Integration and Differential Equations in Computer Algebra

Manuel Bronstein
Institute for Scientific Computation
ETH-Zentrum, CH-8092 Zürich, Switzerland

Abstract

We describe in this paper how the problems of computing indefinite integrals and solving linear ordinary differential equations in closed form are now solved by computer algebra systems. After a brief review of the mathematical history of those problems, we outline the two major algorithms for them (respectively the Risch and Singer algorithms) and the recent improvements on those algorithms which has allowed them to be implemented.

1 Introduction

An *elementary function* of a variable x is a function that can be obtained from the rational functions in x by repeatedly adjoining a finite number of nested logarithms, exponentials, and algebraic numbers or functions. Since $\sqrt{-1}$ is elementary, the trigonometric functions and their inverses are also elementary, as well as all the "usual" functions of calculus. A *Liouvillian function* of x is a function that can be obtained like an elementary function except that arbitrary antiderivatives (integral signs) are allowed instead of logarithms in their construction. For example,

$$\sin(x + \tan(x^3 - \sqrt{x^3 - x + 1}))$$

is elementary, while

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int e^{-x^2} dx$$

is Liouvillian but not elementary, and the Bessel functions defined as the fundamental solutions of

$$x^2 \frac{d^2y}{dx^2} + x\frac{dy}{dx} + (\epsilon x^2 - v^2)y(x) = 0$$

where $\epsilon = \pm 1$, are not Liouvillian (and therefore not elementary). This article describes solutions to the *problem of solving linear ordinary differential equations in closed form*, i.e. given an equation of the form

$$A_n(x)y^{(n)} + \ldots + A_1(x)y' + A_0(x)y = 0 \qquad (1)$$

where the A_i's are Liouvillian functions, find whether it has Liouvillian solutions and a basis over the constants for them. This problem has been of great interest to mathematicians since the second half of the 19^{th} century, but has been mathematically solved only in the past 20 years. Even now, finding practical algorithms for this problem is still a current research topic. Some particular cases have however effective algorithms and are described in this paper.

In the first part we outline the history and recent solutions to the *integration problem* which can be seen as the first order case of the general problem. These solutions are now becoming standard and are implemented in most of the major computer algebra systems. In the second part, we describe very recent solutions to the higher order cases, solutions which have been experimentally implemented for the case when the A_i's of (1) are polynomials in the variable x.

2 Integration

In the *first-order* case ($n = 1$), equation (1) always has Liouvillian solutions, namely

$$y = ce^{-\int (A_0/A_1)dx}$$

for any constant c, so the problem of interest here is really whether there are *elementary* solutions, i.e. solutions which can be expressed without the integral sign. This is essentially the *problem of integration in finite terms*: to decide in a finite number of steps whether for a given elementary function f, the equation $y' = f$ has an elementary solution, and to compute it explicitly if it exists. This problem was studied extensively in the 19^{th} century, mostly when f was either a rational function [18] or an algebraic function [11, 12, 22, 23]. While the methods used were very constructive, no complete algorithm was found for the algebraic function case (Hardy [17] even stated that "there is reason to suppose that no such method can be given"). The development of algebraic geometry and differential algebra in the 20^{th} century finally allowed Risch to prove that the problem can be solved for arbitrary elementary functions [27, 28]. Risch's algorithm was implemented soon after it appeared [24], but only for the *purely transcendental elementary functions*, i.e. those elementary functions which do not contain any algebraic function. Indeed, Risch's proof for the general case [25, 26] did not represent a practical algorithm that could be used to solve integration problems. Davenport [15] then published an algorithm for the integration of purely algebraic functions which he partially implementated in the REDUCE computer algebra system. Finally, Trager [41] gave a "rational" algorithm for the integration of purely algebraic functions which has been implemented in the AXIOM and MAPLE computer algebra systems. That algorithm was then generalized to handle arbitrary elementary functions in [6], and partially implemented in AXIOM.

2.1 Liouville's theorem

We now present the basic principle that forms the foundation of Risch's algorithm: that if an elementary function f has an elementary integral, then it has one of the form $\int f = v + \sum_{i=1}^{n} c_i \log(u_i)$ where the c_i's are constants and any quantity appearing in v or the u_i's must already appear in f. Before formalizing this principle and describing Risch's algorithm, we need some definitions from elementary differential algebra.

A *differential field* is a field k with a given map $a \to a'$ from k into k, satisfying $(a+b)' = a' + b'$ and $(ab)' = a'b + ab'$. Such a map is called a *derivation* on k. An element $a \in k$ which satisfies $a' = 0$ is called a *constant*. The constants of k form a subfield of k.

A differential field K is a *differential extension of* k if $k \subseteq K$, and the derivation on K extends the one on k. Let K be a differential extension of k, and $\theta \in K$. We say that θ is *elementary over* k, if $k(\theta)$ and k have the same subfield of constants, and either:

(i) $\theta' = \eta'/\eta$ for $\eta \in k^*$, in which case we say that θ is logarithmic over k, and write $\theta = \log(\eta)$,

(ii) $\theta' = \eta'\theta$ for $\eta \in k$, in which case we say that θ is exponential over k, and write $\theta = \exp(\eta)$,

(iii) θ is algebraic over k, i.e. $P(\theta) = 0$ for some non-zero polynomial $P \in k[X]$.

A differential extension K of k is an *elementary extension of* k, if there exist $\theta_1, \ldots, \theta_m \in K$ such that $K = k(\theta_1, \ldots, \theta_m)$ and θ_i is elementary over $k(\theta_1, \ldots, \theta_{i-1})$ for $i = 1 \ldots m$.

With those definitions, we can state Liouville's theorem, as proven by Risch [27]:

Theorem 1 *Let F be a differential field of characteristic zero, K be the constant subfield of F, and $\alpha \in F$. If there is an element y in some elementary extension of F such that $y' = \alpha$, then there exist $v \in F$, $c_1, \ldots, c_n \in \overline{K}$, and $u_1, \ldots, u_n \in \overline{K}F$ such that*

$$\alpha = v' + \sum_{i=1}^{n} c_i \frac{{u_i}'}{u_i}. \qquad (2)$$

See [29] for an elementary proof of this result. Note that this Theorem does not state that any integral of α must be of the form (2), but that if there is an elementary integral, then there must also be one of that form, although it is not always the most natural one. For example,

$$\int \frac{dx}{1+x^2} = \arctan(x) = \frac{\sqrt{-1}}{2} \log(\frac{\sqrt{-1}+x}{\sqrt{-1}-x}).$$

The Risch algorithm always looks for an integral of the form (2), and Liouville's Theorem guarantees that if there is none of that form, then the integrand has no elementary integral at all.

2.2 The Risch algorithm

Elementary extensions are useful for modeling any function as a rational or algebraic function of one main variable over the other variables. Given an elementary integrand $f(x)dx$, the algorithm constructs first a field K containing all the constants appearing in f, then the rational function field $K(x)$, and finally builds a tower $L = K(x)(\theta_1, \ldots, \theta_m)$ where the θ_i's are all the elementary functions needed to express f. The derivation used at every step is $' = d/dx$.
Example: to find $\int \cos(x)dx$, we rewrite $cos(x)$ as $(e^{x\sqrt{-1}} + e^{-x\sqrt{-1}})/2$, so the constant field is $K = \mathbf{Q}(\sqrt{-1})$ and the tower is $L = K(x)(\theta)$ where $\theta = e^{x\sqrt{-1}}$ is exponential over $K(x)$. The integrand is then $(\theta^2+1)/2\theta$, which we view as a univariate rational function in θ over $F = K(x)$.

Once we have our integrand $f \in K(x)(\theta_1, \ldots, \theta_m)$ expressed in a tower of elementary extensions, four cases can happen:

1. $m = 0$ (base case) in which case $f \in K(x)$ is a rational function of x,
2. $\theta = \theta_m$ is transcendental and logarithmic over $F = K(x)(\theta_1, \ldots, \theta_{m-1})$,
3. $\theta = \theta_m$ is transcendental and exponential over $F = K(x)(\theta_1, \ldots, \theta_{m-1})$,
4. $\theta = \theta_m$ is algebraic over $F = K(x)(\theta_1, \ldots, \theta_{m-1})$.

The original Risch algorithm treated those 4 cases separately, and we outline them in the next four sections.

2.2.1 The base case

In this case $m = 0$, so $f \in K(x)$ is a rational function. Risch uses the following traditional method: write $f = P + A/D$ where $P, A, D \in K[x]$, and $\deg(A) < \deg(D)$. Let $D = \prod_{i=1}^n P_i^{e_i}$ be a prime factorisation of D over K. Then, using a partial fraction decomposition of f:

$$\int f = \int P + \sum_{i=1}^{n} \sum_{j=1}^{e_i} \int \frac{A_{ij}}{P_i^j} \tag{3}$$

where $A_{ij} \in K[x]$ and $\deg(A_{ij}) < \deg(P_i)$. Computing $\int P$ poses no problem (it will for any other class of functions), and for each i and $j > 1$, we use the extended Euclidean algorithm to find B_{ij} and C_{ij} in $K[x]$ such that

$$B_{ij} P_i{}' + C_{ij} P_i = \frac{A_{ij}}{1-j}. \tag{4}$$

It is then easy to verify that

$$\int \frac{A_{ij}}{P_i^j} = \frac{B_{ij}}{P_i^{j-1}} + \int \frac{-B_{ij}{}' + (1-j)C_{ij}}{P_i^{j-1}}. \tag{5}$$

Doing this for each i and for $j = e_i$ down to $j = 2$, we are eventually left to compute integrals of the form $\int E_i/P_i$ where P_i is irreducible, $E_i \in K[x]$ and $\deg(E_i) < \deg(P_i)$. By computing a factorisation $P_i = \prod_{j=1}^{d_i}(x - \alpha_{ij})$ over $\overline{K}$ and a partial fraction decomposition of those integrands, we get

$$\int \frac{E_i}{P_i} = \int \sum_{j=1}^{d_i} \frac{a_{ij}}{x - \alpha_{ij}} = \sum_{j=1}^{d_i} a_{ij} \log(x - \alpha_{ij})$$

which completes the algorithm.

2.2.2 The logarithmic case

In this case $\theta = \theta_m$ is transcendental over $F = K(x)(\theta_1, \ldots, \theta_{m-1})$, $f \in F(\theta)$, and there exists $\eta \in F^*$ such that $\theta' = \eta'/\eta$ (i.e. $\theta = \log(\eta)$). We still write $f = P + A/D$ where $P, A, D \in F[\theta]$, and $\deg(A) < \deg(D)$. Let $D = \prod_{i=1}^{n} P_i^{e_i}$ be a prime factorisation of D over F. Formula (3) still holds, but this time $\int P$ is computed differently: write $P = A_d\theta^d + \ldots A_1\theta + A_0$ where $A_i \in F$. If f has an elementary integral, then there exist $B_0, \ldots, B_{d+1} \in F$ such that

$$\int A_d\theta^d + \ldots A_1\theta + A_0 = B_{d+1}\theta^{d+1} + \ldots + B_1\theta + \int B_0.$$

Differentiating and comparing coefficients of θ^d, we get ${B_{d+1}}' = 0$ and

$$A_d = {B_d}' + (d+1)B_{d+1}\frac{\eta'}{\eta}. \tag{6}$$

Since $A_d \in F$, we can recursively apply the integration algorithm to it (A_d contains one less monomial than the original integrand, so this recursion must stop). Recursively integrating A_d, we either find that (6) has no solution, in which case f has no elementary integral, or we get the constant B_{d+1}, and B_d up to an additive constant. Write then $B_d = \overline{B_d} + b_d$ where $\overline{B_d} \in F$ is known and $b_d \in F$ is an undetermined constant. Comparing coefficients of θ^{d-1} yields

$$A_{d-1} - d\overline{B_d}\frac{\eta'}{\eta} = {B_{d-1}}' + db_d\frac{\eta'}{\eta}.$$

Recursively integrating the left hand side computes b_d and B_{d-1} up to an additive constant. Repeating this process, either one of the recursive integration fails, in which case f has no elementary integral, or we get $B_0, \ldots, B_{d+1}$.

The integrals $\int A_{ij}/P_i^j$ for $j > 1$ are computed exactly as in the base case since formulas (4) and (5) are also valid in $F[\theta]$. So, after integrating P and removing all the terms with multiple powers of P_i in the denominator, we are left with computing

$$\int B_0 + \int \sum_{i=1}^{n} \frac{E_i}{P_i}.$$

The integral of $B_0 \in F$ is computed recursively and f has no elementary integral if B_0 does not. By computing a factorisation $P_i = \prod_{j=1}^{d_i} Q_{ij}$ over $\overline{K}F$, and a partial fraction decomposition of E_i/P_i we get $E_i/P_i = \sum_{j=1}^{d_i} E_{ij}/Q_{ij}$ where $E_{ij}, Q_{ij} \in \overline{K}F[\theta]$ and $\deg(E_{ij}) < \deg(Q_{ij})$. Write $a_{ij} = E_{ij}/Q'_{ij}$. From Liouville's Theorem, if $a'_{ij} \neq 0$ for some i, j, then f has no elementary integral, otherwise

$$\int \sum_{i=1}^{n} \frac{E_i}{P_i} = \sum_{i=1}^{n}\sum_{j=1}^{d_i} a_{ij}\log(Q_{ij}).$$

2.2.3 The exponential case

In this case $\theta = \theta_m$ is transcendental over $F = K(x)(\theta_1, \ldots, \theta_{m-1})$, $f \in F(\theta)$, and there exists $\eta \in F$ such that $\theta' = \eta'\theta$ (i.e. $\theta = \exp(\eta)$). We again write $f = P + A/D$ where $A, D \in F[\theta]$, and $\deg(A) < \deg(D)$, but we allow $P \in F[\theta, \theta^{-1}]$ so that we can assume that θ does not divide D in $F[\theta]$. Let $D = \prod_{i=1}^{n} P_i^{e_i}$ be a prime factorisation of D over F. Formula (3) holds again, and $\int P$ is computed as follows: write $P = A_d\theta^d + \cdots A_1\theta + A_0 + A_{-1}\theta^{-1} + \cdots + A_{-e}\theta^{-e}$ where $A_i \in F$. If f has an elementary integral, then each $A_i\theta^i$ must have an elementary integral, so

$$\int P = \int \sum_{i=-e}^{d} A_i\theta^i = \sum_{i=-e}^{d} \int A_i\theta^i.$$

For $i \neq 0$, $A_i\theta^i$ has an elementary integral if and only if there exists $B_i \in F$ such that

$$\int A_i\theta^i = B_i\theta^i.$$

Differentiating both sides, we get

$$A_i = B_i{}' + i\eta' B_i \tag{7}$$

which is called a *Risch differential equation over F*. Although solving it looks more complicated than solving $g' = f$, it is actually simpler than an integration problem since we look for solutions B_i in F only. Risch [25, 26, 27] gave an algorithm for solving this type of equation when F is an elementary extension of the rational function field. In the transcendental cases (base, logarithmic and exponential), he writes the coefficients A and η' and the unknown B as univariate rational functions, then bounds the order of B at all its potential singularities, and plugs in their partial fraction decompositions in equation (7). Equating the coefficients on both sides yield systems of linear algebraic equations for the unknown coefficients of the decomposition of B.

Solving equation (7) for every $i \neq 0$ either yields all the B_i's except B_0, or shows that f has no elementary integral if one of those equations has no solution in F. So we are left with computing

$$\int A_0 + \int \sum_{i=1}^{n} \frac{E_i}{P_i}$$

where θ does not divide any P_i. The integral of $A_0 \in F$ is computed recursively and f has no elementary integral if A_0 does not. By computing a factorisation $P_i = \prod_{j=1}^{d_i} Q_{ij}$ over $\overline{K}F$, and a partial fraction decomposition of E_i/P_i we get $E_i/P_i = \sum_{j=1}^{d_i} E_{ij}/Q_{ij}$ where $E_{ij}, Q_{ij} \in \overline{K}F[\theta]$ and $\deg(E_{ij}) < \deg(Q_{ij})$. Write $a_{ij} = E_{ij}/(Q'_{ij} - \deg(Q_{ij})\eta' Q_{ij})$. From Liouville's Theorem, if $a'_{ij} \neq 0$ for some i, j, then f has no elementary integral, otherwise

$$\int \sum_{i=1}^{n} \frac{E_i}{P_i} = \sum_{i=1}^{n} \sum_{j=1}^{d_i} a_{ij}(\log(Q_{ij}) - \deg(Q_{ij})\eta').$$

2.2.4 The algebraic case

In this case $\theta = \theta_m$ is algebraic over $K(x)(\theta_1, \ldots, \theta_{m-1})$. Let t be the "highest" transcendental among $x, \theta_1, \ldots, \theta_{m-1}$, which means that t is transcendental over the previous field in the tower, and all the successive extensions are by adjoining algebraic elements, θ among them. By the primitive element Theorem, we can write $K(x)(\theta_1, \ldots, \theta_m)$ as $F(t, y)$ where t is transcendental over F, y is algebraic over $F(t)$ and F is either K (in which case $t = x$), or of the form $F = K(x)(\theta_1, \ldots, \theta_q)$ where $0 \leq q < m - 1$. Since y is algebraic over $F(t)$, there exists an irreducible polynomial $P \in F[T, Y]$ such that $\deg_Y(P) > 0$ and $P(t, y) = 0$. The integrand $f \in F(t, y)$ is then seen as a univariate algebraic function over F.
Example: to find $\int \sqrt{x^3 + 1}dx$, the tower is $\mathbf{Q}(x)(\theta)$ where $\theta = \sqrt{x^3 + 1}$ is algebraic over $\mathbf{Q}(x)$. In this case, we have $F = \mathbf{Q}$, $t = x$, $y = \theta$, $P = Y^2 - T^3 - 1$ and the integrand is $f = y$.

Here also, by Liouville's Theorem, the integral of $f \in F(t, y)$ must be of the form $v + \sum_i c_i \log(u_i)$ if it is elementary, where the c_i's are constants and $v, u_i \in \overline{K}F(t, y)$. While partial fraction decompositions do not exist for algebraic functions, there exist tools, called *Puiseux series expansions* [3, 13] which replace them in the algebraic case. Those series are similar to the usual Taylor or Laurent series of rational funtions around their poles, except that the exponents are rational numbers instead of integers. Furthermore, the notion of a pole or zero of an algebraic function is well defined only on the so-called *Riemann surface associated with* $P(T, Y)$. There are algorithms for computing on Riemann surfaces and computing those series [3, 13, 43].

Let $P_1, \ldots, P_q$ be all the poles of the integrand on the Riemann surface (i.e. possibly including points at infinity). Risch [25, 26] proceeds by computing the Puiseux series expansion of the integrand at all the P_i's. He then shows that the terms of order less than -1 can be integrated piecewise, yielding the principal parts of the Puiseux series for v, and also bounds on the order of v at all its poles. Using the Bliss-Coates algorithm [3, 15], he gets a basis $\{b_1, \ldots, b_m\}$ over $\overline{K}$ for the vector space of all the functions in $\overline{K}F(t, y)$ having those bounds on their poles. Setting $v = a_1 b_1 + \ldots + a_m b_m$ where the a_i's are undetermined constants, and comparing the Puiseux expansions of this ansatz with the known expansions of v yields a system of linear algebraic equations for the a_i's. If it has no solution in $\overline{K}$ then f has no elementary integral, otherwise v is found (up to an additive constant).

The residues of the integrand at the P_i's can be computed from the coefficients of order -1 of its Puiseux series there. Risch computes then a basis $\{q_1, \ldots, q_r\}$ for the $\mathbf{Z}$-module generated by the residues, as well as integers $(n_{ij})_{i=1,\ldots,r}^{j=1,\ldots,q}$ such that if f has an elementary integral, then it must also have one of the form

$$\int f = v + \sum_{i=1}^{r} \frac{q_i}{e_i} \log(u_i) + \int f_0 \tag{8}$$

where $f_0 \in \overline{F}$, the e_i's are positive integers, and each $u_i \in \overline{K}F(t, y)$ has order exactly $e_i n_{ij}$ at each P_j and no other pole or zero outside the P_j's.

For a given $e_i > 0$, the Bliss-Coates algorithm decides whether there exists $u_i \in \overline{K}F(t, y)$ with the required orders, so the problem is reduced to find whether there exists $e_i > 0$ such that for that particular e_i a corresponding u_i exists. Risch [28] completes his algorithm by showing how reducing the algebraic curve $P(T, Y) = 0$ to one over a finite field produces bounds $B_1, \ldots, B_r$ such that if there is no appropriate u_i for $e_i = 1, \ldots, B_i$, then there is none also for $e_i > B_i$, in which case f has no elementary integral. Otherwise, the e_i's and u_i's of equation (8) are found, and setting

$$f_0 = f - v' - \sum_{i=1}^{r} \frac{q_i}{e_i} \frac{u_i{}'}{u_i}$$

we have either $f_0 \in \overline{F}$ and it can be integrated recursively, or $f_0 \notin \overline{F}$ and f has no elementary integral.

2.3 Modern variants

It can be seen from the previous sections that there are several serious computational and programming problems associated with the original Risch algorithm, in particular in the case of algebraic functions. Research on integration algorithms in the twenty years following the publication of Risch's algorithm has been directed either towards extending the algorithm to allow more general functions, or to make the algorithm more effective, avoiding unnecessary factorisations and algebraic extensions. We outline in this section the improvements made in that second direction. The algorithms described here are in fact the ones used today in the more sophisticated computer algebra systems.

2.3.1 Squarefree factorisation

The main tool that replaces factorisation of polynomials into irreducibles is the notion of a *squarefree factorisation*: let F be a field and $D \in F[X]$ a polynomial. We say that D is *squarefree* if D has no multiple irreducible factor, i.e. if all the zeros of D in $\overline{F}$ are distinct. If F has characteristic 0, this is equivalent to $gcd(D, dD/dX) = 1$. A *squarefree factorisation of D* is a factorisation $D = D_1 D_2^2 \ldots D_m^m$ where each D_i is squarefree and $\gcd(D_i, D_j) = 1$ for $i \neq j$. Such a factorisation can be computed as follows: let $R = \gcd(D, dD/dX)$, then $R = D_2 D_3^2 \ldots D_m^{m-1}$, so $D/R = D_1 D_2 \ldots D_m$, so $\gcd(R, D/R) = D_2 \ldots D_m$, so

$$D_1 = \frac{D/R}{\gcd(R, D/R)}.$$

We then recursively compute a squarefree factorisation of R in order to complete the one for D. Note that this is not the best algorithm for squarefree factorisation: Yun [44] has a faster variant. No matter which algorithm is used, computing a squarefree factorisation over any ground field F is not harder than computing gcd's of polynomials in $F[X]$. This is currently quite easier to program and faster than irreducible factorisation, in particular when F is a non-trivial extension of $\mathbf{Q}$.

2.3.2 The Hermite Reduction

Hermite [18] published the first major algorithmic improvement on integration more than a century ago. He proved that formulas (3), (4) and (5) are still valid when one uses a squarefree factorisation of the denominator $D = \prod_{i=1}^{n} D_i^i$ instead of the more expensive prime factorisation. So his algorithm is the same as we have presented in the base case, except that the irreducible P_i's are replaced by the squarefree D_i's. In the base case, this reduces the problem to integrating a fraction with a squarefree denominator, so this process is called the *Hermite reduction*. A little more than a century later, Rothstein [30] proved in his thesis that the Hermite reduction can also be applied to the logarithmic and exponential cases of the Risch algorithm, also reducing the integration problem to an integrand with a squarefree denominator. In the 1980's the focus turned to algebraic functions, and Trager [41] proved in his thesis that the Hermite reduction is applicable to pure algebraic functions (i.e. when the tower is of the form $K(x, y)$ where K is the constant field, x the integration variable and y an algebraic function). In that case, the Hermite reduction reduces the problem to integrating a function with only simple affine poles. Finally, Bronstein [6, 7] showed that the Hermite reduction is applicable to arbitrary elementary functions (i.e. to all the cases of the Risch algorithm), and even to some more general classes of transcendental functions.

The Hermite reduction was also generalized and applied to the Risch differential equation problem, yielding non-factoring algorithms for its resolution. Fast, Hermite-based algorithms for the various cases of this equation have been published in [5, 8, 16, 30] are are now used in computer algebra systems.

2.3.3 The Rothstein-Trager algorithm

The Hermite reduction leaves an integrand with a squarefree denominator. In the transcendental cases of the Risch algorithm, integrating such fractions required factoring this denominator over $\overline{K}$ or $\overline{KF}$ where K is the constant field. Such a factorisation is in general difficult to compute, and is in fact not implemented in some systems, like Mathematica 2.0, which causes them not to be able to integrate fractions with "difficult" denominators like $\int dx/(x^5 + 3x + 1)$. Trager [40] published the following algorithm which did not eliminate factoring completely, but reduced the degree of the polynomial to factor: given a rational function $f = A/D$ with $\deg(A) < \deg(D)$ $gcd(A, D) = 1$ and D squarefree,

- Let z be a new indeterminate and compute
$$R(z) = resultant_x(A - zD', D). \tag{9}$$
- Then,
$$\int h = \sum_{R(\alpha)=0} \alpha \log(\gcd(A - \alpha D', D)). \tag{10}$$

Of course, this algorithm still requires $R(z)$ to be factored over K, so that the gcd's inside the logarithms can be computed in $K(\alpha)[x]$ where α is any root of a given irreducible factor of R. Not only is this quite easier than factoring D over $\overline{K}$, but in addition $\deg_z(R) \le \deg_x(D)$ and it is possible for R to have multiple factors, while D is squarefree.

Trager later noticed that it is possible to avoid factoring entirely as follows: compute the resultant $R(z)$ using the subresultant PRS algorithm of Collins [14] and write $S_i(x, z)$ for the remainder of degree i in x appearing in the remainder sequence. Let $R = \prod_{i=1}^{m} R_i^i$ be a squarefree factorisation of R. Then,
$$\int h = \sum_{i=1}^{m} \sum_{R_i(\alpha)=0} \alpha \log(S_i(x, \alpha)). \tag{11}$$

If we are willing to return the integral in this form, no factorisation is required since the S_i's are computed in $K[x, z]$ during the resultant computation. While Trager implemented this algorithm in the AXIOM computer algebra system, he did not publish it, and it was independently discovered by Rioboo who published with Lazard in [21].

Rothstein [30] proved in his thesis that this algorithm can also be applied after the Hermite reduction in the logarithmic and exponential cases of the Risch algorithm, with the following modification: after computing $R(z)$ with formula (9) (using θ instead of x), then either one of the roots of $R(z)$ is not constant, in which case there is no elementary integral, or all its roots are constant, in which case substracting the derivative of the right-hand side of (10) from the integrand yields a new integrand in $\overline{K}F$ which is integrated recursively. Finally, Bronstein [7] proved that formulas (9) and (10) can be used in more general transcendental extensions. Formula (11) can be used in all these cases instead of (10).

All of those results together give a factor-free algorithm for integrating purely transcendental elementary functions, and this is essentially the algorithm implemented in the AXIOM and MAPLE computer algebra systems.

2.3.4 Algebraic functions

When algebraic functions are present in the integrand, the Hermite reduction yields a new integrand with only simple finite poles. Trager [41] gave a new resultant formula which generalized (9) to pure algebraic functions and returns a polynomial $R(z)$. There is however no known equivalent of formula (10) for algebraic functions, so it is currently still necessary to factor $R(z)$ over $\overline{K}$ in this case. Trager also describes algorithms for finding the logarithms which compute using only rational operations in the splitting field of R. Finally, Bronstein [6] proved that Trager's formula and most of his algorithms are applicable to arbitrary elementary functions. Both of those algorithms have been partially implemented in the AXIOM and MAPLE computer algebra systems. Currently, the REDUCE system has a partial implementation of the algorithm of [15] for integrating the pure algebraic functions which can be expressed by nested square roots.

3 Higher-order equations

We now return to the general problem of finding the Liouvillian solutions of the general linear ordinary differential equation (1) when the coefficients are themselves Liouvillian functions. Like the integration problem, this problem was extensively studied by mathematicians around the end of the 19^{th} century, mostly when the coefficients were polynomials. Here again, the methods used were very algorithmic, but while those techniques were successful in finding some specific classes of solutions [4, 32], researchers at that time did not have the necessary tools to prove that a given equation had no Liouvillian solutions, so they did not find a complete algorithm. It took the development of differential Galois theory in the 20^{th} century for a criterion similar to Liouville's Theorem to be discovered. Effective criteria with associated algorithms were published by Kovacic [20] for second order equations and Singer [34] for equations of arbitrary order in the 1980's. While the Kovacic algorithm has been partially implemented in several computer algebra systems [31, 39], removing the computational difficulties in those algorithms is the subject of very active current research. We outline Singer's algorithm and its recent improvements in the next sections. See [36] for a more complete history and description of other work on this type of equations.

3.1 Singer's Theorem

We now present the fundamental Theorem behind Singer's algorithm. Like Liouville's Theorem for integration, this result allows algorithms to search for one particular form of solution, and lets them prove that no Liouvillian solution exists if there is none of that particular form. For the same reason that we needed to formally define elementary extensions earlier, we must now define Liouvillian extensions, which can be seen as elementary extensions with arbitrary integrals allowed instead of just logarithms.

Let k be a differential field, K a differential extension of k, and $\theta \in K$. We say that θ is *Liouvillian over* k, if $k(\theta)$ and k have the same subfield of constants, and either:

(i) $\theta' = \eta$ for $\eta \in k$, in which case we say that θ is primitive over k, and write $\theta = \int \eta$,

(ii) $\theta' = \eta\theta$ for $\eta \in k$, in which case we say that θ is exponential over k, and write $\theta = \exp(\int \eta)$,

(iii) θ is algebraic over k.

A differential extension K of k is a *Liouvillian extension of* k, if there exist $\theta_1, \ldots, \theta_m \in K$ such that $K = k(\theta_1, \ldots, \theta_m)$ and θ_i is Liouvillian over $k(\theta_1, \ldots, \theta_{i-1})$ for $i = 1 \ldots m$.

We can now state Singer's Theorem:

Theorem 2 *There exists a function $I : \mathbf{N} \rightarrow \mathbf{N}$ such that: if F is a differential field of characteristic zero with an algebraically closed constant field and L a linear ordinary differential operator of order $n > 0$ with coefficients in F, then if there is an element y in some Liouvillian extension of F such that $L(y) = 0$, then there exists z such that z'/z is algebraic over F, $[F(z'/z) : F] \leq I(n)$ and $L(z) = 0$.*

This Theorem states that if equation (1) has a Liouvillian solution, then it must also have one of the form $z = e^{\int u}$ where u is algebraic over the field generated by the A_i's and there is a computable bound on the degree of u over that field. Of course, it does not state that every solution must be of that form. See [34, 35] for a proof.

3.2 The Singer algorithm

Singer [34] gives an explicit upper bound on the function $I(n)$, so his algorithm proceeds as follows: let F be the field generated by the coefficients A_i of equation (1) which is of order $n > 0$, and let K be the constant field of F. For $d = 1 \ldots I(n)$ look for a solution z such that z'/z has degree exactly d over $\overline{K}F$. If one is found, then use it to reduce the order of the equation and continue, otherwise go to the next d. If no solution has been found for $d \leq I(n)$, then Singer's Theorem guarantees that equation (1) has no Liouvillian solution.

This reduces the problem to finding a solution z such that z'/z has degree exactly $d > 0$ over $\overline{K}F$. Let $P = X^d + b_{d-1}X^{d-1} + \ldots + b_1 X + b_0 \in \overline{K}F[X]$ be the minimal irreducible polynomial for $u = z'/z$, and let $u_1, \ldots, u_d$ be all the zeros of P in $\overline{F}$. Write $z_i = e^{\int u_i}$. Singer notes first that

$$b_{d-1} = -\sum_{i=1}^{d} u_i = -\sum_{i=1}^{d} \frac{z_i{}'}{z_i} = -\frac{(z_1 \ldots z_d)'}{z_1 \ldots z_d}$$

and shows that one can construct a new ordinary linear differential operator L_d with coefficients in F such that $L_d(z_1 \ldots z_d) = 0$. Since $b_{d-1} \in \overline{K}F$, this implies that $L_d(y) = 0$ has a solution a such that $a'/a \in \overline{K}F$. He also shows that one can construct new ordinary linear differential operators $L_0, \ldots, L_{d-2}$ such that $L_i(b_i z_1 \ldots z_d) = 0$. Since

$$\frac{(b_i z_1 \ldots z_d)'}{b_i z_1 \ldots z_d} = \frac{b_i{}'}{b_i} - b_{d-1}$$

and $b_i, b_{d-1} \in \overline{K}F$, this implies that each $L_i(y) = 0$ has a solution a_i such that $a_i{}'/a_i \in \overline{K}F$. So the whole problem for a given degree d is now reduced to the case $d = 1$ (but at the cost of increasing the order of the equation to solve), i.e. given a linear ordinary differential equation $L(y) = 0$ with coefficients in F, find whether it has a solution z such that $z'/z \in \overline{K}F$. We call such a solution an *exponential solution* of $L(y) = 0$ in the rest of this paper.

The problem of finding the exponential solutions of $L(y) = 0$ was algorithmically solved in the last century in the case when the coefficients of L were rational functions [32]. Singer [37] recently gave an algorithm which is complete when the coefficient field of L is either an elementary or purely transcendental Liouvillian extension of the rational functions. Since this algorithm is currently beyond the capabilities of computer algebra systems, we describe only the base case ($F = K(x)$) which is where most of the recent improvements have been made. It turns out that it is necessary (and nowadays convenient) to search first for rational solutions (i.e. solutions in $\overline{K}(x)$) of equation (1) and then for exponential non-rational solutions. Following the same plan as for integration, we describe first Singer's algorithm and then the improvements.

3.2.1 Rational solutions

From now on, we are concerned only with equation (1) when the A_i's are polynomials over some constant field K. Furthermore we assume that $n > 0$ and $A_n \neq 0$. We make extensive use of the order function at a polynomial which is defined as follows: for $P, Q \in \overline{K}[x] \setminus \{0\}$ with $\deg(P) > 0$, the *order of Q at P*, denoted $\nu_P(Q)$ is the unique integer $m \geq 0$ such that $P^m \mid Q$ and $P^{m+1} \not\mid Q$. It can be easily computed by repeated exact divisions of Q by P. We also write $\mathrm{lc}(P)$ for the leading coefficient of a polynomial P.

The classical algorithm [32] proceed as follows: let $A_n = c\prod_{i=1}^m (x-\alpha_i)^{e_i}$ be a linear factorisation of A_n over $\overline{K}$. Then, any rational solution of (1) can be written in the form

$$y = \frac{R}{\prod_{i=1}^m (x-\alpha_i)^{d_i}} \tag{12}$$

where $R \in \overline{K}[x]$ and the d_i's are integer to be determined.

The classical way of computing d_i is via the construction of the *Newton polygon of (1) at $x-\alpha_i$*. We first construct the set of points

$$NP_{x-\alpha_i}(A_0,\ldots,A_n) = \{(j, \nu_{x-\alpha_i}(A_j) - j) \text{ for } j = 0\ldots n\}. \tag{13}$$

Let J_i be the set of indices j such that $\nu_{x-\alpha_i}(A_j)-j$ is minimal, i.e. $J_i = \{j \text{ s.t. } \nu_{x-\alpha_i}(A_j)-j = \min_{k=0}^n(\nu_{x-\alpha_i}(A_k)-k)\}$. It can then be shown that either $d_i = 0$ or $-d_i$ is a root of the polynomial

$$P_i(z) = \text{remainder of } \left(\sum_{j\in J_i} \frac{A_j}{(x-\alpha_i)^{\nu_{x-\alpha_i}(A_j)}}[z]_j\right) \text{ modulo } (x-\alpha_i) \tag{14}$$

where

$$[z]_j = z(z-1)\ldots(z-j+1).$$

Note that $P_i(z) \in \overline{K}[z]$ since the variable x is eliminated by the remainder operation.

So the algorithm proceeds by first factoring A_n over its splitting field, then computing $P_1,\ldots,P_m$ by the above formula. If P_i has no negative integer root, then we set $d_i = 0$. Otherwise we compute n_i = smallest negative integer root of P_i and set $d_i = -n_i$. This gives all the d_i's and the quantity

$$\mu = \sum_{i=1}^m d_i. \tag{15}$$

The degree of the numerator R can also be bounded by a similar process. We construct the set of points

$$NP_\infty(A_0,\ldots,A_n) = \{(j, \deg(A_j)-j) \text{ for } j = 0\ldots n\}.$$

Let J be the set of indices j such that $\deg(A_j)-j$ is maximal, i.e. $J = \{j \text{ s.t. } \deg(A_j)-j = \max_{k=0}^n(\deg(A_k)-k)\}$. It can then be shown that either $\deg(R) \geq \mu + n - 1$ or $\mu - \deg(R)$ is a root of the polynomial

$$P(z) = \sum_{j\in J}(-1)^j \mathrm{lc}(A_j)[z]^j \in K[z]$$

where

$$[z]^j = z(z+1)\ldots(z+j-1).$$

So the algorithm computes P as above and q = smallest negative integer root of P, and sets $d = \deg(R) = \mu + \max(n-1,-q)$. We then write $R = \sum_{j=0}^d a_j x^j$ where the a_j's are undetermined constants, and replace R by this expression in the right hand side of (12). Plugging the resulting expression in equation (1) and equating to 0 yields a homogeneous system of linear algebraic equations over $\overline{K}$ for the a_i's. Any basis for the solution space of this system yields a basis for the rational solutions of equation (1).

We note that Singer [34] pointed out that a factorisation of A_n into linear terms was not necessary, and that a prime factorisation $A_n = \prod_{i=1}^m Q_i^{e_i}$ of A_n over K was enough. In this case, formulas (12) and (13) remain valid with $x - \alpha_i$ replaced by Q_i, formula (14) becomes

$$P_i(z) = \text{resultant}_x(\text{remainder of } \left(\sum_{j\in J_i} \frac{A_j Q_i'^j}{Q_i^{\nu_{Q_i}(A_j)}}[z]_j\right) \text{ modulo } Q_i, Q_i) \tag{16}$$

and formula (15) becomes

$$\mu = \sum_{i=1}^m d_i \deg(Q_i). \tag{17}$$

3.2.2 The associated Riccati equation

We now turn to the search for the exponential solutions of (1). Since z'/z is a rational function when z is an exponential solution of equation (1), it is convenient to construct a new equation that z'/z must satisfy. This equation, called the *associated Riccati equation of (1)* is defined as follows: let u be a new differential indeterminate and define the sequence of differential polynomials $(P_i)_{i \geq 0}$ by

$$\left\{ \begin{array}{lcll} P_0 & = & 1 & \\ P_i & = & {P_{i-1}}' + uP_{i-1} & \text{for } i > 0. \end{array} \right.$$

i.e. $P_0 = 1, P_1 = u, P_2 = u' + u^2, P_3 = u'' + 3uu' + u^3$, etc...

The associated Riccati equation of (1) is the non-linear ordinary differential equation:

$$A_n(x)P_n + \ldots + A_1(x)P_1 + A_0(x)P_0 = 0. \tag{18}$$

Example: The associated Ricatti equation of $y'' = ry, r \in K(x)$ (the equation solved by Kovacic's algorithm) is $u' + u^2 = r$, which is a classical Ricatti equation.

The key property of the associated Ricatti equation, which is proven by substituting $e^{\int u}$ for y in equation (1) is: *if $z \neq 0$ is any solution of (1), then $u = z'/z$ is a solution of (18); conversely, if u is any solution of (18), then $z = e^{\int u}$ is a solution of (1).*

Since for an exponential solution z of (1), $u = z'/z \in \overline{K}(x)$, finding the exponential solutions of (1) is equivalent to finding the rational solutions of (18).

3.2.3 Exponential solutions

From the results of the previous section, we now look for the solutions u of (18) which are in $\overline{K}(x)$. Note that due to the non-linearity of the equation, there is in general a finite number of distinct solutions, and not a unique one or a linear vector space of solutions. For example $u' + u^2 = -1$ has the 2 solutions $\pm\sqrt{-1}$. Like for rational solutions, the classical algorithm computes first a factorisation $A_n = c\prod_{i=1}^{m}(x - \alpha_i)^{e_i}$ of A_n over $\overline{K}$. Then, any rational solution of (18) can be written in the form

$$u = P + \frac{Q'}{Q} + \sum_{i=1}^{m}\sum_{j=1}^{d_i} \frac{a_{ij}}{(x - \alpha_i)^j} \tag{19}$$

where $P, Q \in \overline{K}[x]$, $\gcd(Q, A_n) = 1$, the d_i's are integer to be determined and the a_{ij}'s are in $\overline{K}$.

The determination of the d_i's is done via a method quite similar to the Newton polygon. We first construct the set of integers

$$\Gamma_{x-\alpha_i}(A_0, \ldots, A_n) = \{\frac{\nu_{x-\alpha_i}(A_i) - \nu_{x-\alpha_i}(A_j)}{i - j} \text{ for } 1 \leq i \neq j \leq n\} \cap \mathbf{N} \setminus \{0\}. \tag{20}$$

It can then be shown that

$$d_i \leq \left\{ \begin{array}{cl} 1 & \text{if } \Gamma_{x-\alpha_i}(A_0, \ldots, A_n) = \emptyset \\ \max(\Gamma_{x-\alpha_i}(A_0, \ldots, A_n)) & \text{otherwise.} \end{array} \right. \tag{21}$$

The algorithm first sets d_i to the above upper bound. Let J_i be the set of indices j such that $\nu_{x-\alpha_i}(A_j) - jd_i$ is minimal, i.e. $J_i = \{j \text{ s.t. } \nu_{x-\alpha_i}(A_j) - jd_i = \min_{k=0}^{n}(\nu_{x-\alpha_i}(A_k) - kd_i)\}$. It can then be proven that a_{i,d_i} must be a root of the polynomial

$$P_i(z) = \text{remainder of } \left(\sum_{j \in J_i} \frac{A_j}{(x - \alpha_i)^{\nu_{x-\alpha_i}(A_j)}} S_{j,d_i}(z) \right) \text{ modulo } (x - \alpha_i) \tag{22}$$

where

$$S_{j,d}(z) = \left\{ \begin{array}{ll} z^j & \text{if } d > 1 \\ [z]_j & \text{if } d = 1. \end{array} \right. \tag{23}$$

Here also, $P_i(z) \in \overline{K}[z]$ because of the remainder operation.

So the algorithm computes P_i as above, then for each root $\beta \in \overline{K}$ of P_i it sets $\gamma = \beta/(x-\alpha_i)^{d_i}$ and performs the change of variable $\tilde{L}(y) = L(ye^{\int \gamma})/e^{\int \gamma}$. This yields a new Ricatti equation for which we replace the bound d_i by $d_i - 1$ and look for the potential a_{i,d_i-1} as above. This process eventually returns a finite set of fractions $\{u_1, \ldots, u_q\}$ such that any solution of (18) in $\overline{K}[x]$ must be of the form $u = u_k + P + Q'/Q$ for some k with P and Q as in (19).

A finite set of candidates for P can be found by a similar process. We construct the set of integers

$$\Gamma_\infty(A_0, \ldots, A_n) = \{\frac{\deg(A_i) - \deg(A_j)}{i-j} \text{ for } 1 \le i \ne j \le n\} \cap \mathbf{N} \setminus \{0\}.$$

It can then be shown that

$$\deg(P) \le \begin{cases} 0 & \text{if } \Gamma_\infty(A_0, \ldots, A_n) = \emptyset \\ \max(\Gamma_\infty(A_0, \ldots, A_n)) & \text{otherwise.} \end{cases}$$

Set d to be the above upper bound, and let J be the set of indices j such that $\deg(A_j) + jd$ is maximal i.e. $J = \{j \text{ s.t. } \deg(A_j) + jd = \max_{k=0}^n (\deg(A_k) + kd)\}$. It can then be proven that $\mathrm{lc}(P)$ must be a root of the polynomial

$$P(z) = \sum_{j \in J} \mathrm{lc}(A_j) z^j \in K[z].$$

So the algorithm computes P as above, then for each root $\beta \in \overline{K}$ of P it performs the change of variable $\tilde{L}(y) = L(ye^{\int \beta x^d})/e^{\int \beta x^d}$. This yields a new Ricatti equation for which we replace the bound d by $d-1$ and look for the potential coefficients of x^{d-1} as above. This process eventually returns a finite set of candidates $\{P_1, \ldots, P_r\}$ for the polynomial part P of (19).

Combining the potential polynomial parts with the potential singular parts found before, we now have a finite number of fractions $\{f_1, \ldots, f_s\}$ such that any solution $u \in \overline{K}[x]$ of (18) must be of the form $u = f_k + Q'/Q$ for some k and a polynomial $Q \in \overline{K}[x]$. For each f_k we perform the change of variable $\tilde{L}(y) = L(ye^{\int f_k})/e^{\int f_k}$. This yields a new Ricatti equation for which we look for solutions of the form $u = Q'/Q$. But $Q = e^{\int u}$ is then a polynomial solution of $\tilde{L}(y) = 0$ so the potential Q's can be found by looking for the rational solutions of $\tilde{L}(y) = 0$ with the algorithm of the previous section. This completes the search for all the exponential solutions of equation (1).

Here also, Singer [37] pointed out that a factorisation of A_n into linear terms was not necessary, and that a prime factorisation $A_n = \prod_{i=1}^m Q_i^{e_i}$ of A_n over K was enough. In this case, formula (19) remains valid with $x - \alpha_i$ replaced by Q_i except that we now have $a_{ij} \in \overline{K}[x]$ and $\deg(a_{ij}) < \deg(Q_i)$. Formulas (20), (21) and (22) remain valid with $x - \alpha_i$ replaced by Q_i, while formula (23) becomes

$$S_{j,d}(z) = \begin{cases} z^j & \text{if } d > 1 \\ \prod_{k=0}^{j-1}(z - kQ_j{}') & \text{if } d = 1. \end{cases} \tag{24}$$

A new problem occurs in that we now have $P_i(x,z) \in K[x,z]$ instead of $K[z]$ as earlier, and instead of finding the roots of P_i in $\overline{K}$, we must now find all the polynomials $a_{ij} \in \overline{K}[x]$ such that $\deg(a_{ij}) < \deg(Q_i)$ and

$$P_i(x, a_{ij}(x)) \equiv 0 \pmod{Q_i}. \tag{25}$$

Singer shows that this equation can be solved by setting β to be an arbitrary root of Q_i and then factoring $P_i(\beta, z)$ over $K(\beta)$. All the potential solutions in $\overline{K}[x]$ of degree less than $\deg(Q_i)$ of equation (25) must be of the form $a(x)$ where $z - a(\beta)$ divides $P_i(\beta, z)$ in $K(\beta)[z]$.

3.3 Recent improvements

Like for the Risch algorithm, there are also computational and programming problems associated with Singer's algorithm, and in fact it has not been implemented even in the order 2 case. There are two separate bottlenecks with this algorithm: (i) Singer's upper bound on the algebraic degree of z'/z is too large to be used in practice, and (ii) the algorithm still makes unnecessary factorisations of polynomials. We outline in this section the improvements made in the past few years in those directions, improvements which are being used in the next wave of differential equation solvers in computer algebra systems.

3.3.1 Bounding the algebraic degree

Singer's [34] upper bound for the function $I(n)$ of Theorem 2 is

$$I(n) \leq \max((\sqrt{8n}+1)^{2n^2} - (\sqrt{8n}-1)^{2n^2}, n!I(n-1))$$

and he mentions there that this bound is certainly not optimal (it shows however that the problem is decidable in a finite number of steps). In the second order case ($n = 2$), Kovacic [20] gave optimal bounds in an algorithm based on the same principles, and his algorithm has been implemented, with some difficulties arising from the need to factor polynomials linearly over the complex numbers.

There have been considerable improvements made in the past 10 years towards improving this bound. Ulmer [42] showed that computing an optimal bound $\mathcal{F}(n)$ for the general case reduces to a problem in group theory that is solvable at least for $n \leq 11$, and used this to compute $\mathcal{F}(2)$ and $\mathcal{F}(3)$. Those are small enough to make Singer's algorithm effective for second and third order equations. Finally, Singer and Ulmer [38] gave the list of possible degrees for z'/z between 1 and $\mathcal{F}(n)$ for $n = 2$ and $n = 3$. This improves the algorithm even further since few from the integers between 1 and $\mathcal{F}(n)$ need to be checked. They also give necessary conditions for some of those degrees to actually correspond to solutions. It should be noted that those improvements all assume that the differential equation to solve is irreducible, which means that it is necessary to factor the equation before using those bounds. But factoring such equations can be reduced to finding the rational and exponential solutions of new equations of higher order [32, 33], so it can be done, by the algorithms of the previous (or next) section, before the algebraic solutions are looked for.

3.3.2 Balanced factorisation

We now turn to the problem of avoiding unnecessary factorisations. We have seen earlier that squarefree factorisations can be used instead of prime factorisations for the integration problem. The corresponding tool for linear differential equations is Abramov's *balanced factorisation* [1]: let F be a field and $A, B \in F[X]$ polynomials. We say that A is *balanced w.r.t* B if any 2 irreducible factors of A appear with the same exponent in B, in other words if $\nu_P(B) = \nu_Q(B)$ for any 2 irreducible factors P and Q of A. Let $\mathcal{S} \subseteq F[X]$ be a set of polynomials. We say that A is *balanced w.r.t* $\mathcal{S}$ if it is balanced w.r.t. any $B \in \mathcal{S}$. A *balanced factorisation of* A *w.r.t.* $\mathcal{S}$ is a factorisation $A = A_1^{e_1} A_2^{e_2} \dots A_m^{e_m}$ where each A_i is squarefree and balanced w.r.t. $\mathcal{S}$, and $\gcd(A_i, A_j) = 1$ for $i \neq j$. When $\mathcal{S}$ is finite, such a factorisation can be computed as follows: suppose first that A is squarefree, the following algorithm computes a balanced factorisation of A w.r.t. any polynomial B.

balanced_fact_sqfr(A, B)
INPUT: $A \in F[X] \setminus \{0\}$ monic and squarefree, $B \in F[X]$.
OUTPUT: $A = A_1 \dots A_m$, a balanced factorisation of A w.r.t B.

∘ **if** $\deg(B) = 0$ **then** return A

∘ $G \leftarrow \gcd(A, B)$

∘ **if** $\deg(G) = 0$ **then** return A

∘ $G_1 \dots G_m \leftarrow$ **balanced_fact_sqfr**$(G, B/G^{\nu_G(B)})$

∘ return $(A/G)G_1 \dots G_m$

For an arbitrary A and finite $\mathcal{S}$, we compute first a squarefree factorisation $A = A_1 A_2^2 \dots A_m^m$. If $\mathcal{S} = \emptyset$ then we are done, otherwise pick some $B \in \mathcal{S}$ and replace each A_i by a balanced factorisation of A_i w.r.t. B (obtained by the above algorithm). This yields a balanced factorisation $A = C_1^{e_1} \dots C_q^{e_q}$ of A w.r.t. B. Recursively replace each C_i by a balanced factorisation of C_i w.r.t. $\mathcal{S} \setminus \{B\}$. This yields a complete balanced factorisation of A w.r.t. $\mathcal{S}$.

3.3.3 Rational solutions

Abramov [1] proved that formulas (12), (13), (16) and (17) are still valid when one uses a balanced factorisation of the leading coefficient $A_n = \prod_{i=1}^m C_i^{e_i}$ with respect to $\{A_0, \ldots, A_n\}$ instead of the more expensive linear or prime factorisations. So his algorithm is the same as the search for the rational solutions in the base case of Singer's algorithm, except that the linear $x - \alpha_i$'s and the irreducible P_i's are replaced by the balanced factors C_i's. This reduces the problem to finding the numerator R of the ansatz (12) and this requires no factorisation. Abramov and Kvashenko [2] further improved the efficiency of this algorithm by showing that one can avoid solving a potentially large system of linear algebraic equations over K for the coefficients of R, by computing those coefficients in smaller groups. They implemented their improved algorithm in the REDUCE computer algebra system. Recently, Bronstein [9] showed that balanced factorisation can be used to find the denominator of the in-field solutions when the coefficients are arbitrary Liouvillian functions, and also for some more general classes of transcendental functions, and implemented it in the AXIOM computer algebra system.

3.3.4 Exponential solutions

From the above improvements, the only factorisation steps remaining are found in the search for non-rational exponential solutions. Due to the non-linearity of the problem, it is sometimes required to extend the initial constant field by new algebraic numbers in order to express the solutions, so some amount of factorisation remains necessary. The research efforts here are being directed to minimize the amount of factorisations to be done. Bronstein [9] has shown that balanced factorisation can also be used for solving Ricatti equations and that formulas (19), (20), (21), (22) and (24) remain valid when one uses a balanced factorisation of the leading coefficient $A_n = \prod_{i=1}^m C_i^{e_i}$ with respect to $\{A_0, \ldots, A_n\}$ instead of the more expensive linear or prime factorisations. So the search for the exponential solutions in the base case of Singer's algorithm can be used, replacing the linear $x - \alpha_i$'s and the irreducible Q_i's by the balanced factors C_i's.

A new problem that appears when using balanced factorisation is that the modulus C_i in equation (25) is not irreducible anymore, so $K(\beta)[z]$ when β is a root of C_i is not a unique factorisation domain. One solution is to factor C_i into irreducibles and use Singer's algebraic factoring method [37]. This requires an implementation of factoring polynomial over an algebraic extension of K (e.g. as in [40]). Another approach used in [10] is to set $a_{ij} = c_d x^d + \ldots c_1 x + c_0$ where $d = \deg(C_i)$ and the c_i's are undetermined constants. Plugging this for a_{ij} in equation (25) and equating to 0 yields a system of non-linear algebraic equations for the c_i's (in the base case). This system can then be solved using the Gröbner basis method or resultants. This method has been implemented in the AXIOM computer algebra system and is highly successful on numerous examples from [19]. Whether it is faster than the algebraic factoring approach probably depends on the actual constant field K and remains to be investigated.

4 Final comments

We have presented here an overview of the algorithms that form the backbone of the integration and differential equations facilities found in today's advanced computer algebra systems. There are of course numerous related extensions which have not been mentioned here: extending the Risch algorithm to handle special functions, finding the solutions of linear ordinary differential equations in other type of closed forms (for example series), solving other types of differential equations or systems of them, etc... All of those problems have been studied in connection with computer algebra, many software packages for them have been successfully implemented, and several of them will become part of the standard distributed libraries of computer algebra systems in the coming years (they are currently prototypes running in one particular system). We recommend the excellent survey [36] by Singer and its exhaustive bibliography to the interested or curious reader.

References

[1] S.A. Abramov, *Rational Solutions of Linear Differential and Difference Equations with Polynomial Coefficients (in russian)*, Journal of Computational Mathematics and Mathematical Physics **29**, No.11, 1611–1620, 1989.

[2] S.A. Abramov & K. Yu. Kvashenko, *Fast Algorithms to Search for the Rational Solutions of Linear Differential Equations with Polynomial Coefficients*, in "Proceedings of ISSAC'91", Bonn, BRD, ACM Press, 267–270, 1991.

[3] G.A. Bliss, *Algebraic Functions*, Dover Publications, New York, 1966.

[4] A. Boulanger, *Contributions à l'étude des équations différentielles linéaires homogènes intégrables algébriquement*, Journal de l'Ecole Polytechnique (2^{eme} serie) **4**, 1–122, 1898.

[5] M. Bronstein, *The Transcendental Risch Differential Equation*, Journal of Symbolic Computation **9**, No.1, 49–60, 1990.

[6] M. Bronstein, *Integration of Elementary Functions*, Journal of Symbolic Computation **9**, No.2, 117–173, 1990.

[7] M. Bronstein, *A Unification of Liouvillian Extensions*, Applicable Algebra in Engineering, Communication and Computing **1**, No.1, 5–24, 1990.

[8] M. Bronstein, *The Risch Differential Equation on an Algebraic Curve*, in "Proceedings of ISSAC'91", Bonn, BRD, ACM Press, 241–246, 1991.

[9] M. Bronstein, *On Solutions of Linear Ordinary Differential Equations in their Coefficient Field*, Journal of Symbolic Computation **13**, No.4, 413–439, 1992.

[10] M. Bronstein, *Linear Ordinary Differential Equations: breaking through the order 2 barrier*, in "Proceedings of ISSAC'92", Berkeley, USA, ACM Press, 42–48, 1992.

[11] P.L. Chebyshev, *Sur l'intégration des différentielles qui contiennent une racine carrée d'un polynôme du troisième ou du quatrième degré*, Journal de Mathématiques Pures et Appliquées (2^{eme} série) **2**, 1–42, 1857.

[12] P.L. Chebyshev, *Sur l'intégration de la différentielle* $(x + A)dx/\sqrt{x^4 + ax^3 + bx^2 + cx + d}$, Journal de Mathématiques Pures et Appliquées (2^{eme} série) **9**, 225–246, 1864.

[13] C. Chevalley, *Algebraic Functions of One Variable*, Math. Surveys VI, AMS, New York, 1951.

[14] G.E. Collins, *Subresultants and reduced polynomial remainder sequences*, Journal of the ACM **14**, 128–142, 1967.

[15] J.H. Davenport, *On the Integration of Algebraic Functions*, Lecture Notes in Computer Science, 102, Springer-verlag, New-York, 1981.

[16] J.H. Davenport, *The Risch Differential Equation Problem*, SIAM Journal on Computing **15**, No.4, 903–906, 1986.

[17] G.H. Hardy, *The Integration of Functions of a Single Variable*, Cambridge U. Press, Cambridge, England, 1916.

[18] E. Hermite, *Sur l'intégration des fractions rationelles*, Nouvelles Annales de Mathématiques (2^{eme} série) **11**, 145–148, 1872.

[19] E. Kamke, *Differentialgleichungen, Lösungmethoden und Lösungen, I. Gewöhnliche Differentialgleichungen*, Akad. Verlag., Leipzig, 1959.

[20] J.J. Kovacic, *An Algorithm for Solving Second Order Linear Homogeneous Differential Equations*, Journal of Symbolic Computation **2**, No.1, 3–43, 1986.

[21] D. Lazard & R. Rioboo *Integration of rational functions, Rational computation of the logarithmic part*, Journal of Symbolic Computation **9**, No.2, 113–116, 1990.

[22] J. Liouville *Premier mémoire sur la détermination des intégrales dont la valeur est algébrique*, Journal de l'Ecole Polytechnique **14**, cahier 22, 124–148, 1833.

[23] J. Liouville *Second mémoire sur la détermination des intégrales dont la valeur est algébrique*, Journal de l'Ecole Polytechnique **14**, cahier 22, 149–193, 1833.

[24] J. Moses, *Symbolic Integration: the Stormy Decade*, Communications of the ACM **14**, 548–560, 1971.

[25] R. Risch, *On the Integration of Elementary Functions which are built up using Algebraic Operations*, Report SP-2801/002/00, System Development Corp., Santa Monica, CA, 1968.

[26] R. Risch, *Further Results on Elementary Functions*, Research Report RC-2402, IBM Corp., Yorktown Heights, NY, 1969.

[27] R. Risch, *The Problem of Integration In Finite Terms*, Transactions of the American Mathematical Society **139**, 167–189, 1969.

[28] R. Risch, *The Solution of the Problem of Integration in Finite Terms*, Bulletin of the American Mathematical Society **76**, 605–608, 1970.

[29] M. Rosenlicht, *Integration in Finite Terms*, American Mathematical Monthly **79**, 963–972, 1972.

[30] M. Rothstein, *A New Algorithm for the Integration of Exponential and Logarithmic Functions*, in "Proceedings of 1977 MACSYMA Users Conference", NASA Pub. CP-2010, 263–274, 1977.

[31] B.D. Saunders, *An Implementation of Kovacic's Algorithm for Solving Second Order Linear HomogeneousDifferential Equations*, in "Proceedings of SYMSAC'81", ACM Press, New York, 1981.

[32] L. Schlesinger, *Handbuch der Theorie der linearen Differentialgleichungen*, 3 vols., Teubner, Leipzig, 1895,97,98.

[33] F. Schwarz, *A Factorization Algorithm for Linear Ordinary Differential Equations*, in "Proceedings of ISSAC'89", Portland, Oregon, ACM Press, 17–25, 1989.

[34] M.F. Singer, *Liouvillian Solutions of* n^{th} *Order Homogeneous Linear Differential Equations*, American Journal of Mathematics **103**, No.4, 661–682, 1981.

[35] M.F. Singer, *An Outline of Differential Galois Theory*, in "Computer Algebra and Differential Equations", E.Tournier Ed., Academic Press, 1989.

[36] M.F. Singer, *Formal Solutions of Differential Equations*, Journal of Symbolic Computation **10**, No.1, 59–94, 1990.

[37] M.F. Singer, *Liouvillian Solutions of Linear Differential Equations with Liouvillian Coefficients*, Journal of Symbolic Computation **11**, 251–273, 1991.

[38] M.F. Singer & F. Ulmer, *Bounds and Necessary Conditions for Liouvillian Solutions of (Third Order) Linear Differential Equations*, manuscript, 1991.

[39] C. Smith, *A Discussion and Implementation of Kovacic's Algorithm for Ordinary Differential Equations*, Res. Report CS 84-35, Dpt. of Computer Science, Univ. of Waterloo, Ontario, 1984.

[40] B. Trager, *Algebraic Factoring and Rational Function Integration*, in "Proceedings of SYMSAC'76", ACM Press, 219–226, 1976.

[41] B. Trager, *Integration of Algebraic Functions*, Ph.D. thesis, Dpt. of EECS, Massachusetts Institute of Technology, 1984.

[42] F. Ulmer, *On Liouvillian Solutions of Linear Differential Equations*, Applicable Algebra in Engineering, Communication and Computing **2**, No.3, 171–193, 1992.

[43] R.J. Walker, *Algebraic Curves*, Springer-Verlag, New York, 1978.

[44] D.Y.Y. Yun, *On Square-Free Decomposition Algorithms*, in “Proceedings of SYMSAC’76”, ACM Press, 26–35, 1976.