Minimal Completely Factorable Annihilators*

Sergei A. Abramov

Computer Center of
the Russian Academy of Science
Vavilova 40, Moscow 117967, Russia
abramov@ccas.ru
sabramov@cs.msu.su

Eugene V. Zima

Dept. of Comp. Math. & Cyberns.
Moscow State University
Moscow 119899
Russia
zima@cs.msu.su

Abstract

We propose an algorithm to construct the minimal annihilating operator of a function or a sequence, when the operator is completely factorable (i.e. can be decomposed in first order factors). The algorithm is designed in the frame of the Ore rings theory and can be used in the differential, difference and q-difference cases. We describe also a Maple implementation of the algorithm.

1 Introduction

Constructing a linear ordinary differential operator annihilating a function (an annihilator of the function) is necessary when solving many computer algebra problems. We list some of these problems.

P1. Expanding a function as a power series and subsequently investigating the expansion. An annihilator lets one construct the recurrence for the series coefficients and manipulate them ([14, 17]).

P2. Solving linear inhomogeneous equations. Some methods use annihilators of the right-hand side ([4, 8]).

P3. Integrating. If the minimal annihilator L, $\operatorname{ord} L = n$, of f is given, then one can check whether there exists a primitive of f with an n-th order minimal annihilator. If yes, then it is possible to express the primitive explicitly via f ([9]).

P4. Recognizing the equivalence of two given functions. If the common annihilator of both the functions is given, then it suffices to check the agreement between the corresponding "initial conditions" (a classical approach).

The minimal annihilator, i.e. the annihilator of the lowest order, is the most informative. Note that to solve **P3** only the minimal annihilator of f is suitable. Applying algorithm [8] to an equation with a d'Alembertian right-hand side guarantees that all d'Alembertian solutions will be found only in the situation when the minimal annihilator of the right hand side, decomposed in first order factors, is given. (A function is d'Alembertian if it has a completely factorable annihilator, i.e. an annihilator, which can be decomposed in first-order factors.) In **P1**, **P4** manipulating the minimal annihilators reduces the cost of the investigation.

In this paper we propose an algorithm to search for the minimal annihilator L of a given d'Alembertian f. Our main result is an algorithm which, given an expression E, composed of d'Alembertian elements by the signs of the operations of addition and multiplication, constructs the completely factored minimal annihilator of E. It is assumed that all the d'Alembertian components of E are given with their completely factored (i.e. decomposed in first order factors) minimal annihilators. Together with the algorithm we describe its Maple implementation.

It is easy to observe that **P1** - **P4** can be considered not only in the differential case but also in the difference and the q-difference cases. The concept of Ore rings ([16, 10, 11, 12, 13]) lets one design universal algorithms which can be adjusted on one or another concrete case. The algorithm and the program described below are universal and find the minimal annihilators in all cases covered by the Ore ring approach.

2 Generalities

Let k be a field of characteristic zero, X an indeterminate over k, σ an automorphism of k, and $\delta : k \to k$ a map satisfying

$$\delta(a+b) = \delta a + \delta b,\ \delta(ab) = \sigma(a)\,\delta b + \delta a\, b \qquad (1)$$

for any $a, b \in k$. Then we can consider the *Ore ring* $k[X;\sigma,\delta]$ of polynomials in X over k with the usual polynomial addition $+$ and multiplication $\circ$ given by $X \circ a = \sigma(a)X + \delta a$ for any $a \in k$ ([16, 10, 11]).

Let K be σ,δ-*compatible* extension ring of the field k, i.e. σ can be extended to an automorphism of K and δ can be extended to a map $K \to K$ holding (1) for any $a, b \in K$. A map $\theta : K \to K$, is *pseudo-linear* with respect to σ,δ, if

$$\theta(u+v) = \theta u + \theta v,\ \theta(uv) = \sigma(u)\,\theta v + \delta u\, v.$$

for any $u, v \in K$.

We assume that the constant subring of K (i.e. the set of all $a \in K$ such that $\sigma(a) = a$, $\delta a = 0$) is a field. Furthemore, we assume that this field is equal to the constant field of k and denote it *Const*.

*Work reported herein was supported in part by RFBR under Grant 95-01-01138.

We can consider the ring $k[\theta]$ of operators $K \to K$ of the form $p(\theta), p(X) \in k[X; \sigma, \delta]$. These operators are linear over *Const*.

It is assumed that if a first-order equation $Fy = 0, F \in k[\theta]$, has a nonzero solution in a σ, δ-compatible extension of the field k, then the equation has a nonzero solution in K. These solutions form the set $\mathcal{H}_k \subset K$ of *hyperexponential* elements. Any element of $\mathcal{H}_k$ is invertible in K. An equation $Py = 0$ and the operator P are called *completely factorable* if P can be decomposed as the product of first-order operators over k. Solutions of all completely factorable equations form the linear (over *Const*) space $\mathcal{A}_k \subset K$ of *d'Alembertian elements*. It is easy to see that $\mathcal{H}_k \subset \mathcal{A}_k$.

Let $L \in k[\theta]$ and S be the space of solutions of $Ly = 0$ belonging to K. We assume that $\dim S \leq \operatorname{ord} L$ in this situation.

We will also use the following notions which have been defined in [7, 8].

- the operator ∇, which is the minimal monic operator such that $\nabla 1 = 0$ (an analog of $\frac{d}{dx}$ and Δ). It is obvious that $\nabla f = 0 \Leftrightarrow f \in Const$;
- the set $I(f)$, $f \in K$, which is an analog of the indefinite integral and sum:

  $$I(f) = \{d \mid \nabla d = f\}.$$

  If $f \in K$, $d \in I(f)$, $c_0 \in Const$, then $d + c_0 \in I(f)$ and, vice versa, for any $d_1, d_2 \in I(f)$ we have $d_1 - d_2 \in Const$. We assume that $I(f)$ is not empty for any $f \in K$. If $\mathcal{U}$ is a set of elements of K, then $I(\mathcal{U})$ denotes the set of all d such that $\nabla d \in \mathcal{U}$. We write for brevity $I(f_1, \ldots, f_m)$ instead of $f_1 I(f_2 I(f_3 \ldots f_{m-1} I(f_m) \ldots))$;
- *d'Alembertian space*

  $$I(\varphi_1, \ldots, \varphi_m, 0), \tag{2}$$

  where $\varphi_1, \ldots, \varphi_m \in \mathcal{H}_k$ is a generalization of

  $$\varphi_1 \int (\varphi_2 \int (\ldots \int (\varphi_m \int 0) \ldots))$$

  and

  $$\varphi_1 \sum (\varphi_2 \sum (\ldots \sum (\varphi_m \sum 0) \ldots)).$$

  It is easy to see that

  $$I(\varphi_1, 0) \subset I(\varphi_1, \varphi_2, 0) \subset \ldots \subset I(\varphi_1, \ldots, \varphi_m, 0).$$

  If a d'Alembertian space A is given, then we can construct the completely factorable operator over k such that A is its solution space and vice versa (see (5),(6),(7) below).

Let $\theta 1 = c \in k$, then $\theta = \theta_c = c\sigma + \delta$ and $\nabla = \theta - c$. If $\sigma \neq 1$, then we assume that there exists $\alpha \in k$ such that $\delta = \alpha(\sigma - 1)$. From now on we will use the notations α for this value only.

We will consider the following two cases:

$$\sigma = 1;\ c = 0 \tag{3}$$

and

$$\sigma \neq 1;\ c = 0, \alpha = 1. \tag{4}$$

In these two cases $\theta = \nabla$ and we will use the notation $k[\nabla]$ rather then $k[\theta]$. It is easy to show that the general cases can be reduced to the cases (3),(4). In the general case the substitution $\theta = \nabla + c$ transforms an element of $k[\theta]$ to an element of $k[\nabla]$. The substitution $\nabla = \theta - c$ returns operators to $k[\theta]$.

If $L \in k[\theta]$ then going this way we transform L to $L' \in k[\nabla]$ such that $\operatorname{ord} L = \operatorname{ord} L'$ and $Lf = L'f$ for any $f \in K$.

We denote by $\mathcal{F}_k$ the set of all completely factorable elements of $k[\nabla]$. An element of $\mathcal{F}_k$ can be presented in the form $g(\nabla - h_m) \circ \cdots \circ (\nabla - h_1)$, where $g, h_1, \ldots, h_m \in k$. We will assume (unless otherwise stated) that the leading coefficients of elements of $\mathcal{F}_k$ under consideration are equal to 1 or, equivalently, the elements are monic. Such a completely factorable operator can be written as

$$(\nabla - h_m) \circ \cdots \circ (\nabla - h_1) \tag{5}$$

with $h_1, \ldots, h_m \in k$.

Let $L \in \mathcal{F}_k$ have the form (5) and $\eta_1, \ldots, \eta_m \in \mathcal{H}_k$ be such that

$$\frac{\nabla \eta_i}{\eta_i} = h_i,\ i = 1, \ldots, m.$$

Then the general solution of the equation $L(y) = 0$ can be written (see [7]) in the form (2), where

$$\varphi_1 = \eta_1,\ \varphi_i = \frac{\eta_i}{\sigma(\eta_{i-1})},\ i = 2, \ldots, m; \tag{6}$$

and, respectively,

$$\eta_1 = \varphi_1,\ \eta_i = \sigma^{i-1}(\varphi_1)\sigma^{i-2}(\varphi_2)\ldots\sigma(\varphi_{i-1})\varphi_i,\quad i = 2, \ldots, m. \tag{7}$$

We call $L \in k[\nabla]$ an *annihilator* of $f \in K$ if $L(f) = 0$. In this paper we will consider the questions of constructing a completely factorable annihilator of one or other concrete element of $\mathcal{A}_k$. The problem is equal to the problem of constructing a d'Alembertian space which includes this element. It is desirable to get the *minimal* annihilator (i.e. the annihilator of the lowest possible order) of a given $a \in \mathcal{A}_k$.

The question of minimality will be discussed in Section 4. Now we limit our attention to the search for any completely factorable annihilator. If any completely factorable annihilator exists then the minimal annihilator will be completely factorable.

Let $L_a, L_b \in \mathcal{F}_k$ be completely factorable annihilators of $a, b \in \mathcal{A}_k$. Recall that then $a + b$ is annihilated by the operator

$$M = lLCM(L_a, L_b) \tag{8}$$

(from here on $lLCM$ is the left least common multiple and, respectively, $rGCD$ is the right greatest common divisor).

Due to Ore's theory ([16, 7]) the operator M is completely factorable. This theory lets one find M in the completely factored form if the corresponding factorizations of L_a and L_b are known. More precisely, this theory says that if we let $R, S \in k[\nabla]$ and R/S be the left quotient of $lLCM(R, S)$ by S:

$$lLCM(R, S) = (R/S) \circ S = (S/R) \circ R,$$

then any solution of $R(y) = 0$ is mapped by operator S into a solution of the equation $(R/S)(y) = 0$, and the following propositions hold:

O1. $\operatorname{ord}(R/S) = \operatorname{ord} R - \operatorname{ord} rGCD(R, S)$;

O2. If R is irreducible then so is R/S;

O3. If $R = R_1 \circ R_2 \circ \cdots \circ R_m$ and $R_1, R_2, \ldots, R_m \in k[\theta]$ are monic then

$$R/S = (R_1/S_1) \circ (R_2/S_2) \circ \cdots \circ (R_m/S_m),$$

where $S_j = S/(R_{j+1} \circ R_{j+2} \circ \cdots \circ R_m)$ for $j = 1, ..., m$.

We can find (8) in the completely factorable form as follows: using the factorization of L_a apply **O3**, to get a factorization of L_a/L_b in first order factors; then take $(L_a/L_b) \circ L_b$.

Observe, that this approach gives a d'Alembertian space which contains $a + b$. Let two d'Alembertian spaces A, B of the form (2) be given. Then using (7), (6) and $lLCM$ procedure we can construct such a d'Alembertian space which is equal to the set

$$A + B = \{ a + b \,|\, a \in A, b \in B \}.$$

It is interesting that $\mathcal{A}_k$ is not only a linear space over $Const$, but a ring as well:

$$a, b \in \mathcal{A}_k \Rightarrow ab \in \mathcal{A}_k \tag{9}$$

([6]). In the next section we give a proof of this and propose a procedure to construct a completely factorable annihilator of ab, if annihilators $L_a, L_b \in \mathcal{F}_k$ of $a, b \in \mathcal{A}_k$ are given.

3 Annihilators of products

We can prove (9) using induction on $m + n$, where m and n are the orders of $L_a, L_b \in \mathcal{F}_k$ which annihilate a and b. But unlike the case $a + b$, here it is most convenient to use d'Alembertian spaces instead of annihilators.

Show that the product of d'Alembertian spaces A and B, i.e. the set

$$AB = \{ ab \,|\, a \in A, b \in B \}$$

is a d'Alembertian space. Let A and B be given in the form $I(\varphi_1, \ldots, \varphi_m, 0)$ and, respectively,

$$I(\psi_1, ..., \psi_n, 0). \tag{10}$$

The case $m + n = 0$ is the case $m = 0, n = 0$ and there is nothing to prove. If either $m = 0$ or $n = 0$ then the statement is also obvious. Consider the case $m > 0, n > 0$. First of all observe that

$$\begin{gathered}\varphi_1\psi_1 I(1, \varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0) = \\ = \varphi_1\psi_1 I(\nabla(I(1, \varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0))),\end{gathered}$$

because

$$C\varphi_1\psi_1 \in \varphi_1\psi_1 I(1, \varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0)$$

for any $C \in Const$. Hence we have

$$\begin{gathered}AB = \varphi_1\psi_1 I(1, \varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0) = \\ = \varphi_1\psi_1 I(\nabla(I(1, \varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0))) = \\ = \varphi_1\psi_1 I(\sigma(I(1, \varphi_2, ..., \varphi_m, 0)) \nabla(I(1, \psi_2, ..., \psi_n, 0)) + \\ + \nabla(I(1, \varphi_2, ..., \varphi_m, 0)) I(1, \psi_2, ..., \psi_n, 0)) = \\ = \varphi_1\psi_1 I(\sigma(I(1, \varphi_2, ..., \varphi_m, 0)) I(\psi_2, ..., \psi_n, 0) + \\ + I(\varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0)).\end{gathered} \tag{11}$$

Note that $\sigma I(1, \varphi_2, ..., \varphi_m, 0)$ is the d'Alembertian space

$$I(1, \sigma(\varphi_2), ..., \sigma(\varphi_m), 0).$$

The products

$$\sigma(I(1, \varphi_2, ..., \varphi_m, 0)) I(\psi_2, ..., \psi_n, 0)$$

and

$$I(\varphi_2, ..., \varphi_m, 0) I(1, \psi_2, ..., \psi_n, 0)$$

by the induction hypothesis are some d'Alembertian spaces G_1 and G_2. The sum $G_1 + G_2$ is a d'Alembertian space G, which can be constructed as we saw in the previous Section. Since

$$AB = I(\varphi_1\psi_1, G) \tag{12}$$

the statement is proved.

Let $R, S \in \mathcal{F}_k$ be such that A is the solution space of R while B is the solution space of S (i.e., $\operatorname{Ker} R = A$ and $\operatorname{Ker} S = B$).

The symmetric product $R \otimes S$ of two operators is the minimal operator annihilating all products ab such that $a \in \operatorname{Ker} R$, $b \in \operatorname{Ker} S$. By (11) we have that if $R, S \in \mathcal{F}_k$ then $R \otimes S$ is spanned by ab, $a \in \operatorname{Ker} R$, $b \in \operatorname{Ker} S$, and $R \otimes S \in \mathcal{F}_k$. Before giving a formula for $R \otimes S$ we write down the following properties of completely factorable operators ([7]). Let

$$\operatorname{Ker}((\nabla - \frac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \frac{\nabla\eta_1}{\eta_1})) = I(\varphi_1, \ldots, \varphi_l, 0), \tag{13}$$

where $\eta_1, \ldots, \eta_l, \varphi_1, \ldots, \varphi_l \in \mathcal{H}_k$, $l \geq 1$. Then

$$\eta_1 = \varphi_1 \tag{14}$$

and

$$\begin{gathered}\operatorname{Ker}((\nabla - \tfrac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \tfrac{\nabla\eta_2}{\eta_2}) \circ \sigma(\eta_1)) = \\ = I(\varphi_2, ..., \varphi_l, 0),\end{gathered} \tag{15}$$

$$\begin{gathered}\operatorname{Ker}((\nabla - \tfrac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \tfrac{\nabla\eta_2}{\eta_2}) \circ \sigma(\eta_1) \circ \nabla) = \\ = I(1, \varphi_2, ..., \varphi_l, 0),\end{gathered} \tag{16}$$

$$\begin{gathered}\operatorname{Ker}((\nabla - \tfrac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \tfrac{\nabla\eta_1}{\eta_1}) \circ \\ \circ \sigma(\tfrac{1}{\varphi_0}) \circ (\nabla - \tfrac{\nabla\varphi_0}{\varphi_0})) = \\ = I(\varphi_0, \varphi_1, ..., \varphi_l, 0),\end{gathered} \tag{17}$$

$\varphi_0 \in \mathcal{H}_k$. Note that the left-hand sides of (15),(16),(17) have the form $\operatorname{Ker}(L_1)$, $\operatorname{Ker}(L_2)$, $\operatorname{Ker}(L_3)$. In the general case L_1, L_2, L_3 are not operators over k, because $\sigma(\eta_1)$ and $\sigma(\frac{1}{\varphi_0})$ could be not in k. But if we normalize these operators, i.e. present each of them in the form

$$\xi(\nabla - h_t) \circ \cdots \circ (\nabla - h_1)$$

with $\xi \in \mathcal{H}_k$, $h_1, ..., h_t \in k$ and take

$$(\nabla - h_t) \circ \cdots \circ (\nabla - h_1)$$

then an element of $\mathcal{F}_k$ will be obtained. Denote by $[P]_{norm}$, where P is an operator, the result of such a normalization of P. If P is given in the usual (not factored) form, then $[P]_{norm}$ will be the result of dividing P by its leading coefficient.

Now we can give a formula for $R \otimes S \in \mathcal{F}_k$ as follows. If $R = 1$ or $S = 1$ then $R \otimes S = 1$. Otherwise let

$$R = R_1 \circ (\nabla - \frac{\nabla\eta}{\eta}),\ S = S_1 \circ (\nabla - \frac{\nabla\zeta}{\zeta}).$$

Given the above description of the right-hand side of (12) lets us write

$$R \otimes S = [lLCM(P,Q) \circ \frac{1}{\sigma(\eta\zeta)} \circ (\nabla - \frac{\nabla(\eta\zeta)}{\eta\zeta})]_{norm} \quad (18)$$

where

$$P = ([R_1 \circ \sigma(\eta) \circ \nabla]_{norm}/\sigma) \otimes [S_1 \circ \sigma(\zeta)]_{norm},$$

$$Q = [R_1 \circ \sigma(\eta)]_{norm} \otimes [S_1 \circ \sigma(\zeta) \circ \nabla]_{norm}.$$

In all these cases the normalization gives elements of $\mathcal{F}_k$ (it is a consequence of $\eta, \zeta \in \mathcal{H}_k$). Thanks to **O3** these elements can be computed in the completely factored form.

The preceding can be formulated as the following

Proposition 1 *Let an expression E be composed of elements of $\mathcal{A}_k$ by means of the operations of addition and multiplication. Let for any d'Alembertian element found in E its annihilator in $\mathcal{F}_k$ be given. Then one can construct an annihilator of E belonging to $\mathcal{F}_k$.* □

Additionally if k is a functional field (e.g. $k = \mathbf{C}(x)$ and $\nabla = \delta = ' = \frac{d}{dx}$) then $f(g(x)) \in \mathcal{A}_k$ for any $f(x) \in \mathcal{A}_k$, $g(x) \in k$. Indeed, let

$$f(x) \in \varphi_1(x) \int \varphi_2(x) \dots \int \varphi_m(x) \int 0$$

then

$$f(g(x)) \in \varphi_1(g(x)) \int \varphi_2(g(x))g'(x) \dots \int \varphi_m(g(x))g'(x) \int 0,$$

resp., if

$$(\frac{d}{dx} - h_m(x)) \circ \dots \circ (\frac{d}{dx} - h_2(x)) \circ (\frac{d}{dx} - h_1(x))$$

is an annihilator of $f(x)$ then

$$(\tfrac{d}{dx} - h_m(g(x))g'(x) + (m-1)\tfrac{g''(x)}{g'(x)})) \circ$$
$$(\tfrac{d}{dx} - h_{m-1}(g(x))g'(x) + (m-2)\tfrac{g''(x)}{g'(x)})) \circ \dots$$
$$\dots \circ (\tfrac{d}{dx} - h_2(g(x))g'(x) + \tfrac{g''(x)}{g'(x)})) \circ (\tfrac{d}{dx} - h_1(g(x))g'(x))$$

is an annihilator of $f(g(x))$. It is obvious that

$$\varphi_1(g(x)), \varphi_2(g(x))g'(x), \dots, \varphi_m(g(x))g'(x) \in \mathcal{H}_k$$

and

$$h_i(g(x))g'(x) - (i-1)\frac{g''(x)}{g'(x)} \in k, \; i = 1, \dots, m$$

if $\varphi_1(x), \dots, \varphi_m(x) \in \mathcal{H}_k$ and $h_1, \dots, h_m \in k$.

Thus in the differential case with $k = \mathbf{C}(x)$, the expression E above may be constructed on the base of some known d'Alembertian functions not only by the operations of addition and multiplication but also by substitutions of rational functions in x for x. It is easy to see that $\sin x, \cos x, \arcsin x, \arctan x, \mathrm{sh}x, \mathrm{ch}x, e^x, \ln x, x^c$ ($c \in \mathbf{C}$) and any rational function are d'Alembertian over $\mathbf{C}(x)$. Therefore, for example,

$$\ln(x^2+1)\sin\frac{1}{1-x} + \arctan x,$$

$$e^{x^2}\mathrm{sh}x + \sqrt{x^3+2}$$

and so on, are d'Alembertian functions over $\mathbf{C}(x)$ and one can construct completely factorable annihilators for them.

In the case $\sigma \neq 1, \nabla = \delta = \sigma - 1$ the formulated addition to Proposition 1 seems to be false. But in this case $f \in \mathcal{A}_k$ implies $\sigma^t(f) \in \mathcal{A}_k, t \in \mathbf{Z}$. If

$$(\nabla - h_m) \circ \dots \circ (\nabla - h_1)$$

is an annihilator of f then

$$(\nabla - \sigma^t(h_m)) \circ \dots \circ (\nabla - \sigma^t(h_1))$$

is an annihilator of $\sigma^t(f)$. In the case $k = \mathbf{C}(n)$, $\sigma(f(n)) = Ef(n) = f(n+1)$ the following functions (sequences) are obviously d'Alembertian: u_n (the n−th Fibonacci number), $\Gamma(n)$, $\Psi(n)$ (the digamma function), $\Psi_m(n)$ (the m-th polygamma function), $\prod_{t=1}^{n} R(t)$ ($R \in \mathbf{C}(n)$ has no pole in $\mathbf{N} \cup \{0\}$), and any element of $\mathbf{C}(n)$. Therefore, for example,

$$\frac{1}{n!}u_{n+1}^2 + \frac{n}{n^3+1}\Psi(n)$$

is a d'Alembertian function over $\mathbf{C}(n)$ and one can construct a completely factorable annihilator for it. Corresponding examples for the q-difference case $k = \mathbf{C}(x), \sigma(f(x)) = Qf(x) = f(qx)$ also can be given.

But there is no guarantee that the approach proposed above will give the minimal annihilator even in the case where the minimal annihilators for the original elements are used. The example of function $\sin^2 x + \cos^2 x$ is quite revealing. Therefore the problem of simplifying annihilators must be considered.

4 Minimal annihilators

In constructing the minimal annihilator the *accurate integration* procedure ([9]) plays a key role. First we list the necessary facts from that paper.

An element $g \in K$ is a *primitive* of $f \in K$ if $\nabla g = f$. The problem of accurate integration is the following: let $f \in K$ and the minimal annihilator $L \in k[\theta]$ of order n of f be given. Decide whether there exists a primitive g for f such that the minimal annihilator $\widetilde{L}$ of g is of order n. It has been shown that there are three possibilities:

1. such a primitive does not exist,
2. there is a unique such primitive and $\widetilde{L}$ can be constructed,
3. all primitives of f have annihilators of order n, in which case one can construct the family

$$\widetilde{L}_C = \widetilde{L}_0 + CM \circ \nabla \quad (19)$$

of n-th order annihilators of primitives of f (M is a fixed operator, C runs through the set of constants). For any primitive g of f the value of C is

$$-\frac{\widetilde{L}_0(g)}{M(f)}. \quad (20)$$

Now we return to our problem: given $a \in \mathcal{A}_k, V \in \mathcal{F}_k$ such that $V(a) = 0$, construct the minimal annihilator $W \in \mathcal{F}_k$ of a. Let

$$V = (\nabla - \frac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \frac{\nabla\eta_1}{\eta_1})$$

and (13) takes place. By (14) and (15) the operator

$$(\nabla - \frac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \frac{\nabla\eta_2}{\eta_2}) \circ \sigma(\eta_1)$$

annihilates the element

$$f = \nabla(\frac{a}{\eta_1}).$$

It is enough to construct the minimal annihilator L of f and we decide (by the accurate integration procedure) whether there exists an n-th order operator $\widetilde{L}$ which annihilates $\frac{a}{\eta_1}$ or not. The minimal annihilator of a is equal to

$$[\widetilde{L} \circ \frac{1}{\eta_1}]_{norm} \tag{21}$$

if it exists, and

$$[L \circ \sigma(\frac{1}{\eta_1})]_{norm} \circ (\nabla - \frac{\nabla\eta_1}{\eta_1}) \tag{22}$$

otherwise (due to (17)). It is easy to see that a recursive algorithm appears.

Algorithm 1

Input: $a \in \mathcal{A}_k$, $V = (\nabla - \frac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \frac{\nabla\eta_1}{\eta_1})$.

Output: the minimal annihilator W of a.

1. If $l = \operatorname{ord} V = 0$ then the final result is 1.

2. Apply the algorithm recursively to

$$\nabla(\frac{a}{\eta_1}),\ [(\nabla - \frac{\nabla\eta_l}{\eta_l}) \circ \cdots \circ (\nabla - \frac{\nabla\eta_2}{\eta_2}) \circ \sigma(\eta_1)]_{norm}.$$

Let an n-th order operator L be the result of the recursive call.

3. Apply the accurate integration procedure to

$$L,\ \nabla(\frac{a}{\eta_1})$$

and decide whether there exists an n-th order operator $\widetilde{L}$ such that $\widetilde{L}(\frac{a}{\eta_1}) = 0$ or not. The final result W is (21) if it exists, and (22) otherwise. □

One can observe that the result W of the last algorithm is a non-factored operator (because the accurate integration procedure returns non-factored operator $\widetilde{L}$), though W is completely factorable as a divisor of a completely factorable operator. But an easy factorization is possible. In [3] has been noted that $rGCD(V, W)$ with V decomposed in first-order factors can be computed in a completely factored form. In our case W right divides V, thus $rGCD(W, V) = W$; thanks to [3] we get W in the wanted form.

The process was described in [3] for the differential case. It was supposed that the d'Alembertian solution space for V is given:

$$\operatorname{Ker}(V) = I(\varphi_1, ..., \varphi_m, 0),$$

$\varphi_1, ..., \varphi_m \in \mathcal{H}_k$,. Then the d'Alembertian solution space of $rGCD(V, W)$ can be constructed. First we briefly describe the process for the general case and then we will give the algorithm, which uses a factorization of V instead of its d'Alembertian solution space. We will denote by $L^{[\varphi]}$, $L \in k[\nabla], \varphi \in \mathcal{H}_k$, the operator $[L \circ \varphi]_{norm}/\nabla$.

Let $\operatorname{ord} W = n$. If $\operatorname{ord} rGCD(V, W) \geq 1$ then due to [3] one can find among

$$W^{[\varphi_1]}, W^{[\varphi_1][\varphi_2]}, ..., W^{[\varphi_1][\varphi_2]...[\varphi_m]}$$

an operator of order $< n$. Let $W^{[\varphi_1][\varphi_2]...[\varphi_l]}, 0 \leq l \leq m$, be the first such an operator (note that $\operatorname{ord} W^{[\varphi_1][\varphi_2]...[\varphi_i]} \leq \operatorname{ord} W^{[\varphi_1][\varphi_2]...[\varphi_{i+1}]}, i = 1, ..., m-1$). Then

$$W^{[\varphi_1][\varphi_2]...[\varphi_{l-1}]}(\varphi_l) = 0 \tag{23}$$

and all $W \circ \varphi_1, W^{[\varphi_1]} \circ \varphi_2, ..., W^{[\varphi_1][\varphi_2]...[\varphi_{l-2}]} \circ \varphi_{l-1}$ are not right divisible by ∇. This lets one find a common solution of $V(y) = 0, W(y) = 0$ in $\mathcal{H}_k$. Let $\varphi = \varphi_l$ and

$$L = W^{[\varphi_1][\varphi_2]...[\varphi_{l-2}]} \circ \varphi_{l-1} = b_n\nabla^n + ... + b_1\nabla + b_0.$$

Then

$$\hat{\varphi} = -\varphi_{l-1}(\frac{b_n}{b_0}\nabla^{n-1}\varphi + ... + \frac{b_1}{b_0}\varphi)$$

is such that

$$W^{[\varphi_1][\varphi_2]...[\varphi_{l-2}]}(\hat{\varphi}) = 0. \tag{24}$$

Now one can find a solution in $\mathcal{H}_k$ of the equation $W^{[\varphi_1][\varphi_2]...[\varphi_{l-3}]}(y) = 0$ and so on. Finally we will obtain $\psi \in \mathcal{H}_k$ such that $W(\psi) = 0$ and additionally $V(\psi) = 0$ (the last is proved in [3]). Now it is enough to apply the algorithm recursively to $W^{[\psi]}$ and to the operator $V^{[\psi]}$ whose solution space is of order $m-1$ and is the result of applying ∇ to $I(\frac{\varphi_1}{\psi}, \varphi_2, \ldots, \varphi_m, 0)$. Let it give an operator U and $\lambda_1, ..., \lambda_t \in \mathcal{H}_k$ be such that

$$\operatorname{Ker}(U) = I(\lambda_1, ..., \lambda_t, 0).$$

Then

$$\operatorname{Ker}(rGCD(V, W)) = I(\psi, \lambda_1, ..., \lambda_t, 0).$$

Now we can use formulas (15),(16),(17) to describe the following algorithm.

Algorithm 2

Input: operators $V = (\nabla - h_m) \circ \cdots \circ (\nabla - h_1)$, $W \in k[\nabla]$.

Output: $rGCD(V, W)$ in completely factored form.

1. Let $\varphi_1, ..., \varphi_m$ satisfy (6). If $\operatorname{ord} W^{[\varphi_1]} < \operatorname{ord} W$ then set $\psi = \varphi_1$ and go to **4.**

2. Find the last $t, 1 \leq t \leq m$, such that $\operatorname{ord} W^{[\varphi_1][\varphi_1]...[\varphi_t]} = \operatorname{ord} W$. If $t = m$ then the final result will be 1, otherwise set $l = t + 1$.

3. Starting with $W^{[\varphi_1]...[\varphi_{l-2}]} \circ \varphi_{l-1}, \varphi_l$, find $\psi \in \mathcal{H}_k$ such that $W(\psi) = 0, V(\psi) = 0$ as it was described above (see (23),(24)).

4. Apply the algorithm recursively to

$$V^{[\psi]},\ W^{[\psi]}.$$

Let U be the result of the recursive call. The final result will be

$$[U \circ \frac{1}{\sigma(\psi)}]_{norm} \circ (\nabla - \frac{\nabla\psi}{\psi}).$$

□

Constructing elements η_i in explicit form as well as operating on them are eventually open to many difficulties. In the next section we demonstrate how these difficulties can be overcome.

5 Arithmetic of certificates

Any element $\varphi \in \mathcal{H}_k$ can be completely determined by $r = \frac{\nabla\varphi}{\varphi} \in k$ and by a specification of the concrete solution of the corresponding first-order equation. In the case of a functional field k the specification can be, for example, the value $\varphi(p)$ at a point p at which $\varphi(p) \neq 0$. We will call r in this representation the *certificate* of the hyperexponential element φ and write down this representation in the form

$$\{r, \text{a specification of } \varphi\}. \tag{25}$$

The need for operations on hypergeometric elements appears, for example, in formulas (6),(7),(18),(21) and so on. All such computations with hypergeometric elements can be done up to an arbitrary multiplier from k. It means that, in the context of algorithms considered here, it will be enough to use representation $\{r\}$ instead of (25). Table 1 contains definitions of operations on certificates.

Operation	$\sigma \neq 1; \nabla = \sigma - 1$	$\sigma = 1; \nabla = \delta \neq 0$
$\{r\} \cdot \{s\}$	$\{rs + r + s\}$	$\{r + s\}$
$\frac{1}{\{r\}}$	$\{-\frac{r}{r+1}\}$	$\{-r\}$
$\nabla\{r\}$	$r\{r\}$	$r\{r\}$
$\sigma(\{r\})$	$(r+1)\{r\}$	$\{r\}$

Table 1: Arithmetic of certificates

Observe that any $t \in k$ (at the same time $t \in \mathcal{H}_k$) can be represented by $\{\frac{\nabla t}{t}\}$. Thus we are able to perform computations with hyperexponential elements using only operations from k.

Remark that only intermediate results of such computations will contain constructions of the form $\{r\}$, because the final result in all algorithms has to be normalized. For example, consider the expression $W^{[\psi]} = [W \circ \psi]_{norm}/\nabla$ from Algorithm 2. Here $\psi \in \mathcal{H}_k$ and is represented by a certificate r. After multiplying operators W and ψ all nonzero coefficients of the result will have $\{r\}$ as a multiplier. After normalization this multiplier will be reduced and we get an operator with coefficients from k. Thus, we can formulate the following

Proposition 2 *Algorithms for constructing the minimal annihilator, accurate integration and computing rGCD in factored form can be implemented without involving operations on hyperexponential elements but using arithmetic in the field k only.* □

6 Implementation

The algorithms from the previous sections are implemented in **Maple 5.4** [15] with k as the rational function field. An operator $a_n\theta^n + \cdots + a_1\theta + a_0$ is represented by Maple list $[a_0, a_1, \ldots, a_n]$. An operator decomposed in the product of operators is represented as a list of lists. For example, the operator

$$(\theta^2 + x\theta + \frac{1}{x})(\frac{x}{x-1}\theta + x)$$

is represented as

$$[[\frac{1}{x}, x, 1], [x, \frac{x}{x-1}]].$$

A completely factorable operator in the factored form is represented as a list of lists (as above) but with first element of the form [1], for example

$$w := [[1], [0, 1], [-\frac{1}{x}, 1]].$$

The user is provided with the following set of procedures: `set_Ore_ring`, `get_any`, `get_minimal`, `rgcd_fact`, `check_prim`, `get_min_annih`. The main procedures are `set_Ore_ring` and `get_min_annih`. Other ones are rather auxiliary, but could be useful themselves.

Procedure `set_Ore_ring(indvar, case)` sets the program to a concrete $k[X; \sigma, \delta]$ by selecting the independent variable and concrete δ, σ, c and α. It takes unassigned names as parameters. The value of the first parameter stands for the name of the independent variable, the value of the second parameter selects a concrete $k[X; \sigma, \delta]$.

Table 2 shows the collection of standard Ore rings.

case	θ	σ	δ	c	α
differential	$\theta f(x) = \frac{d}{dx} f(x)$	1	$\frac{d}{dx}$	0	-
Eulerian	$\theta f(x) = x\frac{d}{dx} f(x)$	1	$x\frac{d}{dx}$	0	-
recurrent	$\theta f(x) = Ef(x) = = f(x+1)$	E	0	1	0
difference	$\theta f(x) = \Delta f(x) = = f(x+1) - f(x)$	E	Δ	0	1
q-recurrent	$\theta f(x) = Qf(x) = = f(qx)$	Q	0	1	0
q-differential	$\theta f(x) = D_q f(x) = = \frac{f(qx)-f(x)}{qx-x}$	Q	D_q	0	$\frac{1}{x(q-1)}$
q-difference	$\theta f(x) = \Delta_q f(x) = = f(qx) - f(x)$	Q	Δ_q	0	1

Table 2: Standard Ore rings family

In the standard cases adjustment of the program is hidden from the user; for example, the call

```
set_Ore_ring(x,differential)
```

sets the program to the differential case with x as the independent variable.

If the value of the second parameter is `nonstandard`, the user should provide the program with procedural definitions of δ, σ, σ^{-1} and assign concrete numeric values to the constants c and α (if needed). For example:

```
delta:=proc(y) diff(y,v) * v^2/2 end;
sigma:=proc(y) y end;
sigma_1:=proc(y) y end;
c:=1;
set_Ore_ring(v,nonstandard);
```

Additionally the procedure `k_solver` which will solve k-problems appearing in step **3** of Algorithm 1 has to be declared. By the k-problem we mean (as in [8]) the problem of solving in k a linear equation whose coefficients and right-hand side belong to k. User defined procedure `k_solver` should take two parameters: an equation to be solved and an unknown function. It should return the result in one of the following forms:

- { } — if there are no solutions;
- [sol, {}] — if there is a unique solution sol;
- [sol, {sol1,..,soln}] (where sol1,..,soln is the basis for the solution space of the corresponding homogeneous equation, sol – a particular solution of the inhomogeneous equation) — if there are more than one solution.

In the standard cases we use standard Maple tools for defining of the procedure k_solver.

Procedure get_any(expr) takes an expression and returns an annihilator of the expression in factored form if the expression is composed of d'Alembertian elements by the operations $+, -, *$ and powering to a positive integer. For example, after setting to the differential case as above the call

```
get_any(exp(x/(x-1))+sqrt(x^2+1))
```

results in

$$[[1], [\frac{x^4+6x^2-2x+3}{(x-1)(x^3-x^2+x+1)(x^2+1)}, 1], [\frac{1}{(x-1)^2}, 1]]$$

The first component of this list [1] means that the polynomial is presented in the normalized factored form.

Procedure get_minimal(L,expr) takes an expression expr and its annihilator L in the normalized factored form and returns the minimal annihilator of expr.

Procedure rgcd_fact(L,W) is an implementation of **Algorithm 2**. It takes two operators (L in the factored form and W in the non-factored form) and finds their $rGCD$ in the factored form.

Consider an example. After setting the program to the differential case, we get a third order factored annihilator for the expression $x * ln(x) - x + 1$:

```
> expr:=x*ln(x)-x+1;
```

$$expr := x\ln(x) - x + 1$$

```
> a:=get_any(expr);
```

$$a := [[1], [\frac{2}{x}, 1], [0, 1], [-\frac{1}{x}, 1]]$$

which is not minimal. After the call

```
> u:=get_minimal(a,expr);
```

$$\_z(x) = 1$$

$$[1, \{\}]$$

$$2\frac{\_z(x)}{x} - (\frac{\partial}{\partial x}\_z(x)) = 1$$

$$[x, \{x^2\}]$$

$$-\frac{\_z(x)}{x(x-1)} - (\frac{\partial}{\partial x}\_z(x)) = 1$$

$$\{\}$$

$$u := [\frac{1}{x^2(x-1)} + \frac{1}{x^2}, -\frac{1}{x(x-1)} - \frac{1}{x}, 1]$$

we have the minimal annihilator in the non-factored form (first six lines above contain appearing k-problems and their solutions). Finally with the help of rgcd_fact we obtain the factored minimal annihilator of the initial expression:

```
> v:=rgcd_fact(a,u);
```

$$v := [[1], [-\frac{1}{x(x-1)}, 1], [-\frac{1}{x}, 1]]$$

Procedure get_min_annih(expr) calls consecutively procedures get_any, get_minimal, rgcd_fact and (if the call of get_any did not fail) returns the minimal annihilator of the expression expr in factored form. For example

```
> expr:=exp(x^2)*sinh(1/x)+sqrt(x^3+2);
```

$$expr := e^{(x^2)}\sinh(\frac{1}{x}) + \sqrt{x^3+2}$$

```
> a:=get_min_annih(expr);
```

$$a := [[1], [-\frac{1}{2}(-15x^{11} - 96x^8 - 144x^{13} - 720x^{10} + 980x^9 + 656x^6 + 336x^3 - 864x^7 + 112x^{15} + 576x^{12} + 256 + 192x^5) / ((x^3+2)x(15x^8 + 48x^5 - 48x^{10} - 144x^7 + 60x^6 - 16x^3 - 16 - 96x^4 + 16x^{12} + 64x^9)), 1], [-\frac{1-2x+2x^3}{x^2}, 1], [-\frac{-1+2x^3}{x^2}, 1]]$$

This procedure also transforms an element of $k[\theta]$ to an element of $k[\nabla]$ and back.

Procedure check_prim(L) takes an operator L which is an annihilator of some expression f and decides if there exists a primitive of the expression, which can be annihilated by an operator of the same order. If the answer is positive, the procedure returns an annihilator for the primitive (or the parametrized family of such annihilators) and operator r such that $r(f)$ is a primitive of f (or the parametrized family of such operators). For example, the minimal annihilator of $\sqrt{x}\ln^2 x$ in the non-factored form is

```
> expr:=sqrt(x)*log(x)^2;
```

$$expr := \sqrt{x}\ln(x)^2$$

```
> a:=fact_to_expanded(get_min_annih(expr));
```

$$a := [-\frac{1}{8}\frac{1}{x^3}, \frac{1}{4}\frac{1}{x^2}, \frac{3}{2}\frac{1}{x}, 1]$$

The result of the call

```
> check_prim(a);
```

$$[-\frac{27}{8}\frac{1}{x^3}, \frac{13}{4}\frac{1}{x^2}, -\frac{3}{2}\frac{1}{x}, 1], [\frac{26}{27}x, -\frac{4}{9}x^2, \frac{8}{27}x^3]$$

is the sequence of two operators: the first one is the annihilator for the primitive of $\sqrt{x}\ln^2 x$, the second one allows to get this primitive from the given expression (the call fact_to_expanded here stands for conversion of an operator to the non-factored form).

Consider the difference case (fib(n) stands for n-th Fibonacci number):

```
> expr:=fib(n)^2-fib(n-1)*fib(n+1)+(-1)^(n+1);
```

$$expr := \mathrm{fib}(n)^2 - \mathrm{fib}(n-1)\,\mathrm{fib}(n+1) + (-1)^{(n+1)}$$

```
> a:=get_any(expr);
```

$$a := [[1], [-\frac{1}{2} + \frac{1}{2}\sqrt{5}, 1], [2, 1], [-\frac{1}{2} - \frac{1}{2}\sqrt{5}, 1]]$$

```
> u:=get_minimal(a,expr);
```

$$u := [1]$$

This proves that $\mathrm{fib}(n)^2 - \mathrm{fib}(n-1)\,\mathrm{fib}(n+1) + (-1)^{(n+1)} = 0$.

Remark that the time for each of examples above does not exeed **4** sec. on 32 MB Pentium/100 PC.

References

[1] S. A. Abramov (1989): Rational solutions of linear difference and differential equations with polynomial coefficients, *USSR Comput. Maths. Math. Phys.* **29**, 7–12. Transl. from *Zl. Vychislit. matem. mat. fiz.* **29**, 1611–1620.

[2] S. A. Abramov, K. Yu. Kvashenko (1991): Fast algorithm to search for the rational solutions of linear differential equations, In *Proc. ISSAC'91*, 267–270.

[3] S.A. Abramov (1993): On d'Alembert Substitution, In *Proc. ISSAC'93*, 20–33.

[4] S. A. Abramov (1995): Rational solutions of linear difference and q-difference equations with polynomial coefficients, *Programming and Comput. Software*, No 6.

[5] S. A. Abramov (1996): Symbolic algorithms to search for particular d'Alembertian solutions of linear equations, *Programming and Comput. Software*, No 1.

[6] S.A. Abramov, M. Petkovšek (1993): *Unpublished.*

[7] S.A. Abramov, M. Petkovšek (1994): D'Alembertian Solutions of Linear Differential and Difference Equations, In *Proc. ISSAC'94*, 169–174.

[8] S.A. Abramov, E. Zima (1996): D'Alembertian Solutions of Inhomogeneous Linear Equations (differential, difference and some other), In *Proc. ISSAC'96*, 232–240.

[9] S. A. Abramov, M. van Hoeij (1997): A method for the Integration of Solutions of Ore Equations, *These Proceedings.*

[10] M. Bronstein, M. Petkovšek (1994): On Ore rings, linear operators and factorization, *Programming and Comput. Software* **20**, 14–26.

[11] M. Bronstein, M. Petkovšek (1996): An introduction to pseudo-linear algebra, *Theoretical Computer Science* **157**, 3–33.

[12] F. Chyzak (1996): ∂-finite functions, *INRIA. Algorithm seminar 1995–1996*, 43–46.

[13] F. Chyzak, B. Salvy (1996): Non-commutative elimination in Ore algebra proves multivariate holonomic identities, *INRIA Research Report*, No 2799.

[14] W. Koepf (1992): Power series in computer algebra, *J. Symb. Comp.*, **13**, 581–603.

[15] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S. Vorkoetter. *Maple-V. Programming Guide.* Springer-Verlag, 1996.

[16] O. Ore (1933): The theory of non-commutative polynomials, *Ann. Maths.* **34**, 480–508.

[17] B. Salvy, P. Zimmermann (1992): GFUN: a Maple package for the manipulation of generating and holonomic functions in one variable, *INRIA, Rapports Techniques*, No 143.