A Universal Program to Uncouple Linear Systems *

Sergei A. Abramov
Computer Center of
the Russian Academy of Science,
Vavilova 40, Moscow 117967, Russia.
`abramov@ccas.ru`

Eugene V. Zima
Comput. Math. & Cybernetics dept.,
Moscow State University,
Moscow 119899, Russia.
`zima@cs.msu.su`

Abstract

We describe a program to uncouple a linear system of equations. An algorithm is described in terms of an arbitrary Ore polynomial ring. It allows to adjust the program for the differential, difference, q-difference and many other cases.

1 Introduction

The search for solutions of a system of linear ordinary differential equations can be reduced to the search for solutions of scalar (i.e., with one unique

*Work reported herein was supported in part by the RFBR-INTAS under Grant 95-IN-RU-412.

unknown function) linear equations. Such a reduction is possible for systems of linear difference and q-difference equations as well.

Theoretically, this reduction is not hard for any kind of system. Algorithms to uncouple linear differential and difference systems are based on very similar ideas. There is a mathematical abstraction connecting, in particular, linear ordinary differential, difference and q-difference equations. This is Ore polynomial rings [1, 2, 3]. In this paper we describe a **Maple** implementation of a universal (based on Ore polynomials) algorithm to uncouple a linear system to scalar equations. The implementation assumes that coefficients are polynomials (in the usual sense). If coefficients are rational functions then preliminary clearing of denominators is needed.

The algorithm has been described by the authors in [4]. It is a generalization to an arbitrary Ore polynomial ring of an algorithm to uncouple a system of linear ordinary differential equations [5].

In Section 2 we recap basic notions connected with Ore polynomial rings. Then, in Section 3 we describe shortly the uncoupling algorithm. Section 4 discusses a **Maple** implementation. A few examples of uncoupling can be found in Section 5.

2 Basic notions of Ore polynomial rings

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

$$\delta(a+b) = \delta a + \delta b, \quad \delta(ab) = \sigma(a)\, \delta b + \delta a\, b \tag{1}$$

for any $a, b \in k$. The Ore polynomial ring $k[\theta; \sigma, \delta]$ given by σ and δ is the ring of polynomials in θ over k with the usual addition. Multiplication in $k[\theta; \sigma, \delta]$ is defined by the following relation

$$\theta a = \sigma(a)\theta + \delta a \tag{2}$$

for any $a \in k$.

Let V be a linear space over k (if k is a functional field then usually V is a wider set of functions). Let the application of θ to elements of V be defined in such a way that

$$\theta(u+v) = \theta u + \theta v, \quad \theta(au) = \sigma(a)\theta u + \delta a\, u \tag{3}$$

for $u, v \in V, a \in k$. Then we can consider the application of Ore polynomials $A(\theta), B(\theta), \dots$ to elements of V, with $A(\theta)B(\theta)v = A(\theta)(B(\theta)v), (A(\theta) + B(\theta))v = A(\theta)v + B(\theta)v$ and so on. It allows to consider V as a module over $k[\theta; \sigma, \delta]$. From here on $\theta a, \theta u, \dots$ we denote the results of applying θ to $a, u, \dots$ rather than products of the elements of $k[\theta; \sigma, \delta]$ as in (2).

Let $k \subset V$ (this inclusion is natural in the functional case). Thanks to (3)

$$\theta a = \sigma(a)\theta 1 + \delta(a) \tag{4}$$

for $a \in k$. Therefore, in order to define the result of application of θ to an element of k it is enough (in addition to σ and δ) to know $\theta 1$, i.e. the result of application of θ to the unit of k. The following table shows that linear ordinary differential, recurrent, difference and q-analogs of recurrent and differential equations can be considered in the framework of the theory of Ore polynomial rings:

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

The list above is not exhaustive. It can be extended easily by special cases like

a nonstandard	$\theta f(x) = x\frac{d}{dx} f(x)$	1_k	$x\frac{d}{dx}$	0
q-difference	$\theta f(x) = \Delta_q f(x) = f(qx) - f(x)$	Q	Δ_q	0

and so on.

3 The algorithm

Consider a linear system (in general, inhomogeneous)

$$\begin{aligned} &f_1\theta y_1 = a_{11}y_1 + a_{12}y_2 + ... + a_{1n}y_n + r_1, \\ &... \\ &f_n\theta y_n = a_{n1}y_1 + a_{n2}y_2 + ... + a_{nn}y_n + r_n, \end{aligned}$$

$a_{ij} \in k$, $r_i, y_i \in V$, where y_i are unknown. The result of the application of θ is defined in accordance with (3).

The application of the algorithm starts with the following transformations, which give a scalar equation in y_1.

Consider the first equation. If $a_{12} = \ldots = a_{1n} = 0$ then we already have an equation in y_1 alone. Let $a_{12} \neq 0$ (if $a_{12} = 0$, but $a_{1i} \neq 0, i > 2$, we can re-enumerate unknowns). Introduce new unknown z_2 (instead of y_2):

$$z_2 = a_{12}y_2 + ... + a_{1n}y_n. \tag{5}$$

The first equation can be rewritten as

$$f_1\theta y_1 = a_{11}y_1 + z_2.$$

In all the remaining equations we can eliminate y_2 with the help of (5). After this elimination the second equation will have the form

$$g_2\theta y_2 = b_{21}y_1 + b_{22}z_2 + ... + b_{2n}y_n + s_2. \tag{6}$$

Application of θ to both the sides of (5) gives us

$$\theta z_2 = \sigma(a_{12})\theta y_2 + \delta(a_{12})y_2 + \ldots + \sigma(a_{1n})\theta y_n + \delta(a_{1n})y_n. \tag{7}$$

We can use (6), (5) together with

$$\begin{aligned} &g_3\theta y_3 = b_{31}y_1 + b_{32}z_2 + b_{33}y_3 + ... + b_{3n}y_n + s_3, \\ &... \\ &g_n\theta y_n = b_{n1}y_1 + b_{n2}z_2 + b_{n3}y_3 + ... + b_{nn}y_n + s_n \end{aligned}$$

in order to eliminate $y_2, \theta y_2, \theta y_3, \ldots, \theta y_n$ from (7). After that (7) contains only $\theta z_2, y_1, y_3, \ldots, y_n$:

$$h_2\theta z_2 = c_{21}y_1 + c_{22}z_2 + c_{23}y_3 + ... + c_{2n}y_n + t_2.$$

Hence we have as the result of these actions the system

$$\begin{aligned}
&f_1\theta y_1 = a_{11}y_1 + z_2 + r_1,\\
&h_2\theta z_2 = c_{21}y_1 + c_{22}z_2 + c_{23}y_3 + ... + c_{2n}y_n + t_2,\\
&g_3\theta y_3 = b_{31}y_1 + b_{32}z_2 + b_{33}y_3 + ... + a_{3n}y_n + s_3,\\
&...\\
&g_n\theta y_n = b_{n1}y_1 + b_{n2}z_2 + b_{n3}y_3 + ... + a_{nn}y_n + s_n.
\end{aligned} \tag{8}$$

If $c_{23} = c_{24} = ... = c_{2n} = 0$ we can obtain a second order equation in y_1 from the first two equations of (8). The first equation allows to express z_2 via $y_1, \theta y_1$. Applying θ to the first equation we can express θz_2 via $y_1, \theta y_1, \theta^2 y_1$. Substituting in the second equation we have the desired result.

If at least one of $c_{23}, c_{24}, ..., c_{2n}$ (for example c_{23}) is nonzero, then we introduce the new unknown z_3 (instead of y_3)

$$z_3 = c_{23}y_3 + ... + c_{2n}y_n, \tag{9}$$

and so on until for a value $l \leq n$ we obtain a system

$$\begin{aligned}
&\alpha_1\theta y_1 = \beta_{11}y_1 + z_2 + r_1,\\
&\alpha_2\theta z_2 = \beta_{21}y_1 + \beta_{22}z_2 + z_3 + t_2,\\
&...\\
&\alpha_{l-1}\theta z_{l-1} = \beta_{l-1,1}y_1 + \beta_{l-1,2}z_2 + \ldots + \beta_{l-1,l-1}z_{l-1} + z_l + u_{l-1},\\
&\alpha_l\theta z_l = \beta_{l,1}y_1 + \beta_{l,2}z_2 + \ldots + \beta_{l,l-1}z_{l-1} + \beta_{l,l}z_l + u_l.
\end{aligned} \tag{10}$$

We can get an equation of degree l in y_1 with the help of these equations.

In the rest of the initial system we consider $y_1, z_2, \ldots, z_l$ as knowns and perform the same transformations. It gives us an equation of order $m, m \leq n-l$, for y_{l+1} and so on. Finally, we will have several scalar equations (in y_1, y_{l+1} etc). The sum of their orders will be equal to n. Relations, analogous to (10), allow to express all z_i via $y_1, y_{l+1}, \ldots$. Formulas (5), (9) and so on form a triangle system of linear algebraic equations. Hence, we can express all remaining y_i via solutions of scalar equations.

Let the initial conditions for equations of the initial system be given. Then it is easy to get initial conditions for split scalar equations. For example for an l-order equation in y_1 one can find linear expressions of $\theta y_1, \ldots, \theta^{l-1}y_1$ via $y_1, \ldots, y_n$. If $y_1, \ldots, y_n$ are functions and the values $y_1(0), \ldots, y_n(0)$ are known, then these linear expressions allow to compute $\theta y_1(0), \ldots, \theta^{l-1}y_1(0)$. In order to construct the linear expressions mentioned above, we apply θ

to the first equation of the initial system and then eliminate in the right-hand side all θy_i by means of equations of the initial system. This gives a linear expression for $\theta^2 y_1$. Repeating these steps we obtain expressions for $\theta^3 y_1, \theta^4 y_1, \ldots$ Analogous expressions can be obtained for the scalar equation in y_{l+1} and so on.

4 The program

The algorithm from the previous section has been implemented in **Maple 5.3** with several improvements like

- preliminary splitting of scalar equations on each step,
- reducing coefficients by their gcd,
- choosing the most suitable unknown for excluding on each step

and others.

The user is provided with the three main procedures `set_Ore_ring`, `uncpl_sys` and `get_sol`. Procedure `set_Ore_ring(indvar, case)` sets the program to a concrete $k[\theta; \sigma, \delta]$ by selecting the independent variable and concrete δ, σ and $\theta 1$. 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 $k[\theta; \sigma, \delta]$. In the standard cases (`differential`, `difference`, `recurrent`, `q-recurrent`, `q-difference`, `q-differential`) 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 independent variable.

If the value of the second parameter is `nonstandard`, the program prompts the user for input of definitions of δ and σ (in procedural form) and the value of $\theta 1$. Below we show a Maple session setting the program to the nonstandard case:

```
> set_Ore_ring(x, `nonstandard`);
   Enter definition of automorphism sigma
> proc(y) y end;
```

```
   Enter definition of map delta
> proc(y) x^2*diff(y,x) end;
   What is the value of theta_1 ?
> 0
   program is set to the  nonstandard case
```

Procedure `uncpl_sys(sys,unklst)` takes a system of linear equations as the first parameter, a list of unknowns as the second parameter and returns the result of uncoupling. The result is a list of scalar equations, expressions of auxiliary unknowns in terms of solutions of these equations and linear expressions of the remaining unknowns via solutions of the scalar equations and the auxiliary unknowns. There is the optional third parameter `ch_flag` of type Boolean, which is set `true` by default. If `ch_flag=false` then the choice of the most suitable unknown for excluding is switched off.

Procedure `get_sol(unres, unkl)` takes the result of uncoupling and the list of unknowns as parameters and finds a solution of the initial system by means of standard Maple tools. In the differential case the procedure `dsolve` is used for this purpose. If the optional third papameter has the value `rat`, then Bronstein's procedure `ratlode` (from the Maple share library) to find rational solutions is used instead of `dsolve`.

5 Examples of uncoupling

In this section we give several examples of using the program from the previous section together with timings[1].

Example 1. Consider the well known combinatorial problem: n pairs of inimical knights are invited to dinner. How many ways $y_1(n)$ are there to seat them at a round table so that no enemies are beside each other? Let $y_2(n)$ be the number of ways to seat them so that exactly one pair of enemies is beside each other; $y_3(n)$ be the number of ways to seat them so that exactly two pairs of enemies are beside each other. It is possible to show [6], that for $n \geq 2$

$$\begin{aligned} Ey_1(n) &= 2(2n-1)ny_1(n) + 2(2n-1)y_2(n) + 2y_3(n), \\ Ey_2(n) &= 4n(n+1)y_1(n) + 2(n+1)y_2(n), \\ Ey_3(n) &= 2n(n+1)y_1(n) + 2(n+1)y_2(n), \end{aligned} \tag{11}$$

[1] All the reported timings were obtained on 66Mhz IBM PC 486DX.

$y_1(2) = 2, y_2(2) = 0, y_3(2) = 4$. In the session below we first set the program to the recurrent case and then (after initial assignments) call the procedure `uncpl_sys`:

```
> set_Ore_ring(n, ‘recurrent‘):
> eq1:= theta(y1(n),n) =
>       2*(2*n-1)*n*y1(n) + 2*(2*n-1)*y2(n) + 2*y3(n):
> eq2:= theta(y2(n),n) =
>       2*(n+1)*2*n*y1(n) + 2*(n+1)*y2(n):
> eq3:= theta(y3(n),n) =
>       2*(n+1)*n*y1(n) + 2*(n+1)*y2(n):
> eq:=[eq1,eq2,eq3]:
> varl:=[y1(n), y2(n), y3(n)]:
> rez:=uncpl_sys(eq, varl);
```

$$
\begin{aligned}
rez := \Big[&[(64n^2 + 32n + 8n^4 + 40n^3)\,\mathrm{y1}(n) + (-1-n)\,\theta(\theta(\theta(\mathrm{y1}(n))))\\
&+ (20n^2 + 20 + 34n + 4n^3)\,\theta(\theta(\mathrm{y1}(n)))\\
&+ (40 + 108n + 108n^2 + 8n^4 + 48n^3)\,\theta(\mathrm{y1}(n)) = 0,\\
&\mathrm{y3\_z}(n) = \theta(\mathrm{y1}(n)) - (4n^2 - 2n)\,\mathrm{y1}(n), \mathrm{y2\_z}(n) = \%1 - 6\,\theta(\mathrm{y1}(n))\,n\\
&- 2\,\theta(\mathrm{y1}(n)) - 4\,\theta(\mathrm{y1}(n))\,n^2 - (28n^2 + 12n + 16n^3)\,\mathrm{y1}(n)],\\
&\left[\mathrm{y2}(n) = \frac{\mathrm{y2\_z}(n)}{16n + 8 + 8n^2}, \mathrm{y3}(n) = \frac{1}{2}\,\mathrm{y3\_z}(n) - \frac{1}{2}\,\frac{(4n-2)\,\mathrm{y2\_z}(n)}{16n + 8 + 8n^2}\right]\Big]\\
&\%1 := \theta(\theta(\mathrm{y1}(n)))
\end{aligned}
$$

Here (in 2 seconds) we get as the result one scalar equation in $y1(n)$, expressions of auxiliary functions via $y1(n)$ and linear expressions of $y2(n), y3(n)$ via $y1(n)$ and auxiliary functions. Using the remark at the end of Section 3, it is possible to get the initial conditions for the scalar equation $y1(2) = 2$, $y1(3) = 32$, $y1(4) = 1488$. The scalar equation has the following solution with these initial conditions

$$y_1(n) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} 2^k (2n-k-1)!\,.$$

Example 2. In order to demonstrate the work of the program in the q-recurrent case we consider the system (11) using Q instead of E.

```
> set_Ore_ring(n, ‘q-recurrent‘):
> eq1:= theta(y1(n),n) =
>       2*(2*n-1)*n*y1(n) + 2*(2*n-1)*y2(n) + 2*y3(n):
> eq2:= theta(y2(n),n) =
>       2*(n+1)*2*n*y1(n) + 2*(n+1)*y2(n):
> eq3:= theta(y3(n),n) =
>       2*(n+1)*n*y1(n) + 2*(n+1)*y2(n):
> eq:=[eq1,eq2,eq3]:
> varl:=[y1(n), y2(n), y3(n)]:
> rez:=uncpl_sys(eq, varl);
```

$$\begin{aligned}
rez := \Big[& [(8\,n^2\,q + 8\,n^3\,q^2 + 8\,n^2\,q^2 + 8\,n\,q)\,\mathrm{y1}(n) + (2\,q + 4\,n^2\,q^4)\,\theta(\theta(\mathrm{y1}(n))) \\
& + (12\,n^2\,q^3 + 8\,n^3\,q^4 + 4\,n\,q^2 - 4\,n^2\,q^2 - 4\,n\,q)\,\theta(\mathrm{y1}(n)) \\
& - \theta(\theta(\theta(\mathrm{y1}(n)))) = 0, \\
& \mathrm{y3_z}(n) = \theta(\mathrm{y1}(n)) - (4\,n^2 - 2\,n)\,\mathrm{y1}(n), \mathrm{y2_z}(n) = \%1 \\
& - 4\,n^2\,q^2\,\theta(\mathrm{y1}(n)) + 2\,n\,q\,\theta(\mathrm{y1}(n)) \\
& - (-4\,n^2 - 4\,n + 16\,n^3\,q + 16\,n^2\,q)\,\mathrm{y1}(n)], \\
& \left[\mathrm{y2}(n) = \frac{\mathrm{y2_z}(n)}{8\,n^2\,q + 8\,n\,q}, \mathrm{y3}(n) = \frac{1}{2}\,\mathrm{y3_z}(n) - \frac{1}{2}\,\frac{(4\,n-2)\,\mathrm{y2_z}(n)}{8\,n^2\,q + 8\,n\,q}\right]\Big] \\
& \%1 := \theta(\theta(\mathrm{y1}(n)))
\end{aligned}$$

Here (in 4 seconds) we again get one scalar equation in $y1(n)$, and expressions for the remaining unknowns via $y1(n)$.

Example 3. In both the examples above we get one scalar equation. Now we consider a differential linear system [8] which will be split into several scalar equations:

$$\begin{aligned}
x^2y_1' &= (1+4x)y_1 - 5xy_2 + 7xy_3 - 8xy_4 + 8xy_5 - 6xy_6, \\
x^2y_2' &= -10xy_1 + (9x+1)y_2 - 14xy_3 + 16xy_4 - 16xy_5 + 12xy_6, \\
x^2y_3' &= -5xy_1 + 5xy_2 + (1-8x)y_3 + 8xy_4 - 8xy_5 + 6xy_6, \\
x^2y_4' &= 10xy_1 - 10xy_2 + 14xy_3 + (1-17x)y_4 + 16xy_5 - 12xy_6, \\
x^2y_5' &= 5xy_1 - 5xy_2 + 7xy_3 - 8xy_4 + (1+7x)y_5 - 6xy_6, \\
x^2y_6' &= -5xy_1 + 5xy_2 - 7xy_3 + 8xy_4 - 8xy_5 + (1+5x)y_6.
\end{aligned} \tag{12}$$

In the session below we first set the program to the differential case and then (after initial assignments) call the procedure `uncpl_sys`:

```
> set_Ore_ring(x, ‘differential‘):
> eq1:= theta(y1(x),x)*x^2 =
>       (1+4*x)*y1(x) + (-5*x)*y2(x) + (7*x)*y3(x) +
>       (-8*x)*y4(x)  + (8*x)*y5(x)  + (-6*x)*y6(x):
> eq2:= theta(y2(x),x)*x^2 =
>       (-10*x)*y1(x) + (9*x+1)*y2(x) + (-14*x)*y3(x) +
>       (16*x)*y4(x)  + (-16*x)*y5(x)  + (12*x)*y6(x):
> eq3:= theta(y3(x),x)*x^2 =
>       (-5*x)*y1(x) + (5*x)*y2(x) + (-8*x+1)*y3(x) +
>       (8*x)*y4(x)  + (-8*x)*y5(x)  + (6*x)*y6(x):
> eq4:= theta(y4(x),x)*x^2 =
>       (10*x)*y1(x) + (-10*x)*y2(x) + (14*x)*y3(x) +
>       (1-17*x)*y4(x)  + (16*x)*y5(x)  + (-12*x)*y6(x):
> eq5:= theta(y5(x),x)*x^2 =
>       (5*x)*y1(x) + (-5*x)*y2(x) + (7*x)*y3(x) +
>       (-8*x)*y4(x)  + (1+7*x)*y5(x)  + (-6*x)*y6(x):
> eq6:= theta(y6(x),x)*x^2 =
>       (-5*x)*y1(x) + (5*x)*y2(x) + (-7*x)*y3(x) +
>       (8*x)*y4(x)  + (-8*x)*y5(x)  + (1+5*x)*y6(x):
> eq:=[eq1,eq2,eq3,eq4,eq5,eq6]:
> varl:=[y1(x), y2(x), y3(x), y4(x), y5(x), y6(x)]:
> rez:=uncpl_sys(eq, varl);
```

$$
\begin{aligned}
rez := \Big[&[-x^4\,\theta(\theta(\mathrm{y1}(x))) + 3\,x^3\,\theta(\mathrm{y1}(x)) + 2\,x^2\,\theta(\mathrm{y1}(x)) + 5\,x^2\,\mathrm{y1}(x) \\
&- 5\,x\,\mathrm{y1}(x) - \mathrm{y1}(x) = 0, \mathrm{y2\_z}(x) = x^2\,\theta(\mathrm{y1}(x)) - (1+4\,x)\,\mathrm{y1}(x)], [\\
&-x^2\,\theta(\mathrm{y3}(x)) = 5\,x\,\mathrm{y1}(x) + \mathrm{y2\_z}(x) + (x-1)\,\mathrm{y3}(x), \\
&-x^2\,\theta(\mathrm{y4}(x)) = -10\,x\,\mathrm{y1}(x) - 2\,\mathrm{y2\_z}(x) + (x-1)\,\mathrm{y4}(x), \\
&-x^2\,\theta(\mathrm{y5}(x)) = -5\,x\,\mathrm{y1}(x) - \mathrm{y2\_z}(x) + (x-1)\,\mathrm{y5}(x), \\
&-x^2\,\theta(\mathrm{y6}(x)) = 5\,x\,\mathrm{y1}(x) + \mathrm{y2\_z}(x) + (x-1)\,\mathrm{y6}(x)], \\
&\left[\mathrm{y2}(x) = -\frac{1}{5}\,\frac{\mathrm{y2\_z}(x)}{x} + \frac{7}{5}\,\mathrm{y3}(x) - \frac{8}{5}\,\mathrm{y4}(x) + \frac{8}{5}\,\mathrm{y5}(x) - \frac{6}{5}\,\mathrm{y6}(x)\right]\Big]
\end{aligned}
$$

Here we get (in 3 seconds) as the result a scalar equation of the second order in $y1(x)$, an expression of auxiliary function $y2\_z(x)$ via $y1(x)$, a list of four scalar first-order equations and a linear expression of $y2(x)$ via solutions of the scalar equations and the auxiliary function. Now we can find the solution of the system using procedure `get_sol`:

```
> get_sol(rez,varl);
```

$$\left[\frac{e^{\left(-\frac{1}{x}\right)}(\_C1+\_C2\,x^6)}{x}, -\frac{2}{5}\,\frac{e^{\left(-\frac{1}{x}\right)}(-3\,\_C1+5\,\_C2\,x^6)}{x}, -\frac{e^{\left(-\frac{1}{x}\right)}(\_C2\,x^6-\_C1)}{x},\right.$$
$$\left.\frac{e^{\left(-\frac{1}{x}\right)}(2\,\_C2\,x^6+\_C1)}{x}, \frac{e^{\left(-\frac{1}{x}\right)}(\_C1+\_C2\,x^6)}{x}, -\frac{e^{\left(-\frac{1}{x}\right)}(\_C2\,x^6-\_C1)}{x}\right]$$

Example 4. Consider the previous example in the nonstandard case, taking $\delta = x^2\frac{d}{dx}$:

```
> set_Ore_ring(x, `nonstandard`);
   Enter definition of automorphism sigma
> proc(y) y end;
   Enter definition of map delta
> proc(y) x^2*diff(y,x) end;
   What is the value of theta_1 ?
> 0
   program is set to the  nonstandard case
> eq1:= theta(y1(x),x) =
>       (1+4*x)*y1(x) + (-5*x)*y2(x) + (7*x)*y3(x) +
>       (-8*x)*y4(x)  + (8*x)*y5(x)  + (-6*x)*y6(x):
> eq2:= theta(y2(x),x) =
>       (-10*x)*y1(x) + (9*x+1)*y2(x) + (-14*x)*y3(x) +
>       (16*x)*y4(x)  + (-16*x)*y5(x)  + (12*x)*y6(x):
> eq3:= theta(y3(x),x) =
>       (-5*x)*y1(x) + (5*x)*y2(x) + (-8*x+1)*y3(x) +
>       (8*x)*y4(x)  + (-8*x)*y5(x)  + (6*x)*y6(x):
> eq4:= theta(y4(x),x) =
>       (10*x)*y1(x) + (-10*x)*y2(x) + (14*x)*y3(x) +
>       (1-17*x)*y4(x)  + (16*x)*y5(x)  + (-12*x)*y6(x):
> eq5:= theta(y5(x),x) =
>       (5*x)*y1(x) + (-5*x)*y2(x) + (7*x)*y3(x) +
>       (-8*x)*y4(x)  + (1+7*x)*y5(x)  + (-6*x)*y6(x):
```

```
> eq6:= theta(y6(x),x) =
>       (-5*x)*y1(x) + (5*x)*y2(x) + (-7*x)*y3(x) +
>       (8*x)*y4(x)  + (-8*x)*y5(x)  + (1+5*x)*y6(x):
> eq:=[eq1,eq2,eq3,eq4,eq5,eq6]:
> varl:=[y1(x), y2(x), y3(x), y4(x), y5(x), y6(x)]:
> rez:=uncpl_sys(eq, varl);
```

$$
\begin{aligned}
rez := \Big[&[-\theta(\theta(\mathrm{y1}(x))) + 2\,\theta(\mathrm{y1}(x)) + 5\,x\,\theta(\mathrm{y1}(x)) + 5\,x^2\,\mathrm{y1}(x) - 5\,x\,\mathrm{y1}(x) \\
&- \mathrm{y1}(x) = 0, \mathrm{y2\_\_z}(x) = \theta(\mathrm{y1}(x)) - (1 + 4\,x)\,\mathrm{y1}(x)], [\\
&-\theta(\mathrm{y3}(x)) = 5\,x\,\mathrm{y1}(x) + \mathrm{y2\_\_z}(x) + (x - 1)\,\mathrm{y3}(x), \\
&-\theta(\mathrm{y4}(x)) = -10\,x\,\mathrm{y1}(x) - 2\,\mathrm{y2\_\_z}(x) + (x - 1)\,\mathrm{y4}(x), \\
&-\theta(\mathrm{y5}(x)) = -5\,x\,\mathrm{y1}(x) - \mathrm{y2\_\_z}(x) + (x - 1)\,\mathrm{y5}(x), \\
&-\theta(\mathrm{y6}(x)) = 5\,x\,\mathrm{y1}(x) + \mathrm{y2\_\_z}(x) + (x - 1)\,\mathrm{y6}(x)], \\
&\left[\mathrm{y2}(x) = -\frac{1}{5}\,\frac{\mathrm{y2\_\_z}(x)}{x} + \frac{7}{5}\,\mathrm{y3}(x) - \frac{8}{5}\,\mathrm{y4}(x) + \frac{8}{5}\,\mathrm{y5}(x) - \frac{6}{5}\,\mathrm{y6}(x)\right]\Big]
\end{aligned}
$$

Here the result of uncoupling (obtained in 2 seconds) looks slightly different from the previous one. But again we have a second-order scalar equation in $y1(x)$ and four split first-order scalar equations. Using procedure `get_sol` we obtain the same solution as in the standard case:

```
> get_sol(rez,varl);
```

$$
\begin{aligned}
\Big[&\frac{\mathrm{e}^{(-\frac{1}{x})}\,(\_C1 + \_C2\,x^6)}{x}, -\frac{2}{5}\,\frac{\mathrm{e}^{(-\frac{1}{x})}\,(-3\,\_C1 + 5\,\_C2\,x^6)}{x}, -\frac{\mathrm{e}^{(-\frac{1}{x})}\,(\_C2\,x^6 - \_C1)}{x}, \\
&\frac{\mathrm{e}^{(-\frac{1}{x})}\,(2\,\_C2\,x^6 + \_C1)}{x}, \frac{\mathrm{e}^{(-\frac{1}{x})}\,(\_C1 + \_C2\,x^6)}{x}, -\frac{\mathrm{e}^{(-\frac{1}{x})}\,(\_C2\,x^6 - \_C1)}{x}\Big]
\end{aligned}
$$

Example 5. Now we demonstrate how to use the program in order to get rational solutions of linear systems. Consider the system [7]

$$
\begin{aligned}
y_1' &= y_2, \\
y_2' &= y_3 + y_4, \\
y_3' &= y_5, \\
y_4' &= 2y_1 + 2xy_2 + y_5, \\
y_5' &= x^2 y_1 + 2xy_3 + y_6, \\
y_6' &= x^2 y_2 - 2y_3.
\end{aligned}
$$

```
> set_Ore_ring(x, ‘differential‘):
> eq1:= theta(y1(x),x) =   y2(x):
> eq2:= theta(y2(x),x) =   y3(x) + y4(x):
> eq3:= theta(y3(x),x) =   y5(x):
> eq4:= theta(y4(x),x) =   2*y1(x) + (2*x)*y2(x) + y5(x):
> eq5:= theta(y5(x),x) =   x^2*y1(x) + 2*x*y3(x) + y6(x):
> eq6:= theta(y6(x),x) =   x^2*y2(x) -2*y3(x):
> eq:=[eq1,eq2,eq3,eq4,eq5,eq6]:
> varl:=[y1(x), y2(x), y3(x), y4(x), y5(x), y6(x)]:
> rez:=uncpl_sys(eq, varl);
```

$$
\begin{aligned}
rez := \Big[&[-\theta(\theta(\theta(\%1))) + 6\,\%1 + 4\,x\,\theta(\%1) = 0, \mathrm{y2\_z}(x) = \theta(\mathrm{y1}(x)),\\
&\mathrm{y3\_z}(x) = \%1, \mathrm{y5\_z}(x) = \theta(\%1) - 2\,\mathrm{y1}(x) - 2\,x\,\theta(\mathrm{y1}(x)),\\
&\mathrm{y6\_z}(x) = \theta(\theta(\%1)) - 4\,\theta(\mathrm{y1}(x)) - 6\,x\,\%1 - 2\,x^2\,\mathrm{y1}(x)],\\
&[2\,\theta(\mathrm{y4}(x)) = 4\,\mathrm{y1}(x) + 4\,x\,\mathrm{y2\_z}(x) + \mathrm{y5\_z}(x)], \Big[\\
&\mathrm{y6}(x) = \frac{1}{2}\,\mathrm{y6\_z}(x) + 2\,x\,\mathrm{y4}(x), \mathrm{y5}(x) = \frac{1}{2}\,\mathrm{y5\_z}(x),\\
&\mathrm{y3}(x) = \mathrm{y3\_z}(x) - \mathrm{y4}(x), \mathrm{y2}(x) = \mathrm{y2\_z}(x)\Big]\Big]\\
&\%1 := \theta(\theta(\mathrm{y1}(x)))
\end{aligned}
$$

This result was obtained in 4 seconds and consists of a fifth-order scalar equation in $y1(x)$, a first order equation in $y4(x)$ and linear expressions for the remaining unknowns via solutions of the scalar equations. We can use the procedure `get_sol` with the third parameter set to `rat` in order to find all rational solutions of the initial system:

```
> get_sol(rez, varl, rat);
```

$$
\begin{aligned}
\Big[&\_C_0 + \_C_1\,x, \_C_1, -\_C_2 - \_C_2\,x - \_C_2\,x^2, \_C_2 + \_C_2\,x + \_C_2\,x^2, -\_C_0 - 2\,\_C_1\,x,\\
&(-\_C_1 + 2\,\_C_2)\,x^3 + (-\_C_0 + 2\,\_C_2)\,x^2 + 2\,\_C_2\,x - 2\,\_C_1\Big]
\end{aligned}
$$

6 The uncoupling program family

Another universal algorithm to uncouple linear systems is proposed in [7]. Additionally the paper gives a review and an analysis of known methods for uncoupling systems of linear ordinary differential equations. It was observed that only very few of them can be generalized to arbitrary Ore polynomial rings. Once those methods are generalized, different methods give very different uncoupled equations. The generalization of the successive elimination algorithm which has been done in our publication [4] was considered in this context. The conclusion is that the generalization "tends to yield equations with fewer extra singularities than others, which improves the efficiency of the solving process". This remark gives a hope that the program which we describe can be useful.

Acknowledgement

We would like to thank Francis J. Wright who provided us with useful comments on earlier draft.

References

[1] Ore O. (1933): *Theory of non-commutative polynomials,* Annals of Mathematics **34**, 480–508.

[2] M. Bronstein & M. Petkovšek (1994): *On Ore rings, linear operators and factorisation,* Programmirovanie **1**, 27–45.

[3] Bronstein M., Petkovšek M. (1996): *An Introduction to Pseudo-Linear Algebra,* Theoretical Computer Science, **157**, 3–33.

[4] Abramov S.A., Zima E.V. (1995): *Ore polynomial rings and linear systems reduction (in Russian)*, Vestnik MGU, Ser.15, Computat. Math. and Cybernetics **3**, 50–56.

[5] Murray F.J., Miller K.S. (1954): *Existence theorems for ordinary differential equations.* New York Univ. Press, Intersciences, New York.

[6] Vilenkin N.Ja. (1969) *Combinatorics. (in Russian)* Nauka, Moscow.

[7] Bronstein M., Zürcher B. (1996): *Uncoupling Algorithms for Pseudo-Linear Systems,* (submited to AAECC).

[8] Barkatou M.A. *An algorithm for computing a companion block diagonal form for a system of linear differential equations.* Applicable Algebra in Engineering, Communication and Computing, 1993, **4**, pp. 185-195.