Involutive Bases in *MuPAD* II: Polynomial Algebras of Solvable Type

Marcus Hausdorf and Werner M. Seiler
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Universität Heidelberg,
69120 Heidelberg, Germany,
`{marcus.hausdorf,werner.seiler}@iwr.uni-heidelberg.de`
`http://www.iwr.uni-heidelberg.de/groups/compalg/seiler`

This article continues the description of an implementation of involutive basis techniques in MuPAD. We show how the methods for the "monomial case" developed in the first part can be lifted to a large class of polynomial (including non–commutative) algebras and modules over them. We present categories and domains for dealing with such algebras and exploiting the rich combinatorial structure of involutive bases.

Introduction

In the first part [11], the concept of an *involutive division* on multi indices of a fixed length was introduced. This was just a rule restricting the normal divisibility relation by distinguishing for each member of a given finite set of multi indices between allowed (*multiplicative*) and non–allowed (*non–multiplicative*) entries. This restriction led to the notion of *involutive span* of the set consisting of all the multiples of the multi indices with respect only to the multiplicative directions. We showed how to *complete* a set by adding further multi indices such that the normal and the involutive span coincide. As examples, we considered the two divisions most commonly used in practice, the *Janet* and the *Pommaret division*.

Since multi indices are equivalent to the terms of the polynomial ring $\mathbb{k}[x_1, \dots, x_n]$, we may speak of having treated the "monomial case" of involutive bases. But their real power does not unfold until we have lifted these methods to the "polynomial case", that is to ideals of $\mathbb{k}[x_1, \dots, x_n]$ (or more generally, to submodules of free polynomial modules). What we will get are special non–reduced Gröbner bases which are advantageous in certain situations for two reasons. Firstly, it has been shown that the involutive completion algorithm provides a competitive alternative to the Buchberger algorithm (timings can be found, for example, in [10]). Secondly, involutive bases carry a rich combinatorial structure allowing one to easily read off many invariants; especially with Pommaret bases, extensive structure analysis is possible [14]. A *MuPAD*-library for this task will be presented in a follow-up to the current article.

In the first two sections of this article, we provide the necessary theoretical background. Since in the polynomial case involutive bases depend solely on the leading terms with respect to a given term order, they can be defined for a wide class of algebras resembling polynomial rings but not necessarily commutative – the *polynomial algebras of solvable types*. They were originally introduced by Kandry-Rody and Weispfenning [12] and comprise, for example, linear differential and difference operators, the Weyl algebra or universal enveloping algebras of Lie algebras. Here and in the following, we cite only the basic definitions and omit any proofs of the facts we use. There exist several introductory articles on involutive bases containing the missing details [3, 7, 8, 13].

Involutive basis methods are available for a variety of computer algebra systems (like, for example, the MAPLE package "Janet" [1, 2] or the implementation for MATHEMATICA [6]). A very fast and efficient C++ program for computing Janet bases is presented in [9, 10]. It is obvious what an implementation of the involutive techniques making use of the domains concept in *MuPAD* must look like. Polynomial algebras of solvable type become a category, the heart of which is a generic implementation of the involutive completion algorithm. Together with the category and domains for involutive divisions from the first part, a domain for handling free modules and a domain supplying a user interface for computations with involutive bases, the implementation presented here is the most general so far.

Polynomial Algebras of Solvable Type

Let $\mathcal{P} = \Bbbk[x_1, \dots, x_n]$ be a polynomial ring over a field $\Bbbk$. We equip the $\Bbbk$-algebra $\mathcal{P}$ with alternative multiplications, in particular with non-commutative ones. We allow that both the variables x_i do not commute any more and that they operate on the coefficients. The usual multiplication is denoted either by a dot $\cdot$ or by no symbol at all. Alternative multiplications $\mathcal{P} \times \mathcal{P} \to \mathcal{P}$ are always written as $f \star g$.

Like Gröbner bases, involutive bases are defined with respect to a *term order*. It selects in each polynomial $f \in \mathcal{P}$ a *leading term* $\mathrm{lt}_\prec f = x^\mu$ with *leading exponent* $\mathrm{le}_\prec f = \mu$.[1] The coefficient $r \in \Bbbk$ of x^μ in f is the *leading coefficient* $\mathrm{lc}_\prec f$ and the product rx^μ is the *leading monomial* $\mathrm{lm}_\prec f$. Based on the leading exponents we associate to each finite set $\mathcal{F} \subset \mathcal{P}$ a set $\mathrm{le}_\prec \mathcal{F} \subset \mathbb{N}_0^n$ to which we may apply the theory developed in the first part. But this requires a kind of compatibility between the multiplication $\star$ and the chosen term order.

Definition 1 *$(\mathcal{P}, \star, \prec)$ is a* polynomial algebra of solvable type *for the term order $\prec$, if the multiplication $\star : \mathcal{P} \times \mathcal{P} \to \mathcal{P}$ satisfies three axioms.*

(i) *$(\mathcal{P}, \star)$ is a ring with unit* 1.

(ii) $\forall r \in \Bbbk,\ f \in \mathcal{P} :\ r \star f = rf.$

(iii) $\forall f, g \in \mathcal{P} :\ \mathrm{le}_\prec(f \star g) = \mathrm{le}_\prec f + \mathrm{le}_\prec g.$

Of course, the usual multiplication in a polynomial ring satisfies these conditions; for a non-trivial example, consider a function field $\Bbbk$ containing n variables, e. g. the field of rational functions $\Bbbk := \mathbb{C}(y_1, \dots, y_n)$ equipped with the corresponding n derivations with respect to the variables. We can then form the ring $\Bbbk[\partial_1, \dots, \partial_n]$ of *linear differential operators*. We have always $\partial_i \star \partial_j = \partial_j \star \partial_i$ and $y_i \star \partial_j = \partial_j \star y_i$ for $i \neq j$, whereas $\partial_i \star y_i = y_i \partial_i + 1$. Thus the product of two monomial operators is given by

$$a\partial^\mu \star b\partial^\nu = \sum_{\lambda+\kappa=\mu} \binom{\mu}{\lambda} a \frac{\partial^{|\kappa|} b}{\partial y^\kappa} \partial^{\lambda+\nu} \tag{1}$$

where $\binom{\mu}{\lambda}$ is a shorthand for $\prod_{i=1}^n \binom{\mu_i}{\lambda_i}$. For an arbitrary term order $\prec$, we have

$$\mathrm{le}_\prec\bigl(a\partial^\mu \star b\partial^\nu\bigr) = \mu + \nu = \mathrm{le}_\prec\bigl(a\partial^\mu\bigr) + \mathrm{le}_\prec\bigl(b\partial^\nu\bigr)\ , \tag{2}$$

as any term $\partial^{\lambda+\nu}$ appearing on the right hand side of (1) divides $\partial^{\mu+\nu}$ and thus $\partial^{\lambda+\nu} \preceq \partial^{\mu+\nu}$.

Because of Condition (iii) in Definition 1 we can define Gröbner bases for left ideals in algebras of solvable type. The decisive point, explaining the conditions imposed on $\mathcal{P}$, is that normal forms with respect to a finite set $\mathcal{F} \subset \mathcal{P}$ may be computed in precisely the same way as in the ordinary polynomial ring. In the next section we will see that the involutive approach also works in this general setting.

[1] As in the first part, we use the *multi index* notation $\mu = (\mu^1, \dots, \mu^n)$ for n-tuples of nonnegative integers.

Involutive Bases

We proceed to define involutive bases for left ideals in polynomial algebras of solvable type. In principle, we could at once consider submodules of free modules over such an algebra. But this just complicates the notation. So we treat only the ideal case and show at the end of the section how the extension to the submodule case looks like.

How to Define Involutive Bases in the Polynomial Case

Definition 2 *Let $(\mathcal{P}, \star, \prec)$ be an algebra of solvable type over a field $\Bbbk$ and $\mathcal{I} \subseteq \mathcal{P}$ a left ideal. A finite set $\mathcal{H} \subset \mathcal{P}$ is an* involutive basis *of $\mathcal{I}$ for an involutive division L on $\mathbb{N}_0^n$ if no two elements have the same leading exponent and* $\mathrm{le}_\prec \mathcal{H}$ *is an involutive basis of the monoid ideal* $\mathrm{le}_\prec \mathcal{I}$.

In [11], we distinguished between *weak* and *strong* involutive bases; our implementation uses only the latter, so we restrict to them. The definition implies at once that involutive bases are Gröbner bases; in order to delve deeper into the theory, we next lift the notions of multiplicative variables and involutive span to the polynomial case.

Definition 3 *Let $\mathcal{F} \subset \mathcal{P}$ be a finite set and L an involutive division on $\mathbb{N}_0^n$. We assign each element $f \in \mathcal{F}$ a set of* multiplicative variables

$$X_{L,\mathcal{F},\prec}(f) = \left\{x_i \mid i \in N_{L,\mathrm{le}_\prec \mathcal{F}}(\mathrm{le}_\prec f)\right\} . \tag{3}$$

The involutive span *of $\mathcal{F}$ is then the set*

$$\langle\, \mathcal{F}\, \rangle_{L,\prec} = \sum_{f\in\mathcal{F}} \Bbbk[X_{L,\mathcal{F},\prec}(f)] \star f \subseteq \langle\, \mathcal{F}\, \rangle\ . \tag{4}$$

A polynomial $g \in \mathcal{P}$ is involutively reducible *with respect to $\mathcal{F}$, if it contains a term x^μ such that* $\mathrm{le}_\prec f \mid_{L,\mathrm{le}_\prec \mathcal{F}} \mu$ *for some $f \in \mathcal{F}$. It is in* involutive normal form *with respect to $\mathcal{F}$, if it is not involutively reducible. The set $\mathcal{F}$ is* involutively autoreduced*, if no polynomial $f \in \mathcal{F}$ contains a term x^μ such that another polynomial $f' \in \mathcal{F} \setminus \{f\}$ exists with* $\mathrm{le}_\prec f' \mid_{L,\mathrm{le}_\prec \mathcal{F}} \mu$.[2]

It often suffices, if one does not consider all terms in g but only the leading term $\mathrm{lt}_\prec g$: the polynomial g is *involutively head reducible*, if $\mathrm{le}_\prec f \mid_{L,\mathrm{le}_\prec \mathcal{F}} \mathrm{le}_\prec g$ for some $f \in \mathcal{F}$. *Involutively head autoreduced* sets are defined accordingly. It is clear from the definition that an involutive basis possesses this property, and from it important results on the uniqueness of normal forms can be derived. We note that for an involutively head autoreduced set $\mathcal{F}$, the sum in (4) is direct[3]. It follows easily that for such an $\mathcal{F}$ every $g \in \mathcal{P}$ has a *unique* normal form. If F is an involutive basis, then the ordinary normal form from Gröbner basis theory and the involutive normal form even coincide.

An important aspect of Gröbner bases is the existence of standard representations for ideal elements. For involutive bases a similar characterisation exists.

Theorem 4 *Let $\mathcal{I} \subseteq \mathcal{P}$ be a non-zero ideal, $\mathcal{H} \subset \mathcal{I}$ a finite set and L an involutive division on $\mathbb{N}_0^n$. Then the set $\mathcal{H}$ is an involutive basis of $\mathcal{I}$ with respect to L and $\prec$ if and only if every polynomial $f \in \mathcal{I}$ has a unique* involutive standard representation*, i. e. it can be written in the form $f = \sum_{h\in\mathcal{H}} P_h \star h$ where the coefficients $P_h \in \Bbbk[X_{L,\mathcal{H},\prec}(h)]$ satisfy* $\mathrm{lt}_\prec(P_h \star h) \preceq \mathrm{lt}_\prec f$ *for all polynomials $h \in \mathcal{H}$.*

[2] Note that the definition of an involutively autoreduced set cannot be formulated more concisely by saying that each $f \in \mathcal{F}$ is in involutive normal form with respect to $\mathcal{F}\setminus\{f\}$. If we are not dealing with a global division, the removal of f from $\mathcal{F}$ will generally change the assignment of the multiplicative indices.

[3] The converse holds too.

It is now obvious that for an involutive basis $\mathcal{H}$ of $\mathcal{I}$, ordinary and involutive span are equal: $\langle\,\mathcal{H}\,\rangle_{L,\prec} = \langle\,\mathcal{H}\,\rangle = \mathcal{I}$. The uniqueness of the standard representation, which does not hold for Gröbner bases, gives a first glimpse on why involutive bases are very interesting also from a theoretical point of view. This property, among others, allows one to thoroughly analyse the structure of the ideal under consideration.

Completion to Involution

The remaining important question is now whether any finite set $\mathcal{F} \subset \mathcal{P}$ can be *completed* to an involutive basis of the ideal it generates. This problem can be tackled in exactly the same way as in the monomial case, and the same restrictions also apply. Only for a *Noetherian* involutive division, the existence of an involutive bases is always guaranteed. For non-Noetherian divisions, the completion algorithm that we will describe below does not necessarily terminate. As briefly discussed in the first part, this means for our implementation that we can always compute Janet bases, whereas Pommaret bases exist only for δ*-regular* coordinate systems.

For the completion algorithm, we proceed to lift the property of *local involution* from multi indices to polynomial algebras of solvable type. A finite set $\mathcal{F} \subset \mathcal{P}$ is *locally involutive* for the division L, if for every polynomial $f \in \mathcal{F}$ and for every non-multiplicative variable $x_j \in \bar{X}_{L,\mathcal{F},\prec}(f)$ the product $x_j \star f$ has an involutive standard representation with respect to $\mathcal{F}$. Note that for an involutively head autoreduced set $\mathcal{F}$, we may equivalently demand that $x_j \star f \in \langle\,\mathcal{F}\,\rangle_{L,\prec}$ or that its involutive normal form is 0, respectively.

As in the monomial case, two additional conditions required by the used involutive division finally yield the algorithm for computing an involutive basis (provided that it is finite). *Continuity* ensures that involution and local involution are equivalent: if the finite set $\mathcal{F} \subset \mathcal{P}$ is locally involutive for the continuous division L, then $\langle\,\mathcal{F}\,\rangle_{L,\prec} = \langle\,\mathcal{F}\,\rangle$. *Constructivity* is needed for the correctness of the algorithm by telling us that it suffices to consider only the products appearing in the definition of local involution. Again, we omit the rather technical definitions.

Algorithm: *Completion in* $(\mathcal{P},\star,\prec)$
Input: Finite subset $\mathcal{F} \subset \mathcal{P}$, involutive division L
Output: Involutive basis $\mathcal{H}$ of $\mathcal{I} = \langle\,\mathcal{F}\,\rangle$ with respect to L and $\prec$

/1/ $\mathcal{H} \leftarrow \texttt{InvHeadAutoReduce}_{L,\prec}(\mathcal{F})$
/2/ `loop`
/3/ $\qquad \mathcal{S} \leftarrow \big\{x_j \star h \mid h \in \mathcal{H},\ x_j \in \bar{X}_{L,\mathcal{H},\prec}(h),\ x_j \star h \notin \langle\,\mathcal{H}\,\rangle_{L,\prec}\big\}$
/4/ $\qquad \texttt{if}\ \mathcal{S} = \emptyset\ \texttt{then}$
/5/ $\qquad\qquad \texttt{return}\ \mathcal{H}$
/6/ $\qquad \texttt{else}$
/7/ $\qquad\qquad \bar{g} \leftarrow \min_\prec \mathcal{S}$
/8/ $\qquad\qquad g \leftarrow \texttt{NormalForm}_{L,\prec}(\bar{g},\mathcal{H})$
/9/ $\qquad\qquad \mathcal{H} \leftarrow \texttt{InvHeadAutoReduce}_{L,\prec}(\mathcal{H} \cup \{g\})$
/10/ $\qquad$ `end_if`
/11/ `end_loop`

Figure 1: Involutive completion in $\mathcal{P}$

The completion algorithm is shown in Fig. 1. As in the monomial case, in each iteration we check the condition for local involution (line /4/): if there is a product $x_j \star h$ in the set $\mathcal{S}$ constructed in line /3/ which is not contained in the involutive span of $\mathcal{H}$, it is obviously violated. We add to $\mathcal{H}$ the normal form of the smallest such product with respect to $\prec$ (lines /7/ – /9/). This is known as the *normal strategy* in the computation of Gröbner bases; while it is known to work well there with degree compatible term orderings, it is now even crucial in the termination proof of the completion algorithm. The new $\mathcal{H}$ has to be involutively head autoreduced (line /10/) to ensure uniqueness of the normal forms. Since involutive reducibility is a restriction of ordinary reducibility, the subalgorithms `NormalForm` and `InvHeadAutoReduce` are simple modifications of the algorithms known from Gröbner basis theory.

Involutive Bases for Submodules of Free Modules

The generalisation from the ideal to the submodule case is straightforward; one proceeds in the same way as for Gröbner bases (outlined for example in [5]). We consider a free left module $\mathcal{M}$ of rank r over a polynomial algebra of solvable type $(\mathcal{P}, \star, \prec)$ and underlying polynomial ring $\Bbbk[x_1, \dots, x_n]$. This means that $\mathcal{M}$ has a basis $\{\mathbf{e}_1, \dots, \mathbf{e}_r\}$, and every element of $\mathcal{M}$ can be expressed as a finite sum of *monomials* of the form $a_{\mu,s}x^\mu\mathbf{e}_s$. The combinatorial information to be extracted consists thus of the multi index μ and the integer s for the position, i. e. it lies in $\mathbb{N}_0^n \times \{1, \dots, r\}$. We introduce involutive divisions and term orders on it as follows:

- Let L be an involutive division on $\mathbb{N}_0^n$ and suppose we are given a finite set $\mathcal{N} = \{(\mu_1, s_1) \dots, (\mu_k, s_k)\} \subset \mathbb{N}_0^n \times \{1, \dots, r\}$. For $s = 1, \dots, r$, we form the sets $\mathcal{N}_s := \{\mu_i \mid (\mu_i, s) \in \mathcal{N}\}$ and define $N_{L,\mathcal{N}}(\mu_i, s_i)$ to be equal to $N_{L,\mathcal{N}_{s_i}}(\mu_i)$. We then have that $(\mu_i, s_i)|_{L,\mathcal{N}}(\nu, t)$ if and only if $s_i = t$ and $\mu_i|_{L,\mathcal{N}_{s_i}}\nu$.
- After fixing an ordering of the set $\{1, \dots, r\}$, we can compare two elements from $\mathbb{N}_0^n \times \{1, \dots, r\}$ by either first looking at their position indices and breaking ties via applying $\prec$ to the multi indices (*position over term*) or vice versa (*term over position*). In both cases, we get an ordering on $\mathbb{N}_0^n \times \{1, \dots, r\}$ which we will call a *module order* and with the help of which we can define all the usual notions like leading term, exponent, monomial, etc.

This is everything that is needed to apply the theory developed above to submodules of $\mathcal{M}$.

The *MuPAD* Category for Polynomial Algebras of Solvable Type

For a generic implemention of the above described algorithms, especially the computation of involutive bases, the category `Cat::SolvableAlgebra(K)` representing all polynomial algebras of solvable type exists. `K` denotes here the coefficient field of the underlying polynomial ring $\mathcal{P}$. It follows from Def. 1 that the supercategories are `Cat::Ring` and `Cat::LeftModule(K)`. These are the methods provided by the category:

`nForm(f,F,<Head>,<Coeffs>)`: computes a normal form of a domain element `f` with respect to the domain elements contained in the list `F`. This is done by successively reducing `f` with elements of `F` until this is no longer possible.

`autoreduce(F,<Head>)`: returns a list of domain elements spanning the same ideal as those in the original list `F`; no element can be further reduced by the others.

`involutiveNForm(f,F,ID,<Head>,<Coeffs>)`: computes an involutive normal form of `f` modulo `F` with respect to the involutive division `ID`.

`involutiveAutoreduce(F,ID,<Head>)`: performs an involutive autoreduction of `F` using the involutive division `ID`.

`involutiveComplete(F,ID,<Head>,<Output>)`: computes an involutive basis of the ideal spanned by the elements of `F` with respect to the involutive division `ID`.

With the option `Head`, only head reductions are carried out. If the option `Coeffs` is given for `nForm` or `involutiveNForm`, the result is a list `[nf,C]` such that `nf` is the (involutive) normal form and `C` contains the factors by which the elements of `F` have been multiplied during the reduction steps. This means that we obtain a representation `f=C[1]*F[1]+...+C[k]*F[k]+nf`[4]. The option `Output` displays some information on the course of the completion algorithm like the intermediate sets, the multiplicative variables of their elements and which non-multiplicative products are selected.

The following methods have to be supplied by the domains in `Cat::SolvableAlgebra(K)` for extracting and handling the underlying combinatorial information in the form of multi indices:

`leader(f)`: returns the "leading information" of the domain element `f`, i. e. a list `[lc,le]` where `lc` is the leading coefficient and `le` the leading exponent of `f`.

`allMonomials(f)`: returns a list containing for each monomial of `f` an entry of the form `[c,e]` with its coefficient and exponent.

`exponentOrder(mu1,mu2)`: compares the two multi indices with respect to the term order used by the domain; the result is `1` if `mu1>mu2`, `-1` if `mu1<mu2`, and `0` if `mu1=mu2`.

`exponent2monomial(mu)`: converts the multi index `mu` to a domain element consisting of a single monomial with coefficient `1` and exponent `mu`.

The implemented completion algorithm differs in some points from the one described in Fig. 1. `involutiveComplete` actually even computes the unique *minimal* involutive basis, which is defined as follows: for multi indices, a set $\mathcal{N}$ is called a minimal involutive basis if it is contained in every other involutive basis $\mathcal{N}'$ with the same span. Obviously, $\mathcal{N}$ then must be unique. For the polynomial case, we simply lift this property by saying that an involutive basis is minimal if its leading exponents form a minimal involutive basis. In [8] the computation of such bases is described; we follow the algorithm given there. This includes also some optimisations based on an involutive version of Buchberger's second criterion; with its help one can recognise in advance if certain normal form computations will yield zero.

Currently, there exist in the `DOMAINS`-library of *MuPAD* three domains that are members of `Cat::SolvableAlgebra`. `Dom::SolvablePolynomial` is merely an interface so that one can use the involutive methods for polynomials. It inherits directly from `Dom::MultivariatePolynomial` and justs adds the methods needed by the category. `Dom::LinearDifferentialOperator` represents the linear differential operators given as an example for polynomial algebras of solvable type above. `Dom::WeylAlgebra` contains, in some sense, linear differential operators with polynomial coefficients; it is the free associative algebra generated by $x_1, \dots, x_n, \partial_1, \dots, \partial_n$ modulo the commutator relations for these variables. Thus the exponent vectors of elements from `Dom::WeylAlgebra` have length $2n$. The entry `_mult` inherited from `Dom::MultivariatePolynomial` is overwritten by `Dom::LinearDifferentialOperator` and `Dom::WeylAlgebra` with the new non-commutative multiplication. Instances of all three domains are created by specifying the names of the variables as list of identifiers as first and the coefficient field as second argument. The optional third argument fixes the term order; it can either be an element of `Dom::MonomOrdering` or one of the options `LexOrder`, `DegreeOrder`, `DegInvLexOrder` providing the most commonly used orderings[5].

The domain `Dom::SolvableModule` represents free left modules over polynomial algebras of solvable type. It is obviously *not* a member of `Cat::SolvableAlgebra`, since a multiplication of domain elements is not defined (only a scalar multiplication between the coefficient ring and domain elements). Nevertheless, all the methods described above are implemented and can be called in exactly the same way. The arguments required by `Dom::SolvableModule` are the polynomial algebra of solvable type `S` over which it is a module and the rank `r`. The optional third argument defines how the term order from `S` is extended to a module order: if it is `"top"`

[4] `k` is the number of elements in both `C` and `F`.

[5] The default is `DegInvLexOrder`.

(which is the default), then the strategy "term over position" is used; if it is `"pot"`, then "position over term" is used. Internally, the representation is based on that of `Dom::FreeModule`, whereas the output of a domain element consists of its coordinates with respect to the standard basis in `S`$^{\mathtt{r}}$ and thus is a list of length `r` of elements from `S`. Finally we note that everywhere the methods of `Cat::SolvableAlgebra` would expect or return a multi index `mu`, `Dom::SolvableModule` uses a pair `[mu,pos]` consisting of the exponent vector and the position of the term, i. e. the index of the respective basis vector.

Computations with Involutive Bases

Suppose that we are given elements from a polynomial algebra of solvable type or from a free module over it. We want to compute an involutive basis of the ideal or submodule they generate, read off the combinatorial information and carry out some basic commutative algebra computations directly related to basis. For this task, the domain `Dom::InvolutiveBasis` exists. It is called with exactly the same arguments one normally would give to the `involutiveComplete` command, i. e. the list of generators, an involutive division and the desired optional arguments. From that, it computes an involutive basis. Thus an instance of `Dom::InvolutiveBasis` consists of the involutive basis for the ideal or module the spanned by the generators passed as arguments and the various data that can be read off it; the domain provides procedures for computations using this information.

As an example, we create a domain for analysing the ideal spanned by the linear differential operators $\partial_x^2 + y\partial_z^2$ and ∂_y^2:

MuPAD
```
>> LDO := Dom::LinearDifferentialOperator([z,y,x], Dom::ExpressionField(normal)):
>> gens := map([D(z)^2+y*D(x)^2, D(y)^2], LDO):
>> PD := Dom::PommaretDivision(3, [3,2,1]):
>> IB := Dom::InvolutiveBasis(gens, PD):
```

`IB` is a domain representing a Pommaret basis of this ideal; we can print out the basis elements, their leading terms and the corresponding multiplicative indices:

MuPAD
```
>> IB::basisElements;
>> IB::lTerms;
>> IB::multVars
```

Output
```
     2      2          2      2            2           4      2
[D(y) , D(z)  + y D(x) , D(x)  D(y), D(y)  D(z), D(x) , D(x)  D(y) D(z),

        4
   D(x)  D(z)]

     2      2      2            2           4      2                 4
[D(y) , D(z) , D(x)  D(y), D(y)  D(z), D(x) , D(x)  D(y) D(z), D(x)  D(z)]

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

The entry `basisInfo` returns the combined information of these three calls (as a list of triples); the leading exponent vectors can be obtained with `lexps`.

With `involutiveNForm`, one can compute involutive normal forms with respect to the involutive basis `IB`; `involutiveSRep` returns, in addition, the coefficients of the involutive standard representation[6]:

```
MuPAD
>> IB::involutiveSRep(LDO(D(x)^3*D(y)^3*D(z)^3+D(x)*D(y)*D(z)))
Output

         3     3                    5              5
[[0, D(x)  D(y)  D(z), 0, - y D(x)  D(y) - 3 D(x) , 0, 0, 0], D(x) D(y) D(z)]
```

The entries of the first element of the resulting list are the coefficients for the corresponding basis vectors, while the second element is the involutive normal form.

One basic task where one uses Gröbner bases is the computation of *standard monomials*[7], that is to determine the set $\mathbb{N}_0^n \setminus \mathrm{le}_\prec\mathcal{I}$, which forms a $\Bbbk$-vector space basis for the factor module $\mathcal{P}/\mathcal{I}$ when interpreted as monomials. On the level of multi indices, one has to decompose the complement of $\mathrm{le}_\prec\mathcal{I}$ into disjoint cones. This can be done with the method `combinatorialDecomposition` from the category `Cat::InvolutiveDivision`; for the Janet division, a simpler and more efficient algorithm exists, which can also be used for the Pommaret division. The output looks like this:

```
MuPAD
>> IB::standardMonomials()
Output

[[D(y) D(z), {}], [D(x) D(y) D(z), {}], [D(z), {}], [D(x) D(z), {}],
        2                  3
   [D(x)  D(z), {}], [D(x)  D(z), {}], [D(y), {}], [D(x) D(y), {}],
                              2          3
   [1, {}], [D(x), {}], [D(x) , {}], [D(x) , {}]]
```

Each pair consists of a monomial and a set of variables giving the directions into which the cone with that monomial as base point extends. For our example, the $\Bbbk$-vector space $\mathcal{P}/\mathcal{I}$ has finite dimension, and so the cones are zero-dimensional.

We can formulate this another way: if the term order $\prec$ is degree-compatible, the algebra $\mathcal{P}/\mathcal{I}$ can be considered as graded by degree; we have thus computed in the last paragraph a *Stanley decomposition*, i. e. a $\Bbbk$-vector space isomorphism $\mathcal{P}/\mathcal{I} \cong \oplus_{t\in T}\Bbbk[X_t]\cdot t$ where T is a set of terms from $\mathcal{P}$ and $X_t \subseteq \{x_1,\dots,x_n\}$. This fact allows us to determine at once the Hilbert series of the graded module $\mathcal{P}/\mathcal{I}$; it can be read off the involutive basis $\mathcal{H}$ for the involutive division L via the formula

$$\mathcal{H}_{\mathcal{P}/\mathcal{I}}(\lambda) = \frac{1}{(1-\lambda)^n} - \sum_{f\in\mathcal{H}} \frac{\lambda^{|\mathrm{le}_\prec f|}}{(1-\lambda)^{|N_{L,\mathrm{le}_\prec H}(f)|}} \tag{5}$$

`Dom::InvolutiveBasis` possesses the methods `hilbertSeries`, `hilbertFunction`, `hilbertPolynomial` and `hilbertRegularity` (the degree from which on the Hilbert function and the Hilbert polynomial yield the same values) which compute those quantities of the factor module. For our example, the Hilbert polynomial is of course 0, because the ideal under consideration is a zerodimensional ideal; for the Hilbert series and regularity we get:

[6] For these two methods, the option `Head` is again available.

[7] Originally, they were introduced by Buchberger for that purpose.

MuPAD
```
>> IB::hilbertSeries(t); IB::hilbertRegularity()
```
Output
```
                              2     3    4
                 3 t + 4 t  + 3 t  + t  + 1
                                  5
```

Let us consider another example demonstrating the module case. `SM` represents the free module of rank two over the polynomial ring in three variables; the terms are ordered first by position and then degree reverse lexicographically:

MuPAD
```
>> SP := Dom::SolvablePolynomial([x,y,z], Dom::Rational):
>> SM := Dom::SolvableModule(SP, 2, "pot"):
>> mgens := map([[x^2,x^2*z],[z*y,x^2]], SM):
>> JD := Dom::JanetDivision(3):
>> MIB := Dom::InvolutiveBasis(mgens, JD):
>> MIB::basisInfo
```
Output
```
         2                             2   2       2
[[[y z, x ], [y z, 0], {1, 2, 3}], [[x , x  z], [x , 0], {1, 2}],
      2     2  2     2                          4    2    2
   [[x  z, x  z ], [x  z, 0], {1, 3}], [[0, - x  + x  y z ],
        2    2
   [0, x  y z ], {1, 2, 3}]]
```

For the factor module we get as vector space basis:

MuPAD
```
>> MIB::standardMonomials();
```
Output
```
[[z, 0], {z}], [[x z, 0], {z}], [[1, 0], {y}], [[x, 0], {y}],
           2                     2                 2
   [[0, y z ], {y, z}], [[0, x y z ], {y, z}], [[0, z ], {x, z}],

   [[0, 1], {x, y}], [[0, z], {x, y}]]
```

This output yields at once the Stanley decomposition ($\mathcal{M}$ denotes the original module spanned by the two generators in `mgens`):

$$\begin{aligned}\Bbbk[x,y,z]/\mathcal{M} \cong \Bbbk[z]\cdot\binom{z}{0} \oplus \Bbbk[z]\cdot\binom{xz}{0} \oplus \Bbbk[y]\cdot\binom{1}{0} \oplus \Bbbk[y,z]\cdot\binom{0}{yz^2} \\ \oplus \Bbbk[y,z]\cdot\binom{0}{xyz^2} \oplus \Bbbk[x,z]\cdot\binom{0}{z^2} \oplus \Bbbk[x,y]\cdot\binom{0}{1} \oplus \Bbbk[x,y]\cdot\binom{0}{z}.\end{aligned} \tag{6}$$

The corresponding Hilbert series, polynomial and the order of regularity for $\Bbbk[x,y,z]/\mathcal{U}$ are also easily computed from the information in `MIB`:

MuPAD

```
>> MIB::hilbertSeries(t);
>> MIB::hilbertRegularity();
>> MIB::hilbertPolynomial(x);
```

Output

```
                                          4
                                 2 t + t  + 2
                                 ------------
                                  2
                                 t  - 2 t + 1

                                         4

                      poly(5 x - 1, [x], Dom::Rational)
```

Finally, we show how to compute a set of generators of the first syzygy module; for a system of differential equations, this corresponds to determining its compatibility conditions. We consider for our involutive basis $\mathcal{H}$ all products of the form $x \star f$ where $f \in \mathcal{H}$ and x is non-multiplicative for f with respect to $\mathcal{H}$ and compute involutive standard representations for these products. The vectors containing the coefficients are the desired generators for the syzygy module ([14]). For our examples, we get:

MuPAD

```
>> IB::syzygies();
>> MIB::syzygies()
```

Output

```
[[-D(z), 0, 0, 1, 0, 0, 0], [0, 0, -D(z), 0, 0, 1, 0],
        2                                2      2
   [D(x) , 0, -D(y), 0, 0, 0, 0], [- y D(x) , D(y) , -2, -D(z), 0, 0, 0],
                                       2
   [0, 0, 0, 0, -D(z), 0, 1], [0, 0, D(x) , 0, -D(y), 0, 0],
           2              2
   [0, D(x)  D(y), - y D(x) , 0, -1, -D(z), 0],
                 2                       4                2
   [0, 0, 0, D(x) , 0, -D(y), 0], [0, D(x) , 0, 0, - y D(x) , 0, -D(z)],
                       2
   [0, 0, 0, 0, 0, D(x) , -D(y)]]

                                        2
                        [[0, -z, 1, 0], [x , 0, -y, 1]]
```

By completing these syzygies to an involutive basis and iterating this process by again calling `syzygies`, we obtain a free resolution of the module we started with.

The domain `Dom::InvolutiveBasis`, which we have now at our disposal, provides the basic methods for computations with involutive bases in polynomial algebras of solvable type. The next step is to provide a variety of functions for the structure analysis of modules based on involutive techniques. The theoretical framework is contained in [14]; especially Pommaret bases are suitable for reading off many important quantities used in commutative algebra and algebraic geometry. As already mentioned, a library for this purpose will be the topic of an upcoming article.

References

[1] Y.A. Blinkov, C.F. Cid, V.P. Gerdt, W. Plesken, D. Robertz. The MAPLE Package "Janet": I. Polynomial Systems. Preprint, 2003.

[2] Y.A. Blinkov, C.F. Cid, V.P. Gerdt, W. Plesken, D. Robertz. The MAPLE Package "Janet": II. Linear Partial Differential Equations. Preprint, 2003.

[3] 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.

[4] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Springer Verlag, New York, 1992.

[5] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer Verlag, New York, 1998.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] M. Hausdorf and W.M. Seiler. Involutive Bases in *MuPAD* – Part I: Involutive Divisions. mathPAD, 11(1):51–56, 2002.

[12] A. Kandry-Rody and V. Weispfenning. Non-commutative Gröbner Bases in Algebras of Solvable Type. *J. Symb. Comp.*, 9:1–26, 1990.

[13] W.M. Seiler. A Combinatorial Approach to Involution and Delta-Regularity I: Involutive Bases in Polynomial Algebras of Solvable Type. Preprint, 2002. Available from the author's home page.

[14] W.M. Seiler. A Combinatorial Approach to Involution and Delta-Regularity II: Structure Analysis of Polynomial Modules with Pommaret Bases. Preprint, 2002. Available from the author's home page.