The MAPLE Package "Janet": II. Linear Partial Differential Equations

Yuri A. Blinkov[1⋆], Carlos F. Cid[2⋆⋆], Vladimir P. Gerdt[3⋆⋆⋆], Wilhelm Plesken[2†], and Daniel Robertz[2‡]

[1] Department of Mathematics and Mechanics, Saratov University, 410071 Saratov, Russia
[2] Lehrstuhl B für Mathematik, RWTH Aachen, Templergraben 64, D-52062 Aachen, Germany
[3] Laboratory of Information Technologies, Joint Institute for Nuclear Research, 141980 Dubna, Russia

Abstract. The MAPLE package "Janet"[1] comes in two parts, the first dealing with polynomials and commutative algebra, the second with linear PDEs. Here the second part is described. Amongst others it contains the first MAPLE implementation of the Involutive algorithm which brings systems of linear PDEs to a form from which a quantitative analysis of the space of power series solutions becomes possible.

1 Introduction

This is the second of two papers introducing the MAPLE package "Janet". Whereas the first part, [4], commented about polynomial systems being the special case of linear PDEs with constant coefficients, the present part deals with general systems of linear PDEs, also based on the Involutive algorithm in the form presented in [11]. More precisely, our implementation fixes Janet separation of variables into multiplicative and non-multiplicative ones [15] as the input involutive division [10] for the algorithm that produces a PDE Janet basis in the output. By means of this technique the canonical involutive normal forms for systems of linear partial differential equations can be produced.

Below is a list of the commands available in the PDE part of "Janet": With the commands in the first group one can create a Janet basis, produce normal forms of differential expressions, print out the Janet basis with some relevant extra information and get quantitative information about the free Taylor coefficients, called parametric derivatives. The usage with definitions and some typical examples and comments on the algorithms will be given in section 2. The section 3 comments on compatibility conditions for right hand sides of the equations or, equivalently in the language of modules, on syzygies and free resolutions for modules over the ring of differential operators. The next section 4 gives further

⋆ BlinkovUA@info.sgu.ru
⋆⋆ cfcid@momo.math.rwth-aachen.de
⋆⋆⋆ gerdt@jinr.ru
† plesken@momo.math.rwth-aachen.de
‡ daniel@momo.math.rwth-aachen.de
[1] available at http://wwwb.math.rwth-aachen.de/Janet

examples. The first one discusses on the theoretical side a LIE algebra technique to construct all polynomial solutions of linear PDEs with a sufficiently large symmetry LIE algebra and on the practical side the interaction of the present package with the MAPLE package "jets", cf. [1], [2]. The second example is on differential elimination, the third on finding autonomous elements (in the context of control theory). Finally we compare "Janet" to other MAPLE packages w. r. t. time consumption. We give a table of timings.

We have taken big effort to produce detailed online documentation in the form of help pages with examples, so that we can be brief in this account. Some functions were taken over from "jets"; in particular it is possible to use the more convenient jet notation to some extent, as will be demonstrated in this paper. It is also possible to call the PDE-solver function of the MAPLE package "DESOLV", cf. [7], [21], with the output of JanetBasis as input via the function Jpdesolv. In fact, it was one of the early successes of our "Janet" package, that various big systems of linear PDEs coming up in the symmetry analysis of various nonlinear PDE systems could only be solved by "pdesolv" after they had been processed by the JANET algorithm.

The first implementation of the Involutive algorithm [11] in MAPLE was reported in [16]. Then this algorithm was implemented in Mathematica [3]. A slightly different algorithmic approach inspired by [10], [11] is under development in [8] and is also to be implemented in MAPLE. One more implementation of completion of linear PDE systems to the JANET involutive form is done in MuPAD [13]. This implementation combines the algebraic methods [11] with a geometric approach to involution (see [14] and references therein).

Since an involutive system of linear PDEs is a (generally redundant) differential Gröbner basis, the efficiency of the "Janet" package can be compared with that of MAPLE packages "diffalg" and "Rif" which also construct differential Gröbner bases for linear systems. In section 5 we give some timings and notes on the implementation.

2 Basics

Let F be a field of meromorphic functions in $x_1, \ldots, x_n$ over a field of constants $K \subseteq \mathbb{C}$ closed under all partial derivatives $\frac{\partial}{\partial x_i}$ for $i = 1, \ldots, n$, e. g. $F = \mathbb{Q}(x_1, \ldots, x_n)$, and let $R := F\langle \frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n} \rangle$ be a ring of operators on F generated by F and the partial derivatives $\frac{\partial}{\partial x_i}$ for $i = 1, \ldots, n$, where F acts on itself by multiplication. Clearly, R is non-commutative and any element of R can

Basic commands:	
JanetBasis	InvReduce
PrincDeriv (=TabVar)	HilbertSeries
SolSeries	PolySol
ParamDeriv	ZeroSets
Commands for special applications:	
CompCond	Resolution
Autonom (=Torsion)	SyzOp
WeightedHilbertSeries	
Commands for various invariants derivable from HilbertSeries:	
IndexRegularity	CartanCharacter
HilbertPolynomial	HilbertFunction
HP	HF
Auxiliary Commands:	
LeadingDeriv	Jpdesolv
AffEqn	AssertJanetBasis
Diff2Pol	Pol2Diff
CmpOp	JAdjoint
Diff2Op	AppOp
Ind2Diff	Diff2Ind
Pol2Ind	AppOpInd

be written in the form

$$\sum_i a_i \frac{\partial^{|i|}}{\partial x_1^{i_1} \cdots \partial x_n^{i_n}}$$

with $a_i = a_i(x_1, \ldots, x_n) \in F$ and i running through a finite subset of $(\mathbb{Z}_{\geq 0})^n$ with $|i| := \sum_{j=1}^n i_j$. In general K can be any subfield of $\mathbb{C}$, in which one has a normal form for elements and where addition, subtraction, multiplication, and division can constructively be carried out. For our MAPLE implementation it means $K = \mathbb{Q}$ or a finitely generated field extension of $\mathbb{Q}$. Given q-tuples $A_1, \ldots, A_a \in R^{1\times q}$ we are concerned with the linear system of PDEs given by

$$A_i u = 0 \qquad (i = 1, \ldots, a) \;\text{ with }\; u := \begin{pmatrix} u_1 \\ \vdots \\ u_q \end{pmatrix} \tag{1}$$

for the unknown functions $u_1, \ldots, u_q$ in $x_1, \ldots, x_n$ (which one may think of as dependent variables). The aim is to obtain quantitative control over the power series solutions of (1). At this stage a trivial, nevertheless important remark will provide the bridge between the polynomial case in part 1, cf. [4] and the present case.

Remark 1.

$$R^{1\times q} \to \oplus_{i=1}^q R u_i : Z \mapsto Zu$$

is an isomorphism of left R-modules. This isomorphism maps the R-submodule

$$S := \langle A_1, \ldots, A_a \rangle \leq R^{1\times q}$$

of $R^{1\times q}$ spanned by the A_i onto the submodule S' of $\oplus_{i=1}^{q} Ru_i$ which are consequences of the PDE system (1). In particular, the R-factor module

$$M := R^{1\times q}/S$$

is identified with the (analytically more familiar) R-module

$$M' := \oplus_{i=1}^{q} Ru_i/S',$$

whose presentation is given by (1).

So, when we write expressions in MAPLE, we seem to write elements in $\oplus_{i=1}^{q} Ru_i$, but we need to know them modulo S', i. e. as elements of M'. It will soon become clear that the knowledge of M' is really the (quantitative, formal) control over the power series solutions of (1), which is our declared aim. It is achieved by the concepts of JANET basis, multiplicative variables, and parametric derivatives. The JANET basis B consists of finitely many elements $B_1, \ldots, B_d \in R^{1\times q}$, whose equations taken together have the same set of power series solutions as the original equations (1). Moreover, each B_i has attached multiplicative variables from among the x_i to it, such that each equation that is a consequence of (1) can be written uniquely as an F-linear combination of the derivatives of the B_i with respect to B_i-multiplicative variables. In more algebraic terms: The R-submodule S of the free R-module $R^{1\times q}$ spanned by $A_1, \ldots, A_a$ is also spanned by $B_1, \ldots, B_d$ in such a way that the

$$\frac{\partial^{|i|}}{\partial x_1^{i_1} \cdots \partial x_n^{i_n}} B_l$$

where i runs through the subset of those elements of $(\mathbb{Z}_{\geq 0})^n$ with $i_s = 0$, whenever x_s is not multiplicative for B_l, form an F-basis (in the sense of vector spaces) for this submodule. This allows a close comparison to the polynomial case, cf. [4]. The so called monomial basis in the polynomial case corresponds to parametric derivatives here. $\frac{\partial^{|i|} u_l}{\partial x_1^{i_1} \cdots \partial x_n^{i_n}}$ is called a parametric derivative, if it does not occur as the leading derivative in any of the equations $Bu = 0$, where B is derivative of any of the B_i. All other derivatives of the u_l are called principal. The concept of involutive division allows to express any principal derivative in terms of the parametric derivatives. The resulting expression is unique. Concerning the ordering of the derivatives of the u_l, the same possibilities as in the polynomial case are realized, the degree reverse lexicographic option with the additional option

of favouring the component with the highest degree is usually chosen. Here is a rough description of input and output:

Input: $A_1, \ldots, A_a \in R^q$ generating the submodule S of R^q (and a list of indeterminates, e. g. $x_1, \ldots, x_n$.)
Output: The JANET basis $B_1, \ldots, B_d$ of S by the call JanetBasis.
Subsequent commands:
TabVar or equivalently PrincDeriv reproduces each B_i, the leading term of B_i, and the subset $M_i \subseteq \{x_1, \ldots, x_n\}$ of multiplicative variables of B_i with respect to B, i. e. each element s of S has a unique representation as

$$s = \sum_{i=1}^{d} p_i B_i \text{ with } p_i \in F\langle \frac{\partial}{\partial x_i} | x_i \in M_i \rangle.$$

The second name for the command explains itself from the fact that the output allows to read off all principal derivatives, i. e. those derivatives, which do not occur in a reduced differential expression. They are given by the derivatives of the highest terms of the B_i with respect to B_i-multiplicative variables.
ParamDeriv enumerates all parametric derivatives.
HilbertSeries gives the generating function for the numbers of the parametric derivatives according to their order.
With further **input** $v \in \oplus_{i=1}^{q} R u_i$ the command InvReduce produces the normalised representative v' of the coset $v + S' \in M'$, i. e. an expression involving only parametric derivatives. E. g., if v is some partial higher derivative of some u_i, then v gets expressed in terms of parametric derivatives.
ZeroSets describes the points which cannot be taken as center of a power series expansion for the solutions, i. e. lists the functions by which one has divided in the course of the algorithm.
SolSeries computes the power series solutions of (1) up to an order given as (additional) input.
PolySol computes the polynomial solutions of (1) up to a degree given as (additional) input.

The first example deals with linear PDEs with constant coefficients. It is completely parallel to Example (2.2) of the first part [4]. All the functions above are demonstrated except for ZeroSets, because in the constant coefficients case one needs not to divide by non-constant functions.

Example 1. (cf. Example (2.2) of [4])

Specification of independent and dependent variables:

```
> ivar:=[x,y]; dvar:=[u,v];
```

$$ivar := [x,\, y]$$

$$dvar := [u,\, v]$$

Tuples of polynomials and their translation into linear differential expressions with constant coefficients, which constitute the PDE system:

```
> l:=[[x,-y],[y,x],[y^3,0]];
```

$$l := [[x,\, -y],\, [y,\, x],\, [y^3,\, 0]]$$

```
> L:=Pol2Diff(l,ivar,dvar);
```

$$L := [(\tfrac{\partial}{\partial x}\,\mathrm{u}(x,\, y)) - (\tfrac{\partial}{\partial y}\,\mathrm{v}(x,\, y)),\, (\tfrac{\partial}{\partial y}\,\mathrm{u}(x,\, y)) + (\tfrac{\partial}{\partial x}\,\mathrm{v}(x,\, y)),\, \tfrac{\partial^3}{\partial y^3}\,\mathrm{u}(x,\, y)]$$

Computation of the JANET basis:

```
> J:=JanetBasis(L,ivar,dvar);
```

$$J := [[(\tfrac{\partial}{\partial y}\,\mathrm{u}(x,\, y)) + (\tfrac{\partial}{\partial x}\,\mathrm{v}(x,\, y)),\, (\tfrac{\partial}{\partial x}\,\mathrm{u}(x,\, y)) - (\tfrac{\partial}{\partial y}\,\mathrm{v}(x,\, y)),\, \tfrac{\partial^3}{\partial y^3}\,\mathrm{u}(x,\, y),$$
$$\tfrac{\partial^4}{\partial y^4}\,\mathrm{v}(x,\, y)],\, [x,\, y],\, [u,\, v]]$$

Details on the JANET basis: first the basis vector (equation), next the multiplicative variables with the numbers referring to `ivar` and stars indicating non-multiplicative variables, and finally the highest term, thus allowing to read off the principal derivatives. There are four elements in the JANET basis.

```
> PrincDeriv();
```

$$[(\tfrac{\partial}{\partial y}\,\mathrm{u}(x,\, y)) + (\tfrac{\partial}{\partial x}\,\mathrm{v}(x,\, y)),\, [x,\, y],\, \tfrac{\partial}{\partial x}\,\mathrm{v}(x,\, y)]$$

$$[(\tfrac{\partial}{\partial x}\,\mathrm{u}(x,\, y)) - (\tfrac{\partial}{\partial y}\,\mathrm{v}(x,\, y)),\, [x,\, y],\, \tfrac{\partial}{\partial x}\,\mathrm{u}(x,\, y)]$$

$$[\tfrac{\partial^3}{\partial y^3}\,\mathrm{u}(x,\, y),\, [*,\, y],\, \tfrac{\partial^3}{\partial y^3}\,\mathrm{u}(x,\, y)]$$

$$[\tfrac{\partial^4}{\partial y^4}\,\mathrm{v}(x,\, y),\, [*,\, y],\, \tfrac{\partial^4}{\partial y^4}\,\mathrm{v}(x,\, y)]$$

The parametric derivatives and their generating function, the HILBERT series:

```
> ParamDeriv(ivar,dvar);
```

$$[\mathrm{v}(x,\, y),\, \tfrac{\partial}{\partial y}\,\mathrm{v}(x,\, y),\, \tfrac{\partial^2}{\partial y^2}\,\mathrm{v}(x,\, y),\, \tfrac{\partial^3}{\partial y^3}\,\mathrm{v}(x,\, y),\, \mathrm{u}(x,\, y),\, \tfrac{\partial}{\partial y}\,\mathrm{u}(x,\, y),\, \tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,\, y)]$$

```
> HilbertSeries(t);
```

$$2 + 2\,t + 2\,t^2 + t^3$$

Normalising differential expressions:

```
> InvReduce(diff(u(x,y),x,x)+diff(v(x,y),y,y),J);
```

$$-(\tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,\, y)) + (\tfrac{\partial^2}{\partial y^2}\,\mathrm{v}(x,\, y))$$

Commands without analogues in the polynomial case: Computing the TAYLOR expansion of the power series solutions up to a given order (3 in this case) and computing polynomial solutions up to a given degree (also 3 in this case). That there are 7 free parameters for the TAYLOR expansion was to be expected from the HILBERT series. That all the expansions up to order three are already solutions is new information.

```
> SolSeries(J,3,'SO');
```

$$[\mathrm{u}(x,\,y) = C1_{0,0} + C2_{0,1}\,x + C1_{0,1}\,y - \frac{1}{2}\,C1_{0,2}\,x^2 + C2_{0,2}\,x\,y + \frac{1}{2}\,C1_{0,2}\,y^2$$
$$-\frac{1}{6}\,C2_{0,3}\,x^3 + \frac{1}{2}\,C2_{0,3}\,x\,y^2, \quad \mathrm{v}(x,\,y) = C2_{0,0} - C1_{0,1}\,x + C2_{0,1}\,y$$
$$-\frac{1}{2}\,C2_{0,2}\,x^2 - C1_{0,2}\,x\,y + \frac{1}{2}\,C2_{0,2}\,y^2 - \frac{1}{2}\,C2_{0,3}\,x^2\,y + \frac{1}{6}\,C2_{0,3}\,y^3]$$

```
> SO;
```

$$[[\mathrm{u}(x,y) = 1, \mathrm{v}(x,y) = 0],\ [\mathrm{u}(x,y) = x, \mathrm{v}(x,y) = y],\ [\mathrm{u}(x,y) = y, \mathrm{v}(x,y) = -x],$$
$$[\mathrm{u}(x,y) = -\frac{1}{2}x^2 + \frac{1}{2}y^2, \mathrm{v}(x,y) = -xy],\ [\mathrm{u}(x,y) = xy, \mathrm{v}(x,y) = -\frac{1}{2}x^2 + \frac{1}{2}y^2],$$
$$[\mathrm{u}(x,y) = -\frac{1}{6}x^3 + \frac{1}{2}xy^2, \mathrm{v}(x,y) = -\frac{1}{2}x^2y + \frac{1}{6}y^3],\ [\mathrm{u}(x,y) = 0, \mathrm{v}(x,y) = 1]]$$

```
> PolySol(J,3,'P'):
> evalb(P=SO);
```

$$true$$

If in the previous example the linear PDE system with constant coefficients would have been really big, one could proceed as follows: Find the JANET basis for the polynomial system using InvolutiveBasisFast, cf. [4], rewrite the polynomial JANET basis as PDE system using Pol2Diff, and tell the system that this is a PDE JANET basis by invoking AssertJanetBasis.

The next example deals with non-constant coefficients, also demonstrating the difficulties arising from division by non-constant functions.

Example 2. ($n = 3, q = 3$, non-constant coefficients) The following system describes the set of all vector fields in 3-space commuting with the infinitesimal rotations around the z- and the y-axis:

```
> ivar:=[x,y,z]; dvar:=[s,t,u];
```

$$ivar := [x,\,y,\,z]$$
$$dvar := [s,\,t,\,u]$$

```
> L:=[y*diff(s(x,y,z),x)-x*diff(s(x,y,z),y)-t(x,y,z),
> y*diff(t(x,y,z),x)-x*diff(t(x,y,z),y)+s(x,y,z),
> y*diff(u(x,y,z),x)-x*diff(u(x,y,z),y),
> z*diff(s(x,y,z),x)-x*diff(s(x,y,z),z)-u(x,y,z),
> z*diff(t(x,y,z),x)-x*diff(t(x,y,z),z),
> z*diff(u(x,y,z),x)-x*diff(u(x,y,z),z)+s(x,y,z)];
```

$$L := [y(\tfrac{\partial}{\partial x}\mathrm{s}(x,y,z)) - x(\tfrac{\partial}{\partial y}\mathrm{s}(x,y,z)) - \mathrm{t}(x,y,z),$$
$$y(\tfrac{\partial}{\partial x}\mathrm{t}(x,y,z)) - x(\tfrac{\partial}{\partial y}\mathrm{t}(x,y,z)) + \mathrm{s}(x,y,z), y(\tfrac{\partial}{\partial x}\mathrm{u}(x,y,z)) - x(\tfrac{\partial}{\partial y}\mathrm{u}(x,y,z)),$$
$$z(\tfrac{\partial}{\partial x}\mathrm{s}(x,y,z)) - x(\tfrac{\partial}{\partial z}\mathrm{s}(x,y,z)) - \mathrm{u}(x,y,z), z(\tfrac{\partial}{\partial x}\mathrm{t}(x,y,z)) - x(\tfrac{\partial}{\partial z}\mathrm{t}(x,y,z)),$$
$$z(\tfrac{\partial}{\partial x}\mathrm{u}(x,y,z)) - x(\tfrac{\partial}{\partial z}\mathrm{u}(x,y,z)) + \mathrm{s}(x,y,z)]$$

```
> J:=JanetBasis(L,ivar,dvar);
```

$$J := [[\frac{z\,\mathrm{t}(x,\,y,\,z)}{z-y} - \frac{y\,\mathrm{u}(x,\,y,\,z)}{z-y}, -\frac{z\,\mathrm{s}(x,\,y,\,z)}{z-x} + \frac{x\,\mathrm{u}(x,\,y,\,z)}{z-x},$$
$$y\,\mathrm{u}(x,\,y,\,z) + z^2\,(\tfrac{\partial}{\partial y}\,\mathrm{u}(x,\,y,\,z)) - y\,z\,(\tfrac{\partial}{\partial z}\,\mathrm{u}(x,\,y,\,z)),$$
$$x\,\mathrm{u}(x,\,y,\,z) + z^2\,(\tfrac{\partial}{\partial x}\,\mathrm{u}(x,\,y,\,z)) - x\,z\,(\tfrac{\partial}{\partial z}\,\mathrm{u}(x,\,y,\,z))],\,[x,\,y,\,z],\,[s,\,t,\,u]]$$

```
> TabVar();
```

$$[\frac{z\,\mathrm{t}(x,\,y,\,z)}{z-y} - \frac{y\,\mathrm{u}(x,\,y,\,z)}{z-y},\,[x,\,y,\,z],\,\frac{z\,\mathrm{t}(x,\,y,\,z)}{z-y}]$$
$$[-\frac{z\,\mathrm{s}(x,\,y,\,z)}{z-x} + \frac{x\,\mathrm{u}(x,\,y,\,z)}{z-x},\,[x,\,y,\,z],\,-\frac{z\,\mathrm{s}(x,\,y,\,z)}{z-x}]$$
$$[y\mathrm{u}(x,y,z) + z^2(\tfrac{\partial}{\partial y}\mathrm{u}(x,y,z)) - yz(\tfrac{\partial}{\partial z}\mathrm{u}(x,,y,z)),\,[*,y,z],\,z^2(\tfrac{\partial}{\partial y}\mathrm{u}(x,y,z))]$$
$$[x\mathrm{u}(x,y,z) + z^2(\tfrac{\partial}{\partial x}\mathrm{u}(x,y,z)) - xz(\tfrac{\partial}{\partial z}\mathrm{u}(x,y,z)),\,[x,y,z],\,z^2(\tfrac{\partial}{\partial x}\mathrm{u}(x,y,z))]$$

```
> ZeroSets();
```

$$[[y,\,\{y=0\}],\,[x\,z,\,\{x=0,\,z=0\}],\,[z,\,\{z=0\}],\,[z\,(-y+x),\,\{x=y,\,z=0\}],$$
$$[x,\,\{x=0\}],\,[y\,z,\,\{z=0,\,y=0\}],\,[y\,z-y^2,\,\{y=z,\,y=0\}],$$
$$[y\,(z-x),\,\{x=z,\,y=0\}],\,[z^2,\,\{z=0\}]]$$

```
> HilbertSeries(lambda);
```

$$1 + \frac{\lambda}{1-\lambda}$$

```
> ParamDeriv(ivar,dvar);
```

$$[0,\,0,\,\frac{1}{1-z}]$$

Because of the result of ZeroSets one cannot expand in $(0,0,0)$. We choose $(1,2,3)$ instead:

```
> PolySol(J,5,[1,2,3],'d'):

> d;
```

$$[[\mathrm{s}(x,y,z)=x,\ \mathrm{t}(x,y,z)=y,\ \mathrm{u}(x,y,z)=z],[\mathrm{s}(x,y,z)=-\frac{16}{3}x+\frac{1}{6}xz^2$$
$$+\frac{1}{6}y^2x+\frac{1}{6}x^3,\ \mathrm{t}(x,y,z)=-\frac{16}{3}y+\frac{1}{6}yz^2+\frac{1}{6}yx^2+\frac{1}{6}y^3,\ \mathrm{u}(x,y,z)=-\frac{16}{3}z$$
$$+\frac{1}{6}z^3+\frac{1}{6}y^2z+\frac{1}{6}x^2z],[\mathrm{s}(x,y,z)=\frac{125}{6}x+\frac{1}{60}x^3y^2-\frac{5}{6}xz^2+\frac{1}{120}x^5$$
$$+\frac{1}{60}x^3z^2+\frac{1}{120}xy^4+\frac{1}{60}y^2xz^2+\frac{1}{120}xz^4-\frac{5}{6}y^2x-\frac{5}{6}x^3,\ \mathrm{t}(x,y,z)=\frac{125}{6}y$$
$$-\frac{5}{6}yz^2-\frac{5}{6}yx^2+\frac{1}{120}x^4y+\frac{1}{120}yz^4-\frac{5}{6}y^3+\frac{1}{60}yx^2z^2+\frac{1}{60}x^2y^3+\frac{1}{120}y^5$$
$$+\frac{1}{60}y^3z^2,\ \mathrm{u}(x,y,z)=\frac{125}{6}z-\frac{5}{6}z^3+\frac{1}{60}y^2z^3+\frac{1}{120}z^5+\frac{1}{60}x^2z^3-\frac{5}{6}y^2z$$
$$+\frac{1}{120}y^4z-\frac{5}{6}x^2z+\frac{1}{60}y^2x^2z+\frac{1}{120}x^4z]]$$

Note, because of the HILBERT series, SolSeries would have produced 5 independent expansions of solutions. Hence not all solutions can be expanded by polynomial solutions. But since all the functions, whose LIE derivatives with the two infinitesimal rotations are functions of $\sqrt{x^2+y^2+z^2}$ (use the JANET-program to prove this!), the above HILBERT series tells us, that the $R[\sqrt{x^2+y^2+z^2}](xD_x+yD_y+zD_z)$ is dense in the space of all (outside 0 analytic) vector fields commuting with the two (and hence all) infinitesimal rotations.

Whatever has been said in part 1, [4], about orderings and gradings for the variables applies in the present case as well. The role of the various components in the polynomial case has been taken over by the dependent variables. Again term over position is usually much more effective. Unfortunately there is no C++-implementation for the present case available yet.

The way differential expressions are written in MAPLE is rather clumsy for typing input as seen in the last example, whereas the constant coefficient case is dealt with in a satisfactory manner as demonstrated in the first example. Therefore we have taken over the jet notation from the MAPLE package "jets", cf. [1], [2], and provided two functions Ind2Diff and Diff2Ind to translate jet expressions into differential expressions and back again. For instance, if u is a dependent variable and x, y, z are the independent variables, then the jet variable u_{xxyz} stands for the derivative in MAPLE notation $\frac{\partial^4 u(x,y,z)}{\partial x^2 \partial y \partial z}$. In the subsequent examples this more convenient way for producing input will be used.

3 Compatibility Conditions and Syzygies

In this section we discuss (local) compatibility conditions for right hand sides of linear PDEs, a well known example being the characterisation of gradients via

the start of the POINCARÉ sequence. The way one goes about it, is to introduce a name for each right hand side of an equation and to get a compatibility condition each time the left hand side gets zero in the Involutive algorithm. Here is an example, for which JANET already used the corresponding homogeneous system for demonstrating his algorithm:

Example 3. We first define the system to be investigated by using jet notation:

```
> ivar:=[x,y,z]:dvar:=[u]:
> Lj:=[u[z,z]-y*u[x,x],u[y,y]];
```

$$Lj := [u_{z,z} - y\,u_{x,x},\ u_{y,y}]$$

The aim is to check for which right hand sides the system

```
> Lh:=Ind2Diff(Lj,ivar,dvar);
```

$$Lh := [(\tfrac{\partial^2}{\partial z^2}\,\mathrm{u}(x,\,y,\,z)) - y\,(\tfrac{\partial^2}{\partial x^2}\,\mathrm{u}(x,\,y,\,z)),\ \tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,\,y,\,z)]$$

has solutions. Therefore we introduce names $a(x, y, z)$ and $b(x, y, z)$ for the right hand sides as follows and compute a JANET basis for the resulting system in the usual manner (by carrying the new functions along):

```
> L:=AffEqn(Lh,ivar,[a,b]);
```

$$L := [(\tfrac{\partial^2}{\partial z^2}\,\mathrm{u}(x,\,y,\,z)) - y\,(\tfrac{\partial^2}{\partial x^2}\,\mathrm{u}(x,\,y,\,z)) - \mathrm{a}(x,\,y,\,z),\ (\tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,\,y,\,z)) - \mathrm{b}(x,\,y,\,z)]$$

```
> JL:=JanetBasis(L,ivar,dvar):
```

Whenever a left hand side becomes zero, one gets a compatibility condition. They are collected in the global variable COMPA. Other compatibility conditions come from expressing the original equations in `L` in terms of the JANET basis and from reducing prolongations of the elements of the JANET basis by non-multiplicative variables. All these can be obtained with the command CompCond. To save space, the output is turned into jet notation via the command Diff2Ind:

```
> Diff2Ind(CompCond(L,ivar,dvar),ivar,[a,b]);
```

$$\begin{aligned}
&[\frac{1}{2}\,b_{x,y,z,z}\,y^2 - \frac{1}{2}\,a_{x,y,y,y}\,y^2 - \frac{1}{2}\,b_{x,x,x,y}\,y^3 - \frac{3}{2}\,b_{x,x,x}\,y^2, \frac{1}{2}\,b_{z,z,z,z,z,z}\,y \\
&-\frac{1}{2}\,a_{y,y,z,z,z,z}\,y - \frac{3}{2}\,b_{x,x,z,z,z,z}\,y^2 - a_{x,x,y,z,z}\,y + a_{x,x,y,y,z,z}\,y^2 \\
&+\frac{3}{2}\,b_{x,x,x,z,z,z}\,y^3 - a_{x,x,x,x}\,y + a_{x,x,x,x,y}\,y^2 - \frac{1}{2}\,a_{x,x,x,x,y,y}\,y^3 \\
&-\frac{1}{2}\,b_{x,x,x,x,x,x}\,y^4, \frac{3}{2}\,b_{x,x,x,z,z}\,y^2 - \frac{3}{2}\,b_{x,x,x,x,x}\,y^3 - \frac{1}{2}\,b_{x,y,z,z,z,z}\,y^2 \\
&+\frac{1}{2}\,a_{x,y,y,y,z,z}\,y^2 + b_{x,x,x,y,z,z}\,y^3 - \frac{1}{2}\,a_{x,x,x,y,y,y}\,y^3 - \frac{1}{2}\,b_{x,x,x,x,x,y}\,y^4, \\
&\frac{1}{2}\,b_{y,z,z}\,y^2 - \frac{1}{2}\,a_{y,y,y}\,y^2 - \frac{1}{2}\,b_{x,x,y}\,y^3 - \frac{3}{2}\,b_{x,x}\,y^2, \frac{3}{2}\,b_{x,x,z,z}\,y^2 - \frac{3}{2}\,b_{x,x,x,x}\,y^3 \\
&-\frac{1}{2}b_{y,z,z,z,z}y^2 + \frac{1}{2}a_{y,y,y,z,z}y^2 + b_{x,x,y,z,z}y^3 - \frac{1}{2}a_{x,x,y,y,y}y^3 - \frac{1}{2}b_{x,x,x,x,y}y^4]
\end{aligned}$$

```
> HilbertSeries(t);
```

$$1 + 3\,t + 4\,t^2 + 3\,t^3 + t^4$$

Summarizing, we have a 12-dimensional affine space of solutions, whenever $a(x, y, z)$ and $b(x, y, z)$ satisfy the equations above. The space of solutions of the homogeneous system can easily be computed with JanetBasis and PolySol.

Of course, one can also check for given specific right hand sides, whether the system allows solutions in the same way: The command CompCond should produce zeros only, resp. the empty list. In this case, the commands SolSeries and PolySol can be used as in the homogeneous case to look at expansions of solutions and polynomial solutions.

It is worthwhile to have a look at the compatibility from the module point of view. Note, $M = R^{1\times q}/S$ can be viewed as the cokernel of the homomorphism

$$\alpha : R^{1\times a} \to R^{1\times q} : z \mapsto zA$$

of left R-modules, where the rows of $A \in R^{a\times q}$ are $A_1, \ldots, A_a$. Another way of viewing the command CompCond is that it computes the kernel of α. More precisely, if CompCond is performed after the command JanetBasis on input with general right hand side as above, the result can be interpreted as a homomorphism

$$\beta : R^{1\times b} \to R^{1\times a} : z \mapsto zB$$

where $B \in R^{b\times a}$ has rows corresponding (in the sense of 1) to the elements of the output of CompCond. In particular,

$$R^{1\times b} \xrightarrow{\beta} R^{1\times a} \xrightarrow{\alpha} R^{1\times q} \xrightarrow{\nu} M \to 0,$$

where ν is the natural epimorphism, is the beginning of a free resolution of M as left R-module.

Until now it was not necessary to introduce a proper notation for the elements of the ring R, since we got away with describing the result of applying them to a general function. With the matrices now, we need a proper notation, which is again adopted from the "jets" package, cf. [1], [2], and goes as follows: An element of R is always written in square brackets, which are part of the name. Each summand $a_i \frac{\partial^{|i|}}{\partial x_1^{i1} \cdots \partial x_n^{in}}$ is written inside these brackets as $[a_i, [\underbrace{x_1, \ldots, x_1}_{i_1}, \ldots, \underbrace{x_n, \ldots, x_n}_{i_n}]]$. These terms are separated by commas. Things will become clear in the next example. Matrices over R can be constructed from tuples of differential expressions with the command Diff2Op, the reverse command being AppOp.

Continuing with the discussion of the matrices for α and β above, one can start all over with the matrix B and iterate to construct a free resolution of M. But there

is a way to construct the resolution in one go by using the JANET basis as relations for M. This resolution is necessarily finite, more precisely it finishes after k steps, if k is the maximal number of non-multiplicative variables for the members of the JANET basis. An example follows below. There is another version of this resolution producing only with first order differential operators for the higher syzygies, cf. [17], which can also be realized in the package with the command SyzOp, which works only with operator input and will not be demonstrated here.

Example 4. We start again with the following system:

```
> L:=[exp(y)*diff(u(x,y,z),x)-y^2*diff(u(x,y,z),y,z),
> diff(u(x,y,z),x,z)];
```

$$L := [e^y\,(\tfrac{\partial}{\partial x}\,\mathrm{u}(x,\,y,\,z)) - y^2\,(\tfrac{\partial^2}{\partial z\,\partial y}\,\mathrm{u}(x,\,y,\,z)),\ \tfrac{\partial^2}{\partial z\,\partial x}\,\mathrm{u}(x,\,y,\,z)]$$

and specify the independent and dependent variables:

```
> ivar:=[x,y,z]:dvar:=[u]:
> Resolution(L,ivar,dvar);
```

$$\left[\begin{bmatrix} 0 & [[1,\,[x]]] & [[-1,\,[z]]] \\ [[1,\,[x]]] & [[y^2,\,[y]]] & [[-e^y,\,[]]] \end{bmatrix},\ \begin{bmatrix} [[-y^2,\,[y,\,z]],\,[e^y,\,[x]]] \\ [[1,\,[x,\,z]]] \\ [[1,\,[x,\,x]]] \end{bmatrix}\right]$$

The column on the right is to be compared with the JANET basis of L:

```
> JanetBasis(L,ivar,dvar);
```

$$[[e^y(\tfrac{\partial}{\partial x}\mathrm{u}(x,y,z)) - y^2(\tfrac{\partial^2}{\partial z\partial y}\mathrm{u}(x,y,z)),\ \tfrac{\partial^2}{\partial z\partial x}\mathrm{u}(x,y,z),\ \tfrac{\partial^2}{\partial x^2}\mathrm{u}(x,y,z)],\ [x,\,y,\,z],\ [u]]$$

4 Further Examples

The JANET package can very well be used together with the package "jets", which can perform all sorts of jet calculations. Here is an example:

Example 5. The aim is to construct recursively all polynomial solutions of the heat equation by using its symmetry LIE algebra.

We use the package "jets" to set up the equation for the LIE algebra. Here we define the independent and dependent variables and write down the equation in "jet" notation:

```
> with(jets):with(Janet):
> ivar:=[x,t]; dvar:=[u];
```

$$ivar := [x,\,t]$$

$$dvar := [u]$$

```
> eq:=[u[x,x]+u[t]];
```

$$eq := [u_{x,\,x} + u_t]$$

We will only be interested in symmetry vector fields of the following form:

```
> defvec("lin",ivar,dvar);
```

$$[[[\xi_x(x,t),[x]],[\xi_t(x,t),[t]],[u\,\eta_{u,u}(x,t),[u]]],[x,t],[\xi_x(x,t),\xi_t(x,t),\eta_{u,u}(x,t)]]$$

Here is the "jets" command to set up the equations

```
> le:=gengen(eq,"lin",ivar,dvar);
```

$$le := [[\tfrac{\partial}{\partial x}\,\xi_t(x,t),\,(\tfrac{\partial}{\partial t}\,\eta_{u,u}(x,t))+(\tfrac{\partial^2}{\partial x^2}\,\eta_{u,u}(x,t)),$$
$$-(\tfrac{\partial^2}{\partial x^2}\,\xi_x(x,t))+2\,(\tfrac{\partial}{\partial x}\,\eta_{u,u}(x,t))-(\tfrac{\partial}{\partial t}\,\xi_x(x,t)),$$
$$2\,(\tfrac{\partial}{\partial x}\,\xi_x(x,t))-(\tfrac{\partial}{\partial t}\,\xi_t(x,t))-(\tfrac{\partial^2}{\partial x^2}\,\xi_t(x,t))],\,[x,\,t],\,[\xi_x(x,t),\,\xi_t(x,t),\,\eta_{u,u}(x,t)],$$
$$[[x,\,t,\,u],\,[x,\,t,\,u],\,[],\,[\xi_x(x,t),\,\xi_t(x,t),\,u\,\eta_{u,u}(x,t)]]]$$

We get a JANET basis for these equations using the "Janet" package and ask for the number of solutions:

```
> Jle:=JanetBasis(op(le[1..3]));
```

$$Jle := [[-(\tfrac{\partial}{\partial t}\,\xi_x(x,t))+2\,(\tfrac{\partial}{\partial x}\,\eta_{u,u}(x,t)),\,\tfrac{\partial}{\partial x}\,\xi_t(x,t),\,2\,(\tfrac{\partial}{\partial x}\,\xi_x(x,t))-(\tfrac{\partial}{\partial t}\,\xi_t(x,t)),$$
$$-(\tfrac{\partial^2}{\partial t^2}\,\eta_{u,u}(x,t)),\,(\tfrac{\partial^2}{\partial t^2}\,\xi_t(x,t))+4\,(\tfrac{\partial}{\partial t}\,\eta_{u,u}(x,t)),\,\tfrac{\partial^2}{\partial t^2}\,\xi_x(x,t)],\,[x,\,t],$$
$$[\xi_x(x,t),\,\xi_t(x,t),\,\eta_{u,u}(x,t)]]$$

```
> HilbertSeries(t);
```

$$3+3\,t$$

Here we get the polynomial solutions up to degree 2 (for the LIE algebra):

```
> sol:=PolySol(Jle,2);
```

$$sol := [\xi_x(x,\,t) = C1_{0,0}+\frac{1}{2}\,C2_{0,1}\,x+C1_{0,1}\,t-2\,C3_{0,1}\,x\,t,$$
$$\xi_t(x,\,t) = C2_{0,0}+C2_{0,1}\,t-2\,C3_{0,1}\,t^2,$$
$$\eta_{u,u}(x,\,t) = C3_{0,0}+\frac{1}{2}\,C1_{0,1}\,x+C3_{0,1}\,t-\frac{1}{2}\,C3_{0,1}\,x^2]$$

```
> Cons := [C1[0,0],C1[0,1],C3[0,1], C2[0,0],C2[0,1],C3[0,0]];
```

$$Cons := [C1_{0,0},\,C1_{0,1},\,C3_{0,1},\,C2_{0,0},\,C2_{0,1},\,C3_{0,0}]$$

We use a "jets" command to write down the infinitesimal symmetries explicitly:

```
> lvec:=genvec(sol,Cons,le[4]);
```

$$lvec := [[[1,\,[x]]],\,[[t,\,[x]],\,[\frac{1}{2}\,u\,x,\,[u]]],\,[[-2\,x\,t,\,[x]],\,[-2\,t^2,\,[t]],\,[u\,t-\frac{1}{2}\,u\,x^2,\,[u]]],$$
$$[[1,\,[t]]],\,[[\frac{1}{2}\,x,\,[x]],\,[t,\,[t]]],\,[[u,\,[u]]]]$$

We could have used "jets" to analyse the isomorphism type of the LIE algebra, which we skip. What is important for us is the fact that the solutions of the heat equation form a module for this LIE algebra in the following sense: if $f(x,t)$ is a solution of the heat equation, then taking the commutator of a vector field in

the LIE algebra with the vector field $[f(x,t),[u]]$ yields a vector field of the form $[g(x,t),[u]]$, where $g(x,t)$ also is a solution of the heat equation. (This observation carries over to all linear PDE systems.) We start with $f(x,t)=1$ and take the commutators with the basis of the LIE algebra:

```
> l:=[[1, [u]]];
```

$$l := [[1,\ [u]]]$$

```
> map(s->ldvec(s,l,ivar,dvar),lvec);
```

$$[0,\ [[-\frac{1}{2}\,x,\ [u]]],\ [[-t+\frac{1}{2}\,x^2,\ [u]]],\ 0,\ 0,\ [[-1,\ [u]]]]$$

So the second and the third basis vector of the LIE algebra seem to do something useful:

```
> for i from 1 to 4 do
> ldvec(l,lvec[2],ivar,dvar);l:=ldvec(l,lvec[3],ivar,dvar) od;
```

$$[[\frac{1}{2}\,x,\ [u]]]$$

$$l := [[t-\frac{1}{2}\,x^2,\ [u]]]$$

$$[[\frac{3}{2}\,x\,t-\frac{1}{4}\,x^3,\ [u]]]$$

$$l := [[3\,t^2-3\,x^2\,t+\frac{1}{4}\,x^4,\ [u]]]$$

$$[[\frac{15}{2}\,x\,t^2-\frac{5}{2}\,x^3\,t+\frac{1}{8}\,x^5,\ [u]]]$$

$$l := [[15\,t^3-\frac{45}{2}\,x^2\,t^2+\frac{15}{4}\,x^4\,t-\frac{1}{8}\,x^6,\ [u]]]$$

$$[[\frac{105}{2}\,x\,t^3-\frac{105}{4}\,x^3\,t^2+\frac{21}{8}\,x^5\,t-\frac{1}{16}\,x^7,\ [u]]]$$

$$l := [[105\,t^4-210\,x^2\,t^3+\frac{105}{2}\,x^4\,t^2-\frac{7}{2}\,x^6\,t+\frac{1}{16}\,x^8,\ [u]]]$$

It seems that we get always two linearly independent polynomial solutions of lower degree i for $i>0$. This can be checked by applying the JANET algorithm to the heat equation itself (or by hand, of course):

```
> J:=JanetBasis(Ind2Diff(eq,ivar,dvar),ivar,dvar):
> HilbertSeries(t);
```

$$1+2\,t+2\,\frac{t^2}{1-t}$$

We leave it to the reader to prove that the above commutation routine generates all polynomial solutions out of the constant solution.

The next example demonstrates differential elimination.

Example 6. Specification of independent and dependent variables:

```
> ivar:=[x,y,z];dvar:=[u,v];
```

$$ivar := [x,\, y,\, z]$$

$$dvar := [u,\, v]$$

The PDE system:

```
> L:=Ind2Diff([u[z,z]-y*v[x,x], u[y,y]+z*v[y,z]],ivar,dvar);
```

$$L := [(\tfrac{\partial^2}{\partial z^2}\,\mathrm{u}(x,\, y,\, z)) - y\,(\tfrac{\partial^2}{\partial x^2}\,\mathrm{v}(x,\, y,\, z)),\ (\tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,\, y,\, z)) + z\,(\tfrac{\partial^2}{\partial z\,\partial y}\,\mathrm{v}(x,\, y,\, z))]$$

We want to find an equation for v alone. Computation of the JANET basis with the degree reverse lexicographic ordering with position over term:

```
> J:=JanetBasis(L,ivar,dvar,2);
```

$$J := [[(\tfrac{\partial^2}{\partial z^2}\,\mathrm{u}(x,y,z)) - y\,(\tfrac{\partial^2}{\partial x^2}\,\mathrm{v}(x,y,z)),\ (\tfrac{\partial^2}{\partial y^2}\,\mathrm{u}(x,y,z)) + z\,(\tfrac{\partial^2}{\partial z\,\partial y}\,\mathrm{v}(x,y,z)),$$
$$(\tfrac{\partial^3}{\partial z^2\,\partial y}\,\mathrm{u}(x,y,z)) - (\tfrac{\partial^2}{\partial x^2}\,\mathrm{v}(x,y,z)) - y\,(\tfrac{\partial^3}{\partial y\,\partial x^2}\,\mathrm{v}(x,y,z)), -2\,(\tfrac{\partial^3}{\partial z^2\,\partial y}\,\mathrm{v}(x,y,z))$$
$$- 2(\tfrac{\partial^3}{\partial y\,\partial x^2}\,\mathrm{v}(x,y,z)) - y\,(\tfrac{\partial^4}{\partial y^2\,\partial x^2}\,\mathrm{v}(x,y,z)) - z\,(\tfrac{\partial^4}{\partial z^3\,\partial y}\,\mathrm{v}(x,y,z))], [x,\, y,\, z], [u,\, v]]$$

So the last equation in the JANET basis is an equation just for v. We want to check whether every solution for this equation can be complemented by a function u such that one gets a solution of the original equations. Therefore, the parametric derivatives (still in the same ordering) are computed:

```
> ParamDeriv(ivar,dvar);
```

$$[\frac{z}{1-x} + \frac{1}{1-x} + \frac{y\,z}{1-x} + \frac{y}{1-x},$$
$$\frac{x^2\,y}{(1-x)\,(1-z)} + \frac{x^2}{(1-x)\,(1-z)} + \frac{x}{(1-y)\,(1-z)} + \frac{1}{(1-y)\,(1-z)}]$$

Now the JANET basis and the parametric derivatives for the equation for v are computed, also in the degree reverse lexicographical ordering.

```
> J:=JanetBasis([J[1][4]],ivar,[v]);
```

$$J := [[-2\,(\tfrac{\partial^3}{\partial z^2\,\partial y}\,\mathrm{v}(x,\, y,\, z)) - 2\,(\tfrac{\partial^3}{\partial y\,\partial x^2}\,\mathrm{v}(x,\, y,\, z)) - y\,(\tfrac{\partial^4}{\partial y^2\,\partial x^2}\,\mathrm{v}(x,\, y,\, z))$$
$$- z\,(\tfrac{\partial^4}{\partial z^3\,\partial y}\,\mathrm{v}(x,\, y,\, z))],\ [x,\, y,\, z],\ [v]]$$

```
> ParamDeriv(ivar,[v]);
```

$$\frac{x^2\,y}{(1-x)\,(1-z)} + \frac{x^2}{(1-x)\,(1-z)} + \frac{x}{(1-y)\,(1-z)} + \frac{1}{(1-y)\,(1-z)}$$

Comparison with the second component of the corresponding result for the original equations above shows that any holomorphic solution v of the last equation comes up in a solution of the original equations.

We leave it as an exercise to interchange the role of u and v, i. e. to eliminate u and to compare the corresponding generalised HILBERT series. To get an idea

on the coupling between u and v it is helpful to compare the HILBERT series for the original equations and for the separated system having the two equations, one for u and one for v. (Of course, one has to take the same ordering for both cases.)

There is also a different kind of differential elimination, which tries to find a PDE system involving differentiation with respect to fewer variable. For this one has to use the rather expensive lexicographical ordering.

The final example demonstrates the command Autonom, which is tantamount to Torsion. In the module language it finds the torsion submodule, gives a presentation of it, and the HILBERT series with respect to the degrevlex ordering. In the PDE-language it finds the functions killed by some linear differential operator. Autonomous elements are relevant in control theory, cf. [18] and [19], or [22] for the constant coefficient case.

Example 7. The PDE system:

```
> ivar := [s,t]; dvar := [u,v,w];
```

$$ivar := [s,\ t]$$

$$dvar := [u,\ v,\ w]$$

```
> R := Ind2Diff([u[s]+v[t]-w, u+v[s,t]+w], ivar, dvar);
```

$$R := [(\tfrac{\partial}{\partial s}\,\mathrm{u}(s,\ t)) + (\tfrac{\partial}{\partial t}\,\mathrm{v}(s,\ t)) - \mathrm{w}(s,\ t),\ \mathrm{u}(s,\ t) + (\tfrac{\partial^2}{\partial t\,\partial s}\,\mathrm{v}(s,\ t)) + \mathrm{w}(s,\ t)]$$

```
> Torsion(R, ivar, dvar);
```

$$[[\_\mathrm{T1}(s,\ t) = \mathrm{u}(s,\ t) + (\tfrac{\partial}{\partial t}\,\mathrm{v}(s,\ t)),\ \_\mathrm{T2}(s,\ t) = -\mathrm{u}(s,\ t) - (\tfrac{\partial}{\partial t}\,\mathrm{v}(s,\ t))],$$
$$[-\_\mathrm{T1}(s,\ t) - \_\mathrm{T2}(s,\ t),\ \_\mathrm{T1}(s,\ t) + (\tfrac{\partial}{\partial s}\,\_\mathrm{T1}(s,\ t))],\ 1 + \frac{s}{1-s}]$$

5 Timings and Implementation

For implementation issues we refer to the first part [4]. The essential differences of the differential version JanetBasis of the Involutive algorithm [5] to the polynomial one merely lie in the handling of differential expressions.

We compared "Janet" to other MAPLE packages. These packages were "diffalg" [6], [12] and "Rif" [20][2]. Note that "Janet" consists of MAPLE code only, i. e. no precompiled code (C++ etc.) was run in the timing for column "Janet".

The examples were taken from the data bases of Gerdt, Blinkov, Yanovich (`http://invo.jinr.ru`) and Faugère (`http://www-calfor.lip6.fr/~jcf`). They were converted to systems of linear homogeneous PDEs with constant coefficients using the standard association [11] and then run in MAPLE 8.

The computations were done on a machine with 2 GB RAM and 2 processors with 1 GHz each. The calculation was stopped after 50000 seconds.

[2] The input to rifsimp was the same as to the other packages, i. e. differential expressions.

	Janet	Rif	diffalg
cyclic6	2426 s	> 50000 s	24249 s
des18_3	1849 s	526 s	373 s
eco7	653 s	104 s	78 s
eco8	7851 s	1718 s	883 s
extcyc4	96 s	75 s	17 s
extcyc5	18179 s	> 50000 s	49894 s
katsura6	1915 s	1523 s	929 s
katsura7	24395 s	> 50000 s	> 50000 s
noon5	579 s	458 s	155 s
noon6	27138 s	16406 s	6312 s
redcyc6	2434 s	1847 s	> 50000 s
reimer5	5069 s	> 50000 s	3469 s
wang16	81 s	20 s	19 s

6 Acknowledgements

The contribution of two authors (Yu.A.B. and V.P.G.) was partially supported by the grant 01-01-00708 from the Russian Foundation for Basic Research and grant 2339.2003.2 from the Russian Ministry of Industry, Science and Technologies.

References

1. Barakat, M.: Jets. A MAPLE-Package for Formal Differential Geometry. 1 – 12 in [9]
2. Barakat, M. and Hartjen, G.: Jets. A MAPLE-Package for Variational and Jet Calculus. Submitted
3. Berth, M. and Gerdt, V. P.: Computation of Involutive Bases with Mathematica. Proc. Third International Workshop on Mathematica System in Teaching and Research (Siedlce, Poland, September 5-7, 2001), Institute of Mathematics & Physics, University of Podlasie (2001) 29–34
4. Blinkov, Y. A., Cid, C. F., Gerdt, V. P., Plesken, W. and Robertz, D.: The MAPLE Package "Janet": I. Polynomial Systems. These Proceedings
5. Blinkov, Y. A., Gerdt, V. P. and Yanovich, D. A.: Construction of Janet bases II. Polynomial Bases. 249 – 263 in [9]
6. Boulier, F., Lazard, D., Ollivier, F. and Petitot, M.: Computing Representations for Radicals of Finitely Generated Differential Ideals. Proceedings of ISSAC'95, ACM Press (1995) 158 – 166
7. Carminati, J. and Vu, Khai T.: Symbolic computation and differential equations; Lie symmetries. *J. Symb. Comp.* **29** (2000) 95 – 116
8. Chen, Y. F. and Gao, X. S.: Involutive directions and new involutive divisions. *Comp. Math. Appl.* **41** (7-8) (2001) 945 – 956
9. Ganzha, V. G., Mayr, E. W. and Vorozhtsov, E. V. (eds.): Computer Algebra in Scientific Computing CASC 2001. Springer-Verlag, Berlin (2001)
10. Gerdt, V. P. and Blinkov, Y. A.: Involutive bases of polynomial ideals. *Mathem. and Computers in Simulation* **45** (1998) 519 – 541
11. Gerdt, V. P.: Completion of linear differential systems to involution. In: Computer Algebra in Scientific Computing / CASC'99, V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov (Eds.), Springer-Verlag, Berlin (1999) 115 – 137
12. Hubert, E.: Factorization free decomposition algorithms in differential algebra. *J. Symb. Comp. 29* (2000), No. 4–5, 641 – 662
13. Hausdorf, M. and Seiler, W.: Involutive bases in MuPAD I: Involutive divisions. *mathPAD* **11** (2002), 51 – 56
14. Hausdorf, M. and Seiler, W.: An Efficient Algebraic Algorithm for the Geometric Completion to Involution. *Applicable Algebra in Engineering, Communication and Computing* **13** (2002) 163 – 207

15. Janet, M.: Leçons sur les Systèmes des Équationes aux Dérivées Partielles. Cahiers Scientifiques IV, Gauthier-Villars, Paris (1929)
16. Mityunin, V.: Implementation of the differential involutive algorithms in the computer algebra system Maple V5. Proc. IMACS Conference on Applications of Computer Algebra ACA 2000, St. Petersburg (2000) 38–39
17. Pommaret, J.-F.: Partial Differential Equations and Group Theory. Kluwer Academic Publishers (1994)
18. Pommaret, J.-F.: Partial Differential Control Theory. Vol. I: Mathematical Tools, Vol II: Control Systems. Kluwer Academic Publishers (2001)
19. Quadrat, A.: Analyse algébrique des systèmes de contrôle linéaires multidimensionnels. Thèse de Doctorat, Ecole Nationale des Ponts et Chaussées, Paris (1999), cf. `http://www-sop.inria.fr/cafe/Alban.Quadrat`
20. Reid, G. J., Wittkopf, A. D. and Boulton, A.: Reduction of systems of nonlinear partial differential equations to simplified involutive forms. *Eur. J. Appl. Math.* **7** (1996) 635 – 666
21. Vu, Khai T. and Carminati, J.: DESOLV for Maple V Release 5. School of Computing and Mathematics, Deakin University, Geelong, Victoria, Australia, released June 2000
22. Zerz, E.: Topics in Multidimensional Linear System Theory. Lecture Notes in Control and Inform. Sciences 256, Springer (2000)