f = p^{-1}s/t$, so

$$y' + fy = p^{n-1}\frac{np'qr + pq'r + pqr'}{r^2} + p^{n-1}\frac{qs}{rt} = p^{n-1}\frac{np'qrt + pq'rt + pqr't + qrs}{r^2t}$$

Clearly p does not divide the denominator and to divide the numerator it would have to divide $np't + s$. Let $u \in \mathbb{F}[\theta]$ be such that $np't + s = pu$. Then

$$\int f = \int p^{-1}\frac{s}{t} = \int p^{-1}\frac{pu - np't}{t} = \int \frac{u}{t} - n\int \frac{p'}{p} = \int \frac{u}{t} - n\log(p)$$

would have logarithmic parts with positive integer coefficients, contradicting the assumption that f is weakly normalized. □

Theorem 6.10 *(Bronstein)*
Let y be the solution of $y' + fy = g$ in $\mathbb{F}(\theta)$, where f is weakly normalized with respect to θ. Let the canonical representations of f, g and y be

$$f = \frac{A}{D}, \quad g = \frac{B}{E} \quad \textit{and} \quad y = \frac{Q}{T}$$

and let $G = \gcd(D, E)$. Then

1. *The denominator of y is*
$$T = \frac{\gcd\left(E, \frac{dE}{d\theta}\right)}{\gcd\left(G, \frac{dG}{d\theta}\right)}$$

2. *If $y' + fy = g$ has a solution in $\mathbb{F}(\theta)$, then $E \mid DT^2$ and the numerator Q satisfies the equation*
$$DTQ' + (AT - DT')Q = \frac{BDT^2}{E}$$

Proof

1. We want to show that $Q = Ty \in \mathbb{F}\langle\theta\rangle$ by showing that $\nu_p(Q) \geq 0$ for any normal irreducible $p \in \mathbb{F}[\theta]$. If $\nu_p(y) \geq 0$, then $\nu_p(Q) = \nu_p(T) + \nu_p(y) \geq 0$ and we are done. Otherwise $\nu_p(y) < 0$ and we have two cases depending on the sign of $\nu_p(f)$. Notice that $\nu_p(g) = \nu_p(y' + fy) = \nu_p(y) + \min(-1, \nu_p(f)) < 0$, so $\nu_p(E) = -\nu_p(g) > 0$.

   If $\nu_p(f) \geq 0$, then $\nu_p(G) = \nu_p(\gcd(D, E)) = \min(\nu_p(D), \nu_p(E))) = \nu_p(D) = 0$ so
$$\nu_p(T) = \nu_p(\gcd(E, \frac{dE}{d\theta})) - \nu_p(\gcd(G, \frac{dG}{d\theta})) = \nu_p(\gcd(E, \frac{dE}{d\theta})) = \nu_p(E) - 1$$
$$\nu_p(Q) = \nu_p(T) + \nu_p(y) = \nu_p(E) - 1 + \nu_p(y) = -\nu_p(g) - 1 + \nu_p(g) + 1 = 0$$

   On the other hand if $\nu_p(f) < 0$, then $\nu_p(G) = \min(\nu_p(D), \nu_p(E)) = \nu_p(D)$ since $\nu_p(E) = -\nu_p(g) = -\nu_p(y' + fy) = -\nu_p(y) - \nu_p(f) = -\nu_p(y) + \nu_p(D) > \nu_p(D)$. In this case,
$$\nu_p(Q) = \nu_p(T) + \nu_p(y) = (\nu_p(E) - 1) - (\nu_p(G) - 1) + \nu_p(g) - \nu_p(f) = \\ = -\nu_p(g) + \nu_p(f) + \nu_p(g) - \nu_p(f) = 0$$

   We also want to show that T is the least possible denominator, or in other words, that $yT/p \notin \mathbb{F}\langle\theta\rangle$ for any normal p dividing T. We have already established that $\nu_p(yT) = 0$ when $\nu_p(y) < 0$ which would imply that $\nu_p(yT/p) = -1$. Now suppose that $\nu_p(y) \geq 0$. Since p divides T it must divide E too, so $\min(\nu_p(y'), \nu_p(fy)) = \nu_p(y' + fy) = \nu_p(g) = -\nu_p(E) < 0$. $\nu_p(y')$ cannot be negative if $\nu_p(y) \geq 0$, so it must be that $\nu_p(fy) = \nu_p(f) + \nu_p(y) = \nu_p(g)$. But we also know that $\nu_p(T) = (\nu_p(E) - 1) - (\nu_p(G) - 1) = \nu_p(E) - \min(\nu_p(D), \nu_p(E)) > 0$, so $-\nu_p(g) = \nu_p(E) > \nu_p(D) = -\nu_p(f)$. Adding the two expressions $\nu_p(f) + \nu_p(y) = \nu_p(g)$ and $-\nu_p(f) < -\nu_p(g)$ gives $\nu_p(y) < 0$ contradicting our assumption.

2. Recall that $Q = Ty$, $A = fD$ and $g = B/E$.
$$DTQ' + (AT - DT')Q = DT(T'y + Ty') + (fDT - DT')Ty = \\ = DT(Ty' + Tfy) = DT^2 g = \frac{BDT^2}{E}$$

Since $DTQ' + (AT - DT')Q \in \mathbb{F}\langle\theta\rangle$, it follows that $BDT^2/E \in \mathbb{F}\langle\theta\rangle$. If $\theta' = 1$ or θ is logarithmic, then $\mathbb{F}\langle\theta\rangle = \mathbb{F}[\theta]$ and $\gcd(B, E) = 1$, so E must divide DT^2. If θ is exponential, there is an integer n such that $\theta^n BDT^2 \in \mathbb{F}[\theta]$ and $\gcd(\theta, E) = \gcd(B, E) = 1$, so E must divide DT^2.

□

If $\theta' = 1$ or θ is logarithmic over $\mathbb{F}$, then $\mathbb{F}\langle\theta\rangle = \mathbb{F}[\theta]$, so all the coefficients as well as the solution $Q \in \mathbb{F}[\theta]$. If θ is exponential over $\mathbb{F}$, then $\mathbb{F}\langle\theta\rangle = \mathbb{F}[\theta, \theta^{-1}]$, so the solution Q is of the form q/θ^m for some $q \in \mathbb{F}[\theta]$. We have to find m and q. Notice that $DT \in \mathbb{F}[\theta]$. If $\theta^k \mid DT$, we can divide all three coefficients by θ^k, so from now on we assume that $\theta \nmid DT$.

Theorem 6.11 *Let $\theta = \exp(u)$ be exponential over $\mathbb{F}$ and let m be a positive integer such that $q = Q\theta^m \in \mathbb{F}[\theta]$. Then there are $a, b, c \in \mathbb{F}[\theta]$ such that*

$$aq' + bq = c$$

Proof Choose $a = DT$, $b = AT - DT' - mu'DT$, $c = \theta^m BDT^2/E$

$$aq' + bq = DT(mu'\theta^m Q + \theta^m Q') + (AT - DT' - mu'DT)\theta^m Q =$$
$$= \left(mu'DTQ + DTQ' + (AT - DT')Q - mu'DTQ\right)\theta^m = \frac{BDT^2}{E}\theta^m = c$$

The coefficients belong to $\mathbb{F}\langle\theta\rangle$, but can be put in $\mathbb{F}[\theta]$ by multiplying all of them by a suitable power of θ. □

While it is clear that there exists an m such that $q = Q\theta^m \in \mathbb{F}[\theta]$, we have yet to describe how to find it.

Lemma 6.12 *Let $\theta = \exp(u)$ be exponential over $\mathbb{F}$ and let k, l, m be the smallest possible natural numbers such that $q = \theta^m Q$, $a = DT$, $b = \theta^k(AT - DT')$ and $c = \theta^l BDT^2/E$ belong to $\mathbb{F}[\theta]$. Then $m = l - k$ unless both $k = 0$ and $\int \frac{b_0}{a_0} = mu - \log(f)$ for some $f \in \mathbb{F}$ and $m \in \mathbb{Z}^+$.*

Proof The equation $DTQ' + (AT - DT')Q = BDT^2/E$ implies

$$a(-m\theta^{-m-1}\theta' q + \theta^{-m}q') + b\theta^{-k}q\theta^{-m} = (-mau'q + aq')\theta^{-m} + bq\theta^{-m-k} = c\theta^{-l}$$

If $k > 0$, then θ^{-m-k} is the lowest exponent on the left hand side, so $m + k = l$. If $k = 0$, the left hand side is $(-mau'q + aq' + bq)\theta^{-m}$, so $m = l$ $(= l - k)$ unless the coefficient $-ma_0u'q_0 + a_0q_0' + b_0q_0 = 0$. This happens precisely if

$$\frac{b_0}{a_0} = mu' - \frac{q_0'}{q_0}$$

□

6.3 Degree bounds for the numerator

The previous section reduced the Risch differential equation in $\mathbb{F}(\theta)$ to the equation

$$aq' + bq = c$$

where a, b, c and $q \in \mathbb{F}[\theta]$. The next step is to find a bound for the exponents of θ that appear in q. To simplify the notation, $\deg(f)$ will be understood to mean $\deg_\theta(f)$ throughout this section. Similarly, $\mathrm{lc}(f)$ denotes the leading coefficient of f when viewed as a polynomial in θ.

6.3.1 The base case

Lemma 6.13 *Let θ be the variable x such that $x' = 1$.*

1. *If* $\deg(a) > \deg(b) + 1$, *then* $\deg(q) = \deg(c) - \deg(a) + 1$
2. *If* $\deg(a) < \deg(b) + 1$, *then* $\deg(q) = \deg(c) - \deg(b)$
3. *If* $\deg(a) = \deg(b) + 1$, *then* $\deg(q) = \deg(c) - \deg(b)$ *unless*

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = n$$

for some $n \in \mathbb{N}$, in which case $\deg(q)$ *could equal n.*

Proof

1. If $\deg(a) > \deg(b) + 1$, then $\deg(aq') > \deg(bq)$ so $\deg(c) = \deg(aq' + bq) = \deg(aq') = \deg(a) + \deg(q) - 1$.
2. If $\deg(a) < \deg(b) + 1$, then $\deg(aq') < \deg(bq)$ so $\deg(c) = \deg(aq' + bq) = \deg(bq) = \deg(b) + \deg(q)$.
3. If $\deg(a) = \deg(b) + 1$, then $\deg(aq') = \deg(bq)$ so unless the leading terms of aq' and bq cancel, we can conclude that $\deg(c) = \deg(aq' + bq) = \deg(b) + \deg(q)$. Let the leading term of q be $q_n x^n$. The leading terms cancel if and only if $\mathrm{lc}(a)nq_n + \mathrm{lc}(b)q_n = 0$, i.e.

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = n$$

□

6.3.2 Logarithmic extensions

Lemma 6.14 *Let $\theta = \log(u)$ be transcendental over $\mathbb{F}$ and moreover suppose that $\mathbb{F}(\theta)$ has no new constants.*

1. *If* $\deg(a) > \deg(b) + 1$, *then* $\deg(q) \leq \deg(c) - \deg(a) + 1$
2. *If* $\deg(a) < \deg(b)$, *then* $\deg(q) = \deg(c) - \deg(b)$
3. *If* $\deg(a) = \deg(b) + 1$, *then* $\deg(q) \leq \deg(c) - \deg(a) + 1$ *unless*

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = f' + n\theta'$$

   for some $f \in \mathbb{F}$ and $n \in \mathbb{N}$, in which case the $\deg(q)$ *could equal n.*

4. *If* $\deg(a) = \deg(b)$, *then* $\deg(q) = \deg(c) - \deg(b)$ *unless*

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = \frac{q_n'}{q_n} \quad \textit{and} \quad \frac{\mathrm{lc}(\mathrm{lc}(b)a - \mathrm{lc}(a)b)}{\mathrm{lc}(a)^2} = f' + n\theta'$$

   for some $q_n, f \in \mathbb{F}$ and $n \in \mathbb{N}$, in which case $\deg(q)$ *could equal n.*

Proof Recall 4.12, so $\deg(q) - 1 \leq \deg(q') \leq \deg(q)$

1. If $\deg(a) > \deg(b) + 1$, then $\deg(aq') > \deg(bq)$ so $\deg(c) = \deg(aq' + bq) = \deg(aq') = \deg(a) + \deg(q') \geq \deg(a) + \deg(q) - 1$.
2. If $\deg(a) < \deg(b)$, then $\deg(aq') < \deg(bq)$ so $\deg(c) = \deg(aq' + bq) = \deg(bq) = \deg(b) + \deg(q)$.
3. If the leading coefficient of q is non-constant, then $\deg(a) = \deg(b)+1$ implies $\deg(aq') > \deg(bq)$, so $\deg(c) = \deg(aq' + bq) = \deg(aq') = \deg(a) + \deg(q)$. Otherwise $\deg(aq') = \deg(bq)$ so unless the leading terms of aq' and bq cancel, we can conclude that $\deg(c) = \deg(aq' + bq) = \deg(aq') = \deg(a) + \deg(q) - 1$. Let the leading term of q be $q_n\theta^n$. The leading terms cancel if and only if $\mathrm{lc}(a)(q_{n-1}' + nq_n\theta') + \mathrm{lc}(b)q_n = 0$, i.e.

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = \frac{q_{n-1}' + nq_n\theta'}{q_n} = \frac{q_{n-1}'}{q_n} + n\theta' = \left(\frac{q_{n-1}}{q_n}\right)' + n\theta'$$

4. If the leading coefficient of q is a constant, then $\deg(a) = \deg(b)$ implies $\deg(aq') < \deg(bq)$, so $\deg(c) = \deg(aq' + bq) = \deg(bq) = \deg(a) + \deg(q)$. Otherwise $\deg(aq') = \deg(bq)$ so unless the leading terms of aq' and bq cancel, we can conclude that $\deg(aq' + bq) = \deg(a) + \deg(q)$. Let the leading term of q be $q_n\theta^n$. The leading terms cancel if and only if $\mathrm{lc}(a)q_n' + \mathrm{lc}(b)q_n = 0$, i.e.

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = \frac{q_n'}{q_n}$$

Let $p = q/q_n$. Then $aq' + bq = a(q_n p)' + bq_n p = aq_n p' + (aq_n' + bq_n)p = Ap' + Bp$ where $A = aq_n$ has the same degree as a and $B = aq_n' + bq_n$ has degree less than a since $\mathrm{lc}(a)q_n' + \mathrm{lc}(b)q_n = 0$. If $\deg(A) > \deg(B) + 1$, then $\deg(c) - \deg(A) \leq \deg(p) \leq \deg(c) - \deg(A) + 1$ according to case 1. Otherwise $\deg(A) = \deg(B) + 1$, so according to case 3, $\deg(c) - \deg(A) \leq \deg(p) \leq \deg(c) - \deg(A) + 1$ unless

$$-\frac{\mathrm{lc}(B)}{\mathrm{lc}(A)} = f' + nu'$$

But

$$-\frac{\mathrm{lc}(B)}{\mathrm{lc}(A)} = -\frac{\mathrm{lc}(aq_n' + bq_n)}{\mathrm{lc}(a)q_n} = -\,\mathrm{lc}\left(\frac{aq_n' + bq_n}{\mathrm{lc}(a)q_n}\right) = -\,\mathrm{lc}\left(-\frac{a}{\mathrm{lc}(a)}\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} + \frac{b}{\mathrm{lc}(a)}\right) =$$
$$= \mathrm{lc}\left(\frac{\mathrm{lc}(b)a - \mathrm{lc}(a)b}{\mathrm{lc}(a)^2}\right) = \frac{\mathrm{lc}(\mathrm{lc}(b)a - \mathrm{lc}(a)b)}{\mathrm{lc}(a)^2}$$

which proves the lemma.

□

6.3.3 Exponential extensions

Lemma 6.15 *Let $\theta = \exp(u)$ be transcendental over $\mathbb{F}$ and moreover suppose that $\mathbb{F}(\theta)$ has no new constants.*

1. *If $\deg(a) > \deg(b)$, then $\deg(q) = \deg(c) - \deg(a)$*
2. *If $\deg(a) < \deg(b)$, then $\deg(q) = \deg(c) - \deg(b)$*
3. *If $\deg(a) = \deg(b)$, then $\deg(q) = \deg(c) - \deg(b)$ unless*

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = \frac{q_n'}{q_n} + nu'$$

for some $q_n \in \mathbb{F}$ and $n \in \mathbb{N}$, in which case $\deg(q)$ could equal n.

Proof Recall 4.13, so $\deg(q') = \deg(q)$

1. If $\deg(a) > \deg(b)$, then $\deg(aq') > \deg(bq)$ so $\deg(c) = \deg(aq' + bq) = \deg(aq') = \deg(a) + \deg(q)$
2. If $\deg(a) < \deg(b)$, then $\deg(aq') < \deg(bq)$ so $\deg(c) = \deg(aq' + bq) = \deg(bq) = \deg(b) + \deg(q)$

3. If $\deg(a) = \deg(b)$, then $\deg(aq') = \deg(bq)$ so unless the leading terms of aq' and bq cancel, we can conclude that $\deg(c) = \deg(aq' + bq) = \deg(b) + \deg(q)$. Let the leading term of q be $q_n\theta^n$. The leading terms cancel if and only if $\mathrm{lc}(a)(q_n' + nq_nu') + \mathrm{lc}(b)q_n = 0$, i.e.

$$-\frac{\mathrm{lc}(b)}{\mathrm{lc}(a)} = \frac{q_n' + nq_nu'}{q_n} = \frac{q_n'}{q_n} + nu'$$

□

6.4 The SPDE algorithm

The next step is to reduce the equation $aq' + bq = c$ to one with $a = 1$. This is done by Rothstein's SPDE (Special Polynomial Differential Equation) algorithm.

Theorem 6.16 *Let $q \in \mathbb{F}[\theta]$ be a solution of $aq' + bq = c$ where $a, b, c \in \mathbb{F}[\theta]$, $\deg(a) > 0$ and $\gcd(a, b) = d$ where $\deg(d) > 0$. Then $d \mid c$ and*

$$aq' + bq = c \Longleftrightarrow \frac{a}{d}q' + \frac{b}{d}q = \frac{c}{d}$$

Proof Obvious. □

Theorem 6.17 *Let $q \in \mathbb{F}[\theta]$ be a solution of $aq' + bq = c$ where $a, b, c \in \mathbb{F}[\theta]$, $\deg(a) > 0$ and a relatively prime to b. Let n be an upper bound on the degree of q. Then there exists $s, t \in \mathbb{F}[\theta]$ such that $as + bt = c$ and*

$$aq' + bq = c \Longleftrightarrow ah' + (b + a')h = s - t'$$

where $h = (q - t)/a$, so $\deg(h) \leq n - \deg(a)$.

Proof The elements $s, t \in \mathbb{F}[\theta]$ such that $as + bt = c$ are given by the extended euclidean algorithm.

$$\begin{aligned} ah' + (b + a')h = a\left(\frac{q - t}{a}\right)' + (b + a')\frac{q - t}{a} = \\ = q' - t' - \frac{(q - t)a'}{a} + b\frac{q - t}{a} + a'\frac{(q - t)}{a} = \\ = q' + b\frac{q}{a} - t' - b\frac{t}{a} = \frac{c}{a} - t' - \frac{c - as}{a} = s - t' \end{aligned}$$

Clearly $\deg(h) \leq \deg(q) - \deg(a) \leq n - \deg(a)$. □

Theorem 6.16 will reduce the degree of a while 6.17 will transform the equation to a similar one but with a lower bound on the degree of q. The theorems can be applied repeatedly until either $\deg(a) = 0$ or the bound on $\deg(q)$ becomes impossible to fulfill (i.e. negative). If $\deg(a) = 0$, we can divide all terms by a to get a new equation $q' + \tilde{b}q = \tilde{c}$.

6.5 The final solution

What remains is to solve the equation

$$q' + bq = c$$

in $\mathbb{F}[\theta]$. The following tells us that it is sufficient to find the leading term since the remaining terms will satisfy a similar equation but have a lower bound on the degree.

Theorem 6.18 *Let the leading term of q be $q_n\theta^n$ and the remaining terms be $r = q - q_n\theta^n$. Then $\deg(r) < n$ and*

$$q' + bq = c \Longleftrightarrow r' + br = c - (q_n\theta^n)' - bq_n\theta^n$$

Proof Obvious □

Lemma 6.19 *If $\deg(b) > 0$, then $\deg(q) = \deg(c) - \deg(b)$ and $\operatorname{lc}(q) = \operatorname{lc}(c)/\operatorname{lc}(b)$.*

Proof Whether θ is the variable x, a transcendental logarithm or a transcendental exponential, $\deg(q') \leq \deg(q)$. Hence $\deg(c) = \deg(q' + bq) = \deg(bq) = \deg(b) + \deg(q)$ and $\operatorname{lc}(c) = \operatorname{lc}(q' + bq) = \operatorname{lc}(bq) = \operatorname{lc}(b)\operatorname{lc}(q)$. □

The lemma and theorem above will reduce the equation $q' + bq = c$ to one where $\deg(b) = 0$. If $b = 0$, then $q = \int c$ so suppose that $b \neq 0$ from here on.

Lemma 6.20 *Let $\mathbb{F}(\theta)$ be an elementary transcendental extension of $\mathbb{F}$ with the same field of constants. Let $q \in \mathbb{F}[\theta]$ be a solution to $q' + bq = c$ where $b \neq 0$, $\deg(b) = 0$.*

1. *If $\theta' = 1$, then $\deg(q) = \deg(c)$ and $\operatorname{lc}(q) = \operatorname{lc}(c)/b$.*
2. *If $\theta = \log(u)$, then either there exists $f \in \mathbb{F}[\theta]$ such that*

$$b = \frac{f'}{f} \quad \text{and} \quad q = \frac{\int fc}{f}$$

   or $\operatorname{lc}(q)$ satisfies the Risch differential equation $\operatorname{lc}(q)' + b\operatorname{lc}(q) = \operatorname{lc}(c)$.

3. *If $\theta = \exp(u)$, then either there exists $f \in \mathbb{F}[\theta]$ and $n \in \mathbb{N}$ such that*

$$b = \frac{f'}{f} + nu' \quad \text{and} \quad q = \frac{\int fc\theta^n}{f\theta^n}$$

   or $\operatorname{lc}(q)$ satisfies the Risch differential equation $\operatorname{lc}(q)' + (b + \deg(c)u')\operatorname{lc}(q) = \operatorname{lc}(c)$.

Proof

1. If $\theta' = 1$ and $\mathbb{F}$ is the field of constants, then $\deg(q') < \deg(q)$, so $\deg(c) = \deg(q' + bq) = \deg(bq) = \deg(q)$ and $\operatorname{lc}(c) = \operatorname{lc}(q' + bq) = \operatorname{lc}(bq) = b \operatorname{lc}(q)$.

2. If $\theta = \log(u)$ and $b = f'/f$ for some $f \in \mathbb{F}$, then

$$c = q' + bq = \frac{fq' + f'q}{f} \implies (fq)' = fc \implies q = \frac{\int fc}{f}$$

   Otherwise, $f' + bf \neq 0$ for all $f \in \mathbb{F}$. In particular $\operatorname{lc}(q)' + b \operatorname{lc}(q) \neq 0$, so the coefficient of $\theta^{\deg(q)}$ in $q' + bq$ does not vanish. Hence, $\deg(c) = \deg(q' + bq) = \deg(q)$ and $\operatorname{lc}(c) = \operatorname{lc}(q' + bq) = \operatorname{lc}(q)' + b \operatorname{lc}(q)$.

3. If $\theta = \exp(u)$ and $b = f'/f + nu'$ for some $f \in \mathbb{F}$ and $n \in \mathbb{N}$, then

$$c = q' + bq = \frac{fq' + nfu'q + f'q}{f} \implies (fq\theta^n)' = fc\theta^n \implies q = \frac{\int fc\theta^n}{f\theta^n}$$

   Otherwise, $f' + nu'f + bf \neq 0$ for all $f \in \mathbb{F}$. In particular $\operatorname{lc}(q)' + \deg(q)u' \operatorname{lc}(q) + b \operatorname{lc}(q) \neq 0$, so the coefficient of $\theta^{\deg(q)}$ in $q' + bq$ does not vanish. Hence, $\deg(c) = \deg(q' + bq) = \deg(q)$ and $\operatorname{lc}(c) = \operatorname{lc}(q' + bq) = \operatorname{lc}(q)' + \deg(q)u' \operatorname{lc}(q) + b \operatorname{lc}(q)$.

□

By the induction assumption, we can integrate elements in $\mathbb{F}[\theta]$ and solve the Risch differential equation in $\mathbb{F}$, so the lemma above really gives a solution in $\mathbb{F}[\theta]$ to $q' + bq =$ when $\deg(b) = 0$, $b \neq 0$. Since the earlier theorems in this chapter showed that any equation $y' + fy = g$ in $\mathbb{F}(\theta)$ can be reduced to this form, we can solve the Risch differential equation in $\mathbb{F}(\theta)$ too.

Chapter 7

Risch-Norman's parallel algorithm

As we saw in the previous chapters, it is possible to decide whether a transcendental elementary function has an elementary integral, but the algorithm is quite complicated. In 1976, Risch and Norman suggested an alternative method which is simpler than the full Risch decision procedure. The idea is to make a guess of the structure of the integral using Liouville's theorem. Once we have made a guess with some undetermined coefficients, we differentiate it and solve the system of equations for the coefficients. Because it handles all field extensions in parallel rather than recursively, as was the case with the original Risch algorithm, the Risch-Norman algorithm is also known as the parallel Risch algorithm. Although Risch-Norman's method can fail to find an elementary integral even when one exists, it is popular because it is relatively simple to implement and performs well in practice.

7.1 Preliminaries

Recall definition 6.1 of normal and special[1] polynomials which we will now modify to allow more general differential rings which are not necessarily closed under the derivation, but whose field of fractions is.

Definition 7.1 *A polynomial p in a differential ring $\mathbb{F}[\theta_1, \theta_2, \ldots, \theta_n]$ is called normal if* $\gcd(p, d\partial p) = 1$ *and special if* $\gcd(p, d\partial p) = p$*, where d is the least common multiple of the denominators of all θ_i'.*

Lemma 7.2

1. *Every product of special polynomials is special.*

2. *Every factor of a special polynomial is special.*

[1] What we call a special polynomial is also known as a Darboux polynomial after the french mathematician Jean-Gaston Darboux (1842-1917).

3. *Every product of relatively prime normal polynomials is normal.*

4. *Every factor of a normal polynomial is normal.*

Proof

1. Let p and q be special polynomials. Then pq is special since

$$\gcd(pq, d\partial pq) = \gcd(pq, d(p\partial q + q\partial p)) = \gcd\left(pq, pq\left(\frac{d\partial q}{q} + \frac{d\partial p}{p}\right)\right) = pq$$

2. Let $r = p^n q$ be special and p^n the largest power of p dividing r. Since p^n divides $r = \gcd(r, d\partial r) = \gcd(p^n q, p^n d\partial q + np^{n-1} q d\partial p)$, p must divide ∂p. This shows that p is special.

3. Let p and q be normal, relatively prime polynomials and let r be an irreducible factor of $\gcd(pq, d\partial pq)$. Without loss of generality, we assume that r divides p. Since it also divides $pd\partial q + qd\partial p$, it must divide $d\partial p$ contradicting the assumption that p is normal.

4. Let $r = pq$ be normal and p an irreducible factor. Since p is irreducible it must be either special or normal. Suppose that p is special. Then

$$\gcd(r, d\partial r) = \gcd(pq, dp\partial q + dq\partial p) = p \gcd\left(q, d\partial q + d\frac{d\partial p}{p}\right)$$

   contradicting the assumption that r is normal. Thus every irreducible factor must be normal and, since it is easy to see that the factors of a normal polynomial must be relatively prime, the previous case tells us that all factors are normal.

□

Lemma 7.3 *We can compute a factorization $p = p_s p_n$ of p such that p_s is special and every square-free factor of p_n is normal using only computations of derivatives and (multivariate) greatest common divisors. The factors p_s and p_n are sometimes called the special and normal part of p respectively.*

Proof If p is irreducible it must be either normal or special, and we can determine which it is with a differentiation and a gcd computation. In particular, all polynomials of total degree 1 are irreducible.

Suppose that the lemma is true for all polynomials of total degree less than that of p. If the polynomial p is neither normal nor special, then $\gcd(p, d\partial p)$ is a proper, non-trivial divisor of p. By induction, both $q = \gcd(p, d\partial p)$ and $r = p/\gcd(p, d\partial p)$ can be factored into a normal and a special part. We can now find the normal and special parts of p by using lemma 7.2 which shows that $p_s = q_s r_s$ and $p_n = q_n r_n$. □

There are more efficient methods for computing this factorization.

7.2 The algorithm

We begin with a structure theorem due to Bronstein [4, 5].

Theorem 7.4 *(Bronstein) Let $f, g \in \mathbb{F}[\theta_1, \theta_2, \ldots, \theta_n]$ be relatively prime, where $\mathbb{F}$ is the field of constants and each θ_i is transcendental over $\mathbb{F}[\theta_1, \theta_2, \ldots, \theta_{i-1}]$. Furthermore, suppose that $g = g_s g_n$ where g_s is special and every square-free factor of g_n is normal. Then*

$$\int \frac{f}{g} = \frac{a}{s \prod d_j^{j-1}} + \sum_{i=1}^{k_p} \alpha_i \log p_i + \sum_{i=1}^{k_s} \beta_i \log s_i$$

Proof If the integral of f/g is in an elementary extension of $\mathbb{F}$, then Liouville's theorem tells us that

$$\frac{f}{g} = v_0' + \sum_{i=1}^{k} c_i \frac{v_i'}{v_i}$$

for some $v_0 \in \mathbb{F}(\theta_1, \theta_2, \ldots, \theta_n)$, $v_i \in \bar{\mathbb{F}}[\theta_1, \theta_2, \ldots, \theta_n]$ and $c_i \in \bar{\mathbb{F}}$. Without loss of generality, we can assume that the $v_1, \ldots, v_k$ are irreducible and pairwise relatively prime polynomials.

Suppose that $v_0 = \frac{a}{bp^m}$ where p^m is the largest power of a normal irreducible p dividing the denominator. Let δ be the highest power of p dividing $\prod_{i=1}^{k} v_i$. Since the v_i are square-free and relatively prime, δ can be at most 1. Then

$$\frac{f}{g} = \frac{bp\partial a - ap\partial b - mab\partial p}{b^2 p^{m+1}} + \sum_{i=1}^{k} c_i \frac{\partial v_i}{v_i}$$

Multiplication by the denominators of all terms and the least common multiple d of the denominators of all θ_i, gives a polynomial equation

$$fdb^2 p^{m+1} \prod_{j=1}^{k} v_j = \left((bpd\partial a - apd\partial b - mabd\partial p) \prod_{j=1}^{k} v_j + b^2 p^{m+1} \sum_{i=1}^{k} c_i d \partial v_i \prod_{i \neq j} v_j \right) g$$

where $p^{m+1+\delta}$ divides the left hand side. It is clear that p^δ divides the parenthesis on the right hand side and in fact this is the highest possible power since p divides all terms except $mabd\partial p$. Hence p^{m+1} must divide g, so every normal factor in the denominator of v_0 appear in $\prod d_j^{j-1}$. Thus

$$v_0 = \frac{a}{s \prod d_j^{j-1}}$$

where s is special.

Since every $v_i, 1 \leq i \leq k$ is irreducible it is either normal or special. Let $s_1, \ldots, s_{k_s}$ denote the special elements and $p_1, \ldots, p_{k_p}$ the normal ones. To show that each normal v_i is a factor of g, we notice that v_i divides the left hand side

of the equation above, so unless it divides g, it must divide the parenthesis on the right hand side. Clearly, v_i divides all terms except $c_i d\partial v_i \prod_{j\neq i} v_j$ and to divide this term it would have to divide $d\partial v_i$ contradicting the assumption that v_i is normal. Thus every p_i divides g and, since they are normal, also g_n. □

The previous theorem gives the general structure of the integral, but leaves the special polynomials $s, s_1, \ldots, s_{k_s}$, the coefficients $\alpha_1, \ldots, \alpha_{k_p}, \beta_1, \ldots, \beta_{k_s}$ and the polynomial a undetermined. To create a practical integration method, one can make educated guesses for the polynomials $s, s_1, \ldots, s_{k_s}$ and the degree of each variable in a, and then solve a linear system to obtain $\alpha_1, \ldots, \alpha_{k_p}, \beta_1, \ldots, \beta_{k_s}$ and the coefficients in a.

It seems reasonable to choose $s = g_s$ since this contains all the special factors that appear in the denominator of the integrand. This choice also avoids factoring g_s.

For the types of differential extensions we are considering here, there are only finitely many irreducible special polynomials (cf. [5]), so we could take $s_1, \ldots, s_{k_s}$ equal to those polynomials, if they are known. If $\mathbb{F}[\theta_1, \theta_2, \ldots, \theta_n]$ is a tower of transcendental logarithmic and exponential extensions, the special elements are precisely the exponential θ_i together with the irreducible factors of d, the least common multiple of the denominators of $\theta_1', \theta_2', \ldots, \theta_n'$. In more general fields where the full set of special polynomials is unknown, one can take the s_i to be the special factors of g and d.

Finally, we bound the degree of a. The most common choice in the literature [4, 14] seems to be

$$\deg_{\theta_i}(a) \leq \max(\deg_{\theta_i}(f), \deg_{\theta_i}(g)) - 1 + \deg_{\theta_i}(d\partial\theta_i)$$

although other guesses have been used as well.

Now that we have guessed the special polynomials and bounded the degree of a, it is relatively straightforward to solve the linear system for $\alpha_1, \ldots, \alpha_{k_p}, \beta_1, \ldots, \beta_{k_s}$ and the coefficients in a. One should however remember that if this system lack solutions, it may be either because the integral is non-elementary or that one of the guesses was wrong.

Chapter 8

Implementation of an algebra system

Unlike Risch's algorithm, the heuristic rules (cf. chapter 2) and Risch-Norman's parallel algorithm (cf. chapter 7) can fail to find an elementary integral even if one exists. While Risch's algorithm was a very important theoretical discovery and the certainty it gives can be useful, it is not clear how often the other algorithms fail. It was therefore desired to implement the algorithms and compare how well they perform in practice. Due to time constraints, Risch's and Risch-Norman's algorithms were not fully implemented although many of the prerequisites were developed.

8.1 Existing systems

All of the integration algorithms require a number of simpler algorithms such as arithmetic of polynomials and rational functions, greatest common divisors and resultants. The quickest way to implement integration would of course be to let an existing computer algebra system (CAS) supply the basic functionality. Since symbolic integration, particularly when modified to print hints on how to obtain the solution, would primarily be useful as an instruction aid when learning integration, the following characteristics were identified as the most important.

- The CAS needs to supply functions for arithmetic of polynomials and rational functions in several variables, but must also be capable to represent and perform simple calculations with arbitrary compositions of elementary functions.
- The CAS should be available free of charge on all larger operating systems.
- The CAS should be intuitive and easy to use.
- The CAS should be reasonably small, and use system resources efficiently.

A small survey of the existing systems was made with the intention of finding a suitable system to use.

The commercial computer algebra systems (e.g. Mathematica and Maple) are well tested and have the necessary prerequisites but are neither free nor small. On the other hand, there are many free computer algebra systems, but several of them are special purpose systems which lack features that are needed to support integration. Of the systems that aspire to be general purpose computer algebra systems, most of the smaller are not sufficiently developed to be useful. Axiom is probably the best known among the larger free computer algebra systems. Axiom, however, is intended for writing algorithms for general algebraic structures and thus not very intuitive or easy to use. Sage is another free system, written in Python but with interfaces to a number of other computer algebra systems and C/C++ libraries. Because of its interfaces to other systems, it has many features, but it also becomes very large and difficult to understand. Moreover, the documentation is not satisfactory and there does not seem to be any general policy concerning which library is used for which task.[1]

Yacas is a small computer algebra system, written in C++. It has a Mathematica-like syntax, many of the basic functions required and easily accessible documentation. Initially, the Yacas system seemed promising and parts of the integration routines, including parts of the rule-based integration as well as Horowitz-Ostrogradsky's algorithm[2], were written for it. However, a number of problems in Yacas surfaced during the implementation of some of the more advanced rules and algorithms. In particular, some functions could only handle univariate polynomials, some natural programming constructs did not exist or were undocumented, the automatic simplification was unsatisfactory and developer response was very slow.

Rather than trying to patch internal code in an unfamiliar computer algebra system or potentially wasting more time by testing other systems, it was decided to develop the prerequisites directly in C++. This approach has the advantage that one can implement precisely the features one needs without having to worry about the risk of unforeseen language limitations arising later in the development. Having direct access to the internal representation can also be an advantage in some of the algorithms. The disadvantage is that one will have to write and test more code, some of which is rather peripheral to the original problem of integration.

8.2 Representation of expressions

The first design decision one has to make in a computer algebra system is how to represent expressions.

[1] The situation has improved somewhat since the beginning of the project, as some of the smaller projects have matured. The native features and documentation for Sage has also improved, but it still lacks some needed functionality e.g. pattern matching.

[2] Horowitz-Ostrogradsky's algorithm, like Hermite's, finds the rational part of the integral of a rational function. While Hermite's method was to reduce the degree of the denominator of the integrand until it is square free, Horowitz-Ostrogradsky's constructs a system of linear equations for the coefficients in the rational part.

Even for something as simple as a polynomial, which is basically a list of coefficients, there are several possibilities at several layers of abstraction. At the lowest level, there is the choice of whether to represent the list as a linked list or as a contiguous array and whether to use a sparse or dense representation, i.e. if one should explicitly write out all zero coefficients. If one chooses the sparse representation, which is the more common in computer algebra, the list must contain coefficient-exponent pairs, not only coefficients. At a higher level, there is a choice of whether to insist on a particular canonical or normal form. In some problems it can be useful to represent a polynomial as a product of factors rather than a sum of monomials, in which case the representation must be modified accordingly. For multivariate polynomials, there is the choice of whether to use a recursive or a distributed representation, and so on. Although this list of choices is far from complete, it illustrates that representation is non-trivial and that the choices made will affect many subsequent algorithms.

The choice fell on the representation described by Maeder [19] and Wolfram [27] which is used in some algebra systems, Mathematica being the best known. This representation uses different data types for atomic data such as integers, rational numbers, floating-point numbers, strings and symbols. Non-atomic data is represented as an array of pointers where the first element points to the name of the function or type of the expression, and each following element points to an argument or part of the data. Representations in systems based on Lisp are often similar. The greatest asset of this representation is that it is very general, allowing the same structure to store both defined and undefined functions, lists and other types of data. For the same reason, functions for manipulating expressions can do so regardless of whether a particular expression is a list, a function or some user-defined type. The disadvantage is some overhead in both time and memory to store and maintain the type. The code for polynomial operations also becomes somewhat longer compared to what it would have been if one used a representation designed for polynomials.

Rather than using an array of regular pointers, each pointing to its own data, one should use some form of smart pointers which allow parts to be shared between several expressions. Consider the following product and its derivative.

$$f(x) = \prod_{i=1}^{n} f_i(x)$$

$$f'(x) = \sum_{j=1}^{n} \left(f_j'(x) \prod_{i \neq j} f_i(x) \right)$$

Here the derivative contain $n-1$ copies of each factor. Copying each factor $n-1$ times would waste space and slow down the differentiation, while sharing the representation would just require creating $n-1$ pointers to each factor. The smart pointers can also double as a reference counted garbage collector. (This was also described in [19].) The generic argument against reference counting for garbage collection is that it will not detect or collect cycles, potentially causing a memory

leak. This is not a concern in the case of mathematical expressions as they must be acyclic for other reasons, like being possible to evaluate.

8.3 Automatic simplification

Automatic simplification is defined as the collection of transformations that is applied to every part of the input during the evaluation process [8].

Ideally, one would like to have a canonical representation for each expression. However, Richardson [23] has shown that in a sufficiently rich class of functions it is undecidable whether two expressions are mathematically equal. This implies that one cannot *in general* find a canonical form for a symbolic expression. Even in classes in which every expression has a canonical or normal form, it can be rather expensive to actually find it.

There are different opinions concerning the purpose and scope of automatic simplification and Moses [21] quite humorously classified computer algebra systems as radical, new left, liberal, catholic or conservative depending on their approach. In his vocabulary, the radical systems are those that insist on having a canonical form for all expressions. Because of Richardson's theorem, this approach will only work for certain classes of expressions, but it is popular in systems focused on handling polynomials or rational functions. The liberal systems tries many less time-consuming simplifications, but without any guarantee of producing a normal or canonical form. A typical example would be to obtain all the usual simplifications of polynomials except expansion of powers and products. The conservative systems fully recognizes that "simple" is a subjective property, strongly dependent on the context. Because of this, the conservative system does very little automatic simplification, instead supplying the user with functions for building their own simplification algorithm. The new left and catholic systems can be thought of as variations or hybrids between the other.

The automatic simplification algorithm we implemented is similar to the one described by Cohen [8] and the one used in Mathematica [27]. To reduce the number of different representations of mathematically equal expressions, it is important to realize that the simplified form need not contain differences or quotients, since they can be transformed to sums and products by the rules

$$u - v = u + (-1) * v$$
$$u/v = u * v^{-1}$$

Obviously the rules above are only useful if we have ways to simplify sums, products and powers to automatically simplified forms.

The definition of an automatically simplified form will require an ordering of the expressions. While any total order would do, it is common to choose one which orders the terms of a polynomial in a natural way. This usually means placing numbers before other expressions and ordering powers by their exponent. Once we have an order, we can define automatically simplified forms for the arithmetic

1. Atomic types such as numbers and unbound symbols are considered as automatically simplified.
2. A sum with two or more automatically simplified operands is considered automatically simplified if:
   a) No operand is zero and at most one operand is an explicit number.
   b) No operand of a sum is itself a sum
   c) No two operands are numerical multiples of the same expression
   d) All operands are sorted in the standard order
3. A product with two or more automatically simplified operands is considered automatically simplified if:
   a) No operand is zero or one, and at most one operand is an explicit number.
   b) No operand of a product is itself a product
   c) No two operands are powers of the same base
   d) All operands are sorted in the standard order
4. A power with an integer exponent and an automatically simplified base is considered automatically simplified if:
   a) The exponent is not zero or one
   b) The base is neither an explicit number, a product or a power

It should be obvious that the conditions above can be satisfied by systematic use of associativity, commutativity, distributivity and the power laws. There are of course many other simplification rules in a computer algebra system, in particular related to transcendental functions and radicals, but as they are of less importance when it comes to heuristic integration, we will not discuss them further.

In general, our implementation attempts to convert mathematically equivalent expressions to the same form by choosing relatively simple transformations only depending on the local context. As there is no guarantee that this approach will produce a normal or canonical form, the system would be classified as liberal in Moses' terminology.

8.4 General implementation suggestions

A significant part of the implementation time was not spent on coding, but on two other time-consuming tasks; finding the right algorithm to implement and debugging the code. This section will discuss some well-known programming techniques that were used to reduce the amount of time spent on these unproductive tasks. Since the results were favorable, the techniques may be of interest to others who think about writing their own computer algebra system.

8.4.1 Automatic memory management

Computer algebra systems generate a large amounts of temporary data which must be reclaimed when it is no longer used. This suggests some form of garbage collection to keep track of the used data. We chose to use reference counting, handled by a class of smart pointers [19]. This approach worked very well and only a single bug related to memory management occurred during implementation. (The bug was caused by improper use of the smart pointers and quickly resolved.) Considering the number of allocations and deallocations in the program, this must be considered very successful. Apart from managing memory, the smart pointers allow data to be shared among several expressions leading to substantial reductions in running time for some operations.

8.4.2 Algorithm selection

Most of the code in a computer algebra system is concerned with algorithmically non-trivial problems where finding and deciding on a solution is difficult. It is easy to reject simple algorithms in pursuit of asymptotically faster, or otherwise superior, methods. For reasons listed below, it is often better to implement the simplest realistic solution first, even if one believes that a faster method will be needed.

- A simple method can usually be implemented much faster than a more advanced algorithm. This means that if a more advanced algorithm becomes needed, the extra coding time of having written the simple one will be small in comparison.
- A simple method will normally have less overhead than an advanced method, leading to faster execution for small inputs. In addition, the simpler method provides a baseline of performance which the more advanced method must surpass in order to be useful. (Once several algorithms are implemented, one can automatically choose the fastest one depending on the user input.)
- One can often prove the correctness of a simple method just by inspecting the code. When a more advanced method becomes available it can be verified by comparing the results against the correct results from the simpler code. See also the following section on testing.

Our opinion is that the choice of algorithms and their implementations should in general favor correctness, generality, clarity and low complexity of the code, usually in that order.

8.4.3 Regression testing

Developing a computer algebra system is a relatively large project and should be treated as such.

Even relatively simple functions such as the greatest common divisor can involve several thousand lines of code, spread across many functions. This makes it difficult to trace an error from the point where it is detected back to its source. In order to discover errors as early as possible, one should test each function separately before using it in other functions. The tests should be rerun after every major change in the system to verify that the change does not break existing code.

Fortunately, it is relatively easy to develop tests for computer algebra systems, since most functions can be executed directly by the user. Thus, one can collect tests into script files and interpret them with the computer algebra system rather than using a separate framework to test the code at the C/C++ level. This has the additional advantage that the tests are executed in the same setting as all normal user code.

When creating tests, one should attempt to design them to execute every control path in the function. In addition, one should pay extra attention to trivial cases, boundary cases and invalid inputs. Since it is tedious to write tests and difficult to know when all cases have been tested, one might want to generate tests automatically. Automatic generation of tests makes it possible to test more and larger instances than would be convenient to create manually. There are several methods for generating test that could be useful depending on the problem.

- Some mathematical operations can be uniquely defined by the properties of the result. These operations are often easy to test by verifying that the output has all the required properties for a large number of random inputs. For example, euclidean division of polynomials over a field should produce q, r such that $a = qb + r$ and $\deg r < \deg b$.

- If there are several ways to compute a given function, one can verify that they all give the same answer. It is quite unlikely that the same bug will exist in widely different algorithms. Polynomial greatest common divisors, for example, can be computed in a variety of ways including the usual euclidean algorithm, the primitive, reduced and subresultant polynomial remainder sequences and the different modular algorithms.

- There are sometimes strong relationships between different functions even though the functions do not share much code. If the relationship can be verified efficiently, it may be a good test of both functions. A problem with this type of testing is that it may not be possible to verify the relationship. For example, the fundamental theorem of calculus suggests checking that

$$\left(\int f(x)dx \right)' = f(x)$$

  While this should be mathematically true, the two sides may be expressed in different ways thus reducing to a difficult equivalence problem.

- One can sometimes generate inputs in such a way that the correct output is known. A typical example of a problem where this can be useful is polynomial factorization, as one can easily generate a product of known irreducible factors and verify that the factorization contain precisely those factors. One should however notice that this approach may require some care in order to avoid generating inputs of a very specific structure or form.

Although automatically generated tests discover most of the bugs efficiently by testing a large number of inputs, it does not detect all types of errors as randomly generated inputs seldom trigger special cases in the code. Thus, automatically generated tests should not be treated as an alternative to manual testing, but rather as a complement.

8.4.4 Programming by contract

The previous sections discussed how to generate test cases that are likely to reveal defects in the program. Another method is the "programming by contract" paradigm which helps pinpointing the cause of the bug by specifying contracts between the caller and the callee. The caller guarantees that certain preconditions will be satisfied upon entry to a function and assuming the preconditions, the function will ensure that certain postconditions are satisfied when it returns. If the conditions are correctly written and checked, they will tell whether it is the caller or the callee who has broken the contract in case of an error.

Apart from this use to locate an error, having explicit pre- and postconditions might also prevent some types of errors from entering the code in the first place. Consider as an example the automatic simplification introduced in section 8.3 by giving an explicit list of properties that should be satisfied. Without a clear specification of the goal, different functions could easily get slightly different notions about what the simplified form should be. This leads to a type of bug which is difficult to correct since it may be unclear which function made the invalid assumption.

Unfortunately, it is not always easy to write down a set of conditions that is reasonable to check and the ones that can be checked may not be sufficient to guarantee the correctness of the result. Thus, the effectiveness of the programming by contract depends on the function and the type of bugs one is trying to avoid.

Chapter 9

Results

The computer algebra system we implemented has a syntax similar to Mathematica's, the most noticeable difference being that our functions has parenthesis rather than brackets around the arguments. Like Mathematica, the system has a rule-based programming language which relies on pattern matching of expressions. We chose to use the type of pattern matching described by McIsaac [20]. The entire implementation consists of sightly less than 20 000 lines of code.

Once automatic simplification, pattern matching and functions for handling polynomials and rational functions were written, the code for heuristic integration was fairly compact. The C-code for integration of polynomials and rational functions consists of approximately 200 lines of code. The implementation of the integration rules described in chapter 2 uses about 250 lines of code in the interpreted, high-level language designed specifically for this type of transformation rules.

To evaluate how well the integration rules perform on typical calculus problems, problems were selected from exams and tests given in the courses 5B1104/SF1600 and 5B1106/SF1602 between 2006 and 2008 at the Royal Institute of Technology. Since the integral is not always explicit in the question, we have chosen only the problems in which integration is a significant part of the solution. While this selection is somewhat subjective, an effort has been made to create a fair and unbiased evaluation set. The results were, as expected, good but not perfect; a straightforward implementation of the rules in chapter 2 solved slightly more than 80% of the attempted integrals.

9.1 Some simple examples

We will now give some examples of simple integrals that can be computed with the implemented rules and the corresponding results. As one can notice in some of the outputs below, the integration procedure perform only automatic simplification of the integrals. Further simplification would in some cases generate more beautiful results.

```
In(1) := HeuristicIntegrate(1-4*x+6*x^5, x)
Out(1) = x-2*x^2+x^6

In(2) := HeuristicIntegrate(1/(x^2+4*x+3), x)
Out(2) = 1/2*Log(1-1/2*(4+2*x))-1/2*Log(1+1/2*(4+2*x))

In(3) := HeuristicIntegrate(x*Exp(x^2), x)
Out(3) = 1/2*Exp(x^2)

In(4) := HeuristicIntegrate((1+x^2)*Sin(x), x)
Out(4) = 2*(Cos(x)+Sin(x)*x)-Cos(x)*(1+x^2)

In(5) := HeuristicIntegrate(x*Log(x), x)
Out(5) = -1/4*x^2+1/2*Log(x)*x^2

In(6) := HeuristicIntegrate((Exp(x)+2*Exp(x)^2)/(Exp(x)+1), x)
Out(6) = 2*Exp(x)-Log(1+Exp(x))

In(7) := HeuristicIntegrate(Sqrt(2*x-x^2), x)
Out(7) = 1/2*ArcSin(-1+x)+1/4*Sin(2*ArcSin(-1+x))
```

The integrals can be expressed in many different ways and it is not clear which form is the simplest. For example, Mathematica obtains the following expression for the last integral

$$\frac{\sqrt{-(-2+x)x}\left(\sqrt{-2+x}(-1+x)\sqrt{x}-2\log\left(\sqrt{-2+x}+\sqrt{x}\right)\right)}{2\sqrt{-2+x}\sqrt{x}}$$

9.2 Generating hints

It is not difficult to modify the integration procedure to print hints on how to compute the integral. In fact, this modification can even simplify debugging by indicating which rules are being used.

Integration by parts is effective for integrating the product of a polynomial and an elementary function. The system will use rules repeatedly until the integral is computed, or no more rules are found.

```
In(1) := HeuristicIntegrate((1+x^2)*Sin(x), x, Solution->Hint)
Use integration by parts (integrate Sin(x) and differentiate 1+x^2)
Out(1) = 2*(Cos(x)+Sin(x)*x)-Cos(x)*(1+x^2)
```

Usually one would differentiate the polynomial and integrate the elementary function, but sometimes it is better to integrate the polynomial.

```
In(2) := HeuristicIntegrate(x*Log(x), x, Solution->Hint)
```

```
Use integration by parts (integrate x and differentiate Log(x))
Out(2) = -1/4*x^2+1/2*Log(x)*x^2
```

The following example demonstrates how substitutions can simplify relatively complicated integrals. When a substitution like $f(x) \to t$ is made, the new variable t must not be defined previously. To avoid name conflicts, temporary variable names receive a postfix $\$n$ where n is chosen such that the resulting name is unused.

```
In(3) := HeuristicIntegrate((Exp(x)+2*Exp(x)^2)/(Exp(x)+1), x,
...                                                  Solution->Hint)
Substitute Exp(x)->t$6 to get a simpler integrand
(t$6+2*t$6^2)/(t$6*(1+t$6))
Out(3) = 2*Exp(x)-Log(1+Exp(x))
```

Although the integration process is designed to find the same solution as a human would, it will in some cases find a different one. For example, most humans would substitute x^2 for a new variable when computing $\int xe^{x^2}dx$, but the program substitutes the entire expression e^{x^2} since the resulting integral $\int 1/2dt$ is simpler.

```
In(4) := HeuristicIntegrate(x*Exp(x^2), x, Solution->Hint)
Substitute Exp(x^2)->t$9 to get a simpler integrand 1/2
Out(4) = 1/2*Exp(x^2)

In(5) := HeuristicIntegrate(Sqrt(2*x-x^2), x, Solution->Hint)
Substitute -1+x->t$12 to complete the square
Out(5) = 1/2*ArcSin(-1+x)+1/4*Sin(2*ArcSin(-1+x))
```

Performing the substitution suggested above results in a new integral $\int \sqrt{1-x^2}dx$. The heuristic integration procedure can of course be applied repeatedly to generate more hints. In this case, it demonstrates the use of trigonometric substitutions to solve certain algebraic integrals.

```
In(6) := HeuristicIntegrate(Sqrt(1-x^2), x, Solution->Hint)
Substitute x->Sin(t$20) to obtain a trigonometric integrand
Out(6) = 1/2*ArcSin(x)+1/4*Sin(2*ArcSin(x))
```

9.3 Generating complete solutions

Instead of repeatedly generating hints, it is possible to generate complete textbook solutions. In the following examples, the solution is typeset using the actual [1] LaTeX-code produced by the algebra system.

Compared to just printing hints, the code that achieves this is much more involved. This modification almost doubles the size of the integration rules, even without counting the code for converting an expression to LaTeX.

[1] Line breaks have been added to make the solution fit the page layout.

```
In(1) := HeuristicIntegrate((1+x^2)*Sin(x), x, Solution->TeXForm)
```

$$\int \sin(x)\left(1+x^2\right)dx =$$

$$\left\{\text{Use integration by parts (integrate } \sin(x) \text{ and differentiate } 1+x^2\right)\} =$$

$$\int 2\cos(x)\,xdx - \cos(x)\left(1+x^2\right) =$$

$$\{\text{Use integration by parts (integrate } \cos(x) \text{ and differentiate } x)\} =$$

$$2\left(-\int \sin(x)\,dx + \sin(x)\,x\right) - \cos(x)\left(1+x^2\right) =$$

$$2\left(\cos(x) + \sin(x)\,x\right) - \cos(x)\left(1+x^2\right)$$

```
Out(1) = 2*(Cos(x)+Sin(x)*x)-Cos(x)*(1+x^2)
```

```
In(2) := HeuristicIntegrate(x*Log(x), x, Solution->TeXForm)
```

$$\int \log(x)\,xdx =$$

$$\{\text{Use integration by parts (integrate } x \text{ and differentiate } \log(x)\,)\} =$$

$$-\int \frac{1}{2}xdx + \frac{1}{2}\log(x)\,x^2 = -\frac{1}{4}x^2 + \frac{1}{2}\log(x)\,x^2$$

```
Out(2) = -1/4*x^2+1/2*Log(x)*x^2
```

```
In(3) := HeuristicIntegrate((Exp(x)+2*Exp(x)^2)/(Exp(x)+1), x,
...                                                  Solution->TeXForm)
```

$$\int \frac{\left(\exp(x) + 2\exp(x)^2\right)}{(1+\exp(x))}dx = \{\text{Substitute } \exp(x)\text{->t\$6}\} =$$

$$\int \frac{\left(\text{t\$6} + 2\text{t\$6}^2\right)}{\text{t\$6}\,(1+\text{t\$6})}d\text{t\$6} = -\log(1+\text{t\$6}) + 2\text{t\$6} =$$

$$\{\text{Substitute t\$6-> } \exp(x)\} = 2\exp(x) - \log(1+\exp(x))$$

```
Out(3) = 2*Exp(x)-Log(1+Exp(x))
```

```
In(4) := HeuristicIntegrate(x*Exp(x^2), x, Solution->TeXForm)
```

$$\int \exp\left(x^2\right)xdx = \left\{\text{Substitute } \exp\left(x^2\right)\text{->t\$9}\right\} =$$

$$\int \frac{1}{2}d\text{t\$9} = \frac{1}{2}\text{t\$9} = \left\{\text{Substitute t\$9-> } \exp\left(x^2\right)\right\} =$$

$$\frac{1}{2}\exp\left(x^2\right)$$

```
Out(4) = 1/2*Exp(x^2)
```

9.4 Integrals which remain unevaluated

As mentioned several times before, the integration rules does not solve every integral. We will now take a closer look at some of the integrals which fail, and how the integration rules could be enhanced.

Problem The integral

$$\int x^2 \log(1+x^2)dx$$

is not evaluated.

Solution The natural way to solve this integral is to use integration by parts since integrating x^2 and differentiating $\log(1+x^2)$ will reduce the problem to integrating a rational function. This is a straightforward generalization of the rule for integrating $P(x)\log(x)$, but it has not been implemented in order to make a fair comparison of the success rate. (Tweaking the rules to fit a finite evaluation set would of course make the results meaningless as one could in principle add one rule for each problem).

Problem The integral

$$\int \frac{1}{\sqrt[3]{x}+\sqrt{x}}dx$$

is not evaluated.

Solution The problem is caused by the fact that there are no rules for handling multiple algebraic extensions. In this case, it is easy to see that both radicals can be expressed in terms of $\sqrt[6]{x}$. A similar property holds in general since every finite separable extension has a primitive element. As every extension of a field of characteristic 0 is separable, this means that every algebraic extensions we are considering can be obtained by adjoining a single element to the base field.

Problem The integral

$$\int \sqrt{1+\left(\frac{d}{dx}\log(1-x^2)\right)^2}dx = \int \sqrt{1+\left(\frac{-2x}{1-x^2}\right)^2}dx$$

is not evaluated.

Solution This failure is mostly a consequence of insufficient simplification. A good simplification algorithm would realize that

$$\sqrt{1+\left(\frac{-2x}{1-x^2}\right)^2} = \sqrt{\frac{1+2x^2+x^4}{(1-x^2)^2}} = \sqrt{\frac{(1+x^2)^2}{(1-x^2)^2}} = \frac{1+x^2}{1-x^2}$$

which can easily be integrated.

When examining the algorithm's behavior, one notices that most problems are caused by algebraic integrands. This reinforces our feeling that he algebraic case is the hardest even with the rule-based approach.

Chapter 10

Discussion

10.1 Extensions of Risch's algorithm

The Risch algorithm as described in the preceding chapters can be extended in several ways. The easiest such improvement would be to allow arbitrary primitive and hyperexponential extensions in the integrand, rather than just logarithms and exponentials.

Definition 10.1 *Let $\mathbb{E}$ be a differential field extension of $\mathbb{F}$. An element $\theta \in \mathbb{E}$ is said to be*

1. *primitive if there exists $f \in \mathbb{F}$ such that $\theta' = f$*

2. *hyperexponential if there exists $f \in \mathbb{F}$ such that $\theta' = f\theta$*

Notice the similarity between the definition of primitive and hyperexponential extensions, and the definitions of logarithmic (4.9) and exponential (4.10) extensions. Because of this similarity, most of the theorems in chapters 5 and 6 will hold for primitive and hyperexponential extensions with little or no change in the proofs.

It is also possible to modify Risch's algorithm to handle tangents without rewriting them as complex exponentials. This is done by treating tangents as a third type of field extension in the absence of imaginary numbers, but come at the cost of a new case in almost all proofs and algorithms in the preceding chapters. Nevertheless, it could be worth the extra complications since the final output might be more readable.

Unlike the algorithm presented here, the method originally described by Risch could also integrate algebraic functions. Actually creating a practical algorithm for this requires algorithms from algebraic geometry which are beyond the scope of this thesis.

There are generalizations of Liouville's theorem which allow certain classes of special functions in the integral, for example by Singer et al. [25] and by Baddoura [1]. These extensions can sometimes be used to compute integrals in terms of

non-elementary functions, but it is not fully solved how to turn them into efficient decision procedures.

10.2 Concerning the complexity of integration

There are many reasons why it is difficult to perform a complexity analysis of Risch's algorithm. The first problem is that it is not clear what exactly is meant by Risch's algorithm. The method described by Risch was an outline of an algorithm, but many subproblems were not satisfactorily solved at the time. Today, Risch's algorithm is used to denote almost any realization of Risch's outline for integrating elementary functions, and consequently the complexities can vary.

One major problem is that it is not clear how one should measure the size of the input. If one uses the number of bits in the input as a measure, then it is easy to see that even integration of rational functions will have exponential complexity since $\int \frac{x^n-1}{x-1}dx$ is a polynomial with n terms while the input size is proportional to $\log(n)$ where n is a large integer. In this case, the swell could be dismissed as an artifact of using a rational expression to represent a polynomial, but similar swell occur for example when computing $\int x^n e^x dx$. Since $\int x^n e^x dx = p(x)e^x$ where $p(x)$ is a polynomial of degree n with coefficients up to $n!$, it is easy to see that both the number of terms and the bit length of the coefficients grow exponentially with the bit length of n.

Aside from the theoretical difficulties in defining the complexity of Risch's algorithm, there are also practical difficulties when computing it. The most obvious problem is the sheer number of functions or algorithms that are part of the integration procedure. Because of intermediate expression swell, not only would one have to analyze the complexity of all the subalgorithms, but also how their output size depend on input size. One way to deal with this problem is to use modular techniques.

Gerhard [15] gave modular algorithms for several of the subproblems needed for integration, including some asymptotically optimal solutions. For rational functions he obtained the following simple result.

Theorem 10.2 *Let $f, g \in \mathbb{Z} \setminus \{0\}$ be polynomials of degree at most n with integer coefficients bounded by 2^λ. Then the symbolic integral of the rational function f/g can be computed using $\mathrm{O}(n^8 + n^6\lambda^2)$ word operations using classical arithmetic and $\mathrm{O}(n^2 M(n^3+n^2\lambda)\log(n\lambda))$ word operations using fast multiplication with cost $M(n)$.*

10.3 Future work

There are many aspects of symbolic integration that would be interesting to examine further, the most obvious one being how to extend Risch's algorithm to handle new classes of functions. Some less ambitious and perhaps less obvious projects are discussed in the following paragraphs.

10.3.1 A simpler proof of the Lazard-Rioboo-Trager formula

The Lazard-Rioboo-Trager formula 3.5 for integrating rational functions was left without proof because the known proofs depend on how the subresultants change under a homomorphism of the coefficient ring. Resultants and subresultants are traditionally studied by interpreting them as determinants of certain matrices. Since integration is such a fundamental problem, it would be very nice to be able to explain the Lazard-Rioboo-Trager theorem without having to include this extra machinery. This motivates asking whether the theorem can be proven directly using the same definition and approach as in appendix C.

10.3.2 Symbolic integration in numerical computations

Symbolic algorithms are often perceived as rather restricted, compared to its numerical counterparts. While it is true that many problems cannot be solved satisfactorily using only algebraic manipulation, the numerical algorithms also have a number of weaknesses related to round-off and truncation errors. In particular, one will not know whether the computed solution even resemble the exact solution without some form of precision tracking or analysis of how the numerical errors propagate in the algorithm. Although numerical computation was not the subject of this thesis, it would be interesting to know the extent to which symbolic algorithms can be used to automatically transform problems to forms which are more suitable for numerical computation.

Davenport [9] did a small informal study of how well symbolic integration works in practice based on problems suggested by colleagues. Of the problems considered, he found that approximately one third could be solved by symbolic integration, another third was simplified (for example by solving one dimension of a multiple integral) while the remaining third did not yield to a symbolic approach.

Geddes and Fee [13] discussed the use of symbolic techniques to remove singularities in numerical integrals, a method which is used internally by Maple. The authors also remarked that their experiences with hybrid symbolic-numeric algorithms had been very promising.

With some exceptions like the ones mentioned above, there seem to be little work published on how to combine computer algebra with numerical computations and definitely less than the amount published on either subject.

10.3.3 Symbolic integration in education

The heuristic method discussed in chapter 2 can be adapted in a number of ways that might be useful in education. For example it could:

- Provide hints on how to compute an integral
- Provide complete solutions
- Generate examples that demonstrate the use of a certain rule

- Generate exercises that can be solved using a certain set of rules

The feasibility of generating hints and complete solutions was demonstrated in sections 9.2 and 9.3, but it seems that neither these nor the other possibilities have been implemented in the existing computer algebra systems.

The tasks listed above are traditionally done by the teacher or the author of the course book, but could be done interactively by a computer algebra system. The ability to automatize these asks could be utilized for example to create programs which generates exercises depending on the which rules the student has previously had difficulties applying. If the students get stuck, they could obtain instant feedback by asking the program for hints or solutions. In addition, a computer algebra system would typically choose rules in a more systematic way than most humans, which may may make easier to memorize the rules and how they are applied.

It is, however, difficult to predict whether the use of algebra systems would really improve learning in calculus courses. The availability of programs that can do the integrals for them, may make the students more inclined to cheat, or make them perceive the exercises as pointless. The risk of cheating on homeworks can be to some degree be allayed by choosing problems which are either not handled well or handled differently by the integration rules compared to what a human would. In section 9.2 we saw that that xe^{x^2} is an example of such a problem.

Thus, before any serious use in education one would have to investigate whether or not the computer algebra improves learning and, if so, how it should be incorporated into the curriculum.

10.3.4 Improvements of the implementation

From a practical perspective, it would have been very nice to implement Risch-Norman's algorithm. Bronstein has shown that it can be implemented in Maple in less than 100 lines of code [2], and that it performs very well in practice. Just as the major difficulty in implementing heuristic integration is the pattern matching, the major difficulty in implementing Risch-Norman's parallel integration algorithm is the need for factorization of multivariate polynomials, possibly over algebraic number fields. Since our main goal when writing the computer algebra system was to implement the heuristic integration rules, the design choices have favored this approach. No time was spent on developing the advanced polynomial algorithms that are prerequisites for Risch-Norman's algorithm.

10.4 Conclusions

There are several approaches to symbolic integration with different strengths and weaknesses. A set of heuristic integration rules is easy to understand and can be improved by the user when the need arise. On the other hand, the success rate is moderate, typically similar to that of a decent student. The most interesting aspect

of this approach is its ability to generate readable solutions, demonstrated in section 9.3.

The Risch-Norman algorithm is a more systematic approach which offer a high rate of success and can even be applied to non-elementary integrands. Since the algorithm simply does an educated guess of the general form of the integral, differentiates it and then tries to identify the coefficients, it is not very difficult to understand the idea or verify its correctness. However, understanding why the guess is made in a particular way requires some differential algebra. Implementations of the algorithm can be short, but requires several algorithms from computer algebra such as equivalence testing, solution of linear systems and multivariate polynomial factorization.

Being entirely based on differential algebra, the Risch decision procedure is the most difficult to understand of the three discussed methods. It is also the most difficult to implement as it consists of many parts not normally used elsewhere in a computer algebra system. The advantage of this method is that it will find an elementary integral of an elementary function if one exists and otherwise detect that the integral in non-elementary. Thus, the success rate is the best possible in the context of elementary functions, but on the other hand, it cannot easily be generalized to handle non-elementary functions. Except for special applications where detection of non-elementary integrals is necessary, it would be better to implement Risch-Norman's algorithm instead of the full Risch algorithm.

Bibliography

[1] J. Baddoura. Integration in finite terms with elementary functions and dilogarithms. *J. Symb. Comput.*, 41(8):909–942, 2006.

[2] M. Bronstein. The poor man's integrator, v 1.1, 2005. Available online from `http://www-sop.inria.fr/cafe/Manuel.Bronstein/pmint/pmint.txt` (Retrieved October 20, 2008).

[3] M. Bronstein. The transcendental Risch differential equation. *J. Symb. Comput.*, 9(1):49–60, 1990.

[4] M. Bronstein. *Symbolic integration. I: Transcendental functions*, volume 1 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, second edition, 2005.

[5] M. Bronstein. Structure theorems for parallel integration. *J. Symb. Comput.*, 42(7):757–769, 2007.

[6] W. S. Brown. The subresultant PRS algorithm. *ACM Trans. Math. Software*, 4(3):237–249, 1978.

[7] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[8] J. S. Cohen. *Computer algebra and symbolic computation: Mathematical methods*. AK Peters Ltd., Natick, Massachusetts, 2003.

[9] J. H. Davenport. Symbolic and numeric manipulation of integrals. In *Accurate Scientific Computations*, pages 168–180. Springer-Verlag, 1985.

[10] J. H. Davenport, Y. Siret, and E. Tournier. *Computer algebra: Systems and algorithms for algebraic computation*. Academic Press Ltd., London, second edition, 1993.

[11] J. B. Fraleigh. *A first course in abstract algebra*. Addison-Wesley Publishing Company, Inc, 1999.

[12] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer Academic Publishers, Boston, MA, 1992.

[13] K. O. Geddes and G. J. Fee. Hybrid symbolic-numeric integration in MAPLE. In *ISSAC*, pages 36–41, 1992.

[14] K. O. Geddes and L. Y. Stefanus. On the Risch-Norman integration method and its implementation in MAPLE. In *ISSAC '89: Proceedings of the ACM-SIGSAM 1989 international symposium on Symbolic and algebraic computation*, pages 212–217, New York, NY, USA, 1989. ACM.

[15] J. Gerhard. *Modular Algorithms in Symbolic Summation and Symbolic Integration*, volume 3218 of *Lecture Notes in Computer Science*. Springer, 2004.

[16] G. H. Hardy. *The integration of functions of a single variable*. Hafner Publishing Co., New York, 1971. Reprint of the second edition, 1916, Cambridge Tracts in Mathematics and Mathematical Physics, No. 2.

[17] S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, third edition, 2002.

[18] D. Lazard and R. Rioboo. Integration of rational functions: rational computation of the logarithmic part. *J. Symb. Comput.*, 9(2):113–115, 1990.

[19] R. E. Maeder. Algbench: An object-oriented symbolic core system. *Lecture Notes in Computer Science*, 721:56–64, 1993.

[20] K. McIsaac. Pattern matching algebraic identities. *SIGSAM Bull.*, 19(2):4–13, 1985.

[21] J. Moses. Algebraic simplification: a guide for the perplexed. In *SYMSAC '71: Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, pages 282–304, New York, NY, USA, 1971. ACM.

[22] T. Mulders. A note on subresultants and the Lazard/Rioboo/Trager formula in rational function integration. *J. Symb. Comput.*, 24(1):45–50, 1997.

[23] D. Richardson. Some undecidable problems involving elementary functions of a real variable. *J. Symbolic Logic*, 33(4):514–520, 1968.

[24] R. H. Risch. The problem of integration in finite terms. *Trans. Amer. Math. Soc.*, 139:167–189, 1969.

[25] M. F. Singer, B. D. Saunders, and B. F. Caviness. An extension of Liouville's theorem on integration in finite terms. In *Proc. 1981 ACM Symposium on Symbolic and Algebraic Computation*, 1981.

[26] B. L. van der Waerden. *Modern Algebra. Vol. I.* Frederick Ungar Publishing Co., New York, N. Y., 1949.

[27] S. Wolfram. *The Mathematica Book, Fifth Edition*. Wolfram Media, 2003.

Appendix A

Partial fractions decomposition

In this appendix, we prove the existence of a partial fraction decomposition.

Lemma A.1 *Let $\mathbb{F}$ be a field and $p/(qr)$ a proper rational expression with $p, q, r \in \mathbb{F}[x]$ and q, r relatively prime. Then there exists polynomials $a, b \in \mathbb{F}[x]$ such that*

$$\frac{p}{qr} = \frac{a}{q} + \frac{b}{r}$$

where $\deg(a) < \deg(q)$ *and* $\deg(b) < \deg(r)$

Proof The condition that q and r are relatively prime means that $\gcd(q, r) = 1$, so there exist polynomials $\tilde{a}$ and $\tilde{b}$ such that $\tilde{a}r + \tilde{b}q = 1$. Hence,

$$\frac{p}{qr} = \frac{p(\tilde{a}r + \tilde{b}q)}{qr} = \frac{p\tilde{a}}{q} + \frac{p\tilde{b}}{r}$$

In this expression however, $p\tilde{a}$ and $p\tilde{b}$ need not have degrees less than $\deg(q)$ and $\deg(r)$ respectively. Using the division algorithm, we obtain $p\tilde{a} = d_a q + a$ and $p\tilde{b} = d_b r + b$. Now, for

$$\frac{p}{qr} = \frac{d_a q + a}{q} + \frac{d_b r + b}{r} = \frac{(d_a + d_b)qr + ar + bq}{qr}$$

to be a proper fraction, we need $d_a + d_b = 0$. Without loss of generality, we may assume that $d_a = d_b = 0$, so

$$\frac{p}{qr} = \frac{a}{q} + \frac{b}{r}$$

as desired. □

Theorem A.2 *Let $\mathbb{F}$ be a field and p/q a proper rational expression with $p, q \in \mathbb{F}[x]$ and suppose that q has the factorization $q = \prod_{i=1}^{k} q_i^{e_i}$ where the q_i are pairwise relatively prime. Then there exists polynomials $r_{ij} \in \mathbb{F}[x]$ such that*

$$\frac{p}{q} = \sum_{i=1}^{k} \sum_{j=1}^{e_i} \frac{r_{ij}}{q_i^j}$$

where $\deg(r_{ij}) < \deg(q_i)$

Proof Using the previous lemma $k-1$ times gives a partial fraction decomposition

$$\frac{p}{q} = \sum_{i=1}^{k} \frac{r_i}{q_i^i}$$

where $\deg(r_i) < \deg(q_i^j)$. If $\deg(r_i) \geq \deg(q_i)$, euclidean division can express $r_i = \tilde{r}_i q_i + r_{ij}$, so

$$\frac{r_i}{q_i^j} = \frac{r_{ij}}{q_i^j} + \frac{\tilde{r}_i}{q_i^{j-1}}$$

where $\deg(r_{ij}) < \deg(q_i)$ and $\deg(\tilde{r}_i) < \deg(q_i^{j-1})$. Induction on j proves the theorem. □

Appendix B

Square-free factorization

Definition B.1 *A polynomial is square-free if it has no repeated factors.*

Theorem B.2 *A polynomial $p \in \mathbb{F}[x]$ is square-free if and only if* $\gcd(p, p') = 1$.

Proof ($\Leftarrow$) Assume that p has some repeated factor, so $p = f^n h$ for some $n > 1$. Then $p' = nf^{n-1}f'h + f^n h'$, so p and p' have a common factor f^{n-1}.
($\Rightarrow$) On the other hand, if p is square-free, with irreducible factorization

$$p = p_1 p_2 \cdots p_k$$

where all p_i are distinct, then

$$p' = p_1' p_2 \cdots p_k + p_1 p_2' p_3 \cdots p_k + \ldots + p_1 p_2 \cdots p_{k-1} p_k'$$

Now, any nontrivial divisor of p must be a multiple of some p_i, so if $\gcd(p, p')$ is nontrivial we can assume that p_i divides both p and p'. Since p_i clearly divides all terms of p' except the first, it must also divide p_i' in order to divide p'. This is a contradiction since $\deg(p_i) > \deg(p_i')$, so $\gcd(p, p')$ must be 1. □

Definition B.3 *A square-free factorization of p is a factorization $p = \prod_{i=0}^{k} (p_i)^i$, such that all p_i are square-free and* $\gcd(p_i, p_j) = 1$ *if $i \neq j$.*

While it is difficult to compute the full factorization into irreducibles, computing a square-free factorization is not.

Let $p \in \mathbb{F}[x]$ have the square-free factorization $p = p_1 p_2^2 p_3^3 \cdots p_k^k$, so

$$p' = \sum_{i=1}^{k} \left((i-1) p_i^{i-1} p_i' \prod_{i \neq j} p_j^j \right)$$

Let

$$a_1 = \gcd(p, p') = p_2 p_3^2 p_4^3 \cdots p_k^{k-1}$$

and

$$b_1 = \frac{p}{\gcd(p, p')} = p_1 p_2 p_3 \cdots p_k$$

Now we can iterate

$$\begin{aligned} c_i &= \gcd(a_i, b_i) = p_{i+1} p_{i+2} \cdots p_k \\ a_{i+1} &= \frac{a_i}{c_i} = p_{i+2} p_{i+3}^2 \cdots p_k^{k-i-1} \\ b_{i+1} &= c_i \end{aligned}$$

The square-free factors p_i are given by $p_i = \frac{b_i}{c_i}$.

Theorem B.4 *If $\mathbb{F}$ has characteristic 0, then a square-free factorization over $\mathbb{F}$ is simultaneously a square-free factorization over $\bar{\mathbb{F}}$, where $\bar{\mathbb{F}}$ denotes the algebraic closure of $\mathbb{F}$.*

Proof This classic result in field theory is usually stated by saying that all fields of characteristic 0 are perfect. The theorem and its proof can be found in most abstract algebra textbooks, for example Fraleigh's [11] and van der Waerden's [26]. □

It is interesting that this holds for all finite fields too, so the only fields for which a square-free factorization over $\mathbb{F}$ can fail to be a square-free factorization over $\bar{\mathbb{F}}$ are infinite fields of prime characteristic. Such fields are not very common in practice.

Appendix C

Greatest common divisors and the resultant

If $\mathbb{F}$ is a field, then $\mathbb{F}[x]$ is an euclidean domain i.e. an integral domain with euclidean division. It is well known that the euclidean algorithm can be used to compute greatest common divisors in any euclidean domain. For any unique factorization domain D, $D[x]$ will be also be a unique factorization domain, but it need not be euclidean because we cannot divide by the coefficients. In practice, this happens for polynomials over $\mathbb{Z}$ and for polynomials in several variables over any field.

Since we cannot use the familiar euclidean algorithm to compute greatest common divisors, we will define a pseudo-division which share some important properties with the ordinary polynomial division, but which can be computed without divisions in the coefficient ring.

Definition C.1 *Let $a, b \in D[x]$ be polynomials over a unique factorization domain D and let $\operatorname{lc}(b)$ denote the leading coefficient of b. Then there are polynomials $q, r \in D[x]$ such that*

$$\operatorname{lc}(b)^{\deg(a)-\deg(b)+1} a = qb + r$$

and $\deg(r) < \deg(b)$. We call $q = \operatorname{pquo}(a, b)$ the pseudo-quotient and $r = \operatorname{prem}(a, b)$ the pseudo-remainder of a and b.

Definition C.2 *Let D be a unique factorization domain, and let $p = \sum_{i=0}^{n} p_i x^i$ be a polynomial in $D[x]$. We define the content and primitive part of p as*

$$\begin{aligned} \operatorname{cont}(p) &= \gcd(p_0, p_1, \ldots, p_n) \\ \operatorname{pp}(p) &= p/\operatorname{cont}(p) \end{aligned}$$

Definition C.3 *Let D be a unique factorization domain, and p, q polynomials in $D[x]$ with $\deg(p) \geq \deg(q)$. We define the polynomial remainder sequence or PRS*

for p and q as the finite sequence defined by

$$\begin{aligned} R_0 &= p(x) \\ R_1 &= q(x) \\ \beta_i R_{i+1} &= \operatorname{prem}(R_{i-1}, R_i) \quad \text{if } R_i \neq 0 \end{aligned}$$

where β_i is chosen to limit coefficient growth. The particular PRS with $\beta_i = 1$ is called the euclidean PRS and the one with $\beta_i = \operatorname{cont}(\operatorname{prem}(R_{i-1}, R_i))$ is called the primitive PRS.

Definition C.4 *The polynomials p and q in $D[x]$ are said to be similar if there exist $a, b \in D\backslash\{0\}$ such that $ap = bq$*

Theorem C.5 *If p and q are polynomials in $D[x]$ where D is a unique factorization domain and $(R_0, R_1, \ldots, R_k, 0)$ a PRS for p and q, then $\gcd(p, q)$ is similar to R_k. Furthermore, if p and q are primitive, then*

$$\gcd(p, q) = \operatorname{pp}(R_k)$$

Proof Inserting the definition of the pseudo-remainder in the definition of the PRS we see that there exist $\alpha, \beta \in D$ and $Q \in D[x]$ such that

$$\alpha R_{i-1} = QR_i + \beta R_{i+1}$$

Hence

$$\alpha \gcd(R_{i-1}, R_i) = \gcd(\alpha R_{i-1}, R_i) = \gcd(R_i, \beta R_{i+1}) = \beta \gcd(R_i, R_{i+1})$$

and iterating this, we see that there are elements $a, b \in D$ such that

$$a \gcd(p, q) = b \gcd(R_k, 0) = bR_k$$

which proves the first part of the statement. If p and q are primitive then $\operatorname{pp}(\gcd(p, q)) = \gcd(p, q)$ so

$$\gcd(p, q) = \operatorname{pp}(\gcd(p, q)) = \operatorname{pp}(a \gcd(p, q)) = \operatorname{pp}(bR_k) = \operatorname{pp}(R_k)$$

□

This last theorem shows how to use polynomial remainder sequences to compute greatest common divisors. We will now turn the attention to the resultant and its relationship with the greatest common divisor.

Definition C.6 *Let D be an integral domain, and let p and q be polynomials in $D[x]$, with factorizations*

$$\begin{aligned} p(x) &= a \prod_{i=1}^{m} (x - \alpha_i) \\ q(x) &= b \prod_{j=1}^{n} (x - \beta_j) \end{aligned}$$

in $\bar{D}[x]$. *We define the resultant with respect to* x *as*

$$\begin{aligned}\operatorname{res}_x(p,q) &= a^n b^m \prod_{i=1}^{m}\prod_{j=1}^{n}(\alpha_i - \beta_j)\\ &= a^n \prod_{i=1}^{m} q(\alpha_i)\\ &= (-1)^{mn} b^m \prod_{j=1}^{n} p(\beta_j)\end{aligned}$$

We will not prove the equivalence of the three expressions above, nor that the seemingly unnecessary multiplications by a^n and b^m in fact ensure that $\operatorname{res}_x(p,q) \in D$. The interested reader can consult almost any algebra textbook (for example [17]) for a discussion of the mathematical foundations of the resultant, and [6] or [7] for nice descriptions of the subresultant algorithm for computing resultants. When there can be no confusion about the variable, we omit the explicit reference to it and write $\operatorname{res}(p,q)$ instead of $\operatorname{res}_x(p,q)$.

Theorem C.7 *Let* D *be a unique factorization domain, and* p, q *polynomials in* $D[x]$. *Then*

$$\operatorname{res}_x(p,q) = 0 \iff \deg(\gcd(p,q)) > 0$$

Proof ($\Leftarrow$) Suppose that $\deg(\gcd(p,q)) > 0$. Then p and q have a common zero, so $\operatorname{res}_x(p,q) = a^n b^m \prod_{i=1}^{m}\prod_{j=1}^{n}(\alpha_i - \beta_j) = 0$.
($\Rightarrow$) On the other hand, suppose that $\operatorname{res}_x(p,q) = a^n b^m \prod_{i=1}^{m}\prod_{j=1}^{n}(\alpha_i - \beta_j) = 0$. Since D is an integral domain, one of the factors $\alpha_i - \beta_j$ must be zero, so p and q have a common root and thus $\deg(\gcd(p,q)) > 0$. □

Definition C.8 *The subresultant polynomial remainder sequence for the polynomials* $p, q \in D[x]$ *is the PRS defined by* $\beta_i = g_{i-1} h_{i-1}^{\delta_{i-1}}$ *where*

$$\begin{aligned}d_i &= \deg(R_i)\\ \delta_i &= d_i - d_{i+1}\\ g_0 &= h_0 = 1\\ g_i &= \operatorname{lc}(R_i)\\ h_i &= g_i^{\delta_{i-1}} h_{i-1}^{1-\delta_{i-1}}\end{aligned}$$

The subresultant PRS can be used to compute greatest common divisors and, as we shall see, resultants. The following elegant proof is essentially from Cohen [7].

Theorem C.9 *Let p and q be two primitive polynomials and let R_k be the last element of (strictly) positive degree in the subresultant PRS defined above. Then*

$$\operatorname{res}(p,q) = (-1)^{\left(\sum_{i=1}^{k} d_{i-1}d_i\right)} h_{k+1}$$

where d_i is the degree of R_i

Proof Let d_i be the degree of R_i and let $\{\alpha_j\}$ be the set of zeros to the polynomial R_i

$$\begin{aligned}
\operatorname{res}(R_{i-1},R_i) &= (-1)^{d_{i-1}d_i}\operatorname{lc}(R_i)^{d_{i-1}} \prod_{j=1}^{\deg R_i} R_{i-1}(\alpha_j) \\
&= (-1)^{d_{i-1}d_i}\operatorname{lc}(R_i)^{d_{i-1}} \prod_{j=1}^{\deg R_i} \frac{\operatorname{prem}(R_{i-1},R_i)(\alpha_j)}{\operatorname{lc}(R_i)^{\delta_{i-1}+1}} \\
&= (-1)^{d_{i-1}d_i}\operatorname{lc}(R_i)^{d_{i-1}} \prod_{j=1}^{\deg R_i} \frac{g_{i-1}h_{i-1}^{\delta_{i-1}}R_{i+1}(\alpha_j)}{\operatorname{lc}(R_i)^{\delta_{i-1}+1}} \\
&= (-1)^{d_{i-1}d_i}\operatorname{lc}(R_i)^{d_{i-1}} \prod_{j=1}^{\deg R_i} \frac{g_{i-1}h_{i-1}R_{i+1}(\alpha_j)}{\operatorname{lc}(R_i)h_i} \\
&= (-1)^{d_{i-1}d_i}\frac{\operatorname{lc}(R_i)^{\delta_{i-1}}g_{i-1}^{d_i}h_{i-1}^{d_i}}{h_i^{d_i}} \prod_{j=1}^{\deg R_i} R_{i+1}(\alpha_j) \\
&= (-1)^{d_{i-1}d_i}\frac{\operatorname{lc}(R_i)^{\delta_{i-1}}g_{i-1}^{d_i}h_{i-1}^{d_i}}{h_i^{d_i}} \frac{\operatorname{res}(R_i,R_{i+1})}{\operatorname{lc}(R_i)^{d_{i+1}}} \\
&= (-1)^{d_{i-1}d_i}\frac{g_{i-1}^{d_i}h_{i-1}^{d_{i-1}-1}}{g_i^{d_{i+1}}h_i^{d_i-1}} \operatorname{res}(R_i,R_{i+1})
\end{aligned}$$

Hence

$$\begin{aligned}
\operatorname{res}(p,q) &= \left(\prod_{i=1}^{k}(-1)^{d_{i-1}d_i}\frac{g_{i-1}^{d_i}h_{i-1}^{d_{i-1}-1}}{g_i^{d_{i+1}}h_i^{d_i-1}}\right)\operatorname{res}(R_k,R_{k+1}) \\
&= (-1)^{\left(\sum_{i=1}^{k} d_{i-1}d_i\right)}\frac{g_0^{d_1}h_0^{d_0-1}}{g_k^0 h_k^{d_k-1}} \operatorname{res}(R_k,R_{k+1}) \\
&= (-1)^{\left(\sum_{i=1}^{k} d_{i-1}d_i\right)} h_k^{1-d_k}\operatorname{lc}(R_k) \\
&= (-1)^{\left(\sum_{i=1}^{k} d_{i-1}d_i\right)} h_{k+1}
\end{aligned}$$

□

Notice that many authors include extra signs in the β_i factors. Although this can simplify some formulas, it is often faster to keep track of signs separately as

we did above. Since the purpose of the β_i is to limit coefficient growth, it does not really matter which convention one uses.

TRITA-CSC-E 2009:095
ISRN-KTH/CSC/E--09/095--SE
ISSN-1653-5715

www.kth.se