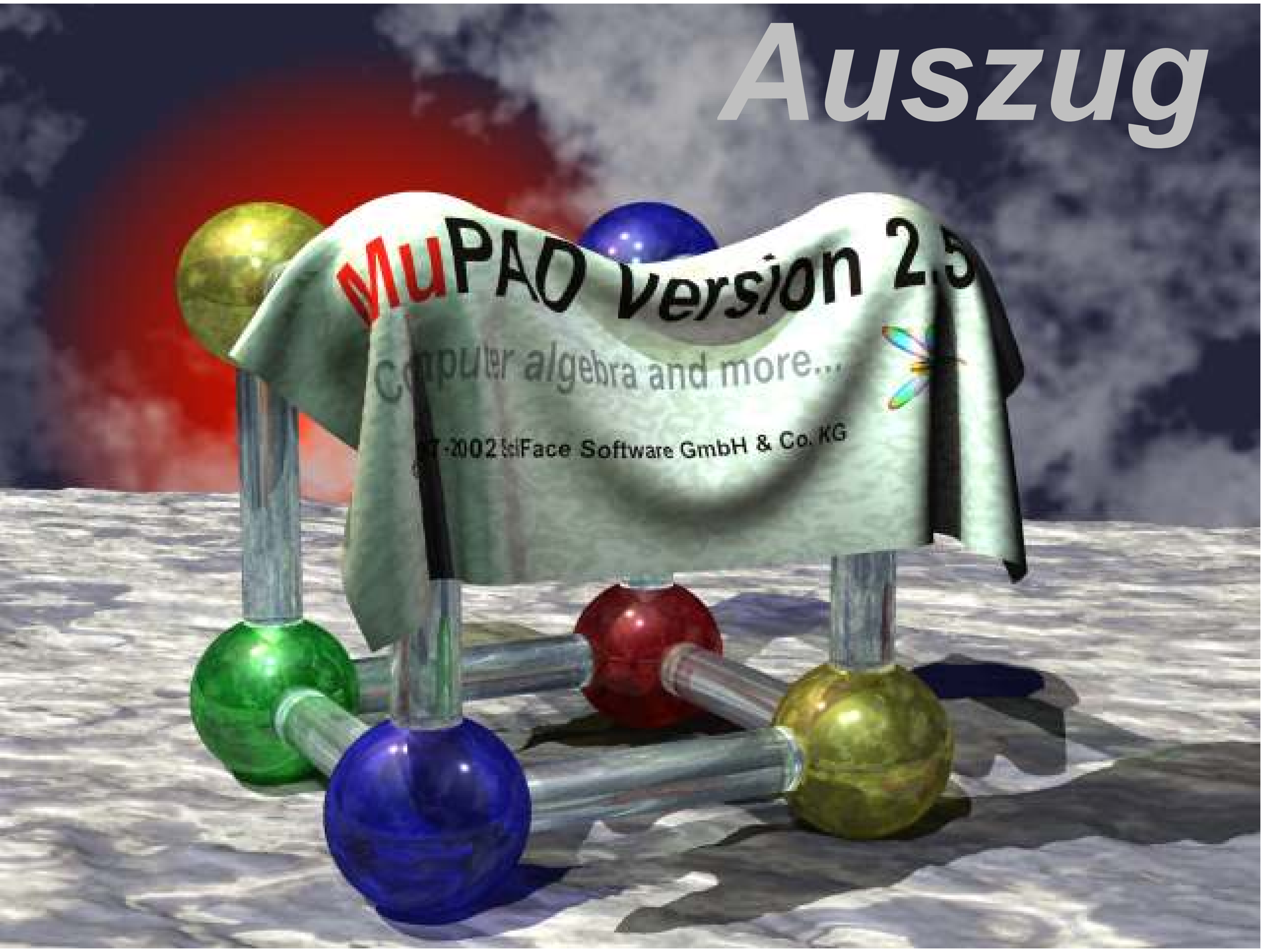
mathPAD

Das MuPAD-Magazin der MuPAD Research Group

Band 11, Ausgabe 1
August 2002



In dieser Ausgabe:

Die neue DUBBEL CD - powered by MuPAD
Individuelle Leistungskontrolle bei mathem. Massenvorlesungen
Einführung in die Finanzmathematik mit MuPAD
➤ Involutive Bases in MuPAD: Involutive Divisions
Numerics with Hardware Floats in MuPAD 2.5
The new Library for Statistics in MuPAD 2.5
Sparse Matrices in MuPAD 2.5

Involutive Bases in *MuPAD* – Part I: Involutive Divisions

Marcus Hausdorf and Werner M. Seiler
Lehrstuhl für Mathematik I, Universität Mannheim,
68131 Mannheim, Germany,
mailto:hausdorf@math.uni-mannheim.de
mailto:werner.seiler@math.uni-mannheim.de
http://www.math.uni-mannheim.de/~wms

This is the first of two articles on an implementation of involutive bases techniques in MuPAD. Whereas the computation of involutive bases in polynomial algebras of solvable type is treated in the second part, we introduce here the concept of an involutive division on multi indices, show how to complete a given set to involution and present MuPAD domains for the two most important involutive divisions, the Janet and the Pommaret division.

Introduction

Involutive bases are a special kind of non-reduced Gröbner bases. They have been introduced for polynomial ideals by Gerdt and collaborators (see e.g. [3, 4]) based on ideas from the Janet-Riquier theory of differential equations. Involutive bases are distinguished by special combinatorial properties and allow for a structural analysis of the ideals they span; in particular, they define Stanley decompositions. For more details, the reader is referred to [1, 9] and references therein.

The Janet-Riquier theory also motivated an explicit algorithm for the determination of involutive bases. It was first implemented in REDUCE and `C` by Blinkov and Gerdt; there followed packages for MAPLE [8] and MATHEMATICA [2]. A highly competitive implementation of Janet bases in `C++` using intricate data structures by Gerdt and Yanovich [5, 6] could in many instances produce a Gröbner basis faster than traditional algorithms.

While not making full use of the above mentioned optimisation, the here presented implementation is the most general so far. Utilising *MuPAD*'s categories and domains capabilities, we can provide a completion algorithm for so-called polynomial algebras of solvable type. This comprises, for example, rings of linear differential or difference operators, the Weyl algebra or universal enveloping algebras of Lie algebras. We are going to look at the definition of a algebra of solvable type (the "polynomial" case) and the completion of ideal bases to involutive bases in the second part of this article; first, we want to explain the combinatorial ideas lying underneath by studying involutive divisions on multi indices (the "monomial" case). The transfer will then be rather straightforward.

The Idea behind Involutive Divisions

Involutive division were originally introduced for monomials, i. e. elements of the ring $\mathbb{K}[x_1, \dots, x_n]$. But it suffices, at first, to work only with the exponent vectors. Therefore, we consider multi indices of a fixed length n, being elements of the Abelian monoid $(\mathbb{N}_0^n, +)$. Unfortunately, this leads to a slight confusion in terminology: instead of *multiplying* and *dividing* terms, we are now actually *adding* and *subtracting* multi indices.

For two multi indices $\mu = (\mu^1, \dots, \mu^n)$ and $\nu = (\nu^1, \dots, \nu^n)$, we say that $\mu|\nu$ (μ *divides* ν), if $\mu^i \leq \nu^i$ for all i. The set $C(\mu) = \mu + \mathbb{N}_0^n$ of all *multiples* of a given multi index is called its *cone*. For a finite set $\mathcal{N} = \{\mu_1, \dots, \mu_r\}$ its *span* is the union $\langle \mathcal{N} \rangle = \bigcup_{\mu_i \in \mathcal{N}} C(\mu_i)$ of all the cones of its elements. Of course, the cones of $\mathcal{N}$ will always overlap (Fig. 11 on the left). If we desire a disjoint union of the span of $\mathcal{N}$, we have to restrict the directions in which the cones stretch; this is exactly the task of an involutive division. In the following definition, let $C_N(\mu)$, the *restricted cone* of μ with respect to $N \subseteq \{1, \dots, n\}$, denote the set $\mu + \{\nu \in \mathbb{N}_0^n : \nu^i = 0 \text{ for } i \notin N\}$.

Definition 1. *An* involutive division L *on* $\mathbb{N}_0^n$ *is given by prescribing for each finite subset* $\mathcal{N} \subset \mathbb{N}_0^n$ *and for each multi index* $\mu \in \mathcal{N}$ *a set* $N_{L,\mathcal{N}}(\mu)$ *of* multiplicative indices *such that the following holds: (i) If* $C_{L,\mathcal{N}}(\mu)$ *is used as a shorthand for* $C_{N_{L,\mathcal{N}}(\mu)}(\mu)$*, then for all* $\mu, \nu \in \mathcal{N}$ *with* $C_{L,\mathcal{N}}(\mu) \cap C_{L,\mathcal{N}}(\nu) \neq \emptyset$ *either* $C_{L,\mathcal{N}}(\mu) \subseteq C_{L,\mathcal{N}}(\nu)$ *or* $C_{L,\mathcal{N}}(\nu) \subseteq C_{L,\mathcal{N}}(\mu)$*; (ii) if* $\mathcal{M} \subset \mathcal{N}$*, then* $\forall \mu \in \mathcal{M} : N_{L,\mathcal{N}}(\mu) \subseteq N_{L,\mathcal{M}}(\mu)$.

$C_{L,\mathcal{N}}(\mu)$ *is called the* involutive cone *of* μ *with respect to* L *and* $\mathcal{N}$*. We denote the complement of* $N_{L,\mathcal{N}}(\mu)$ *in* $\{1, \dots, n\}$*, the* non-multiplicative indices *of* μ*, by* $\bar{N}_{L,\mathcal{N}}(\mu)$*. Finally, for* $\mu \in \mathcal{N}$ *and* $\nu \in \mathbb{N}_0^n$ *we write* $\mu \,|_{L,\mathcal{N}}\, \nu$ *(*μ involutively divides *or is an* involutive divisor *of* ν*), if and only if* $\nu \in C_{L,\mathcal{N}}(\mu)$.

Let us rephrase the two conditions for an involutive division L in the definition above: The first one says that if the involutive cones of two multi indices intersect, one cone must lie completely in the other one. The second condition requires that if we remove a multi index from $\mathcal{N}$, there must be at least the same multiplicative indices with respect to L for the remaining elements.

The right half of Fig. 11 shows the situation for an involutive division where the multiplicative indices are $\{1\}$ for $[2, 0]$ and $\{1, 2\}$ for $[0, 2]$. Actually, these are exactly the multiplicative indices for both the Janet and the Pommaret division, which will be defined below. Obviously, the two involutive cones do not intersect, but instead now we are missing the half-line starting at $[1, 2]$. We will thus have to *complete* the set $\{[0, 2], [2, 0]\}$.

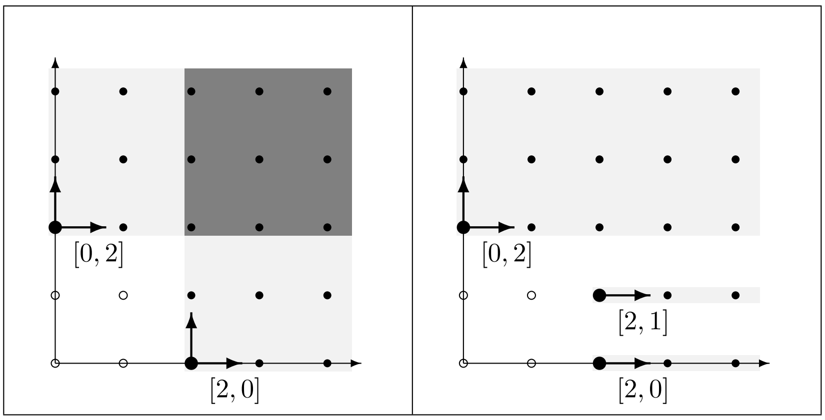


Figure 11: *Left:* intersecting cones. *Right:* involutive cones.

Definition 2. *Let* L *be an involutive division on* $\mathbb{N}_0^n$ *and* $\mathcal{N} \subset \mathbb{N}_0^n$ *a finite set.*

1. *The* involutive span *of $\mathcal{N}$ is the union of the involutive cones of its elements:*
$$\langle\,\mathcal{N}\,\rangle_L = \bigcup_{\mu\in\mathcal{N}} C_{L,\mathcal{N}}(\mu)\,. \tag{19}$$
2. *The set $\mathcal{N}$ is* (L-)involutively autoreduced, *if $C_{L,\mathcal{N}}(\mu) \cap C_{L,\mathcal{N}}(\nu) = \emptyset$ for all $\mu \neq \nu \in \mathcal{N}$.*
3. *$\mathcal{N}$ is called* weakly (L)-involutive, *if $\langle\,\mathcal{N}\,\rangle_L = \langle\,\mathcal{N}\,\rangle$ where $\langle\,\mathcal{N}\,\rangle$ denotes the ordinary span of $\mathcal{N}$, i. e. the monoid ideal $\langle\,\mathcal{N}\,\rangle = \mathcal{N} + \mathbb{N}_0^n$.*
4. *A finite subset $\hat{\mathcal{N}} \subset \langle\,\mathcal{N}\,\rangle$ is called a* weak involutive basis *of $\langle\,\mathcal{N}\,\rangle$, if $\langle\,\hat{\mathcal{N}}\,\rangle_L = \langle\,\mathcal{N}\,\rangle$. If $\hat{\mathcal{N}}$ contains $\mathcal{N}$, it is a* weak (L)-completion *of $\mathcal{N}$. If the set $\hat{\mathcal{N}}$ is furthermore autoreduced (i. e., no multi index is a multiple of another one in the set), it is a* (strong) involutive basis *of $\langle\,\mathcal{N}\,\rangle$ or a* (strong) completion *of $\mathcal{N}$, respectively.*
5. *An involutive division L is called* Noetherian, *if every finite set $\mathcal{N} \subset \mathbb{N}_0^n$ of multi indices has a completion.*

For an involutively autoreduced set the union in (19) is disjoint. It is a nice exercise to show that any weak involutive basis contains a strong involutive basis as a subset. Only the two properties in Def. 1 are needed for this. The three multi indices $\{[0,2],[1,2],[2,0]\}$ are a strong involutive basis and a completion of the initial set $\{[0,2],[2,0]\}$ with respect to both the Janet and the Pommaret division.

The actual algorithm for the completion of a set of multi indices to involution is very similar to the Buchberger algorithm for the construction of Gröbner bases. Buchberger's criterion for S-polynomials is replaced by the criterion of *local involution*: A set $\mathcal{N}$ of multi indices is called *locally involutive* with respect to the involutive division L, if for every $\mu \in \mathcal{N}$ and $i \in \bar{N}_{L,\mathcal{N}}(\mu)$, there exists some $\nu \in \mathcal{N}$ with $\mu + 1_i \in C_{L,\mathcal{N}}(\nu)$. That means, if we go one step from a multi index $\mu \in \mathcal{N}$ into a non-multiplicative direction, we always arrive in the multiplicative cone of another multi index. In [3], it is proved that local involution of a set implies its involution, provided the involutive division L satisfies some rather technical property called continuity. For almost all involutive divisions used in practice (especially the Janet and Pommaret divisions), this is the case.

Algorithm: *Involutive completion in $\mathbb{N}_0^n$*
Input: Finite subset $\mathcal{N} \subset \mathbb{N}_0^n$, involutive division L, term order $\preceq$ on $\mathbb{N}_0^n$
Output: Involutive basis of $\langle\,\mathcal{N}\,\rangle$

/1/ `repeat`
/2/ $\quad\mathcal{S} \leftarrow \{\mu + 1_i \;:\; \mu \in \mathcal{N},\, i \in \bar{N}_{L,\mathcal{N}}(\mu),\, \mu + 1_i \notin \langle\,\mathcal{N}\,\rangle_L\}$
/3/ $\quad\mathcal{N} \leftarrow \mathcal{N} \cup \{\min_{\preceq} \mathcal{S}\}$
/4/ `until` $\mathcal{S} = \emptyset$
/5/ `return` $\texttt{InvAutoReduce}_L(\mathcal{N})$

Figure 12: Completion in $\mathbb{N}_0^n$

The algorithm for the completion of a set of multi indices to completion is shown in Fig. 12. For it to work, in addition to continuity the involutive division L is required to be constructive (an even more complicated condition, which can be read about in [3] but nevertheless is valid for all divisions we consider here). Notice the parallels between this and the Buchberger algorithm: Instead of examining S-polynomials of critical pairs, we investigate all non-multiplicative multiples; instead of computing normal forms, we check whether the multiples can be obtained multiplicatively. The polynomial algorithm, which will be formulated in the second part, will work in the same way; then actually involutive normal forms will be computed. The choice of the next element to be treated in line $/3/$ with respect to a term order is necessary for the proof of termination. Of course, the algorithm does in general not terminate for a non-Noetherian involutive division.

The Category of Involutive Divisions

`Cat::InvolutiveDivision(n)` contains the basic operations with involutive divisions. The parameter `n` is the length of the multi indices the involutive divisions is defined on. The category is only a member of `Cat::BaseCategory`. The following undefined entries must be provided by the domains representing the different involutive divisions:

`sepListMult(mu_l)`: returns a list of lists of the form `[mu, mv_s]` giving for each multi index `mu` in `mu_l` its multiplicative indices `mv_s` with respect to the involutive division and the multi indices in `mu_l`.

`sepListMultAdd(sep_l,mu)`: given a separation list (as returned by `sepListMult`), the multi index `mu` is inserted into `sep_l`; the multiplicative indices of the multi indices in the new list are recomputed (with respect to all elements).

`sepListMultRem(sep_l,mu)`: same as `sepListMultAdd(sep_l,mu)`, only that `mu` is removed from `sep_l`.

`multDirs(mu_l,mu)`: returns the set of multiplicative directions for the multi index `mu` with respect to the multi indices in `mu_l`.

For the following entries, default implementations are present in the category:

`sepListNonMult`, `sepListNonMultAdd`, `sepListNonMultRem`, `nonMultDirs`: same as the respective methods above, only for non-multiplicative indices.

`isInvDivisor(mu,nu,mv_s)`: returns `nu−mu` if `nu` lies in the cone restricted by the directions in the set `mv_s` and 0 otherwise.

`autoreduce(mu_l)`: returns an autoreduced list of multi indices with the same span as the multi indices in the original list `mu_l`.

`invAutoreduce(mu_l)`: returns an involutively autoreduced list of multi indices with the same involutive span as the multi indices in the original list `mu_l`.

`complete(mu_l)`: returns an involutively autoreduced list of multi indices with the same span as the multi indices in the original list `mu_l`, i. e., a (strong) involutive basis.

The Janet Division

At long last, we give the definition of the first of the two important involutive divisions most frequently encountered. As before, let $\mathcal{N} = \{\mu_1, \dots, \mu_r\}$ denote a set of multi indices of length n; furthermore, we write $\mathcal{T}_{\mathcal{N},k}(\nu) = \{\mu \in \mathcal{N} : \mu^i = \nu^i, k \leq i \leq n\}$ for the subset of $\mathcal{N}$ consisting of those multi indices of $\mathcal{N}$ the last k indices of which coincide with those of ν. For the Janet division $\mathcal{J}$, we then have:

- $n \in N_{\mathcal{J},\mathcal{N}}(\nu)$, if $\nu^n = \max_{\mu\in\mathcal{N}}\{\mu^n\}$;
- $n > m \in N_{\mathcal{J},\mathcal{N}}(\nu)$, if $\nu^m = \max_{\mu\in\mathcal{T}_{\mathcal{N},m+1}(\nu)}\{\mu^m\}$

This seemingly convoluted definition unfolds into a rather straightforward algorithm for computing the multiplicative indices of $\mathcal{N}$ with respect to the Janet division.

Algorithm: *Multiplicative Indices for the Janet Division*
Input: list $\mathcal{N} = [\mu_1, \dots, \mu_r]$ of pairwise different multi indices of length n,
Output: list $\mathcal{M} = [N_{\mathcal{N},\mathcal{J}}(\mu_1), \dots, N_{\mathcal{N},\mathcal{J}}(\mu_r)]$ of multiplicative indices

$\mathcal{N} \leftarrow$ `sort`$(\mathcal{N}, \succeq_{\text{ilex}})$; $\nu \leftarrow \mathcal{N}[1]$; $p_1 \leftarrow n$; $N \leftarrow \{1, \dots, n\}$; $\mathcal{M}[1] \leftarrow N$;
`for` k `from` 2 `to` r `do`
  $p_2 \leftarrow \max\{i : (\nu - \mathcal{N}[k])^i \neq 0\}$; $N \leftarrow N \setminus \{p_2\}$;
  `if` $p_1 < p_2$ `then`
    $N \leftarrow N \cup \{p_1, \dots, p_2 - 1\}$;
  `end_if`;
  $\mathcal{M}[k] \leftarrow N$; $\nu \leftarrow \mathcal{N}[k]$; $p_1 \leftarrow p_2$;
`end_for`;
`return`$(\mathcal{M})$;

Figure 13: *Multiplicative Indices for the Janet Division*

We start by ordering the list $\mathcal{N}$ decreasingly inverse lexicographically. For the first multi index in $\mathcal{N}$, obviously all directions are multiplicative. We now proceed by comparing each element of $\mathcal{N}$ with its successor. The position at which they differ is p_2, while p_1 holds the appropriate value from the last iteration. Clearly, $\mathcal{N}[k-1]$ and $\mathcal{N}[k]$ both lie in $\mathcal{T}_{\mathcal{N},p_2-1}$, and because of the ordering and the definition of the Janet division, p_2 must be non-multiplicative for $\mathcal{N}[k]$. If $p_2 > p_1$, the determination whether the indices $p_1, \dots, p_2 + 1$ are multiplicative or not is not affected by $\mathcal{N}[k-1]$, so they must become multiplicative (if they have not been before).

The domain `Dom::JanetDivision(n)` takes as optional argument a multi index of length n representing a permutation exerted on all multi indices when computing the multiplicative indices. So if the permutation is given by $[\pi_1, \dots, \pi_n]$, a multi index μ is transformed into $[\mu^{\pi_1}, \dots, \mu^{\pi_n}]$. This is especially useful since different authors define the Janet division differently (some start at the front, some at the back). The same holds for the Pommaret division, and so this optional parameter is also valid there.

The algorithm of Fig. 13 has the advantage that if a multi index is inserted (resp. removed) from a list of multi indices for which the multiplicative indices are already known, a recomputation is only required from the position of the new (resp. the removed) multi index in the list. The starting values for N and p_1 are readily computed from the preceding two multi indices.

The Pommaret Division

As a second domain for an involutive division, `Dom::PommaretDivision(n)` is implemented. The definition of the Pommaret division is rather simple: For a multi index μ, the class $cls(\mu) = \min\{i : \mu^i \neq 0\}$ is its leftmost non-vanishing entry; we take as multiplicative indices for μ all those smaller than or equal to $cls(\mu)$. An important difference to the Janet division is that the multiplicative directions of a multi index are fixed *a priori* and thus independent of the set $\mathcal{N}$ currently considered. Such an involutive division is called *globally defined*. While this property allows for a much simpler computation of multiplicative indices, the Pommaret division is unfortunately not Noetherian. Completions exist only in special coordinate systems called δ-regular (see [7, 9] for more details and a constructive solution of this problem). The Pommaret division is of great theoretical importance for the structure analysis of polynomial modules and also used in the hybrid geometric-algebraic completion algorithm described in [7] and implemented as part of the *MuPAD* DETools-library.

Let us finish with a simple example of how all these things work. We create domains for the Janet and the Pommaret division and consider the situation of Fig. 11. The completion is the same for both divisions.

MuPAD

```
>> JD:= Dom::JanetDivision(2): PD:= Dom::PommaretDivision(2): mu_l:= [[0,2],[2,0]]:
>> PD::complete(mu_l);
```

Output

```
[[2, 0], [0, 2], [2, 1]]
```

If we translate the multi indices of the starting set one step to the right, the multiplicative indices differ. Now there does not exist a finite completion with respect to the Pommaret division anymore: infinitely many multi indices would have to be added ($[3,1], [1,3], [1,4], [1,5], \ldots$). The Janet division does not have any problems.

MuPAD

```
>> mu_l:= [[1,2],[3,0]]: PD::sepListMult(mu_l); JD::sepListMult(mu_l);
>> JD::complete(mu_l);
```

Output

```
[[[1, 2], {1}], [[3, 0], {1}]]
[[[1, 2], {1, 2}], [[3, 0], {1}]]
[[1, 2], [3, 0], [3, 1]]
```

References

[1] J. Calmet, M. Hausdorf, and W.M. Seiler. A constructive introduction to involution. In R. Akerkar, editor, *Proc. Int. Symp. Applications of Computer Algebra — ISACA 2000*, pages 33–50. Allied Publishers, New Delhi, 2001.

[2] V.P. Gerdt, M. Berth. Computation of Involutive Bases with Mathematica. *Proc. Workshop on Mathematica System in Teaching and Research*, pages 29–34. University of Podlasie, Sieldce, 2001.

[3] V.P. Gerdt and Yu.A. Blinkov. Involutive bases of polynomial ideals. *Math. Comp. Simul.*, 45:519–542, 1998.

[4] V.P. Gerdt and Yu.A. Blinkov. Minimal involutive bases. *Math. Comp. Simul.*, 45:543–560, 1998.

[5] V.P. Gerdt and D.A Yanovich. Construction of Janet Bases I. Monomial Bases. In V.G. Ganzha, E.W. Mayr, E.V.Voroszhtsov, editors, *Computer Algebra in Scientifing Computing – CASC 2001*, pages 233–247. Springer, Berlin, 2001.

[6] V.P. Gerdt and D.A Yanovich. Construction of Janet Bases II. Polynomial Bases. In V.G. Ganzha, E.W. Mayr, E.V.Voroszhtsov, editors, *Computer Algebra in Scientifing Computing – CASC 2001*, pages 249–263. Springer, Berlin, 2001.

[7] M. Hausdorf and W.M. Seiler. An Efficient Algebraic Algorithm for the Geometric Completion to Involution. *Appl. Alg. Eng. Comp. Comm.*, 2002, to appear.

[8] V.A. Mityunin. Implementation of Differential Involutive Algorithm in Maple V5. *Proc. IMACS Conference on Applications of Computer Algebra ACA 2000*, pages 38–39, St. Petersburg, 2000.

[9] W.M. Seiler. Involution — The Formal Theory of Differential Equations and Its Applications in Computer Algebra and Numerical Analysis. Habilitation thesis, Dept. of Mathematics, Universität Mannheim, 2001.