J. Symbolic Computation (1997) **11**, 1–000

Formal Solutions and Factorization of Differential Operators with Power Series Coefficients

Mark van Hoeij

Department of mathematics
University of Nijmegen
6525 ED Nijmegen
The Netherlands
e-mail: hoeij@sci.kun.nl

(Received December 12, 1995)

The topic of this paper is formal solutions of linear differential equations with formal power series coefficients. The method proposed for computing these solutions is based on factorization of differential operators. The notion of exponential parts is introduced to give a description of factorization properties and to characterize the formal solutions. The algorithms will be described and their implementation is available.

1. Introduction

Factorization of differential operators is a powerful computer algebra tool for handling ordinary linear differential equations. It can be applied to compute formal solutions and to study the structure of a differential equation. A differential equation

$$y^{(n)} + a_{n-1}y^{(n-1)} + \ldots + a_1 y' + a_0 y = 0$$

corresponds to a differential operator

$$f = \partial^n + a_{n-1}\partial^{n-1} + \ldots + a_0\partial^0$$

acting on y. Here the coefficients a_i are elements of the differential field $k((x))$ and ∂ is the differentiation d/dx. The field k is the field of constants. It is assumed to have characteristic 0. The differential operator f is an element of the non-commutative ring $k((x))[\partial]$. This is an example of an Ore ring (Ore, 1933).

Sections 6 and 8 contain the main results of this paper. These results are expressed using the notion of exponential parts. The exponential parts will be studied in section 6 from the viewpoint of factorization, and in section 8 from the viewpoint of formal solutions. They form the key ingredient for our factorization algorithm for $k(x)[\partial]$ in chapter 3 in (van Hoeij, thesis). Another application is found in section 9. Here the question is: when is a given vector space a solution space of a certain differential operator. This question can easily be answered using the direct sum splitting in section 8.

0747–7171/90/000000 + 00 $03.00/0

The algorithms in this paper are given in sections 4, 5 and 8.4. From an algorithmic point of view the factorization in $k((x))[\partial]$ is the central problem because the other algorithms in this paper require this tool. We will discuss it in the rest of this section.

Note that in general elements of $k((x))$ consist of infinitely many terms. Only a finite number of them can be computed. This means that a factorization can only be determined up to some finite accuracy. The notion of accuracy will be formalized later. Increasing the accuracy of a factorization will be called *lifting* a factorization.

From (Malgrange, 1979) we know that an element of $k((x))[\partial]$ which has only 1 slope in the Newton polygon (cf. section 3.3) and which has an irreducible Newton polynomial (cf. section 3.4) is irreducible in $k((x))[\partial]$. In (Malgrange, 1979) Malgrange shows that in the following two cases a differential operator $f \in k((x))[\partial]$ is reducible in this ring and how a factorization can be computed:

1 An operator with a broken Newton polygon (i.e. more than 1 slope).
2 An operator with one slope > 0 where the Newton polynomial is reducible and not a power of an irreducible polynomial.

In our method these two cases of factorization and the factorization of regular singular operators are called *coprime index 1 factorizations.* Coprime index 1 means that the factorization can be lifted by the usual Hensel lifting (cf. any book on computer algebra) procedure. For a definition of the coprime index see section 2.

Example:

$$f = \partial^4 + \frac{1}{x^2}\partial^3 + \frac{2}{x^4}\partial^2 + \frac{1}{x^6}\partial + \frac{1}{x^8}.$$

The Newton polynomial is $T^4+T^3+2T^2+T+1$. This polynomial can be factored over $\mathbb{Q}$ as $(T^2+1)(T^2+T+1)$. Because T^2+1 and T^2+T+1 in $\mathbb{Q}[T]$ are coprime (i.e. the gcd is 1) we can conclude from (Malgrange, 1979) that f is reducible in $\mathbb{Q}((x))[\partial]$. A factorization of $f = LR$ is obtained in two steps. The first step is to compute the factorization up to accuracy 1 (definitions follow later, this integer 1 is related to the coprime index). This accuracy is obtained when we have the Newton polynomials T^2+1 and T^2+T+1 of L and R (here T^2+1 and T^2+T+1 can be interchanged to obtain a different factorization). The next step is to lift the factorization up to the desired accuracy. Because T^2+1 and T^2+T+1 are coprime this lifting can be done by the usual Hensel lifting procedure. In each lift step the extended Euclidean algorithm is used. Note that in this example the reducibility of f can be concluded from very few coefficients of f in k; the coefficients which determine the Newton polynomial are sufficient.

Now there remains one hard case of factorization in $k((x))[\partial]$. Here f has one slope $s \neq 0$ and the Newton polynomial is of the form P^d, where P is an irreducible polynomial over k and d is an integer > 1. In this case it is more difficult to decide if f is reducible or not. A factorization of f will have coprime index > 1.

Example:

$$f = \partial^4 + \frac{2+x^4}{x^4}\partial^2 - \frac{8}{x^5}\partial + \frac{1+20x^2}{x^8}.$$

The Newton polynomial of f is $T^4+2T^2+1 = (T^2+1)(T^2+1)$. Because the two factors T^2+1 and T^2+1 are not coprime we can not apply Hensel lifting to find a factorization over $\mathbb{Q}((x))[\partial]$. Malgrange provides a factorization method in $\overline{\mathbb{Q}((x))}[\partial]$ for this case. We want to find a factorization in $\mathbb{Q}((x))[\partial]$. In this example f is reducible in $\mathbb{Q}((x))[\partial]$.

However, $f + 1/x^6$ (replace the coefficient 20 by 21) is irreducible in $\mathbb{Q}((x))[\partial]$. In the previous example adding $1/x^6$ would have no influence on the reducibility of f because the reducibility could already be decided from the Newton polynomial. We see that this example is more complicated because more coefficients of f are relevant for deciding reducibility. We shall proceed as follows:

- Compute a first order right-hand factor $\partial - r$ of f where $r \in \overline{k((x))}$. We use a variant on the method in (Malgrange, 1979) for this.
- Compute an operator $R \in k((x))[\partial]$ of minimal order such that $\partial - r$ is a right-hand factor of R.
- Perform a division to find a factorization $f = LR$.

For some applications, like factorization in $k(x)[\partial]$, we need to compute the factors L and R up to a high accuracy. The method sketched for computing L and R is not very suitable for this because it is slow. We will use this slow method to compute L and R up to a certain accuracy (up to the coprime index) and then use a different method to lift the factorization. Coprime index > 1 means that the usual Hensel lifting does not work because the Newton polynomials of L and R have gcd $\neq 1$. For this case we give a variant on the Hensel lifting method in section 4.

The factorization of a differential operator f is done recursively. If f can be factored $f = LR$ then the factorization algorithm is applied to the factors L and R (or only to R when we are only interested in right-hand factors) until f is factored in irreducible factors. This causes a difficulty; if a factorization is required with a given accuracy it is not clear how accurate the intermediate factorizations should be. To solve this problem we use *lazy evaluation* in our implementation. This is a computer algebra trick which makes exact computation in $k((x))$ possible. It does not use truncations of some finite accuracy. Instead, an expression $a \in k((x))$ is denoted as the name and arguments of a procedure that computes coefficients of a. These coefficients are automatically computed and stored when they are needed. This method of computing in $k((x))$ is very efficient because coefficients which are not used will not be computed.

The use of factorization for computing formal solutions is beneficial for the efficiency in case the solutions involve algebraic extensions, c.f. the comments after algorithm formal solutions in section 8.4.

2. Valuations and the coprime index

A discrete *valuation* on a ring D is a map $v : D \to \mathbb{Z} \bigcup \{\infty\}$ such that for all a and b in D we have: $v(ab) = v(a) + v(b)$, $v(a+b) \geq \min(v(a), v(b))$ and $v(a+b) = \min(v(a), v(b))$ if $v(a) \neq v(b)$. $v(0) = \infty$. An example: D is the field of p-adic numbers $\mathbb{Q}_p$ or D is a polynomial ring $\mathbb{Q}_p[x]$ over the p-adic numbers. Define the valuation $v(a)$ of $a \in \mathbb{Q}_p[x]$ as the largest integer n such that $a \in p^n\mathbb{Z}_p[x]$. Another example: $s \in \mathbb{Q}$ and $D = k((x))[y]$ where k is a field. Write $s = n/d$ where n and d are integers, $\gcd(n, d) = 1$ and $d > 0$. Now the valuation $v_s(\sum_{i,j} a_{i,j} x^i y^j)$ is defined as the minimum $id - jn$ for which $a_{i,j} \neq 0$.

A third example: k is a field, $s \in \mathbb{Q}$, $s \geq 0$ and $D = k((x))[\delta]$. Here δ is defined as $x\partial \in k((x))[\partial]$, cf. section 3.2. Write $s = n/d$ where n and d are integers, $\gcd(n, d) = 1$ and $d > 0$. Now the valuation $v_s(\sum_{i,j} a_{i,j} x^i \delta^j)$ is defined as the minimum $id - jn$ for which $a_{i,j} \neq 0$.

A *filtered ring* is a ring D with a chain of additive subgroups $\cdots \supset D_{-1} \supset D_0 \supset D_1 \cdots$

such that: $1 \in D_0$, $D = \bigcup_{n \in \mathbb{Z}} D_n$ and $D_n D_m \subset D_{n+m}$ for all integers n and m. The chain $(D_n)_{n \in \mathbb{Z}}$ is called a *filtration* of D. The *associated graded ring* $\mathrm{gr}D$ is defined as $\oplus_n D_n / D_{n+1}$. The symbol map $\sigma : D \to \mathrm{gr}D$ is defined as: $\sigma(0) = 0$, $\sigma(f) = f + D_{n+1}$ if $f \in D_n \setminus D_{n+1}$. For more information about filtrations see (Björk J.E, 1979). A valuation v defines a filtration on a ring D as follows

$$D_n = \{f \in D | v(f) \geq n\}.$$

For positive integers a the set D_0/D_a has the structure of a ring.

For a ring D with a valuation v we can define a *truncation* σ_a with *accuracy* a for non-zero elements f of D and positive integers a as follows

$$\sigma_a(f) = f + D_{v(f)+a} \in D_{v(f)}/D_{v(f)+a}.$$

The symbol map is σ_1.

Suppose $f \in D$ can be written as $f = LR$ where $L, R \in D$. For invertible elements $u \in D$ we have $f = LR = (Lu)(u^{-1}R)$. We will call the ordered pair L, R equivalent with the pair $Lu, u^{-1}R$. Let t be a positive integer. Then the ordered pair L, R is called *coprime* with *index* t if for all $a \geq t$ the pair $\sigma_{a+1}(L), \sigma_{a+1}(R)$ is determined up to the above equivalence by $\sigma_a(L)$, $\sigma_a(R)$ and $\sigma_{a+t}(f)$. Assume t is minimal, then t is called the *coprime index* of L, R. If L, R is not coprime for any integer t then the coprime index is ∞.

For our examples $\mathbb{Q}_p[x]$, $k((x))[y]$ and $k((x))[\delta]$ the notion of equivalence for pairs L, R can be avoided by restricting ourselves to monic elements f, L and R. Then we can define the coprime index of the factorization $f = LR$ as the smallest positive integer t for which the following holds: For all integers $a \geq t$ and monic elements L' and R' of D, if

$$\sigma_a(L') = \sigma_a(L) \quad \text{and} \quad \sigma_a(R') = \sigma_a(R) \quad \text{and} \quad \sigma_{a+t}(L'R') = \sigma_{a+t}(f)$$

then

$$\sigma_{a+1}(L') = \sigma_{a+1}(L) \quad \text{and} \quad \sigma_{a+1}(R') = \sigma_{a+1}(R).$$

Example: Suppose we want to factor $f = x^2 + x + 3 \in D = \mathbb{Q}_3[x]$. First we look at the truncation $\sigma_1(f) = x^2 + x \in D_0/D_1$ which factors as $x(x+1) \in D_0/D_1$. Because x and $x+1$ have gcd 1 in $D_0/D_1 \simeq F_3[x]$ we can apply Hensel lifting to find a factorization $f = LR$ in D. To determine L and R up to some accuracy a we only need to know f up to accuracy a. So the coprime index is 1 in this example.

Example: $f_1 = x^4 - x^2 - 2 = L_1 R_1 = (x^2+1)(x^2-2) \in \mathbb{Q}_3[x]$ and $f_2 = x^4 - x^2 - 20 = L_2 R_2 = (x^2+4)(x^2-5) \in \mathbb{Q}_3[x]$. Now f_1 and f_2 are the same up to accuracy 2 (i.e. are congruent modulo 3^2) but the factorizations L_1, R_1 and L_2, R_2 are different up to this accuracy. It follows that to determine the factorization of f_1 up to some accuracy a it is not sufficient to know $\sigma_a(f_1)$. This means that the coprime index of L_1, R_1 is > 1. We cannot apply ordinary Hensel lifting to find a factorization of f_1 because $\sigma_1(L_1)$ and $\sigma_1(R_1)$ have gcd $\neq 1$.

The name coprime index is explained from the case $k((x))[y]$. In this ring L, R have finite coprime index if and only if L and R are coprime in the usual sense (i.e. $\gcd(L, R) = 1$). It is shown in chapter 5 in (van Hoeij, thesis) that the coprime index of a factorization $f = LR$ in $k((x))[\delta]$ is always finite.

3. Preliminaries

This section summarizes the concepts and notations we will use in this paper. Definitions will be brief; references to more detailed descriptions are given.

3.1. The field $k((x))$

k is a field of characteristic 0, $\overline{k}$ is its algebraic closure. $k((x))$ is the field of formal Laurent series in x with finite pole order and coefficients in k. $\overline{k((x))}$ is the algebraic closure of $k((x))$. It is (cf. (Bliss, 1966)) contained in the algebraically closed field $\bigcup_{n \in \mathbb{N}} \overline{k}((x^{1/n}))$, the field of Puiseux series with coefficients in $\overline{k}$.

A *ramification* of the field $k((x))$ is a field extension $k((x)) \subset k((r))$ where r is algebraic over $k((x))$ with minimum polynomial $r^n - ax$ for some non-zero $a \in k$ and positive integer n (cf. (Sommeling, 1993)). If $a = 1$ this is called a *pure ramification.*

For $r \in \overline{k((x))}$ (not necessarily with minimum polynomial $r^n - ax$) we call the smallest integer n for which $r \in \overline{k}((x^{1/n}))$ the *ramification index* $\mathrm{ram}(r)$ of r. If L is a finite algebraic extension of $k((x))$ then the ramification index of L is the smallest n for which $L \subset \overline{k}((x^{1/n}))$.

$k((x))$ is a *differential field* with differentiation d/dx. If $k((x)) \subset L$ is an algebraic extension then d/dx can be extended in a unique way to L. All finite algebraic extensions $k((x)) \subset L$ are of the following form:

$$L = l((r))$$

where $k \subset l$ is a finite extension and $l((x)) \subset l((r))$ is a ramification (cf. (Sommeling, 1993), proposition 3.1.5).

3.2. The ring $k((x))[\delta]$

Define $\delta = x\partial \in k((x))[\partial]$. We have $\delta x = x\delta + x$ in $k((x))[\delta]$. Since $k((x))[\partial] = k((x))[\delta]$ we can represent differential operators in the form $f = a_n\delta^n + \ldots + a_0\delta^0$. This form has several advantages. The multiplication formula

$$(\sum_i x^i P_i(\delta))(\sum_j x^j Q_j(\delta)) = \sum_n x^n \sum_{i+j=n} P_i(\delta + j)Q_j(\delta)$$

and the definition of the Newton polygon (cf. section 3.3) are easier for operators with this syntax. The operators we consider are usually *monic.* This means $a_n = 1$. The *order* of a differential operator f is the degree of f as a polynomial in δ.

f is called the *least common left multiple* of a sequence of differential operators $f_1, \ldots, f_r$ if all f_i are right factors of f, the order of f is minimal with this property, and f is monic. Notation: $f = \mathrm{LCLM}(f_1, \ldots, f_r)$ (cf. (Singer, 1996)). The solution space of f is spanned by the solutions of $f_1, \ldots, f_r$. So $V(f) = \sum V(f_i)$ where $V(f)$ stands for the solution space of f. In order to speak about the solutions of differential operators a differential extension of $k((x))$ is required that contains a fundamental system of solutions of $f_1, \ldots, f_r$. For this we can use the so-called *universal extension* that we will denote as V. This V is constructed as follows (this construction is obtained from (Hendriks, van der Put, 1995), our V is called R in lemma 2.1.1 in (Hendriks, van der Put, 1995)). Define the set

$$E = \bigcup_{n \in \mathbb{N}} \overline{k}[x^{-1/n}].$$

First view $\mathrm{Exp}(e)$ and $\log(x)$ as variables and define the free $\overline{k((x))}$-algebra W in these variables $W = \overline{k((x))}[\{\mathrm{Exp}(e)|e \in E\}, \log(x)]$. Then define the derivatives $\mathrm{Exp}(e)' = \frac{e}{x}\mathrm{Exp}(e)$ and define the derivative of $\log(x)$ as $1/x$. This turns W into a differential ring. We can think of $\mathrm{Exp}(e)$ as

$$\mathrm{Exp}(e) = \exp(\int \frac{e}{x})$$

because $x\frac{d}{dx}$ acts on $\mathrm{Exp}(e)$ as multiplication by e. Now define V as the quotient ring $V = W/I$ where the ideal I is generated by the following relations:

$$\mathrm{Exp}(e_1 + e_2) = \mathrm{Exp}(e_1)\mathrm{Exp}(e_2) \quad \text{for} \quad e_1, e_2 \in E$$

and

$$\mathrm{Exp}(q) = x^q \in \overline{k((x))} \quad \text{for} \quad q \in \mathbb{Q}.$$

Note that this ideal is closed under differentiation. Hence V is a differential ring. It is proven in (Hendriks, van der Put, 1995) that V is an integral domain and that $\overline{k}$ is the set of constants of V. We denote the set of solutions of f in V as $V(f)$. This is a $\overline{k}$-vector space. Since every $f \in k((x))[\delta]$ has a fundamental system of solutions in V (cf. (Hendriks, van der Put, 1995)) it follows that

$$\dim(V(f)) = \mathrm{order}(f).$$

The *substitution map* $S_e : k((x))[\delta] \to k((x))[\delta]$ is a $k((x))$-homomorphism defined by $S_e(\delta) = \delta + e$ for $e \in k((x))$. S_e is a ring automorphism. The following is a well-known relation between the solution spaces:

$$V(f) = \mathrm{Exp}(e) \cdot V(S_e(f)).$$

The algorithm "Riccati solution" in section 5.1 introduces algebraic extensions over $k((x))$. This requires computer code for algebraic extensions of the constants $k \subset l$. But we can avoid writing code for ramifications. Given a field extension $k((x)) \subset k((r))$ where $r^n = ax$ for some $a \in k$ we will use the following ring isomorphism

$$\theta_{a,n} : k((r))[\delta] \to k((x))[\delta]$$

defined by $\theta_{a,n}(r) = x$ and $\theta_{a,n}(\delta) = \frac{1}{n}\delta$. This map enables us to reduce computations in $k((r))[\delta]$ to computations in $k((x))[\delta]$.

3.3. The Newton polygon

The *Newton polygon* of a monomial $x^i y^j$ in the commutative polynomial ring $k((x))[y]$ is defined as the set $\{(j, b) \in \mathbb{R}^2 | i \leq b\}$. The Newton polygon $N(f)$ of a non-zero polynomial $f \in k((x))[y]$ is defined as the convex hull of the union of the Newton polygons of the monomials for which f has a non-zero coefficient (cf. (Bliss, 1966), p. 36). The main property is $N(fg) = N(f) + N(g)$ for f and g in $k((x))[y]$. A rational number s is called a *slope* of f if s is the slope of one of the edges of the polygon $N(f)$. If s is a slope of fg then s is a slope of f or s is a slope of g.

For the non-commutative case $f \in k((x))[\delta]$ definitions of the Newton polygon are given in (Malgrange, 1979), (Tournier, 1987) and (Sommeling, 1993), p. 48. $N(x^i\delta^j)$ is defined as $\{(a, b) \in \mathbb{R}^2 | 0 \leq a \leq j, i \leq b\}$ and $N(f)$ is again defined as the convex hull of the union of the Newton polygons of the monomials that appear in f. This definition is slightly different from the commutative case. As a consequence all slopes are ≥ 0. This is needed to ensure $N(fg) = N(f) + N(g)$. If f has only one slope $s = 0$ then f is called *regular singular.*

3.4. The Newton polynomial

Let $s \geq 0$ be a rational number. We have defined a valuation v_s and a truncation σ_a for non-zero elements of $k((x))[\delta]$ in section 2. σ_a depends on s and will from now on be denoted as $\sigma_{a,s}$.

If $s > 0$ then $\sigma_{1,s}(L)\sigma_{1,s}(R) = \sigma_{1,s}(LR) = \sigma_{1,s}(R)\sigma_{1,s}(L)$ for all L and R in $k((x))[\delta]$. If $s = 0$ then $\sigma_{1,s}(L)\sigma_{1,s}(R) = \sigma_{1,s}(LR) = S_{-v_s(L)}(\sigma_{1,s}(R)) \cdot S_{v_s(R)}(\sigma_{1,s}(L))$.

So $\sigma_{1,s}$ is commutative (i.e. is the same for LR and RL) if $s > 0$. If $s = 0$ then $\sigma_{1,s}$ is commutative up to substitutions $S_{-v_s(L)}$ and $S_{v_s(R)}$ which map δ to δ plus some integer.

To $\sigma_{1,s}(f)$ for $f \in k((x))[\delta]$ corresponds a certain polynomial, the so-called *Newton polynomial* $N_s(f)$ (the definition is given after the example) of f for slope s. The Newton polynomial is useful for factorization in $k((x))[\delta]$ because if $f = LR$ then $\sigma_{1,s}(L)\sigma_{1,s}(R) = \sigma_{1,s}(f)$. So a factorization of f induces a factorization of the Newton polynomial.

Example: Consider the following differential operator

$$\begin{aligned} f = {} & 7x^{-5} + 2x^{-6}\delta + 2x^{-5}\delta + 3x^{-5}\delta^2 - 3x^{-5}\delta^3 + 5x^{-4}\delta^3 + x^{-4}\delta^5 \\ & +2x^{-2}\delta^5 + 2x^{-3}\delta^6 + 3x^{-2}\delta^7 + 2x^{-1}\delta^8 + \delta^9 \end{aligned}$$

In figure 1 we have drawn every monomial $x^i\delta^j$ which appears in f by placing the coefficient of this monomial on the point (j, i). This gives a set of points (j, i). For all points (j, i) for which $x^i\delta^j$ has a non-zero coefficient in f we can draw the rectangle with vertices $(0, i)$, (j, i), (j, ∞) and $(0, \infty)$. The Newton polygon is the convex hull of the union of all these rectangles. It is the part of the plane between the points $(0, \infty)$, $(0, -6)$, $(1, -6)$, $(5, -4)$, $(9, 0)$ and $(9, \infty)$. In the commutative case (i.e. if we had written y instead of δ in f) the definition of the Newton polygon is slightly different and the point $(0, -6)$ would have been $(0, -5)$ in this example. But for $k((x))[\delta]$ the Newton polygon is defined in such a way that there are no negative slopes.

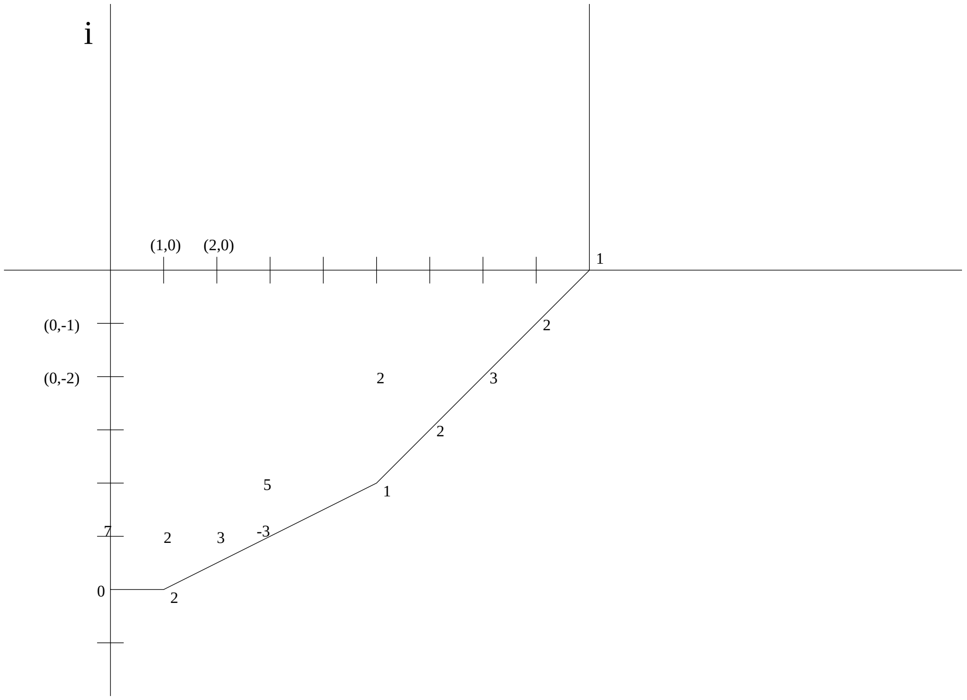

Fig. 1

The slopes of f are 0, 1/2 and 1. The Newton polynomials are $N_0(f) = 2T$, $N_{1/2}(f) = T^2 - 3T + 2$ and $N_1(f) = T^4 + 2T^3 + 3T^2 + 2T + 1$. Here T is used as a variable. $N_s(f)$ will be defined for all non-negative $s \in \mathbb{Q}$. However, we will only use the Newton polynomial for those values s which are a slope in the Newton polygon because for other values the Newton polynomial is trivial (i.e. degree 0).

Write $s = n/d$ where n and d are integers, $\gcd(n, d) = 1$ and $d > 0$. The valuation v_s gives a filtration (D_i), $i \in \mathbb{Z}$. $\sigma_{1,s}(f)$ is an element of $\overline{D} = \bigcup_{i \in \mathbb{Z}} D_i/D_{i+1}$. A multiplication is defined for elements of $\overline{D}$. An addition is only defined for $a, b \in \overline{D}$ which are element of the same D_i/D_{i+1}.

D_0 and $k[x^n\delta^d]$ are equal modulo D_1. There is a k-linear bijection

$$N'_s : D_i/D_{i+1} \to k[T]$$

which is also a ring isomorphism if $i = 0$. If $i = 0$ then N'_s is defined by $N'_s(x^n\delta^d) = T$.

For every $i \in \mathbb{Z}$ there is a unique pair of integers n_i, d_i such that the map $\phi_i : D_0/D_1 \to D_i/D_{i+1}$ defined by $\phi(a) = x^{n_i}\delta^{d_i}a$ is a bijection. The integers n_i, d_i can be determined from the conditions $0 \leq d_i < d$ and $v_s(x^{n_i}\delta^{d_i}) = i$. Now $N'_s(a)$ for $a \in D_i/D_{i+1}$ can be defined as $N'_s(\phi_i^{-1}(a))$. N'_s is also defined for non-zero elements of $f \in k((x))[\delta]$ as $N'_s(\sigma_{1,s}(f))$. In our example $N'_0(f) = 2T$, $N'_{1/2}(f) = T^2 - 3T + 2$ and $N'_1(f) = T^9 + 2T^8 + 3T^7 + 2T^6 + T^5$.

For slope $s = 0$ we define the Newton polynomial $N_0(f)$ as $N'_0(f)$. From the multiplication formula in section 3.2 the following property follows for $L, R \in k((x))[\delta]$

$$N_0(LR) = S_{T=T+v_0(R)}(N_0(L))N_0(R).$$

Here $S_{T=T+v_0(R)}(N_0(L))$ means $N_0(L)$ with T replaced by $T+v_0(R)$. For our example f we get $N_0(f\cdot f)=4(T-6)T$.

For slope $s>0$ we have the following property for $L,R\in k((x))[\delta]$

$$N'_s(LR)=T^p N'_s(L)N'_s(R).$$

Here p is either 0 or 1, depending on the slope s and the valuations $v_s(L)$ and $v_s(R)$. Let $i=v_s(L)$ and $j=v_s(R)$. Then $\phi_i(1)\cdot\phi_j(1)=x^{n_i+n_j}\delta^{d_i+d_j} \bmod D_{i+j+1}$. This is either equal to $\phi_{i+j}(1)$ or $x^n\delta^d\phi_{i+j}(1) \bmod D_{i+j+1}$, depending on whether d_i+d_j is smaller than d or not. In the first case $p=0$, in the second case $p=1$. For our example $N'_{1/2}(f\cdot f)=T\cdot(N'_{1/2}(f))^2$ and $N'_1(f\cdot f)=(N'_1(f))^2$. Now define $N_s(f)$ as $N'_s(f)$ divided by T to the power the multiplicity of the factor T in $N'_s(f)$. Then

$$N_s(LR)=N_s(L)N_s(R)$$

for $s>0$ and for all $L,R\in k((x))[\delta]$.

Note that our definition does not correspond to the usual definition of the Newton polynomial. It corresponds to the definition of the reduced characteristic polynomial in (Barkatou, 1988). The roots of $N_0(f)$ in $\overline{k}$ are called the *exponents* of f. If $f\in k((x))[\delta]$ is regular singular (i.e. f has only one slope $s=0$, or equivalently degree$(N_0(f))=$ order(f)) and all exponents of f are integers then f is called *semi-regular.*

Property: If $f=LR$ then the Newton polynomial of the right-hand factor $N_s(R)$ divides $N_s(f)$. However, for a left-hand factor this need not hold. But if $s>0$ or if $v_0(R)=0$ (for example if R is regular singular and monic) then $N_s(f)=N_s(L)N_s(R)$ so in such cases $N_s(L)$ divides $N_s(f)$.

4. The lift algorithm

Suppose $f\in k((x))[\delta]$ is monic and that $f=LR$ is a non-trivial factorization, where L and R are monic elements of $k((x))[\delta]$. Let $s\geq 0$ be a rational number. Recall that there is a valuation v_s on $D=k((x))[\delta]$, a filtration $(D_{n,s})$, $n\in\mathbb{Z}$ and a truncation map $\sigma_{a,s}$ depending on s. In this section we will assume that L and R have been computed up to some accuracy a. How to compute this $\sigma_{a,s}(L)$ and $\sigma_{a,s}(R)$ will be the topic of the sections 5 and 7. In this section we deal with the question how to compute $\sigma_{a+1,s}(L)$ and $\sigma_{a+1,s}(R)$ from $\sigma_{a,s}(L)$, $\sigma_{a,s}(R)$ and f in an efficient way. The goal is an algorithm with the following specification:

Lift Algorithm:
Assumption: $f=LR$ where f,L,R are monic elements of $k((x))[\delta]$.
Input: $a\geq 1$, s, $\sigma_{a,s}(L)$, $\sigma_{a,s}(R)$ and f.
Output: Either $\sigma_{a+1,s}(L)$ and $\sigma_{a+1,s}(R)$ or "failed", where "failed" can only occur if $t>a$ where t is the coprime index.

We use this algorithm to lift a factorization. If the output is "failed" then we will use the less efficient method in section 7 to lift the factorization. Note that since $a\geq 1$ this can only happen if the coprime index is >1.

Suppose $t\leq a$. We will use indeterminates for those coefficients of $\sigma_{a+t,s}(L)$ and $\sigma_{a+t,s}(R)$ which are not yet known. Then the equation $\sigma_{a+t,s}(LR)=\sigma_{a+t,s}(f)$ gives a set of equations in these unknowns (more details on how to find these equations are given below). $t \leq a$ is needed to ensure that all these equations are linear. Coprime index t means that $\sigma_{a+1,s}(L)$ and $\sigma_{a+1,s}(R)$ can be uniquely determined from these linear equations.

Except if the coprime index is 1, our algorithm usually does not know the coprime index in concrete situations. Then the lift algorithm will use a guess for the coprime index. If the lift algorithm is called for the first time, it takes $t = 2$. Otherwise it takes the guess for t that was used in the previous lift step. Then it will try, by solving linear equations, if there is a unique solution for $\sigma_{a+1,s}(L)$ and $\sigma_{a+1,s}(R)$ from $\sigma_{a,s}(L)$, $\sigma_{a,s}(R)$ and $\sigma_{a+t,s}(f)$. If so, t remains unchanged and the accuracy of the factorization increases; the output of the lift algorithm is $\sigma_{a+1,s}(L)$ and $\sigma_{a+1,s}(R)$. If the solution for $\sigma_{a+1,s}(L)$ and $\sigma_{a+1,s}(R)$ is not unique (there is at least one solution because of the assumption that the factorization $f = LR$ exists) the number t will be increased by 1. If t is still $\leq a$ then we can use recursion with our increased guess t for the coprime index. Otherwise, if $t > a$, the output of the lift algorithm is "failed", and we will have to use the less efficient method in section 7 to lift the factorization. Note that the efficiency of our lift algorithm depends on the coprime index, if this number is very high then it may not provide any speedup over the method from section 7.

A truncation $\sigma_{a,s}(R) = R + D_{v_s(R)+a}$ is stored as an element $R' \in k[x, 1/x, \delta]$ with no terms in $D_{v_s(R)+a}$. Now write

$$r = \sum_{i,j} r_{ij} x^i \delta^j$$

where the sum is taken over all i, j such that $v_s(R) + a \leq v_s(x^i\delta^j) < v_s(R) + a + t$ and $j \leq \text{order}(R)$. Here r_{ij} are indeterminates. We set $r_{ij} = 0$ for $j = \text{order}(R)$, $i \neq 0$, and set $r_{ij} = 1$ for $j = \text{order}(R)$, $i = 0$. Similarly write L' and l. Now we look for values for the l_{ij} and r_{ij} such that $R' + r$ and $L' + l$ approximate R and L up to accuracy $a + 1$. If the coprime index is t, the accuracy is at least $a + 1$ if the following holds: $\sigma_{a+t,s}((L' + l)(R' + r)) = \sigma_{a+t,s}(f)$, or equivalently

$$(L' + l)(R' + r) \equiv f \mod D_{v_s(f)+a+t}.$$

$(L' + l)(R' + r) = L'R' + lR' + L'r + lr$. To determine lR' mod $D_{v_s(f)+a+t}$ it suffices to have R' up to accuracy t because $v_s(l) + v_s(R') \geq v_s(f) + a$. Similarly $\sigma_{t,s}(L')$ suffices to compute $L'r$ mod $D_{v_s(f)+a+t}$. $v_s(lr) \geq v_s(f) + a + a \geq v_s(f) + a + t$ so lr vanishes modulo $D_{v_s(f)+a+t}$. Hence

$$f \equiv L'R' + l\sigma_{t,s}(R') + \sigma_{t,s}(L')r \mod D_{v_s(f)+a+t}.$$

By equating the coefficients of the left-hand side to the coefficients of the right-hand side (the coefficients of all monomials of valuation $< v_s(f) + a + t$) we find the linear equations in l_{ij} and r_{ij}. To determine these equations we must multiply l by $\sigma_{t,s}(R')$, ($= \sigma_{t,s}(R)$ because R' equals R up to accuracy a and $t \leq a$) which is the lowest block of R with slope s and width t in the Newton polygon of R. Similarly we must compute $\sigma_{t,s}(L')r$.

Usually the most time consuming part is the multiplication $L'R'$ modulo $D_{v_s(f)+a+t}$. One approach is the following. Compute $L'R'$ in $k[x, 1/x, \delta]$ and store the result together with L' and R'. In the next lift step a similar multiplication must be performed, but then L' and R' are slightly changed. Suppose we must compute the product $(L' + e_1)(R' + e_2)$ in the next lift step. Here L' and R' are large expressions and e_1 and e_2 are small. Using the previous multiplication $L'R'$ we can speed up this multiplication by writing

$(L' + e_1)(R' + e_2) = L'R' + e_1R' + L'e_2 + e_1e_2$. The result of this multiplication is again stored for use in the next lift step.

In this approach $L'R'$ has been computed exactly. This is not efficient since we only need it up to accuracy $a+t$, i.e. modulo $D_{v_s(f)+a+t}$. Computing modulo $D_{v_s(f)+a+t}$ is not as convenient as computing modulo a power of x when using the multiplication formula in section 3.2. We compute $L'R'$ modulo a suitable power of x such that $L'R'$ can still be determined modulo $D_{v_s(f)+a+t}$. Unless the slope s is zero, however, a few more terms of the product $L'R'$ than needed have been computed then. These terms are stored to speed up the multiplication the next time that the lift algorithm is called.

5. Coprime index 1 factorizations

The lifting process for coprime factorizations can be done by solving linear equations. However, for coprime index 1 solving linear equations can be avoided. In this case we must solve a system (see section 4) of the form $l\sigma_{1,s}(R) + \sigma_{1,s}(L)r = g$ where g is computed by multiplying the so far computed truncations (called L' and R' in section 4) of L and R and subtracting this product from f. As in section 3.4 this equation can be converted to an equation $lR_0 + rL_0 = g$ for certain $l, r, L_0, R_0, g \in k[T]$ and l, r unknown. Such an equation can be solved by the Euclidean algorithm.

Consider the example f in section 3.4. f has slopes 0, 1/2 and 1 in this example. In (Malgrange, 1979) a method is given to compute a right-hand factor f_1 with only slope 0 and order 1, a right factor f_2 with slope 1/2 and order 4 and a right factor f_3 with slope 1 and order 4. The Newton polynomial of f_2 is the same as the Newton polynomial $N_{1/2}(f)$ of f for slope 1/2. It is $2-3T+T^2 = (T-1)(T-2)$. Because $\gcd(T-1, T-2) = 1$ this f_2 is again reducible in $\mathbb{Q}((x))[\delta]$, cf. (Malgrange, 1979). It has a right factor g_1 of order 2 and slope 1/2 with Newton polynomial $T-1$ and a right factor g_2 with Newton polynomial $T-2$. So to obtain g_1 two factorization were needed. In one application, our algorithm for factorization in $\mathbb{Q}(x)[\partial]$, we are mainly interested in one of the irreducible right-hand factors of f in $\mathbb{Q}((x))[\delta]$. That is why we want to be able to compute g_1 directly without using an intermediate factorization to compute f_2. This is done by the following algorithm.

Algorithm Coprime Index 1 Factorizations:
Input: $f \in k((x))[\delta]$, f monic
Output: All monic coprime index 1 factorizations $f = LR$ in $k((x))[\delta]$ such that R does not have a non-trivial coprime index 1 factorization.
Note: the definition of coprime index depends on the valuation that is chosen on $k((x))[\delta]$. Here only the valuations v_s that are defined in section 2 are considered.

```
for all slopes $s$ of $f$ do
    $g := N_s(f)$
    Compute a prime factorization of $g$ in $k[T]$, $g = cg_1^{e_1} \cdots g_r^{e_r}$,
        where $g_i$ are the different monic irreducible factors and $c \in k$.
    if $s = 0$ then
        $M := \{g_1, \ldots, g_r\}$
        $N := M \setminus \{g | g(T) = h(T+i), h \in M, i \in \mathbb{N}, i > 0\}$
    else
        $N := \{g_1^{e_1}, \ldots, g_r^{e_r}\}$
    end if
    for $h$ in $N$ do
        Write $h = T^p + h_{p-1}T^{p-1} + \ldots + h_0T^0$.
        Write $s = n/d$ with $d > 0$ and $\gcd(n, d) = 1$ (if $s = 0$ then $n = 0$, $d = 1$)
        $R' := \delta^{pd} + h_{p-1}x^{-n}\delta^{(p-1)d} + h_{p-2}x^{-2n}\delta^{(p-2)d} + \ldots + h_0x^{-pn}\delta^0$.
        Now $R'$ has Newton polynomial $h$. We want to lift $R'$ to a right
            factor $R$ such that $R'$ is $R$ modulo $D_{v_s(R')+1}$.
        $L'$ := an operator such that $\sigma_{1,s}(f) = \sigma_{1,s}(L'R')$.
            $L'$ is uniquely determined by requiring that $L'$ has no
            monomials of valuation $> v_s(L')$.
        $f$, $L'$, $R'$ with the lift algorithm gives a factorization $f = LR$
    end do
end do
```

We need to prove the following:

1. L' and R' lift to a unique coprime index 1 factorization $f = LR$.
2. The right factors R do not allow a non-trivial coprime index 1 factorization.
3. All such coprime index 1 factorizations $f = LR$ (f, L and R monic) are obtained this way.

Suppose $\sigma_{a,s}(L'R') = \sigma_{a,s}(f)$, meaning that the factorization has been lifted up to accuracy a. If no lift steps were done yet, then $a = 1$. Now we look for $l \in D_{v_s(L')+a}$ and $r \in D_{v_s(R')+a}$ such that $\sigma_{a+1,s}((L'+l)(R'+r)) = \sigma_{a+1,s}(f)$ and $\mathrm{order}(r) < \mathrm{order}(R')$. To prove statement 1 we have to show that l, r exist and that $\sigma_{a+1,s}(L'+l)$ and $\sigma_{a+1,s}(R'+r)$ are uniquely determined. This means that $l \bmod D_{v_s(L)+a+1} \in \overline{D}$ (cf. section 3.4) and $r \bmod D_{v_s(R)+a+1} \in \overline{D}$ are uniquely determined. Then L' and R' are replaced by $L' + l$ and $R' + r$ and the accuracy of the approximations L' and R' for L and R has increased by 1. l and r must satisfy the following equation in $\overline{D}$

$$\sigma_{1,s}(L)r + l\sigma_{1,s}(R) = f - L'R' \bmod D_{v_s(f)+a+1}.$$

By applying N_s' we obtain the following equation in $k[T]$ if $s = 0$

$$S_{T=T+a}(L_0)r_0 + l_0R_0 = g$$

and

$$L_0r_0 + l_0R_0 = g \ \text{ or } \ TL_0r_0 + l_0R_0 = g$$

if $s > 0$. Here $l_0 = N_s'(l \bmod D_{v_s(L)+a+1})$, $r_0 = N_s'(r \bmod D_{v_s(R)+a+1})$, $L_0 = N_s'(L)$, $R_0 = N_s'(R)$ and $g = N_s'(f - L'R' \bmod D_{v_s(f)+a+1})$. Note that $v_s(R)$ is 0 if $s = 0$. The requirement $\mathrm{order}(r) < \mathrm{order}(R)$ means $\mathrm{degree}(r_0) < \mathrm{degree}(R_0)$. To prove statement 1 we now have to show that $l_0, r_0 \in k[T]$ exist and are uniquely determined. For this it is sufficient to show that $\gcd(TL_0, R_0) = 1$ if $s > 0$ and $\gcd(S_{T=T+a}(L_0), R_0) = 1$ if $s = 0$. First the case $s > 0$. R_0 is the factor h of the Newton polynomial in the algorithm. $L_0R_0 = N_s'(f) = T^iN_s(f)$ for some integer i. The set N of factors h of $N_s(f)$ is chosen in such a way in the algorithm that $\gcd(h, N_s(f)/h) = 1$. Also $\gcd(h, T) = 1$ because $N_s(f)$ does not contain a factor T by definition and h is a factor of $N_s(f)$. So $\gcd(TL_0, R_0) = 1$. Now the case $s = 0$. We have $L_0R_0 = N_s(f)$ because $v_s(R) = 0$ (see the multiplication formula for N_0 in section 3.4). R_0 is the factor h of $N_s(f)$ in the algorithm. We have to show that $\gcd(S_{T=T+a}(L_0), R_0) = 1$. The set N containing these factors h was chosen in such a way that this holds for all $a \geq 1$.

To prove the second statement we distinguish 2 cases. Suppose $s = 0$. Then the Newton polynomial of R is irreducible. Hence R must be irreducible because a factorization of R gives a factorization of the Newton polynomial. Now suppose $s > 0$. Then the Newton polynomial is of the form p^i where p is irreducible and i is an integer. If $i > 1$ then it is not clear whether R is reducible or not. Suppose R can be factored $R = R_1 R_2$. Then the Newton polynomials of R_1 and R_2 are both factors of p^i. So the gcd of these Newton polynomials is not equal to 1. Coprime index 1 would mean that $\sigma_{a+1,s}(R_1)$ and $\sigma_{a+1,s}(R_2)$ can be uniquely determined from $\sigma_{a,s}(R_1)$, $\sigma_{a,s}(R_2)$ and $\sigma_{a+1,s}(f)$. To determine $\sigma_{a+1,s}(R_1)$ and $\sigma_{a+1,s}(R_2)$ requires solving an equation $l_0 N_s(R_1) + r_0 N_s(R_2) = g$ in $k[T]$. Such an equation has a unique solution if and only if the gcd of the Newton polynomials $N_s(R_1)$ and $N_s(R_2)$ is 1. So a possible factorization $R = R_1 R_2$ cannot be a coprime index 1 factorization, which proves statement 2.

Suppose $f = LR$ is a monic factorization satisfying statement 2. Now we need to show that the algorithm finds this factorization. R can have only one slope s, otherwise it could be factored by the given algorithm (which contradicts the assumption that statement 2 holds). First consider the case $s = 0$. Then $N_s(R)$ must be an irreducible polynomial, otherwise R can be factored by the algorithm. So $N_s(R)$ must be an element of the set M in the algorithm. It cannot be an element of $\{g | g(T) = h(T+i), h \in M, i \in \mathbb{N}, i > 0\}$ because then $\gcd(S_{T=T+a}(L_0), R_0) = 1$ does not hold for all $a \geq 1$ which was shown to be a necessary and sufficient condition for having coprime index 1 if $s = 0$. So $N_s(R) \in N$. This means that $\sigma_{1,s}(R)$ and hence also $\sigma_{1,s}(L)$ are the same as $\sigma_{1,s}(R_1)$ and $\sigma_{1,s}(L_1)$ for a factorization L_1, R_1 of f given by the algorithm. Because the coprime index is 1 this factorization L_1, R_1 is completely determined by $\sigma_{1,s}(R_1)$, $\sigma_{1,s}(L_1)$ and f. Hence these two factorizations $L_1 R_1$ and LR are the same and so the third statement holds. In the same way the case $s > 0$ is proven.

□

Remark: the given method can also be applied for factorization in the ring $L[\delta]$ where L is a finite extension of $k((x))$, because

- The method is not different for algebraic extensions of the constants $k \subset l$.
- Ramifications over $l((x))$ can be handled using the map $\theta_{a,n}$ in section 3.2.
- All finite field extensions of $k((x))$ are obtained as an algebraic extension of the constants followed by a ramification, cf. section 3.1.

Consider again the example f in section 3.4 and let $k = \mathbb{Q}$. The given algorithm produces a right-hand factor R_1 with slope 0, order 1 and Newton polynomial T, a right factor R_2 with slope $1/2$, order 2 and polynomial $T - 1$, a right factor R_3 with slope $1/2$, order 2 and polynomial $T - 2$ and a right factor R_4 with slope 1, order 4 and Newton polynomial $(T^2 + T + 1)^2$. Now R_1, R_2 and R_3 are irreducible in $\mathbb{Q}((x))[\delta]$ because their Newton polynomials are irreducible. But it is not yet clear whether R_4 is irreducible or not. The second example in section 1 remains unfactored as well. Reducible operators f that remain unfactored by the given factorization algorithm are of the following form: f has one slope $s > 0$ and $N_s(f)$ is a power > 1 of an irreducible polynomial. The given algorithm will compute only a trivial factorization $L = 1$, $R = f$ for this case. If such an operator is reducible then a factorization must have coprime index > 1. In section 6 the notion of exponential parts will be introduced. Using exponential parts a description of the irreducible elements of $k((x))[\delta]$ will be given.

If f has one slope $s > 0$, $s \in \mathbb{N}$ and the Newton polynomial is a power of a polynomial of degree 1, then compute $S_{cx^{-s}}(f)$ where c is the root of the Newton polynomial (see also case 4 of the algorithm in section 5.1). Then apply the factorization algorithm to $S_{cx^{-s}}(f)$ and find a factorization of f by applying $S_{-cx^{-s}}$ to the factors of $S_{cx^{-s}}(f)$. For all other cases (i.e. $s \notin \mathbb{N}$ or degree$(N_s(f)) > 1$) we apply the method in section 7. The factorization obtained that way lifts rather slowly, i.e. it costs much time to compute more terms. The lifting will be speeded up using the lift method of section 4 whenever that is possible (when its output is not the message "failed").

A differential operator can have infinitely many different factorizations. For example ∂^2 which equals $1/x^2$ times $\delta^2 - \delta$ has $ax + b$ as solutions, where a and b are constants. Hence it has $\partial - (ax+b)'/(ax+b) = \partial - a/(ax+b)$ as right factors. Note that algorithm coprime index 1 factorizations produces only a finite number of different factorizations. In the semi-regular case (cf. section 3.4) it computes only 1 unique factorization, although like the example ∂^2 shows other factorizations could exist as well.

5.1. Computing first order factors over $\overline{k((x))}$

An element r of some differential extension of $k((x))$ is by definition a *Riccati solution* of $f \in k((x))[\delta]$ if $\delta - r$ is a right factor of f. The reason this is called a Riccati solution is that they are solutions of the so-called *Riccati equation*. This is a non-linear differential equation. The Riccati equation of $f \in k((x))[\delta]$ can be found by computing a right division of f by $\delta - u$, where u is an indeterminate. The remainder of this right division is the Riccati equation. It is a polynomial in u and the derivatives of u. It vanishes precisely when we substitute for u an element r such that $\delta - r$ is a right-hand factor of f. The Riccati solutions are of the form xy'/y where y is a solution of f. In the usual definition the Riccati solutions are the logarithmic derivatives y'/y of solutions of f. The definition in this paper differs by a factor x because we work with $\delta = x\partial$ instead of ∂. In this paper only Riccati solutions in $\overline{k((x))}$ are considered. In general there exist more Riccati solutions in larger differential fields. The implementation does not determine the Riccati equation itself because this can be a large expression. Instead we use factorization to find Riccati solutions. Computing first order right-hand factors of f is the same as computing Riccati solutions.

The following algorithm is similar to the Rational Newton algorithm (cf. (Barkatou, 1988)) which is a version of the Newton algorithm (cf. (Tournier, 1987; Della Dora, di Crescenzo, Tournier, 1982)) that computes formal solutions using a minimal algebraic extension of the constants field k. A difference between the Rational Newton algorithm and the following algorithm Riccati solution is that we use factorization of differential operators. So the order of the differential operator decreases during the computation.

Algorithm Riccati solution:
Input: $f \in k((x))[\delta]$
Output: a first order right-hand factor in $\overline{k((x))}[\delta]$

```
1  If order($f$) $= 1$ then the problem is trivial.
2  If one of the following holds
```

(a) f is regular singular and the $N_0(f)$ is reducible.
(b) The Newton polygon has more than 1 slope.
(c) f has one slope $s > 0$ and $N_s(f)$ is not a power ≥ 1 of an irreducible polynomial.

then compute a coprime index 1 factorization and apply recursion to the right-hand factor.

3 If f has one slope s and the Newton polynomial $N_s(f)$ is of the form p^e with p irreducible, $e \geq 1$ and p is of degree $d > 1$. Then extend k by one root r of p. Now compute a right factor of order $\mathrm{order}(f)/d$ with $(T-r)^e$ as Newton polynomial using a coprime index 1 factorization as in the algorithm in section 5. This is a coprime index 1 factorization because the gcd of $(T-r)^e$ and $p^e/(T-r)^e$ (this is the Newton polynomial of the left hand factor) is 1. Now apply recursion to the right-hand factor.

4 If f has one slope $s > 0$, $s \in \mathbb{N}$ and $N_s(f)$ is a power of a polynomial of degree 1, then compute $S_{cx^{-s}}(f)$ where c is the root of $N_s(f)$. Use recursion (this recursion is valid because the slopes of $S_{cx^{-s}}(f)$ are smaller than the slope of f) to find a first order factor of $S_{cx^{-s}}(f)$. Then apply $S_{-cx^{-s}}$.

5 If f has one slope $s > 0$, $s \notin \mathbb{N}$ and the Newton polynomial is a power of a polynomial of degree 1, then write $s = n/d$ with $\gcd(n,d) = 1$, $n > 0$. Now we will apply a ramification of index d. Instead of extending the field $k((x))$ we apply the isomorphism $\theta_{a,d} : k((r))[\delta] \to k((x))[\delta]$ of section 3.2. First we need to compute a suitable value $a \in k$. $\theta_{a,d}(x) = \theta_{a,d}(r^d/a) = x^d/a$. Write the Newton polynomial of f as $(T-c)^e$, where $c \in k$ and $e \in \mathbb{N}$. Then the Newton polynomial of $\theta_{a,d}(f)$ equals a constant times $(T^d - d^d c a^n)^e$. Now choose a equal to c^p, $p \in \mathbb{Z}$, such that $d^d c a^n$ is a d-th power of an element $b \in k$. This is done by choosing p such that $pn + 1$ is a multiple of d. Then the Newton polynomial $(T^d - d^d c a^n)^e$ equals $(T^d - b^d)^e$ and can be factored as $(T-b)^e g^e$ with $\gcd((T-b)^e, g^e) = 1$. Now use a coprime index 1 factorization as in section 5 with $(T-b)^e$ as Newton polynomial for the right-hand factor. This provides a right factor R of order $e = \mathrm{order}(f)/d$. Now use recursion on R to find a first order factor and apply $\theta_{a,d}^{-1}$.

Note that there are two cases where a field extension of $k((x))$ is applied. One case was an extension of k of degree d, and the other case was a ramification of index d. Both these cases were extensions of $k((x))$ of degree d. In both cases the algorithm finds a right factor of order $\mathrm{order}(f)/d$ over this algebraic extension. In the three other cases field extensions were not needed. We can conclude

LEMMA 5.1. *Every $f \in k((x))[\delta]$ has a Riccati solution which is algebraic over $k((x))$ of degree $\leq \mathrm{order}(f)$.*

6. Exponential parts

A commutative invariant is a map ϕ from $k((x))[\delta]$ to some set such that $\phi(fg) = \phi(gf)$ for all $f, g \in k((x))[\delta]$. An example is the Newton polygon, i.e. $N(fg) = N(gf)$ for all non-zero f and g. However, there are more properties of differential operators that remain invariant under changing the order of multiplication. We want a commutative invariant which contains as much information as possible. In (Sommeling, 1993) Sommeling defines *normalized eigenvalues* and *characteristic classes* for matrix differential operators. The topic of this section is the analogue of normalized eigenvalues for differential operators in $k((x))[\delta]$. We will call these *exponential parts.* The exponential parts are useful for several topics. They appear as an exponential integral in the formal solutions (this explains the name exponential part). They describe precisely the algebraic extensions over $k((x))$ needed to find the formal solutions. The exponential parts are also used in our method of factorization in the ring $k(x)[\partial]$ in chapter 3 in (van Hoeij, thesis). For factorization in $k((x))[\delta]$ the exponential parts will be used to describe the irreducible elements, (cf. theorem 6.2).

Differential operators (in this paper that means elements of $k((x))[\delta]$ or $\overline{k((x))}[\delta]$) can be viewed as a special case of matrix differential operators. So our definition of exponential parts could be viewed as a special case of the definition of normalized eigenvalues in (Sommeling, 1993). A reason for giving a different definition is that the tools for computing with matrix differential operators are not the same as for differential operators. Important tools for matrix differential operators are the splitting lemma and the Moser algorithm. The tools we use for differential operators are the substitution map and the Newton polynomial. That is why we want to have a definition of exponential parts expressed in these tools. Because then the definition allows the computation of exponential parts using a variant of the "algorithm Riccati solution", namely the "algorithm semi-regular parts" in section 8.4. A second reason for our approach is that it allows the definition of semi-regular parts of differential operators.

Let L be a finite extension of $k((x))$. Since $L \subset \overline{k}((x^{1/n}))$ for some integer n we can write every $r \in L$ as $r = e + t$ with $e \in E$ and $t \in x^{1/n}\overline{k}[[x^{1/n}]]$. Now e is called the *principal part* $\mathrm{pp}(r)$ of $r \in L$. Using the following lemma we can conclude $e \in k((x))[r] \subset L$.

Lemma 6.1. *Let $n \in \mathbb{Q}$ and $r \in \overline{k((x))}$ be equal to $r_n x^n$ plus higher order terms. Then $r_n x^n$ is an element of the field $k((x))[r]$.*

Proof: Write $r = r_n x^n + r_m x^m$ plus higher order terms, where $m \in \mathbb{Q}$, $m > n$. We want to prove that there exists an $s \in k((x))[r]$ of the form $r_n x^n$ plus terms higher than x^m. Then we can conclude $r_n x^n \in k((x))[r]$ by repeating this argument and using the fact that the field $k((x))[r]$ is complete (cf. (Bourbaki, 1953) Chap I, §2, thm. 2). We can find this s as a $\mathbb{Q}$-linear combination of r and $x\frac{dr}{dx}$.

□

Definition 6.1. *Let $f \in k((x))[\delta]$, $e \in E$ and $n = \mathrm{ram}(e)$. Let $P = N_0(S_e(f))$, the Newton polynomial corresponding to slope 0 in the Newton polygon of $S_e(f) \in \overline{k}((x^{1/n}))[\delta]$. Now $\mu_e(f)$ is defined as the number of roots (counted with multiplicity) of P in $\frac{1}{n}\mathbb{Z}$ and $\overline{\mu}_e(f)$ is defined as the number of roots (counted with multiplicity) of P in $\mathbb{Q}$.*

Recall that $\mathrm{ram}(e)$ denotes the ramification index of e. Note that we have only defined the Newton polynomial for elements of $k((x))[\delta]$, not for ramifications of $k((x))$. Define $N_0(f)$ for $f \in \overline{k}((x^{1/n}))[\delta]$ as follows. Write $f = \sum_i x^{i/n} f_i$ with $f_i \in \overline{k}[\delta]$. Then $N_0(f)$ is (written as a polynomial in δ instead of T) defined as f_i where i is minimal such that $f_i \neq 0$.

We define an equivalence $\sim$ on E as follows: $e_1 \sim e_2$ if $e_1 - e_2 \in \frac{1}{n}\mathbb{Z}$ where n is the ramification index of e_1. Note that the ramification indices of e_1 and e_2 are the same if

$e_1 - e_2 \in \mathbb{Q}$. If $e_1 \sim e_2$ then $\mu_{e_1}(f) = \mu_{e_2}(f)$ for all $f \in k((x))[\delta]$ so we can define μ_e for $e \in E/\sim$. Similarly $\overline{\mu}_e(f)$ is defined for $e \in E/\mathbb{Q}$.

DEFINITION 6.2. *The* exponential parts *of an operator $f \in k((x))[\delta]$ are the elements $e \in E/\sim$ for which $\mu_e(f) > 0$. The number $\mu_e(f)$ is the* multiplicity *of e in f.*

LEMMA 6.2. *Let $f = LR$ where f, L and R are elements of $\overline{k}((x^{1/n}))[\delta]$. Let N_f be the number of roots of $N_0(f)$ in $\frac{1}{n}\mathbb{Z}$, counted with multiplicity. Similarly define N_L and N_R. Then $N_f = N_L + N_R$.*

The proof of this lemma is not difficult; we will skip it. Note that if $n = 1$ then $N_f = \mu_0(f)$.

LEMMA 6.3. *If $f = LR$ where f, L and R are elements of $k((x))[\delta]$ and e in E or in $E/\sim$ then $\mu_e(f) = \mu_e(L) + \mu_e(R)$.*

If $f = LR$ where f, L and R are elements of $\overline{k((x))}[\delta]$ and e in E or in $E/\mathbb{Q}$ then $\overline{\mu}_e(f) = \overline{\mu}_e(L) + \overline{\mu}_e(R)$.

Proof: If n is the ramification index of e, then $\mu_e(f)$ is the number of roots in $\frac{1}{n}\mathbb{Z}$ of $N_0(S_e(f))$. Now the first statement follows using the previous lemma and the fact that $S_e(f) = S_e(L)S_e(R)$. The proof for $\overline{\mu}$ is similar.

□

THEOREM 6.1. *Let f be a non-zero element of $k((x))[\delta]$, then the sum of the multiplicities of all exponential parts is:*

$$\sum_{e \in E/\sim} \mu_e(f) = \mathrm{order}(f).$$

Let f be a non-zero element of $\overline{k((x))}[\delta]$, then

$$\sum_{e \in E/\mathbb{Q}} \overline{\mu}_e(f) = \mathrm{order}(f).$$

Proof: If order$(f) = 1$ then both statements hold. If f is reducible then we can use induction and lemma 6.3 so then both statements hold. In $\overline{k((x))}[\delta]$ every operator of order > 1 is reducible (see also the algorithm in section 5.1 which computes a first order right-hand factor in $\overline{k((x))}[\delta]$) so the second statement holds.

To prove the first statement we need to show that the sum of the multiplicities is the same for μ over all $e \in E/\sim$ and $\overline{\mu}$ over all $e \in E/\mathbb{Q}$. Suppose $\overline{e}$ is an element of $E/\mathbb{Q}$. The sum of $\mu_e(f)$ taken over all $e \in E/\sim$ such that $\overline{e} \equiv e \bmod \mathbb{Q}$ is equal to $\overline{\mu}_{\overline{e}}(f)$ because they are both equal to the number of rational roots of the same Newton polynomial. So we can see that the sum of the multiplicities $\overline{\mu}$ is the same as sum of the multiplicities μ by grouping together those exponential parts of f that are congruent modulo $\mathbb{Q}$.

□

6.1. Semi-regular part

An operator $f \in k((x))[\delta]$ is called *semi-regular* over $k((x))$ if f has only one exponential part which is equal to $0 \in E/\sim$. A semi-regular operator is a regular singular operator with only integer roots of the Newton polynomial. In other words $\mu_0(f) = \mathrm{order}(f)$. An operator $f \in k((x))[\delta] = k((x))[\partial]$ is *regular* (or: non-singular) if f can be written as a product of an element of $k((x))$ and a monic element of $k[[x]][\partial]$. A regular operator is regular singular and the roots of the Newton polynomial are $0, 1, \ldots, \mathrm{order}(f) - 1$. So a regular operator is semi-regular. We can generalize the notion of semi-regular for algebraic extensions $k((x)) \subset L$.

Definition 6.3. *$f \in L[\delta]$ is called* semi-regular *over L if it is regular singular and all roots of $N_0(f)$ are integers divided by the ramification index of L.*

For a ramification $r^n = ax$ an isomorphism $\theta_{a,n} : k((r))[\delta] \to k((x))[\delta]$ was given in section 3.2. Now $f \in k((r))[\delta]$ is semi-regular over $k((r))$ if and only if $\theta_{a,n}(f) \in k((x))[\delta]$ is semi-regular over $k((x))$.

Definition 6.4. *Let $f \in k((x))[\delta]$. Then the* semi-regular part R_e *of f for $e \in E$ is the monic right-hand factor in $k((x))[e, \delta]$ of $S_e(f)$ of order $\mu_e(f)$ which is semi-regular over $k((x))[e]$.*

R_e can be computed by a coprime index 1 factorization of $S_e(f)$ as in section 5 using slope $s = 0$. The Newton polynomial (called h in the algorithm) is the largest factor of $N_0(S_e(f))$ for which all roots are integers divided by the ramification index. Since such coprime index 1 factorizations for a given Newton polynomial are unique (see the comments after Algorithm Coprime Index 1 Factorizations) it follows that R_e is uniquely defined. Note that if the ramification index n is > 1 then in fact our algorithm does not compute with $S_e(f)$ but with $\theta_{a,n}(S_e(f))$ for some constant a, cf. the remark in section 5. Then we have to compute the highest order factor of $\theta_{a,n}(S_e(f))$ of which the roots of the Newton polynomial are integers, instead of integers divided by n.

$S_{-e}(R_e)$ is a right-hand factor of f. Note that if $e_1 \sim e_2$ then $S_{-e_1}(R_{e_1}) = S_{-e_2}(R_{e_2})$. Hence the operators $S_{-e_1}(R_{e_1}), \ldots, S_{-e_p}(R_{e_p})$ in the following lemma are up to a permutation uniquely determined by f.

Lemma 6.4. *Let f be an element of $k((x))[\delta]$. Let $e_1, \ldots, e_p \in E$ be a list of representatives of all exponential parts in $E/\sim$ of f. Then*

$$f = \mathrm{LCLM}(S_{-e_1}(R_{e_1}), \ldots, S_{-e_p}(R_{e_p})).$$

Remark: A similar statement (expressed in the terminology of D-modules) is given in corollaire 4.3.1 in (Malgrange, 1979). There is, however, a subtle but important difference namely that in our lemma the operators R_i are semi-regular instead of regular singular. To this difference corresponds a different notion of exponential parts as well; in corollaire 4.3.1 in (Malgrange, 1979) a notion appears which, in our terminology, can be viewed as elements of $E/\overline{k}$ instead of our $E/\sim$. One often distinguishes the two notions irregular singular and regular singular. In this paper we propose to drop the notion of regular singular as much as possible and only to make a distinction between semi-regular and not semi-regular, and measure the "non-semi-regularity" using the exponential parts in $E/\sim$. The motivation for doing this is to generalize algorithms that work for regular singular operators to the irregular singular case. In (van Hoeij, thesis) the benefits of this approach are shown.

Proof: Let $R = \mathrm{LCLM}(S_{-e_1}(R_{e_1}), \ldots, S_{-e_p}(R_{e_p}))$. Conjugation over $k((x))$ only permutes $S_{-e_1}(R_{e_1}), \ldots, S_{-e_p}(R_{e_p})$. Hence R is invariant under conjugation over $k((x))$ and so $R \in k((x))[\delta]$. $S_{-e_i}(R_{e_i})$ is a right factor of R, so R_{e_i} is a right factor of $S_{e_i}(R)$. So $N_0(R_{e_i})$ is a factor of $N_0(S_{e_i}(R))$, hence $\mu_{e_i}(R) \geq \mathrm{degree}(N_0(R_{e_i})) = \mu_{e_i}(f)$ because all roots of $N_0(R_{e_i})$ are integers divided by the ramification index. Then by theorem 6.1 we can conclude $\mathrm{order}(R) \geq \mathrm{order}(f)$. R is a right-hand factor of f because the $S_{-e_i}(R_{e_i})$ are right factors of f. Hence $f = R$.

□

This provides a method to compute a fundamental system of solutions of f. The solutions of $f = \mathrm{LCLM}(S_{-e_1}(R_{e_1}), \ldots, S_{-e_p}(R_{e_p}))$ are spanned by the solutions of $S_{-e_1}(R_{e_1}), \ldots, S_{-e_p}(R_{e_p})$. The solutions of $S_{-e_1}(R_{e_1})$ are obtained by multiplying the solutions of R_{e_1} by $\mathrm{Exp}(e_1)$ (recall that $\mathrm{Exp}(e_1) = \exp(\int \frac{e_1}{x} dx)$, cf. section 3.2). Consequently, when all e_i and R_{e_i} have been computed, then the problem of finding the solutions of f is reduced to solving semi-regular differential operators, which is a much easier problem (cf. section 8.1).

Define $\overline{R}_e$ for $e \in E$ and $f \in \overline{k((x))}[\delta]$ as the largest regular singular factor of $S_e(f)$ for which all roots of the Newton polynomial are rational numbers. Now we can show in the same way for $f \in \overline{k((x))}[\delta]$ that

$$f = \mathrm{LCLM}(S_{-e_1}(\overline{R}_{e_1}), \ldots, S_{-e_q}(\overline{R}_{e_q})) \tag{6.1}$$

where $e_1, \ldots, e_q \in E$ is a list of representatives for all $e \in E/\mathbb{Q}$ for which $\overline{\mu}_e(f) > 0$.

6.2. Irreducible elements of $k((x))[\delta]$

If $r \in \overline{k((x))}$ is a Riccati solution of $f \in k((x))[\delta]$ then the principal part $e = \mathrm{pp}(r) \in k((x))[r]$ modulo $\sim$ is an exponential part of f. Conversely, if $\mu_e(f) > 0$ then f has a Riccati solution $r_e \in k((x))[e]$ of which the principal part is e modulo $\sim$. Though there may be infinitely many such Riccati solutions, we can compute one such r_e in a canonical way. The algorithm in section 5 provides (although infinitely many different factorizations could exist) only 1 unique factorization of semi-regular operators (namely the one that has coprime index 1). This way we can compute a unique right factor $\delta - r_e$ of $S_{-e}(R_e)$ by computing a first order factor of R_e and applying S_{-e}. If $e_1 \sim e_2$ then $r_{e_1} = r_{e_2}$. So r_e is defined for exponential parts $e \in E/\sim$ of f.

Suppose $e_1 \in E$ is algebraic over $k((x))$ of degree d and suppose $\mu_{e_1}(f) > 0$. Suppose $e_1, \ldots, e_d \in \overline{k((x))}$ are the conjugates of e_1 over $k((x))$. If L is a Galois extension of $k((x))$ then conjugation over $k((x))$ is an automorphism of $L[\delta]$. So $\mu_{e_i}(f) = \mu_{e_j}(f)$ for all i, j. We can find unique right factors $\delta - r_{e_i} \in k((x))[e_i, \delta] \subset \overline{k((x))}[\delta]$ of f as just described. Then $R = \mathrm{LCLM}(\delta - r_{e_1}, \ldots, \delta - r_{e_d})$ is a right-hand factor of f. Because conjugation is an automorphism the r_{e_i} are all conjugates of r_{e_1} over $k((x))$. So the set $\{\delta - r_{e_1}, \ldots, \delta - r_{e_d}\}$ is invariant under conjugation which implies that R is invariant under conjugation over $k((x))$. Hence $R \in k((x))[\delta]$. In general

$$\mathrm{order}(\mathrm{LCLM}(f_1,\ldots,f_n)) \leq \sum_i \mathrm{order}(f_i)$$

because the order of an operator is equal to the dimension of the solution space, and the solution space of $\mathrm{LCLM}(f_1,\ldots,f_n)$ is spanned by the solutions of $f_1,\ldots,f_n$. So $\mathrm{order}(R) \leq d$. Since $\mu_{e_i}(R) = \mu_{e_1}(R) > 0$ for all $i = 1,\ldots,d$ we can conclude by theorem 6.1 that $\mathrm{order}(R) \geq d$ if all e_i represent different exponential parts. For this we must prove $e_i - e_j \notin \mathbb{Q}$ if $i \neq j$. Suppose $e_i - e_j \in \mathbb{Q}$. We now have to prove that $e_i = e_j$. The Galois group G of $k((x))[e_1,\ldots,e_d]$ over $k((x))$ acts transitively on $e_1,\ldots,e_d$. Hence $\gamma(e_i) = e_j$ for some $\gamma \in G$. If $\gamma(e_i) = e_i + (e_j - e_i)$ where $(e_j - e_i) \in \mathbb{Q}$ then $\gamma^{\#G}(e_i) = e_i + (\#G)(e_j - e_i)$. Here $\#G$ denotes the number of elements of G. However, for any finite group G and element $\gamma \in G$ the equation $\gamma^{\#G} = 1$ holds so $\gamma^{\#G}(e_i) = e_i$. Hence $(\#G)(e_j - e_i) = 0$ and $e_i = e_j$. Now we can conclude $\mathrm{order}(R) = d$. We have partly proven the following

Theorem 6.2. *$f \in k((x))[\delta]$ has an exponential part e which is algebraic over $k((x))$ of degree d if and only if f has an irreducible right-hand factor $R \in k((x))[\delta]$ of order d.*

Note: In a different terminology (normalized eigenvalues, characteristic classes and D-modules) this result is found in (Sommeling, 1993) as well.

Proof: Given an exponential part of degree d over $k((x))$ we have already shown how to construct R as $\mathrm{LCLM}(\delta - r_{e_1},\ldots,\delta - r_{e_d})$. Now we must show that R is irreducible in $k((x))[\delta]$. Suppose R has a non-trivial right-hand factor R_1 of order $d_1 < d$. By induction we can conclude that R_1 has an exponential part e which is algebraic over $k((x))$ of degree d_1. Lemma 6.3 shows that e is an exponential part of R. Then $e, e_1,\ldots,e_d$ are $d+1$ different exponential parts of R contradicting theorem 6.1. So R is irreducible.

Now suppose f has an irreducible right factor R of order d. The exponential parts of R are exponential parts of f by lemma 6.3. We will show that all exponential parts of R are conjugated over $k((x))$ and algebraic of degree d over $k((x))$. Let e_1 be an exponential part of R algebraic of degree p over $k((x))$. So the conjugates $e_1,\ldots,e_p$ are exponential parts of R and by our construction we find an irreducible factor R_1 of R of order p. Since R is irreducible we have $R_1 = R$ and hence $p = d$. Now $e_1,\ldots,e_d$ are d different exponential parts of R. Because of theorem 6.1 there cannot be more exponential parts, so all exponential parts of R are conjugated with e_1.

□

7. Coprime index > 1 factorization

How can one compute an irreducible factor of a polynomial $f \in \mathbb{Q}[y]$? A method is to compute a root r and the minimum polynomial of r. This is not the usual factorization method for the ring $\mathbb{Q}[y]$. But for the ring $k((x))[\delta]$ this idea supplies a method for the cases we have not yet treated. The role of the root is played by a Riccati solution. The analogue of the minimum polynomial for a Riccati solution r is the least common left multiple of $\delta - r$ and its conjugates. A minimum polynomial is the product of $y - r$ and its conjugates. One does not need to compute the conjugates to determine this product. The same holds for the least common left multiple. To see this write the LCLM as an operator

R with undetermined coefficients $R = a_n\delta^n + \cdots + a_0\delta^0$. Now the statement that $\delta - r$ is a right factor of R translates into a linear equation in $a_0, \ldots, a_n$. This is an equation over $k((x))[r]$. We know that all conjugated equations (which we do not compute) hold as well. Then this system of equations can be converted to a system over $k((x))$. We show how this can be done in a slightly more general situation. Suppose α is algebraic of degree d over a field K and we have an equation $b_0\alpha^0 + \cdots b_{d-1}\alpha^{d-1} = 0$ (in our application $K = k((x))$, $\alpha = r$ and the b_i are $k((x))$-linear expressions in a_i). The system formed by this linear equation and all its conjugates is equivalent with $b_0 = b_1 = \cdots = b_{d-1} = 0$. The reason is that the transition matrix (which is a Vandermonde matrix) between these two systems of linear equations is invertible.

This method for computing R is not very efficient for two reasons. The right-hand factor R is computed by solving linear equations over $k((x))$ which is rather complicated. The computation of these linear equations involves an algebraic extension over $k((x))$. So we prefer to lift R with the algorithm in section 4 whenever possible.

Example:

$$f = \delta^4 + 2\delta^3 - \frac{2}{x}\delta^2 + \frac{9}{4x} + \frac{1}{x^2} \in k((x))[\delta]$$

The exponential parts are $e_1 = \frac{1}{\sqrt{x}} + \frac{\sqrt{-1}}{2}$ in $E/\sim$ and the conjugates e_2, e_3, e_4 of e_1 over $\mathbb{Q}((x))$. If $\sqrt{-1} \notin k$ then e_1 is algebraic of degree 4 over $k((x))$ and then f is irreducible in $k((x))[\delta]$. Now assume that $\sqrt{-1} \in k$. Then e is algebraic of degree 2 over $k((x))$ and hence f has an irreducible right-hand factor $R \in k((x))[\delta]$ of order 2. To e_1 corresponds the following right-hand factor in $\overline{k((x))}$

$$r = \delta - x^{-1/2} - \frac{\sqrt{-1}}{2}x^0 + (-\frac{27}{80} - \frac{3\sqrt{-1}}{40})x^{1/2} + (\frac{1587}{12800} + \frac{4141\sqrt{-1}}{12800})x^1 + \cdots$$

Write $R = \delta^2 + a_1\delta + a_0$ where $a_0, a_1 \in k((x))$ are to be determined. Dividing R by r results in a remainder of the form $a_0b_{0,0} + a_1b_{0,1} + b_{0,2} + x^{1/2}(a_0b_{1,0} + a_1b_{1,1} + b_{1,2})$ for some $b_{i,j} \in k((x))$. By equating this to zero, the following linear equations are obtained: $a_0b_{i,0} + a_1b_{i,1} + b_{i,2} = 0$, $i = 1, 2$. Solving these equations over $k((x))$ gives

$$\begin{aligned} R \quad = \quad & \delta^2 + \left((\frac{1}{2} - \sqrt{-1})x^0 + (-\frac{573}{6400} + \frac{3661\sqrt{-1}}{6400})x^1 + \cdots\right)\delta^1 + \\ & \left(-x^{-1} + (-\frac{2\sqrt{-1}}{5} - \frac{37}{40})x^0 + (-\frac{12291\sqrt{-1}}{64000} + \frac{48663}{64000})x^1 + \cdots\right)\delta^0. \end{aligned}$$

It is not efficient to compute many coefficients of a_0, a_1 in this way. It suffices to determine R in this way up to accuracy 2 (i.e. to determine the coefficient of x^0 in a_1 and the coefficient of x^{-1} in a_0). Then the higher terms can be computed more efficiently by the lift algorithm in section 4.

8. Formal solutions of differential equations

8.1. Solutions of semi-regular equations

Let $f \in k((x))[\delta]$ be a semi-regular operator of order $n \geq 1$. Then we can apply section 5 to factor $f = L(\delta - r)$ where r is an element of $\mathbb{Z} + x \cdot k[[x]]$. $S_r(f) = S_r(L)\delta$. We can recursively compute a fundamental system of solutions $a_1, \ldots, a_{n-1} \in k((x))[\log(x)]$

of $S_r(L)$. Define $s_i = \int \frac{a_i}{x}dx$ for $i = 1, \ldots, n-1$ and $s_n = 1$. Then $s_1, \ldots, s_n$ is a fundamental system of solutions of $S_r(f)$. These s_i are elements of $k((x))[\log(x)]$ because $a_i/x \in k((x))[\log(x)]$ and every element of $k((x))[\log(x)]$ has an anti-derivative in this ring. By requiring that the coefficients of $x^0\log(x)^0$ in $s_1, \ldots, s_{n-1}$ are 0 the s_i are uniquely defined. To obtain the solutions of f we multiply the solutions of $S_r(f)$ by $t = \mathrm{Exp}(r) = \exp(\int \frac{r}{x}dx)$. This $t \in k((x))$ can be computed efficiently as follows. If r is written as $m \in \mathbb{Z}$ plus an element of $x \cdot k[[x]]$ then t can be written as $x^m + t_{m+1}x^{m+1} + t_{m+2}x^{m+2} + \cdots$. The the fact that t is a solution of $\delta - r$ gives a linear equation for t_{m+1}, after solving it we find an equation for t_{m+2}, etcetera.

The same method can also be used for an element f of $L[\delta]$ which is semi-regular over L, where L is an algebraic extension of $k((x))$, for the same reason as in the remark in section 5. This way a uniquely defined basis of solutions $s_1, \ldots, s_n \in L[\log(x)]$ can be computed. By theorem 8.1 in section 8.3 (first apply the theorem to $k((x))[\delta]$, then generalize using the remark in section 5) it follows that f is semi-regular over L if and only if f has a fundamental system of solutions in $L[\log(x)]$.

8.2. The canonical basis of solutions

Let $e_1, \ldots, e_r \in E$ be representatives for the exponential parts of f. Computing e_i and the corresponding semi-regular parts R_{e_i} can be done by the algorithm in section 8.4. Note that the algorithm only computes the e_i up to conjugation over $k((x))$. This means that the formal solutions will also be computed up to conjugation over $k((x))$, i.e. if a number of solutions are conjugated then only one of them will be computed.

The semi-regular $R_{e_i} \in k((x))[e_i, \delta]$ has a basis of solutions $s_{i,j} \in k((x))[e_i, \log(x)]$. So according to section 6.1 we get a basis of solutions of the form

$$y = \mathrm{Exp}(e_i)s_{i,j} \quad \text{where} \quad e_i \in E \quad \text{and} \quad s_{i,j} \in k((x))[e_i, \log(x)] \tag{8.1}$$

(recall that $\mathrm{Exp}(e_i) \in V$ stands for $\exp(\int \frac{e_i}{x}))$. In the LCLM factorization in lemma 6.4 the $S_{-e_i}(R_{e_i})$ are uniquely determined. Furthermore a unique basis of solutions for semi-regular operators was defined in the previous section. As a consequence, the basis of solutions obtained in this way is uniquely defined. We will call this basis the *canonical basis of solutions.*

For a solution in the form (8.1) $s_{i,j}$ is called the *semi-regular part* of (8.1) and e_i is called the *exponential part* of (8.1). The exponential part of (8.1) is an exponential part of the operator as well. The semi-regular part $s_{i,j}$ is a solution of the semi-regular part R_{e_i}. Note that from a given y in the form (8.1), e_i can be determined modulo $\sim$ (without further restrictions on $s_{i,j}$ one cannot determine $e_i \in E$ from y because when replacing for example e_i by $e_i - 1$ and $s_{i,j}$ by $x \cdot s_{i,j}$ in y we obtain an equivalent expression).

A few introductory comments on the next section: Every $f \in \overline{k((x))}[\delta]$ is an element of some $L[\delta]$ where L is a finite extension of $k((x))$. By a suitable transformation $\theta_{a,d}$ as in the remark in section 5 the problem of finding solutions of f can be reduced to finding solutions of an operator $\theta_{a,d}(f) \in l((x))[\delta]$. The solutions of f can be obtained from the solutions of $\theta_{a,d}(f)$. But the elements of the basis of solutions that we find for f are not necessarily in the form (8.1) (in other words: are not necessarily an element of some V_e) but are element of some $\overline{V}_e$, definitions follow in the next section.

Example: $\delta - \sqrt{x}/(2 + 2\sqrt{x})$. Apply $\theta_{1,2}$ to obtain $\frac{1}{2}\delta - \frac{1}{2}x/(1+x)$. A basis for the solutions is $1+x$. This is of the form (8.1) with $e = 0$. Now apply an inverse transformation to find the solution $1 + \sqrt{x}$ of f. This is not of the form (8.1) but it is a sum of two terms of the form (8.1), one with $e = 0$ and one with $e = 1/2$. This example shows that the direct sum decomposition $V(f) = \bigoplus V_e(f)$ in theorem 8.1 in the next section which holds for $f \in k((x))[\delta]$ need not hold for $f \in \overline{k((x))}[\delta]$. For $f \in \overline{k((x))}[\delta]$ a less precise statement is given in theorem 8.1, corresponding to the less precise version $\overline{\mu}$ of exponential parts.

8.3. The solution space and exponential parts

Definition 8.1. *Define for $e \in E$ the set*

$$\overline{V}_e = \mathrm{Exp}(e) \cdot \overline{k((x))}[\log(x)] \subset V$$

and

$$V_e = \mathrm{Exp}(e) \cdot \left((\overline{k} \cdot k((x))[e])[\log(x)]\right) \subset \overline{V}_e$$

If $e_1 \sim e_2$ then $V_{e_1} = V_{e_2}$ so V_e is also defined for $e \in E/\sim$. Similarly $\overline{V}_e$ is defined for $e \in E/\mathbb{Q}$. Define

$$V_e(f) = V_e \bigcap V(f) \quad \text{and} \quad \overline{V}_e(f) = \overline{V}_e \bigcap V(f).$$

Note that $\overline{k} \cdot k((x))[e] = \overline{k} \cdot k((x^{1/n}))$ where $n = \mathrm{ram}(e)$. The reason for writing $\overline{k} \cdot k((x^{1/n}))$ instead of $\overline{k}((x^{1/n}))$ is that in general (namely if $k \neq \overline{k}$) the field $\overline{k}((x^{1/n}))$ is not a subfield of $\overline{k((x))}$.

Theorem 8.1. *For non-zero $f \in k((x))[\delta]$*

$$V(f) = \bigoplus_e V_e(f) \quad \text{and} \quad \dim(V_e(f)) = \mu_e(f)$$

where the sum is taken over all $e \in E/\sim$. For non-zero $f \in \overline{k((x))}[\delta]$

$$V(f) = \bigoplus_e \overline{V}_e(f) \quad \text{and} \quad \dim(\overline{V}_e(f)) = \overline{\mu}_e(f)$$

where the sum is taken over all $e \in E/\mathbb{Q}$.

This theorem enables us to give an alternative definition of exponential parts and their multiplicities $\mu_e(f)$ in terms of the solution space of f.

Proof: Let $f \in k((x))[\delta]$. Each element of the basis of solutions in the previous section is an element of some V_e where e is an exponential part of f. So the sum of the $V_e \bigcap V(f)$ contains a complete basis of solutions of f. In this basis of solutions, $\mu_e(f)$ elements are in the form (8.1), i.e. $\mu_e(f)$ elements are in $V_e(f)$. Hence

$$V(f) = \sum_e V_e(f) \quad \text{and} \quad \dim(V_e(f)) \geq \mu_e(f)$$

where the sum is taken over all exponential parts of f. It follows from the following lemma 8.1 that this is a direct sum. Then $\mathrm{order}(f) = \dim(V(f)) = \sum_e \dim(V_e(f)) \geq \sum_e \mu_e(f) = \mathrm{order}(f)$ hence the $\geq$ must be an equality. The second statement follows in the same way.

□

Lemma 8.1.

$$V = \bigoplus_{e\in E/\sim} V_e \qquad \text{and} \qquad V = \bigoplus_{\overline{e}\in E/\mathbb{Q}} \overline{V}_{\overline{e}}$$

Proof: Let $n \in \mathbb{N}$. Then $\overline{k((x))} = \bigoplus_q \mathrm{Exp}(q)\cdot(\overline{k}\cdot k((x^{1/n})))$ where the sum is taken over all $q \in \mathbb{Q}$ with $0 \le q < 1/n$. So for $\overline{e} \in E/\mathbb{Q}$

$$\overline{V}_{\overline{e}} = \bigoplus_{e} V_e$$

where the sum is taken over all $e \in E/\sim$ such that $\overline{e} = e \bmod \mathbb{Q}$. This reduces the first direct sum to the second one. Because of the relations $\mathrm{Exp}(e_1)\mathrm{Exp}(e_2) = \mathrm{Exp}(e_1+e_2)$ every element of V can be written as a polynomial in the $\mathrm{Exp}(e)$ of degree 1. So $V = \sum_{\overline{e}} \overline{V}_{\overline{e}}$. We will show that this is a direct sum which finishes the proof of this lemma.

Let $e_1,\ldots,e_d \in E$ be different modulo $\mathbb{Q}$. Let $s_i \in \overline{k((x))}[\log(x)]$ and $s = \sum_i \mathrm{Exp}(e_i)s_i = 0$. To prove that the sum is direct we need to show that all s_i are zero. Assume that not all $s_i = 0$ and that $d > 1$ is minimal with this property. Then all $s_i \neq 0$. Now $x\frac{ds}{dx} = \sum_i \mathrm{Exp}(e_i)(e_i s_i + x s_i')$. Suppose the vectors $(s_1,\ldots,s_d)$ and $(e_1 s_1 + x s_1', \ldots, e_d s_d + x s_d')$ are linearly independent over $\overline{k((x))}(\log(x))$. Then we can find a linear combination in which at least one (but not all) of the components vanishes. This contradicts the fact that d is minimal (multiply with a suitable element of $\overline{k((x))}[\log(x)]$ to eliminate $\log(x)$ from the denominator). So these two vectors must be linearly dependent over $\overline{k((x))}(\log(x))$. It follows that

$$\frac{e_1 s_1 + x s_1'}{s_1} = \frac{e_2 s_2 + x s_2'}{s_2} \in \overline{k((x))}(\log(x))$$

so

$$e_2 - e_1 = x s_1'/s_1 - x s_2'/s_2 = x b'/b$$

where $b = s_1/s_2 \in \overline{k((x))}(\log(x))$. But $e_2 - e_1 \in E$ and $e_2 - e_1 \notin \mathbb{Q}$ which contradicts lemma 8.2.

□

Lemma 8.2. *Let $b \in k((x^{1/n}))(\log(x))$. Suppose that the logarithmic derivative $c = xb'/b$ is an element of $\overline{k((x))}$. Then $c \in \frac{1}{n}\mathbb{Z} + x^{1/n}\cdot k[[x^{1/n}]]$.*

Proof: Write $b = p/q$ with $p,q \in k((x^{1/n}))[\log(x)]$. Write $p = p_l \log(x)^l + \cdots$ and $q = q_m \log(x)^m + \cdots$ where $p_l, q_m \in k((x^{1/n}))$. The dots stands for an element of $k((x^{1/n}))[\log(x)]$ of lower degree as a polynomial in $\log(x)$.

$$c = \frac{xb'}{b} = \frac{xp'}{p} - \frac{xq'}{q} = \frac{xp_l'\log(x)^l + \cdots}{p_l\log(x)^l + \cdots} - \frac{xq_m'\log(x)^m + \cdots}{q_m\log(x)^m + \cdots}$$

$$= \frac{x(p_l'q_m - q_m'p_l)\log(x)^{l+m} + \cdots}{p_l q_m \log(x)^{l+m} + \cdots} \in \overline{k((x))}.$$

Then $x(p_l'q_m - q_m'p_l)/(p_l q_m)$ must be the same element c of $\overline{k((x))}$. Write $r = p_l/q_m \in k((x^{1/n}))$. Then $c = x(p_l'q_m - q_m'p_l)/(p_l q_m) = xr'/r \in \frac{1}{n}\mathbb{Z} + x^{1/n}\cdot \overline{k}[[x^{1/n}]]$.

□

8.4. Coprime index 1 LCLM factorization

Lemma 8.3. *Let $f_1, \ldots, f_d \in k((x))[\delta]$, $e \in E/\sim$ and $f = \mathrm{LCLM}(f_1, \ldots, f_d)$. Then*

$$\max_i \, \mu_e(f_i) \leq \mu_e(f) \leq \sum_i \mu_e(f_i).$$

In particular every exponential part e of f is an exponential part of at least one of the f_i.

Proof: These inequalities follow from the dimensions of $V_e(f)$ and $V_e(f_i)$ in the following equation: $V_e(f) = V_e \bigcap (\sum_i V(f_i)) = \sum_i V_e(f_i)$. The second equality holds because the $V(f_i)$ are direct sums of $V(f_i) \bigcap V_{e_1}$ taken over all $e_1 \in E/\sim$.

□

Lemma 8.4. *Let $f \in k((x))[\delta]$ be monic and let $f_1, \ldots, f_d \in k((x))[\delta]$ be right hand factors of f. Suppose that $\sum_i \mathrm{order}(f_i) = \mathrm{order}(f)$ and that the f_i have no exponential parts in common. Then*

- $f = \mathrm{LCLM}(f_1, \ldots, f_d)$
- *If $e \in E/\sim$ and $\mu_e(f) > 0$ then there is precisely one f_i such that $V_e(f) \subset V(f_i)$.*
- *For this e and f_i the semi-regular part R_e of f is the semi-regular part of f_i as well.*

Proof: Using the previous lemma, the fact that the f_i have no exponential part in common and theorem 6.1 we can conclude that $\mathrm{order}(\mathrm{LCLM}(f_1, \ldots, f_d)) = \sum \mathrm{order}(f_i)$, and this equals $\mathrm{order}(f)$ by the assumption in this lemma. Since all f_i, and hence this LCLM, are right-hand factors of f the first statement follows. If e is an exponential part of f then for precisely one i we have $\mu_e(f_i) > 0$. Then $\mu_e(f_i) = \mu_e(f)$ because of the previous lemma and because the μ_e of the other f_j are zero. For the second statement note that $V_e(f_i) \subset V_e(f)$, because f_i is a right-hand factor of f. Since $\mu_e(f_i) = \mu_e(f)$ the dimensions are the same. Hence $V_e(f) = V_e(f_i) \subset V(f_i)$. The third statement follows because $V(S_{-e}(R_e)) = V_e(f) \subset V(f_i)$, hence $S_{-e}(R_e)$ is a right-hand factor of f_i and so R_e is a right-hand factor of $S_e(f_i)$.

□

Lemma 8.5. *Let $f, g \in k((x))[\delta]$ and suppose $\gcd(N_s(f), N_s(g)) = 1$ holds for all $s \in \mathbb{Q}$, $s > 0$. Suppose $\gcd(N_s(f), S_{T=T+n}(N_s(g))) = 1$ holds for $s = 0$ and all $n \in \mathbb{Z}$. Then f and g have no exponential parts in common.*

Proof: For every exponential part e of f there exists a Riccati solution r_e of f such that e is the principal part of r_e modulo $\sim$, cf. section 6.2. Now the proof follows from the next lemma.

□

LEMMA 8.6. *Let $r \in \overline{k((x))}$ be a Riccati solution of $f \in k((x))[\delta]$. Suppose r, viewed as an element of $\bigcup_n \overline{k}((x^{1/n}))$, can be written as $r_s x^s$ plus higher order terms, where $s \in \mathbb{Q}$, $s \leq 0$ and $r_s \neq 0$ if $s < 0$. Write $s = n/d$ with $n, d \in \mathbb{Z}$, $\gcd(n,d) = 1$ and $d > 0$. Then $-s$ is a slope of f and r_s^d is a root of the Newton polynomial $N_{-s}(f)$.*

Proof: $\delta - r$ is a right-hand factor of f. If the ramification index of r is 1 the lemma can easily be proved using the fact that the slopes of factors of f are slopes of f and the Newton polynomials of right factors of f are factors of the Newton polynomials of f, cf. section 3.4. However, we have not defined the Newton polygon and Newton polynomial over ramifications of $k((x))$. Choose $d' \in \mathbb{N}$ such that $\theta_{1,d'}(\delta - r) \in \overline{k}((x))[\delta]$. Then d must divide d'. Now $\theta_{1,d'}(\delta - r)$ is a right-hand factor of $\theta_{1,d'}(f)$. The slope of $\theta_{1,d'}(\delta - r)$ is $-sd'$ so $\theta_{1,d'}(f)$ has this slope as well. Hence f has a slope $-s$. The Newton polynomial of $\theta_{1,d'}(\delta - r)$ is $\frac{1}{d'}T - r_s$. If $N_{-s}(f) = c(T^p + a_{p-1}T^{p-1} + \cdots + a_0T^0)$ where c is a constant then $N_{-sd'}(\theta_{1,d'}(f))$ is a constant times $T^{pd} + {d'}^d a_{p-1}T^{(p-1)d} + \cdots + {d'}^{pd} a_0 T^0$. So $\frac{1}{d'}T - r_s$ is a factor of this Newton polynomial hence r_s^d is a root of $T^p + a_{p-1}T^{p-1} + \cdots + a_0T^0$.

□

Now we can write *algorithm LCLM factorization* as follows. Take algorithm Coprime Index 1 Factorizations in section 5. Replace the lines

```
if $s = 0$ then
    $M := \{g_1, \ldots, g_r\}$
    $N := M \setminus \{g | g(T) = h(T+i), h \in M, i \in \mathbb{N}, i > 0\}$
else
```

by the lines

```
if $s = 0$ then
    $M := \{g_1, \ldots, g_r\}$
    $M' := M \setminus \{g | g(T) = h(T+i), h \in M, i \in \mathbb{N}, i > 0\}$
    $M'' := \{\{n | \exists_{i \in \mathbb{Z}} g_n(T+i) = h\} | h \in M'\}$
    $N := \{\prod_{i \in h} g_i^{e_i} | h \in M''\}$
else
```

The resulting algorithm produces a number of factorizations. The sum of the orders of the right factors is equal to the order of f. The different right factors $f_1, \ldots, f_d$ have no exponential parts in common because of lemma 8.5. Hence $f = \mathrm{LCLM}(f_1, \ldots, f_d)$. This variant on the algorithm in section 5 produces an *LCLM factorization*, i.e. it produces a number of right-hand factors $f_1, \ldots, f_d$ such that $f = \mathrm{LCLM}(f_1, \ldots, f_d)$. The orders of the f_i need not be minimal because we only apply the "easy" (i.e. coprime index 1) factorization method.

Algorithm semi-regular parts:
Input: $f \in k((x))[\delta]$
Output: representatives $e_1, \ldots, e_d \in E$ for all exponential parts up to conjugation over $k((x))$ and the corresponding semi-regular parts $R_{e_i} \in k((x))[e_i, \delta]$.

1 Same as case 1 in Algorithm Riccati solution. This is also a special case of case 6 below after a suitable substitution map S_e.
2 If algorithm LCLM factorization produces a non-trivial (i.e. $d > 1$) LCLM factorization $f = \mathrm{LCLM}(f_1, \ldots, f_d)$ then apply recursion to the right factors $f_1, \ldots, f_d$.
3 If the condition of case 3 in Algorithm Riccati solution holds, and furthermore the slope of f is non-zero, then proceed as in case 3 in Algorithm Riccati solution; apply recursion to the right-hand factor.
4 Same as case 4 in Algorithm Riccati solution, apply recursion to $S_{cx^{-s}}(f)$.
5 Same as case 5 in Algorithm Riccati solution, apply recursion on R.
6 If f has one slope $s = 0$ and the Newton polynomial has the following form $N_s(f) = g \cdot S_{T=T+i_1}(g) \cdots S_{T=T+i_n}(g)$ where $n \geq 0$ and i_j are integers, and g is an irreducible polynomial. Let $r \in \overline{k}$ be a root of g. Extend the field k with r (note that g can have degree 1 in which case $r \in k$). Define $h = T \cdot (T + i_1) \cdots (T + i_n)$. This is the largest factor of $N_0(S_r(f))$ which has only integer roots. Now use a coprime index 1 factorization (cf. Algorithm Coprime Index 1 Factorizations in section 5) to compute a right factor R of $S_r(f)$ that has Newton polynomial h.

The right-hand factors R that this algorithm produces in case 6 are the semi-regular parts of f (actually such R is an image of a semi-regular part under certain maps $\theta_{a,d}$ that were used in case 5). The corresponding exponential parts are obtained by keeping track of the substitution maps S_e and ramification maps $\theta_{a,d}$ that were performed. The recursion in case 2 of the algorithm is valid because of lemma 8.4.

In the cases 3 and 5 of the algorithm a field extension over $k((x))$ is applied (also in case 6 if degree$(g) > 1$ but the argument is almost the same for this case). Suppose the degree of the of this field extension is d. Then the algorithm computes a right factor f_1 of f and uses recursion on this right factor. Let $f_1, \ldots, f_d \in L[\delta]$ be the conjugates of f_1 over $k((x))$ where L is some finite extension of $k((x))$. Lemma 8.5 and lemma 8.4 were formulated for $k((x))[\delta]$ instead of $L[\delta]$, but they are still applicable when using the less precise notion of exponential parts $\overline{\mu}$. We must replace the condition "for all $n \in \mathbb{Z}$" by "for all $n \in \mathbb{Q}$" in lemma 8.5 in order for this lemma to hold for the case of $\overline{\mu}$ instead of μ. So our algorithm would produce all exponential parts and semi-regular parts if we would use recursion on not only f_1 but also on $f_2, \ldots, f_d$. However, this could introduce very large algebraic field extensions (worst case d factorial) which could make the algorithm too slow to be useful. If we would use recursion on $f_2, \ldots, f_d$ we will only find conjugates of the exponential parts and semi-regular parts that are obtained from f_1. So there is no need to do the recursion on $f_2, \ldots, f_d$ because the result of that computation can also be obtained as the conjugates (which are not computed, however) of the output of the recursion on f_1.

Algorithm formal solutions:
Input: $f \in k((x))[\delta]$
Output: a basis of solutions, up to conjugation over $k((x))$
Step 1: this is the main step: apply algorithm semi-regular parts.
Step 2: compute the solutions $s_{i,j}$ of R_{e_i} as in section 8.1.
Step 3: Return the set of $\mathrm{Exp}(e_i)s_{i,j}$.

Our method for computing formal solutions cannot avoid the use of field extensions over $k((x))$ because these field extensions appear in the output. It does, however, delay the use of algebraic extensions as long as possible. The use of algorithm LCLM factorization reduces the problem of finding solutions of f to operators of smaller order. This way the order of the operator is as small as possible at the moment that an algebraic extension is introduced, and so the amount of computation in algebraic extensions is minimized. Lazy evaluation is used to minimize the number of operations in the field of constants.

9. A characterization of the solution spaces

The symbol $\log(x)$ is viewed as an element of a differential extension of $k((x))$ which satisfies the equation $y' = 1/x$. The corresponding linear differential equation is $y'' + \frac{1}{x}y' = 0$. We do not view $\log(x)$ as a function on an open subset of the complex plane, but as a formal expression which is defined by the property that the derivative is $1/x$. From this viewpoint it is clear that the $\overline{k((x))}$-homomorphism

$$S_{\log} : \overline{k((x))}[\log(x)] \to \overline{k((x))}[\log(x)]$$

defined by $S_{\log}(\log(x)) = \log(x) + 1$ is a differential automorphism, because the derivative of $\log(x) + 1$ is also $1/x$, and hence all differential properties of $\log(x) + 1$ and $\log(x)$ are the same. This automorphism can be extended to the ring V by defining $S_{\log}(\mathrm{Exp}(e)) = \mathrm{Exp}(e)$. If $f \in V[\delta]$ and $y \in V$ is a solution of f then $S_{\log}(y)$ is a solution of $S_{\log}(f)$. Note that the differential Galois group G of the Picard-Vessiot extension $\overline{k((x))} \subset \overline{k((x))}(\log(x))$ contains more elements than just $S_{\log}$. However, we will see that it is sufficient to consider only $S_{\log}$. This is explained from the fact that G is equal to the Zariski closure of the group generated by $S_{\log}$.

Let $f \in k((x))[\delta]$. The questions of this section are: what are the possible right-hand factors of f in $k((x))[\delta]$, or in $\overline{k((x))}[\delta]$, what are the semi-regular and regular right factors. Every right factor R corresponds to a subspace of solutions $V(R) \subset V(f)$. But not every linear subspace $W \subset V(f)$ corresponds to a right factor of f because we do not look for right factors in $V[\delta]$ but only in smaller rings like $k((x))[\delta]$. So the question now is the following. Given a finite dimensional $\overline{k}$-vector space $W \subset V$, when is W the solution space of either

1 a semi-regular operator in $k((x))[\delta]$
2 a regular operator in $k((x))[\delta]$
3 any operator in $\overline{k((x))}[\delta]$
4 any operator in $k((x))[\delta]$.

Example: Let $\log(x)$ be a basis of W. Now there cannot be any $f \in \overline{k((x))}[\delta]$ such that $W = V(f)$. Because then $S_{\log}(\log(x))$ would be a solution of $S_{\log}(f) = f$. So f has $\log(x)$ and $S_{\log}(\log(x)) - \log(x) = 1$ as solutions. Hence the dimension of $V(f)$ is at least 2.

LEMMA 9.1. *Let W be a n dimensional $\overline{k}$-subspace of V. Then $W = V(f)$ for some semi-regular $f \in k((x))[\delta]$ if and only W has a basis $b_1, \ldots, b_n \in k((x))[\log(x)]$ and $S_{\log}(W) = W$.*

Proof: Let $f \in k((x))[\delta]$ be semi-regular. Then it follows from section 8.1 that $V(f)$ has a basis of solutions in $k((x))[\log(x)]$. Furthermore $S_{\log}(V(f)) = V(S_{\log}(f)) = V(f)$.

Now suppose $S_{\log}(W) = W$ and suppose $b_1, \ldots, b_n \in k((x))[\log(x)]$ is a basis of W as a $\overline{k}$-vector space. We want to construct a semi-regular operator f such that $V(f) = W$.

Let b be an element of W of minimal degree d as a polynomial in $\log(x)$. Suppose $d > 0$. Then $S_{\log}(b) - b \in W$ has degree $d-1$ which contradicts the minimality of d. Hence $d = 0$, so b is an element of $\overline{k} \cdot k((x))$. Then $b \in l \cdot k((x))$ where l is some finite extension of k. After multiplication by a constant we may assume that one of the coefficients of b is 1. Then, by taking the trace over the field extension $k \subset l$, we may assume $b \in k((x))$ and $b \in W$ (use here that W has a basis of elements in $k((x))[\log(x)]$, hence the trace over k of an element $b \in W$ is an element of W). Now $b \neq 0$ because the trace of the coefficient 1 is not 0. Because $b \in V(f)$ for the operator f that we want to construct it follows that $R = \delta - xb'/b$ must be a right factor of f. This operator R is a $\overline{k}$-linear map from V to V. The kernel is the solution space of R. It has dimension 1. Because the kernel is a subspace of W it follows that $\dim(R(W)) = n-1$. It is easy to check that $R(W)$ satisfies the conditions of this lemma, hence by induction there is a semi-regular operator $L \in k((x))[\delta]$ such that $V(L) = R(W)$. Now define $f = LR$. This is a semi-regular operator in $k((x))[\delta]$ because $L, R \in k((x))[\delta]$ are semi-regular. $f(W) = L(R(W)) = \{0\}$ and $\dim(W) = \mathrm{order}(f)$ so $V(f) = W$.

□

From the remark in section 5 it follows that the lemma is also valid when $k((x))$ is replaced by a finite extension L of $k((x))$.

Lemma 9.2. *Let W be a n dimensional $\overline{k}$-subspace of V. Then $W = V(f)$ for some regular $f \in k((x))[\delta]$ if and only W has a basis $b_1, \ldots, b_n \in k[[x]]$ and all non-zero elements of W have valuation $< n$.*

Proof: If $f \in k((x))[\delta]$ is regular it is known by the Cauchy theorem that there exists a basis $b_1, \ldots, b_n \in k[[x]]$ of solutions such that b_i is x^{i-1} modulo x^n. It is easy to compute these b_i as follows. The equation $f(b_i) = 0$ (writing f as an element of $k[[x]][\partial]$ is more convenient for this) gives a linear equation in the coefficient of x^n in b_i, a linear equation for the coefficient of x^{n+1}, etcetera. From these equations the coefficients of b_i can be computed.

To prove the reverse statement let $b_1, \ldots, b_n \in k[[x]]$ be a basis of W and suppose that all non-zero elements of W have valuation (i.e. the smallest exponent of x which has a non-zero coefficient) smaller than n. Then, after a basis transformation, we may assume that b_i is x^{i-1} modulo x^n. Now define $R_1 \in k[[x]][\partial]$ as $R_1 = \partial - b_1'/b_1$. Define for $1 \leq d < n$ the operator $R_{d+1} \in k[[x]][\partial]$ as follows: define $y_{d+1} = R_d(b_{d+1})$. Note that $v(R_i(b_{d+1})) = d - i$ for $1 \leq i \leq d$. So $v(y_{d+1}) = 0$ and hence $\partial - y_{d+1}'/y_{d+1} \in k[[x]][\partial]$. Now define $R_{d+1} = (\partial - y_{d+1}'/y_{d+1})R_d$. Now $f = R_n$ is a monic element of $k[[x]][\partial]$, hence regular, with $V(f) = W$.

□

From the lemma we see that right factors of regular operators need not be regular. Suppose for example that $1, x, x^2$ is a basis of solutions of f. Then the right-hand factor given by the basis of solutions $1, x^2$ is not regular. But the right factor with the basis $1, x + x^2$ is regular. An LCLM of regular operators is not necessarily regular either. For certain purposes (not for all) semi-regular is a more convenient notion than regular because factors, products, LCLM's and symmetric products of semi-regular operators are semi-regular.

If $W \subset V$ is a solution space of a differential operator $f \in \overline{k((x))}[\delta]$ then $W = \bigoplus_{\overline{e}}(W \bigcap \overline{V}_{\overline{e}})$ because of theorem 8.1. Furthermore $S_{\log}(W)$ must equal W because f is invariant under $S_{\log}$. This proves one part of the following lemma.

Lemma 9.3. *Let W be a finite dimensional $\overline{k}$-subspace of V. Then $W = V(f)$ for some $f \in \overline{k((x))}[\delta]$ if and only $W = \bigoplus_{\overline{e}}(W \bigcap \overline{V}_{\overline{e}}) = S_{\log}(W)$ where the sum is taken over all $\overline{e} \in E/\mathbb{Q}$.*

Proof: Assume $W \neq \{0\}$ is finite dimensional and $W = \bigoplus_{\overline{e}}(W \bigcap \overline{V}_{\overline{e}}) = S_{\log}(W)$. Let $e \in E$ such that $W_e = W \bigcap \overline{V}_e \neq \{0\}$. Note that $S_{\log}(\overline{V}_e) = \overline{V}_e$ hence W_e is invariant under $S_{\log}$. W_e has a basis of the form $\mathrm{Exp}(e) \cdot s_i$, $i = 1, \ldots, t$ where $s_i \in \overline{k((x))}[\log(x)]$ so $s_i \in L[\log(x)]$ for some finite extension L of $k((x))$. Using lemma 9.1 it follows that there exists an operator $R_e \in L[\delta]$ which has s_i, $i = 1, \ldots, t$ as a basis of solutions. So $S_{-e}(R_e)$ has $\mathrm{Exp}(e)s_i$, $i = 1, \ldots, t$ as a basis of solutions and so $S_{-e}(R_e)$ must be a right-hand factor of the operator f that we want to construct. Choose a representative $e \in E$ for every $\overline{e} \in E/\mathbb{Q}$ for which $W \bigcap \overline{V}_{\overline{e}} \neq \{0\}$. Construct the corresponding $S_{-e}(R_e)$ and define f as the LCLM of these $S_{-e}(R_e)$. Then $V(f) = W$.

□

Lemma 9.4. *Let W be a finite dimensional $\overline{k}$-subspace of V. Then $W = V(f)$ for some $f \in k((x))[\delta]$ if and only the conditions of the previous lemma hold, and furthermore W is invariant under the action of the Galois group of the algebraic extension $k((x)) \subset \overline{k((x))}$.*

Proof: if τ is a $k((x))$-automorphism of $\overline{k((x))}$ then τ can be extended to V by setting $\tau(\log(x)) = \log(x)$ and $\tau(\mathrm{Exp}(e)) = \mathrm{Exp}(\tau(e))$. Now for any $f \in \overline{k((x))}[\delta]$ we have $V(\tau(f)) = \tau(V(f))$ because conjugation commutes with differentiation. So if $f \in k((x))[\delta]$ then $V(f) = \tau(V(f))$ which proves one part of the lemma. Now suppose $W = V(f)$ for some monic $f \in \overline{k((x))}[\delta]$ and suppose that $W = \tau(W)$. Now $\mathrm{order}(f - \tau(f)) < \mathrm{order}(f)$ and $W \subset V(f - \tau(f))$ so $\dim(V(f - \tau(f))) > \mathrm{order}(f - \tau(f))$ and hence $f - \tau(f)$ must be 0. So if W is invariant under the Galois group of the algebraic extension $k((x)) \subset \overline{k((x))}$ then f is invariant as well, hence $f \in k((x))[\delta]$.

□

Every $y \in V$ is a finite sum $y = \sum_e b_e$ with $b_e \in V_e$. Define W as the closure under Galois actions and under $S_{\log}$ of the set $\sum_e \overline{k} \cdot b_e$. Now W satisfies the conditions of the previous lemma, hence for every $y \in V$ there is a $g \in k((x))[\delta] \setminus \{0\}$ such that $y \in V(g)$. From this it follows that for any non-zero $f \in k((x))[\delta]$ the map $f : V \to V$ is surjective. This is seen as follows. If the kernel of g is not contained in the image of f then the dimension of the kernel of gf would be smaller than the sum of the dimensions of the kernels of g and f. In other words, $\mathrm{order}(gf) < \mathrm{order}(g) + \mathrm{order}(f)$ which is a contradiction. Hence $V(g) \subset f(V)$ for every g and so f is surjective, $f(V) = V$.

References

Barkatou M.A., *Rational Newton Algorithm for computing formal solutions of linear differential equations*, Proceedings of ISSAC'88, ACM Press, (1988).

Björk J.E., *Rings of Differential Operators*, North-Holland Publishing Company, (1979).

Bliss G.A., *Algebraic Functions*, Dover, (1966).

Bourbaki N., *Espaces vectoriels topologiques*, Paris, (1953).

Coddington E., Levinson N.,*Theory of ordinary differential equations*, MacGraw-Hill, (1955).

Duval A., *Lemmes de Hensel et factorisation formelle pour les opérateurs aux différences*, Funkcial Ekvac, **26**, p. 349-368, (1983).

Duval D., *Rational Puiseux expansions*, Compos. Math. 70, No. 2, p. 119-154, (1989).

Della Dora J., di Crescenzo Cl., Tournier E., *An algorithm to obtain formal solutions of a linear homogeneous differential equation at an irregular singular point*, Proc. Symp. EUROCAM '82 (Lect. Notes Comput. Sci.) **144**, 273-280, (1982).

van den Essen A., Levelt A.H.M., *An Explicit Description of all Simple* $k[[x]][\partial]$*-Modules*, Contemporary Mathematics, **130**, p. 121-131, (1992).

Hendriks P.A., van der Put M., *Galois action on solutions of a differential equation*, J. Symb. Comp., Vol 19, No. 6, p. 559-576, (1995).

van Hoeij M., *Factorization of Linear Differential Operators*, Ph.D. thesis, University of Nijmegen, http://www-math.sci.kun.nl/math/compalg/diffop/ (1996).

Levelt A.H.M., *Jordan decomposition for a class of singular differential operators*, Arkiv för matematik, 13 (1), p. 1-27, (1975).

Levelt A.H.M., *Differential Galois theory and tensor products*, Indagationes Mathematicae, 1 (4), p. 439-450, (1990).

Malgrange B., *Sur la réduction formelle des équations différentielles à singularités irrégulières*, Manuscript, (1979).

Ore O., *Theory of non-commutative polynomial rings*, Ann. of Math. **34** p. 480-508, (1933).

van der Put M., *Singular complex differential equations: an introduction*, Nieuw Achief voor Wiskunde, 4$^{\mathrm{de}}$ serie **13**, No. 3, p. 451-470, (1995).

Robba P., *Lemmes de Hensel pour les operateurs différentiels. Application a la reduction formelle des equations différentielles*, L'Enseignement Mathematique, Ser. II, **26**, p. 279-311, (1980).

Singer M.F., *Testing Reducibility of Linear Differential Operators: A Group Theoretic Perspective*, Appl. Alg. in Eng. Comm. and Comp., **7**, p. 77-104, (1996).

Sommeling R., *Characteristic classes for irregular singularities*, Ph.D. thesis, University of Nijmegen, (1993).

Tournier E., *Solutions formelles d'équations différentielles*, Thèse d'Etat, Faculté des Sciences de Grenoble, (1987).