The MAPLE Package "Janet": I. Polynomial Systems

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"⋆ comes in two parts, the first dealing with polynomials and commutative algebra, the second with linear PDEs. Here the first part, called "Involutive", is described. Amongst others it contains a MAPLE and a C++ implementation of the involutive technique for polynomial modules as an alternative for conventional Gröbner basis techniques.

1 Introduction

This is the first of two papers introducing the MAPLE package "Janet". The package contains the first MAPLE implementation of the Involutive algorithm [4], [5] in the form presented in [6], [7] and covers both the polynomial and the linear differential cases. More precisely, our implementation fixes JANET separation of variables into multiplicative and non-multiplicative ones [8] as the input involutive division [4] for the algorithm that produces a 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. In this first paper we shall only describe a version of implementation of the algorithm for polynomial systems and various related algorithms dealing with commutative algebra. Theoretically speaking, the polynomial version is obtained from the more general differential version [6] if one restricts to linear systems of PDEs with constant coefficients, cf. [10] for more details. From the point of view of implementation, however, quite some effort has been taken to optimise the polynomial version, so that it is highly competitive with the fastest implementations of the classical BUCHBERGER algorithm for computing GRÖBNER bases, cf. [1] for time comparisons. To make the package usable for bigger systems, C++ routines for the most important functions have been built in. Without the C++ code the package has about 4500 lines of MAPLE code.

Below is a list of the commands available in the polynomial part of "Janet":

With the commands in the first group one can create a JANET basis, produce normal forms of elements of the residue class module, print out the JANET basis with some relevant extra information and get quantitative information about the module as well as an explicit basis, even in the infinite dimensional case. The usage with definitions and some typical examples and comments on the algorithms will be given in section 2, more details on steering the main function are given in the subsequent section 3. The section 4 comments on relations between generators (corresponding to compatibility conditions in the PDE case). Again we proceed in the scheme definitions, algorithms, examples. The following section 5 describes various elimination techniques by giving examples, i. e. computing intersections of modules, finding annihilators, finding ring relations. 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. Finally, the last section 6 provides the results of several comparisons

⋆ 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
⋆ available at http://wwwb.math.rwth-aachen.de/Janet

Basic commands:	
InvolutiveBasis	PolInvReduce
PolTabVar	PolHilbertSeries
FactorModuleBasis	
Commands for special applications:	
PolMinPoly	Syzygies
PolResolution	PolRepres
Relations	PolWeightedHilbertSeries
NotHas/Has	
Commands for various invariants derivable from PolHilbertSeries:	
PolIndexRegularity	PolDimension
PolHilbertPolynomial	PolHilbertFunction
PolHP	PolHF
PolCartanCharacter	
Auxiliary Commands:	
LeadingMonomial	Stats
AddRhs	

of "Involutive" to other MAPLE packages and some notes on the implementation.

2 Basics

Let $R := K[x_1, \ldots, x_n]$ be the polynomial ring over the field K in the indeterminates $x_1, \ldots, x_n$. In general K can be any field, 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 $A_1, \ldots, A_a \in R^q$, two problems are treated:

1. Find a K-basis for the R-module $M := R^q/S$, where S is the submodule of R^q generated by $A_1, \ldots, A_a$.
2. Given $t \in M$, express t in terms of this basis.

Note the special case $q = 1$, where S becomes an ideal and M a residue class ring of R. This is also the case most relevant to polynomial equations. However, the general case is also often relevant in physical situations, e. g. for linear systems of PDEs with constant coefficients, where either the x_i become partial derivatives or alternatively are the variables after applying LAPLACE transform.

The above task is dealt with by the involutive algorithmic technique, which is first applied to create a JANET basis B for M that allows to read off a K-basis for M, and then apply the involutive reduction with respect to the JANET basis to have an algorithm for task 2. The JANET basis B is a finite subset of S which is more than a GRÖBNER basis: it does not only have the property that one can constructively represent any element of S as a linear combination of the elements of B, but the concept of multiplicative variables restricts the coefficients of the linear combinations in such a way that not only the coefficients become unique but also the algorithmic procedure to obtain them leaves no space for a long way round but forces every move by strict rules. Like in the ordinary GRÖBNER basis case the result depends to some extent on an ordering of the monomial K-basis of R^q. Various possibilities 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 InvolutiveBasis.
The subsequent call PolTabVar 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

$$s = \sum_{i=1}^{d} p_i B_i \text{ with } p_i \in K[x_i | x_i \in M_i].$$

The further subsequent call FactorModuleBasis produces a subset of all monomial K-basis vectors $(0, \ldots, 0, x_1^{\alpha_1} \cdots x_n^{\alpha_n}, 0, \ldots, 0)$ (with degree $\alpha_1 + \cdots + \alpha_n$) of R^q whose residue classes modulo S form a K-basis of $M = R^q/S$.
The command PolHilbertSeries gives the generating function for the numbers of these basis vectors according to their degrees.
Finally, with further **input** $v \in R^q$ the command PolInvReduce produces the normalised representative of the coset $v + S \in M$ i. e. the unique K-linear combination $\overline{v}$ of the basis vectors above with $\overline{v} + S = v + S$.

Example 1. ($q = 1$, infinite dimensional residue class ring)
Specification of variables:

```
> var:=[x,y,z];
```

$$var := [x, y, z]$$

Ideal generators:

```
> L:=[x^2+y^2-1,x^2+z^2-2];
```

$$L := [x^2 + y^2 - 1, x^2 + z^2 - 2]$$

Computation of the JANET basis:

```
> J:=InvolutiveBasis(L,var);
```

$$J := [-z^2 + 1 + y^2, x^2 + z^2 - 2, -z^2\,x + x + y^2\,x]$$

JANET basis with multiplicative variables and leading terms:

```
> PolTabVar();
```

$$[-z^2 + 1 + y^2, [*, 2, 3], y^2]$$
$$[x^2 + z^2 - 2, [1, 2, 3], x^2]$$
$$[-z^2\,x + x + y^2\,x, [*, 2, 3], y^2\,x]$$

The expansion via the geometric series of the following result yields the sum of the monomials forming a K-basis for the vector space complement of the ideal in the polynomial ring $K[x, y, z]$. Essentially, this is given without further computation.

```
> F:=FactorModuleBasis(var);
```

$$F := \frac{x}{1-z} + \frac{y\,x}{1-z} + \frac{1}{1-z} + \frac{y}{1-z}$$

The count for the basis vectors according to their degrees** (note, since $(1 - t)$ is the highest power of $(1 - t)$ occurring in the denominators, the (maximal) dimension of the variety is 1):

```
> PolHilbertSeries(t);
```

$$1 + 3\,t + 4\,t^2 + 4\,\frac{t^3}{1-t}$$

Computing the normalised representative of the coset of x^3:

```
> PolInvReduce(x^3,J,var);
```

$$-z^2\,x + 2\,x$$

Example 2. ($q = 2$, finite dimensional residue class module)

```
> var:=[x,y];
```

$$var := [x, y]$$

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

** One gets the same result (in a different expansion) by substituting t for x, y, z in F.

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

```
> J:=InvolutiveBasis(L,var);
```

$$J := [[y, x], [x, -y], [0, x^2 + y^2], [0, y^2\, x], [0, y^4]]$$

```
> PolTabVar();
```

$$[[y, x], [*, 2], [y, 1]]$$
$$[[x, -y], [1, 2], [x, 1]]$$
$$[[0, x^2 + y^2], [1, 2], [x^2, 2]]$$
$$[[0, y^2\, x], [*, 2], [y^2\, x, 2]]$$
$$[[0, y^4], [*, 2], [y^4, 2]]$$

```
> FactorModuleBasis(var);
```

$$[[0, 1], [0, y], [0, x], [0, y^2], [0, y\, x], [0, y^3], [1, 0]]$$

```
> PolHilbertSeries(t);
```

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

```
> PolInvReduce([x^2,y^2],J,var);
```

$$[0, y\, x + y^2]$$

3 Modifications and Extensions

There are quite a few possibilities to modify the above computations: In InvolutiveBasis various orderings for the monomial basis can be chosen. The standard choice is the degree reverse lexicographical ordering for the monomials itself. A second possibility is the pure lexicographical ordering, which might be slow for big examples, but is often used for elimination purposes. In both cases the ordering of the variables is implicitly declared by the declaration of the variables. In the proper module case, i. e. $q > 1$, one can in addition choose the ordering of the tuples. The default strategy is "term over position", i. e. the leading term of a tuple is the highest among the leading terms of the components. In case it is not unique the first one is chosen. The strategy "position over term", where the leading term is the leading term of the first non-zero component, is also possible. In any case one can modify the natural succession of the components by a permutation. The default strategy "term over position" is usually much more effective, but the other strategy is sometimes useful, when one computes resolutions. Note, all of these modifications might lead to different JANET bases. Other changes implied are the multiplicative variables and the grading of the residue class module. This latter point shows up in the commands FactorModuleBasis and PolHilbertSeries.

Example 3. (Influence of the orderings)

First we work with the degree reverse lexicographical ordering (note $x > y$):

```
> var:=[x,y];
```

$$var := [x, y]$$

```
> L:=[x-y^3];
```

$$L := [x - y^3]$$

```
> J:=InvolutiveBasis(L,var);
```

$$J := [-x + y^3]$$

```
> FactorModuleBasis(var);
```

$$\frac{1}{1-x} + \frac{y}{1-x} + \frac{y^2}{1-x}$$

```
> PolHilbertSeries(t);
```

$$1 + 2\, t + 3\, t^2 + 3\, \frac{t^3}{1-t}$$

Now we work with the pure lexicographical ordering (note still $x > y$):

```
> J1:=InvolutiveBasis(L,var,1);
```

$$J1 := [x - y^3]$$

```
> FactorModuleBasis(var);
```

$$\frac{1}{1-y}$$

```
> PolHilbertSeries(t);
```

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

The grading is determined by the filtering according to the degrees of the monomials in the basis computed by FactorModuleBasis, which again results from the information contained in PolTabVar. There is yet another possibility to modify this grading, i. e. the choice of the leading terms. One can assign a natural number d_i as degree to the variable x_i and in the proper module case $q > 1$ a non-negative integer c_i to the ith standard R-basis vector $e_i := (0, \ldots, 0, 1, 0, \ldots, 0)$. The degree of the term $x_1^{\alpha_1} \cdots x_n^{\alpha_n} e_i$ is then $c_i + \sum \alpha_j d_j$ instead of $\sum \alpha_j$ in the old regime. When choosing the leading term, this degree is taken into account first, and, in case there are two terms of the same new degree, the old regime is applied for the final decision. If in doubt, one can use the function LeadingMonomial. Note a subtle point at this stage: The functions FactorModuleBasis and PolHilbertSeries start from the information contained in PolTabVar which does no longer know anything about these modified degrees. If one wants the true HILBERT series of the graded ring with the grading according to the new degrees, one has to use the function PolWeightedHilbertSeries, which takes the degrees as part of its input.

The final point in this section concerns speed. For some, but not all of the commands above, there are C++-versions available, which of course can be called from within MAPLE.

Command:	Fast version:
InvolutiveBasis	InvolutiveBasisFast
PolInvReduce	PolInvReduceFast

For these commands up to now only the defaults work. Moreover, the C++-versions assume that the field of constants is the field of rational numbers, whereas the MAPLE-version can deal with extensions of the rationals, e. g. with purely transcendental extensions, which is often useful, cf. Example 8. For big examples, the C++-versions are faster by a factor 1000. If one wants to see the HILBERT series or continue to work with some other of the MAPLE programs, the command AssertInvBasis produces the data contained in PolTabVar, and one can proceed.

4 Syzygies

Having discussed the cokernel $M := R^q/S$ of an R-module homomorphism $R^a \to R^q$ in the last two sections, the main issue here will be the kernel of such a map here, i. e. the relations between the generators of S. The idea for their computation is to introduce right hand sides. We redo Example 2 with right hand sides.

Example 4. (Syzygies in Example 2)

```
> var:=[x,y];
```

$$var := [x, y]$$

Introduce names for the generators of the module and take them as right hand sides:

```
> L:=[[x,-y]=a,[y,x]=b,[y^3,0]=c];
```

$$L := [[x, -y] = a, [y, x] = b, [y^3, 0] = c]$$

Now the Janet basis also has a right hand side, expressing the basis vectors as a combination of the original generators:

```
> J:=InvolutiveBasis(L,var);
```

$$J := [[y, x] = b, [x, -y] = a, [0, x^2 + y^2] = b\,x - a\,y, [0, y^2\,x] = -c + b\,y^2,$$
$$[0, y^4] = c\,x - a\,y^3]$$

Now also a coset representative can be given a name and the standard representative will also be given together with its expression in terms of the original representative and the original generators of the submodule:

```
> PolInvReduce([x^2,y^2]=u,J,var);
```

$$[0, y\,x + y^2] = -a\,x + u$$

Finally the relations between the original generators can be computed:

```
> Syzygies(L,var);
```

$$[c\,x^2 - a\,y^3\,x + c\,y^2 - b\,y^4]$$

If one iterates the computation of the syzygies, i. e. computes the second, third, and higher syzygies, one gets a free resolution. This can be done in one go by the command PolResolution, as the next example shows. However the resolution starts with the Janet basis and not with the original relations.

Example 5. (Resolution in Example 2)

```
> with(Involutive):
> var:=[x,y]:
```

Enter generators without right hand side and compute the Janet basis (with ordering position over term):

```
> L:=[[x,-y],[y,x],[y^3,0]]:
> J:=InvolutiveBasis(L,var,2);
```

$$J := [[y, x], [x, -y], [0, x^2 + y^2], [0, x\,y^2], [0, y^4]]$$

The following command computes a free resolution of the residue class module M (even without invoking the last command first), where, however, one does not work with the original generators of S but with the Janet basis, cf. rows of the second matrix. The rows of the first matrix give the relations between the rows of the second matrix. In general the procedure stops after k steps, where k is the maximal number of non-multiplicative variables of the original Janet basis.

```
> PolResolution(L,var);
```

$$\left[\begin{bmatrix} 0 & 0 & 0 & -y^2 & x \\ 0 & 0 & -y^2 & x & 1 \\ x & -y & -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} y & x \\ x & -y \\ 0 & x^2 + y^2 \\ 0 & x\,y^2 \\ 0 & y^4 \end{bmatrix}\right]$$

5 Further Examples

In this section some examples demonstrating the use of the JANET algorithm will be presented. The first example concerns the **intersection** of submodules of R^q. We employ the well known technique by ZASSENHAUS, cf. [9]: If S_1, S_2 are submodules of R^q, then $S_1 \cap S_2$ is the kernel of the projection

$$R^{2q} \to R^q : (a, b) \mapsto a$$

restricted to $\langle (s_1, s_1), (s_2, 0) | s_1 \in S_1, s_2 \in S_2 \rangle$. The degrevlex ordering with position over term preference for the module case provides the JANET basis for the intersection automatically as the subsequent example demonstrates in the case $q = 1$.

Example 6. The intersection of the ideals generated by `L1` and `L2` is computed as follows:

```
> var:=[x,y]:
> L1:=[x^6-y^6]:
> L2:=[x^9-y^9]:
> l12:=map(a->[a,a],L1):
> l21:=map(a->[a,0],L2):
> L:=[op(l12),op(l21)];
```

$$L := [[x^6 - y^6,\ x^6 - y^6],\ [x^9 - y^9,\ 0]]$$

```
> J:=InvolutiveBasis(L,var,2);
```

$$J := [[x^6 - y^6, x^6 - y^6], [-y^9 + y^6x^3, -x^9 + y^6x^3], [-y^9x + y^6x^4, -x^{10} + y^6x^4],$$
$$[-y^9x^2 + y^6x^5, -x^{11} + y^6x^5], [0, x^{12} - y^{12} + x^9y^3 - y^9x^3]]$$

```
> N:=map(a->if a[1]=0 then a[2] fi,J);
```

$$N := [x^{12} - y^{12} + x^9\,y^3 - y^9\,x^3]$$

`N` is already a JANET basis for the intersection. Here a check:

```
> JN:=InvolutiveBasis(N,var);
```

$$JN := [x^{12} - y^{12} + x^9\,y^3 - y^9\,x^3]$$

Up to this point everything works also for non-principal ideals. Since in the present case the two ideals to be intersected were principal, `JN[1]` is the least common multiple of `L1[1]` and `L2[1]`. Here is the greatest common divisor:

```
> simplify(L1[1]*L2[1]/JN[1]);
```

$$x^3 - y^3$$

The next example concerns the computation of the **annihilator** of the module $M = R^q/S$, i. e. the ideal of all $a \in R$ with $aM = 0$. Obviously this is the intersection of the q ideals $I_i \subseteq R$ such that

$$\underbrace{\{0\} \oplus \cdots \oplus \{0\}}_{i-1} \oplus I_i \oplus \underbrace{\{0\} \oplus \cdots \oplus \{0\}}_{q-i} \subseteq S$$

with $i = 1, \ldots, q$. Note, the command InvolutiveBasis with option "position over term" yields a JANET basis for I_q immediately. Since the procedure for taking intersections of ideals was already demonstrated above, the following example is for a module invariant under the cyclic permutation of the positions so that InvolutiveBasis yields the JANET basis for the intersection in one go.

Example 7. Computation of the annihilator of R^3/S where S is generated by `L` below.

```
> var:=[x,y,z]:
> L:=[[x^2,y,z],[y,z,x^2],[z,x^2,y],[x+y,x+y,x+y]]:
> J:=InvolutiveBasis(L,var,2);
```

$$J := [[z,\ x^2,\ y],\ [y,\ z,\ x^2],\ [x,\ x+y-z,\ x+y-x^2],$$
$$[0,\ -z\,x+y\,x+y^2-z\,y,\ -x^3+y\,x+y^2-x^2\,y],$$
$$[0,\ x^2-y^2+z\,y-y,\ -z+x^2-y^2+x^2\,y],\ [0,\ -z^3+z^2\,y-z\,y+z^2,$$
$$-x^4\,y+z\,y^2-x^2\,z^2-y^2+2\,x^2\,z-x^4+x^2\,y-y^3+x^2\,y^2-z^2+x^2\,y\,z],$$
$$[0,\ z\,y^2-z^3-y^2+z^2,\ -z\,y+x^2\,z+2\,z\,y^2-x^4\,y-x^2\,z^2-y^2+x^2\,y-y^3$$
$$+x^2\,y^2],\ [0,\ y^3-z^3,\ -x^4\,y+2\,z\,y^2-x^2\,z^2],\ [0,\ 0,\ x^5-z\,y\,x+z^2\,x-x^3\,z$$
$$+x^4\,y-z\,y^2+z^2\,y-x^2\,y\,z-x^3\,y+y^2\,x+y^3-x^2\,y^2],\ [0,\ 0,\ -2\,x^2\,y\,z-z\,y^3$$
$$+x^4\,y^2+y^3-x^2\,y^2+y^4-x^2\,y^3+z^2\,y^2-x^2\,z^2+x^4\,z-x^2\,y^2\,z+z^3+x^4\,y]]$$

```
> N:=map(a->if (a[1]=0 and a[2]=0) then a[3] fi,J);
```

$$N := [x^5-z\,y\,x+z^2\,x-x^3\,z+x^4\,y-z\,y^2+z^2\,y-x^2\,y\,z-x^3\,y+y^2\,x+y^3$$
$$-x^2\,y^2,\ -2\,x^2\,y\,z-z\,y^3+x^4\,y^2+y^3-x^2\,y^2+y^4-x^2\,y^3+z^2\,y^2-x^2\,z^2$$
$$+x^4\,z-x^2\,y^2\,z+z^3+x^4\,y]$$

Since `L` and hence also S is closed under cyclic permutation of the three components, `N` is a JANET basis for the annihilator of S. (The first and second components give the same ideal as the third, which was picked here.) Here is a confirmation for the first component:

```
> map(n->PolInvReduce([n,0,0],J,var),N);
```

$$[[0,\ 0,\ 0],\ [0,\ 0,\ 0]]$$

The next example demonstrates the use of the Involutive algorithm for rings rather than modules: By a trick it is able to produce ring relations.

Example 8. Obviously the three polynomials x^2+y^2, x^2y^2, x^3y-y^3x are algebraically dependent. To find a relation between them, three new variables a, b, c are introduced and the following computation is carried out over the field of rational functions in a, b, c over the rationals:

```
> var:=[x,y];
```

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

```
> L:=[x^2+y^2-a,x^2*y^2-b,x^3*y-y^3*x-c];
```

$$L := [x^2+y^2-a,\ x^2\,y^2-b,\ x^3\,y-y^3\,x-c]$$

```
> J:=InvolutiveBasis(L,var);
```

$$J := [-a\,c^3+b\,a^3\,c-4\,b^2\,a\,c]$$

This result means that `L` generates all of R. It can also be interpreted as a relation between the three original polynomials.

The conventional way to proceed in the last example would have been to have a, b, c as additional variables and to work with 5 variables over the rationals using the elimination order to get rid of x and y. This is more time consuming, but in principle it always works, whereas the above trick only gives a first relation.

Example 9. Conventional method to redo the previous example:

```
> var:=[x,y,a,b,c];
```

$$var := [x,\ y,\ a,\ b,\ c]$$

```
> L:=[x^2+y^2-a,x^2*y^2-b,x^3*y-y^3*x-c]:
> J:=InvolutiveBasis(L,var,1);
```

$$J := [-c^2 + b\,a^2 - 4\,b^2,\ -4\,b^2\,y - c^2\,y + a^2\,b\,y,\ y^2\,a^2\,b - c^2\,y^2 - 4\,y^2\,b^2,$$
$$y^3\,a^2\,b - c^2\,y^3 - 4\,y^3\,b^2,\ b + y^4 - a\,y^2,\ c\,x + 2\,b\,y + a\,y^3 - a^2\,y,$$
$$a\,c\,x + y^3\,a^2 - a^3\,y + 2\,b\,a\,y,\ a^2\,c\,x + y^3\,a^3 - a^4\,y + 8\,b^2\,y + 2\,c^2\,y,$$
$$y^3\,c\,a - a^2\,c\,y + a^2\,b\,x - 4\,b^2\,x + 2\,c\,y\,b,\ c\,y\,x + 2\,b\,y^2 - a\,b,\ a\,c\,x\,y + 2\,b\,a\,y^2$$
$$-4\,b^2 - c^2,\ a^2\,y\,x - 4\,b\,y\,x + 2\,c\,y^2 - a\,c,\ c\,x\,y^2 + 2\,b\,y^3 - b\,a\,y,\ 2\,b\,y^2\,x$$
$$-y^3\,c - a\,b\,x + a\,c\,y,\ -2\,b\,x + a\,y^2\,x + c\,y,\ 2\,y^3\,x + c - a\,y\,x,\ x^2 + y^2 - a]$$

```
> NotHas(J,var,[x,y]);
```

$$[-c^2 + b\,a^2 - 4\,b^2]$$

Note, this result differs by a factor ac from the result obtained in the previous example.

Further examples of the package JANET may be found in [2].

6 Timings and Implementation

The main function InvolutiveBasis is an implementation of the Involutive basis algorithm in [1]. However, the "Janet trees" [1] were not built in for lack of availability of such data structures in MAPLE. The MAPLE implementation manages with two lists, one of which contains an intermediate involutive basis and the other one keeping the so called "prolongations" which are still to be reduced modulo the involutive basis. An important technical issue concerning the efficiency in MAPLE 8 was to empty MAPLE's "remember tables" in order to prevent memory overflow.

We include here a table summerizing comparisons of "Involutive" to other MAPLE packages. These packages were "Groebner" and "Rif" [12]***. Note that "Involutive" consists of MAPLE code only, i. e. no precompiled code (C++ etc.) was run in the timing for column "Involutive". We have also added a column "C++" which contains the corresponding time to run the respective example by InvolutiveBasisFast followed by AssertInvBasis, i. e. by calling the C++ implementation and loading the result into MAPLE.

The examples were taken from the data bases of Gerdt, Blinkov, Yanovich (`http://invo.jinr.ru`) and Faugère (`http://www-calfor.lip6.fr/~jcf`) and were 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.

	Groebner	Involutive	C++	Rif
cyclic6	657 s	425 s	1 s	> 50000 s
des18_3	26 s	242 s	1 s	45 s
eco7	9 s	76 s	1 s	53 s
eco8	97 s	1164 s	1 s	> 50000 s
extcyc4	3 s	23 s	1 s	210 s
extcyc5	914 s	3991 s	2 s	> 50000 s
katsura6	50 s	160 s	1 s	> 50000 s
katsura7	514 s	5687 s	2 s	> 50000 s
noon5	21 s	118 s	2 s	139 s
noon6	580 s	2250 s	8 s	24789 s
redcyc6	> 50000 s	347 s	1 s	16986 s
reimer5	53 s	554 s	2 s	> 50000 s
wang16	2 s	21 s	1 s	10 s

*** The input to rifsimp was the same as to the other packages, i. e. no differential expressions, but polynomials.

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. Blinkov, Y. A., Gerdt, V. P. and Yanovich, D. A.: Construction of Janet bases II. Polynomial Bases. 249 – 263 in [3]
2. Cid, C. F. and Plesken, W.: Invariants of finite groups and involutive division. 122 – 135 in [3]
3. Ganzha, V. G., Mayr, E. W. and Vorozhtsov, E. V. (eds.): Computer Algebra in Scientific Computing CASC 2001. Springer-Verlag, Berlin (2001)
4. Gerdt, V. P. and Blinkov, Y. A.: Involutive bases of polynomial ideals. *Mathem. and Computers in Simulation* **45** (1998) 519 – 541
5. Gerdt, V. P. and Blinkov, Y. A.: Minimal involutive bases. *Mathem. and Computers in Simulation* **45** (1998) 543 – 560
6. 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
7. Gerdt, V. P.: Involutive Division Technique: Some Generalizations and Optimizations. *Journal of Mathematical Sciences* **108**(6), (2002) 1034 – 1051
8. Janet, M.: Leçons sur les Systèmes des Équationes aux Dérivées Partielles. Cahiers Scientifiques IV, Gauthier–Villars, Paris 1929
9. Laue, R., Neubüser, J. and Schoenwaelder, U.: Algorithms for finite soluble groups and the SOGOS system. In: *Computational Group Theory: Proc. London Math. Soc. Symposium, Durham 1982* M. Atkinson (Ed.), Academic Press, Florida (1984) 195 – 247
10. Plesken, W.: Janets Algorithm. In *Symmetry and Perturbation Theory SPT2002*. S. Abenda, G. Gaeta, S. Walcher (Eds.), to appear World Scientific (2003)
11. Pommaret, J.-F.: Partial Differential Equations and Group Theory. Kluwer Academic Publishers (1994)
12. 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