OreModules: A symbolic package for the study of multidimensional linear systems

F. Chyzak [*], A. Quadrat [†], D. Robertz [‡]

Keywords Multidimensional linear systems, effective algorithms, symbolic computation, Ore algebras, Gröbner bases, parametrizations, controllability, autonomous elements, flatness, π-freeness.

1 Introduction

In the seventies, the wish to study transfer matrices of time-invariant finite-dimensional linear systems has led to the development of the polynomial approach in which the canonical forms of matrices of univariate polynomials were studied (e.g., Hermite and Smith forms, invariant factors, primeness).

In the middle of the seventies, while generalizing linear systems defined by ordinary differential equations (ODEs) to differential time-delay systems, ordinary differential equations with parameters, 2-D and 3-D filters, systems over a ring..., one had to face the case of systems described by means of matrices with entries in multivariate commutative polynomial rings. All these new systems were called 2-D or 3-D systems and, more generally, *n-D systems* or *multidimensional linear systems* with constant coefficients. It was quickly realized that no canonical forms such as Hermite, Smith and Popov forms existed for polynomial matrices with two and three variables (i.e., with entries in $k[x_1, x_2, x_3]$, where k is a field such as $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$). Moreover, more than only one type of primeness was needed in order to classify n-D systems (e.g., factor/minor/zero primeness [23]). Hence, it is not very much surprising that, in the eighties, *Gröbner bases* were introduced in the study of multidimensional linear systems with constant coefficients. A Gröbner basis defines normal forms for polynomials with respect to a certain ordering of the variables x_i [1]. Given a Gröbner basis, there is a simple algorithm to compute these normal forms effectively. In many ways, the computation of these normal forms can be seen as an extension of the Gaussian elimination algorithm to commutative polynomial rings.

In a pioneering work, R. E. Kalman developed a *module-theoretic approach* to time-invariant ordinary differential linear systems. In the nineties, U. Oberst developed the idea of studying multidimensional linear systems with constant coefficients by means of module theory and Gröbner bases [12]. In particular, using module theory, he showed how to intrinsically reformulate and generalize the different concepts of primeness used for 2-D or 3-D systems. Moreover, using some ideas of B. Malgrange, U. Oberst was able to develop a perfect duality between his module-theoretic approach to multidimensional linear systems and the *behavioural approach* developed by J. C. Willems and his school (see [14, 20] and the references therein). Since the end of the nineties, a behavioural approach to multidimensional linear systems has been successfully developed in the literature. See [13, 22, 23] for more details and references therein.

[*]INRIA Rocquencourt, ALGO project, Domaine de Voluceau BP 105, 78153 Le Chesnay cedex, France. Email: frederic.chyzak@inria.fr.

[†]INRIA Sophia Antipolis, CAFE project, 2004, Route des Lucioles BP 93, 06902 Sophia Antipolis cedex, France. Email: Alban.Quadrat@sophia.inria.fr.

[‡]Lehrstuhl B für Mathematik RWTH - Aachen, Templergraben 64, 52056 Aachen, Germany. Email: daniel@momo.math.rwth-aachen.de.

Also using module theory, the concepts of *flatness* and π*-freeness* were introduced in [8, 10] for differential time-delay linear systems with constant coefficients. The detection of such structural properties is important for the study of motion planning as it is shown in [10, 11] on different concrete examples. Let us notice that the problem of flatness is also related to the problem of computing an *observable image representation* of a multidimensional linear systems in the behavioural approach.

In the same years as [12], J.-F. Pommaret studied under-determined systems of partial differential equations (PDEs) coming from mathematical physics and differential geometry (e.g., elasticity, electromagnetism, hydrodynamics, general relativity). In particular, he showed how his mathematical approach was a generalization of U. Oberst's module-theoretic approach for multidimensional (linear) systems with varying coefficients. See [15] for more details and references therein. In particular, the problem of checking whether or not a multidimensional linear system described by PDEs with varying coefficients can be formally parametrized was solved using a differential operator approach. Moreover, the work of M. Fliess on linear systems defined by ODEs with variable coefficients also illustrated the need to pass from the commutative polynomial viewpoint to the differential operators one.

In the seventies, *algebraic analysis* was developed in order to study general linear systems of PDEs with variable coefficients by means of differential module theory, algebraic geometry, homological algebra [21] and functional analysis. Recently, algebraic analysis has been introduced in [18] for the study of multidimensional linear systems defined by PDEs with varying coefficients. In particular, using the formal theories of PDEs (Spencer's, Riquier-Janet's theories), it was shown in [15, 16, 17, 18] how some structural properties of such systems could be checked by means of effective algorithms.

Finally, using the homological algebra approach developed in [18], we have recently shown in [5, 6] how all the previous results could be generalized to most of the classes of multidimensional linear systems with varying coefficients encountered in the literature (e.g., ODEs, PDEs, differential time-delay systems, multidimensional discrete systems, partial differential delay systems, multidimensional convolutional codes). In order to do that, the concept of multidimensional linear systems over *Ore algebras* was introduced. An Ore algebra is a ring of non-commutative polynomials in functional operators with polynomial or rational coefficients [4]. The characterization of structural properties such as controllability, formal parametrizability and flatness were obtained. In particular, the following methodology for the study of the multidimensional linear systems over Ore algebras was developed:

1. A linear system Σ is defined by means of a $(q \times p)$-matrix R with entries in an Ore algebra D, i.e., Σ corresponds to the system of equations $R\,z = 0$, where z is composed of the system variables.

2. Using the matrix R, we define the left D-module $M = D^{1\times p}/D^{1\times q}\,R$.

3. We develop a dictionary between the structural properties of the system Σ and the properties of the left D-module M. Then, we use module theory in order to classify the properties of the left D-module M.

4. Homological algebra permits to check these properties of the left D-module M.

5. Using Gröbner bases over Ore algebras (i.e., over non-commutative polynomial rings), we develop some effective algorithms which check the properties of the left D-module M, and thus, of the system Σ.

6. Implementations of these algorithms in computer algebra systems.

Hence, using the recent progress of Gröbner bases over Ore algebras (i.e., over some classes of non-commutative polynomial rings) [4], we are now in position to effectively test the algebraic properties of general multidimensional linear systems (e.g., controllability, observability, flatness, poles and zeros, equivalences), to compute different types of parametrizations and to propose some feedback laws (motion planning, tracking, poles placement, optimal controllers, diophantine equations). Finally, let us point out that the link between the Ore algebra-theoretic approach to multidimensional systems and the behavioural one, pioneered by J.C. Willems [14], has not been studied in details yet.

The purpose of this paper is to give an introduction to the new package OreModules for Maple which offers symbolic methods to investigate the structural properties of multidimensional linear systems over Ore algebras. The advantage of describing these properties in the language of homological algebra carries over to the implementation of OreModules: up to the choice of the domain of operators which occur in a given system, all algorithms are stated and implemented in sufficient generality such that ODEs, PDEs, differential time-delay systems, discrete systems... are covered at the same time. The cases of linear systems with constant, polynomial or rational coefficients can be coped with. Hence, OreModules is the first implementation of homological methods in this generality with regard to applications in control theory and engineering sciences.

2 Multidimensional linear systems over Ore algebras

The mathematical framework of this paper is built on Ore algebras which are rings of non-commutative polynomials that represent linear functional operators in a natural way.

Definition 1. 1. [9] Let A be an integral domain (i.e., $a\,b = 0,\ a \neq 0 \Rightarrow b = 0$). The *skew polynomial ring* $A[\partial;\sigma,\delta]$ is the non-commutative ring consisting of all polynomials in ∂ with coefficients in A obeying the commutation rule

$$\forall\, a \in A, \quad \partial\, a = \sigma(a)\,\partial + \delta(a), \tag{1}$$

where σ is a *k-algebra endomorphism of A*, namely $\sigma : A \to A$ satisfies

$$\sigma(1) = 1, \quad \forall\, a, b \in A, \quad \sigma(a+b) = \sigma(a) + \sigma(b), \quad \sigma(a\,b) = \sigma(a)\,\sigma(b),$$

and δ is a *σ-derivation of A*, namely $\delta : A \to A$ satisfies:

$$\forall\, a, b \in A, \quad \delta(a+b) = \delta(a) + \delta(b), \quad \delta(a\,b) = \sigma(a)\,\delta(b) + \delta(a)\,b.$$

2. [4] Let $A = k[x_1, \ldots, x_n]$ be a commutative polynomial ring over a field k (if $n = 0$ then $A = k$). The skew polynomial ring $D = A[\partial_1;\sigma_1,\delta_1]\ldots[\partial_m;\sigma_m,\delta_m]$ is called *Ore algebra* if the σ_i's and δ_j's commute for $1 \leq i, j \leq m$ and satisfy $\sigma_i(\partial_j) = \partial_j, \quad \delta_i(\partial_j) = 0, \quad j < i$.

Example 1. In order to model a time-varying ordinary differential linear system, we use the *Weyl algebra* $A_1 = k[t][\partial;\sigma_1,\delta_1]$ which is the non-commutative k-algebra generated by t and ∂, i.e., contains all polynomials over k in t and ∂, where k is a field (e.g., $k = \mathbb{Q}$, $\mathbb{R}$) and the commutation rule of ∂ with polynomials $a \in k[t]$ is defined by $\partial\, a = a\,\partial + \frac{da}{dt}$, expressing the product rule when ∂ acts as differentiation on t. Therefore, in terms of Definition 1, we have $\sigma = \mathrm{id}_{k[t]}$ and $\delta = \frac{d}{dt}$.

More generally, for the study of partial differential linear systems, we shall use the Weyl algebra $A_n = k[x_1, \ldots, x_n][\partial_1;\sigma_1,\delta_1]\ldots[\partial_n;\sigma_n,\delta_n]$, where σ_i and δ_i on $k[x_1, \ldots, x_n]$ are the maps

$$\sigma_i = \mathrm{id}_{k[x_1,\ldots,x_n]}, \quad \delta_i = \frac{\partial}{\partial\, x_i}, \quad i = 1, \ldots, n,$$

and every other commutation rule is prescribed by Definition 1. In particular, we have:

$$\partial_i\, x_j = x_j\, \partial_i + \delta_{ij}, \quad 1 \leq i, j \leq n, \quad \text{where } \delta_{ij} = 1, \text{ if } i = j, \text{ and } 0 \text{ else.}$$

Example 2. The algebra of *shift operators* with polynomial coefficients is another special case of an Ore algebra. For h in the field k (e.g., $k = \mathbb{Q}$, $\mathbb{R}$), we define $S_h = k[t][\delta_h;\sigma_h,\delta]$ by:

$$\forall\, a \in k[t], \quad \sigma_h(a)(t) = a(t-h), \quad \delta(a) = 0.$$

The commutation rule $\delta_h\, t = (t-h)\,\delta_h$ represents the action of the shift operator on polynomials. Forming equations over S_h, we model time-delay (resp. time-advance) systems if $h > 0$ (resp. $h < 0$).

Example 3. For differential time-delay systems, we mix the constructions of the two preceding examples. For $h \in k$, we define the Ore algebra $D_h = k[t][\partial; \sigma_1, \delta_1][\delta_h; \sigma_2, \delta_2]$, where:

$$\sigma_1 = \mathrm{id}_{k[t]}, \quad \delta_1 = \frac{d}{dt}, \quad \forall\, a \in k[t], \quad \sigma_2(a)(t) = a(t-h), \quad \delta_2 = 0.$$

If the considered system also involves the advance operator, then we may work with the Ore algebra

$$H_h = k[t][\partial; \sigma_1, \delta_1][\delta_h; \sigma_2, \delta_2][\tau_h; \sigma_3, \delta_3],$$

where σ_i, δ_i, $i = 1, 2$, are as above and:

$$\forall\, a \in k[t], \quad \sigma_3(a)(t) = a(t+h), \quad \delta_3 = 0.$$

Example 4. In order to study multidimensional discrete linear systems, we can define the following Ore algebra $D = k[z_1, \ldots, z_n][\partial_1; \sigma_1, \delta_1] \ldots [\partial_n; \sigma_n, \delta_n]$, where σ_i and δ_i on $k[z_1, \ldots, z_n]$ are the maps:

$$\sigma_i(a)(z_1, \ldots, z_n) = a(z_1, \ldots, z_{i-1}, z_i + 1, z_{i+1}, \ldots, z_n), \quad \delta_i = 0, \quad i = 1, \ldots, n.$$

Similarly as in Example 2, we can define an Ore algebra which combines the shift operator σ_i and the inverse shift τ_i defined by $\tau_i(a)(z_1, \ldots, z_n) = a(z_1, \ldots, z_{i-1}, z_i - 1, z_{i+1}, \ldots, z_n)$. Such a construction is then a generalization of the Laurent polynomial ring to non-commutative polynomials.

We refer to [4] for more examples of Ore algebras using for instance the difference, the divided differences, the q-dilation, the q-difference functional operators. Of course, we can "concatenate" different Ore algebras in order to combine different types of functional operators and, by this means, we get Ore algebras for most of linear systems that appear in control theory.

In general, the linear systems studied in control theory are defined by means of systems of ordinary (partial) differential equations, differential time-delay equations, discrete equations... These equations usually come from mathematical models. Hence, we can generally write a system as $R\,z = 0$, where R is a matrix with entries in a certain Ore algebra and z is a set of the system variables including the input, the output, the state, the latent variables...

Example 5.

- The system $P(\frac{d}{dt})\,y = Q(\frac{d}{dt})\,u$, where P and Q are two polynomial matrices in the differential operator $\frac{d}{dt}$ and with coefficients in $k[t]$, can be rewritten as $R\,z = 0$, where the entries of the matrix $R = \big(P(\frac{d}{dt}) \,:\, -Q(\frac{d}{dt})\big)$ belong to the Weyl algebra A_1 and $z = (y^T \,:\, u^T)^T$.
- The differential time-delay system $\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t-h)$, where A and B are two matrices with entries in $k[t]$ and $h > 0$, can be rewritten as $R\,z = 0$, where the entries of the matrix $R = \big(\frac{d}{dt}\,I - A(t) \,:\, -B(t)\,\delta_h\big)$ belong to the Ore algebra D_h and $z = (x^T \,:\, u^T)^T$.
- The partial differential equation (heat equation)

$$\frac{\partial y(t,x)}{\partial t} = \frac{\partial}{\partial x}\left(a(x)\,\frac{\partial y(t,x)}{\partial x}\right) + u(t,\,x),$$

where the conductivity of the bar a is assumed to be polynomial in x, can be rewritten as $R\,z = 0$, where the entries of the matrix $R = \big(\frac{\partial}{\partial t} - \frac{\partial}{\partial x}\left(a(x)\,\frac{\partial}{\partial x}\right) \,:\, -1\big)$ belong to the Weyl algebra A_2 with $x_1 = t$, $x_2 = x$ and $z = (y \,:\, u)^T$.

Finally, let us notice that real systems are generally nonlinear ones, and thus, they do not enter in the theory developed in this paper. However, using a linearization around a (generic/given) trajectory of the system, then the linearized system has varying coefficients. If these coefficients are polynomial or rational, then we can use the approach developed in this paper in order to study the properties of the linearized system and then use the information in order to come back to the nonlinear system.

3 Module-theoretic approach to linear systems over Ore algebras

In the sixties, R. E. Kalman showed that a module-theoretic approach to linear systems was a powerful tool for the study of the structural properties of linear 1-D systems. More recently, the works of U. Oberst [12], M. Fliess [8, 10] and J.-F. Pommaret [15], also based on module theory, have highly contributed to the development of the use of module theory in control theory. This approach was particularly fruitful for the study of n-D systems where the complexity in the classification of the systems by means of primeness grows with the number n [23]. In particular, using some ideas of B. Malgrange, U. Oberst has shown in [12] how the *behavioural approach* to multidimensional linear systems, developed by J.C. Willems and his school (see [14, 20, 23] and the references therein), was dual to the module-theoretic approach. See [13, 22] for more information and references. Moreover, using also some ideas of B. Malgrange, it is explained in [17] how the theory of differential operators with varying coefficients − mainly developed by D. C. Spencer and his school in the seventies − is dual to the theory of algebraic analysis (also called *D-modules theory*) (see [15] for more details). The main idea of algebraic analysis is to study a linear system of the form $R\,z = 0$, where $R \in D^{q\times p}$, by means of the left D-module $M = D^{1\times p}/D^{1\times q}\,R$. Hence, in the past years, a classification of some properties of multidimensional linear systems has been done in terms of the properties of the corresponding left D-module M. For instance, let us summarize some of them in the following table.

Module M	**Structural properties**	**Optimal control**
Torsion	Poles/zeros classifications	
With torsion	Existence of autonomous elements	
Torsion-free	No autonomous elements, Controllability, Parametrizability, π-flatness	Variational problem without constraints (Euler-Lagrange equations)
Reflexive	Filter identification	
Projective	Internal stabilizability, Bézout identities, Stabilizing controllers	Computations of the Lagrange parameters without integrations
Free	Flatness, Poles placement, Doubly coprime factorization, Youla-Kučera parametrization of all stabilizing controllers	Optimal controller

See [8, 12, 10, 13, 15, 16, 17, 18, 20, 22, 23] and the references therein for more details. Some of these properties were obtained for particular classes of multidimensional systems (e.g., differential time-delay systems, multidimensional discrete systems, systems of PDEs, multidimensional convolutional codes). Hence, using the concepts of linear systems over an Ore algebra developed in [5, 6], we can extend the previous table to general linear systems over Ore algebras.

4 Homological tools

In the middle of the nineties, the classification of the properties of multidimensional linear systems was almost completely obtained. However, even if the use of modules allowed us to intrinsically characterize the structural properties of linear systems, i.e., without requiring special forms for the systems such as state-space formulations, input-output formulations, Roesser or Fornasini-Marchesini models..., the main issue of checking effectively the previous system properties via the properties of modules was mainly open. Only the case of multidimensional linear systems with constant coefficients defined by a full row rank matrix R with entries in the commutative polynomial ring $k[x_1, \ldots, x_n]$ was known using the different concepts of primeness [10, 12] developed in the middle of the seventies.

In [18], using the concepts of *syzygy modules*, *free resolutions*, *extension and torsion functors*, *projective and homotopic equivalences*, *projective dimensions...* developed in homological algebra [21], new algorithms allowing us to check the first column of the previous table, and thus, the systems properties of multidimensional linear systems, were obtained for systems of PDEs (see also [15, 16, 17] and the references therein). We have recently shown in [5, 6] how these algorithms could be extended to some classes of Ore algebras including all the main interesting ones used in control theory (e.g., ordinary (partial) differential equations, discrete equations, time-delay systems, multidimensional convolutional codes). The main steps of the algorithms developed in [5, 6, 18] are:

1. Computation of free resolutions of a finitely presented left module over an Ore algebra D,
2. Dualization of free resolutions of left D-modules using the $\hom_D(\cdot, D)$ functor,
3. Use of *involutions* in order to pass from right to left D-modules,
4. Computation of the quotient module of finitely presented (f.p.) left D-modules.

Using the previous four points, we are then able to compute the extension functor of any left D-module of the form $M = D^{1\times p}/D^{1\times q}\, R$ with values in D, namely $\mathrm{ext}^i(M, D)$ for $i \in \mathbb{Z}_{\geq 0}$.

Let us recall that an involution θ of D is a k-linear map $\theta : D \to D$ satisfying:

$$\forall\, a_1, a_2 \in D, \quad \theta(a_1 \cdot a_2) = \theta(a_2) \cdot \theta(a_1), \quad \theta \circ \theta = \mathrm{id}_D. \tag{2}$$

For instance, if $D = A_n$ is the Weyl algebra, then $\theta(R)$ is the classical *formal adjoint* of R obtained by multiplying a column vector of test functions on the left of $R\, z$ and by integrating by parts [15, 16, 17].

Now, if R is a matrix with entries in an Ore algebra having an involution θ (e.g., A_n, S_h, D_h), then we can define $\theta(R) = (\theta(R_{ij}))^T$ and the left D-module $\widetilde{N} = D^{1\times q}/D^{1\times p}\, \theta(R)$. The main idea developed in [5, 6, 15, 17, 18] is that the elements of the first column of the previous table are characterized by the triviality of $\mathrm{ext}^i(\widetilde{N}, D) = 0$ for certain $i \geq 0$, as it is shown in the next table.

Module M	$\mathbf{ext}^i_D(\widetilde{N}, D)$	$d(\widetilde{N})$	**Primeness**
With torsion	$\mathrm{ext}^1_D(\widetilde{N}, D) \cong t(M)$	$n-1$	$\emptyset$
Torsion-free	$\mathrm{ext}^1_D(\widetilde{N}, D) = 0$	$n-2$	Minor left-prime
Reflexive	$\mathrm{ext}^i_D(\widetilde{N}, D) = 0$, $i = 1, 2$	$n-3$	
Projective	$\mathrm{ext}^i_D(\widetilde{N}, D) = 0$, $1 \leq i \leq n$	-1	Zero left-prime

The last column of this table explains the correspondence between some module properties and some notions of primeness for a multidimensional system defined by a full row rank matrix R with entries in the commutative polynomial ring $D = k[x_1, \ldots, x_n]$. The third column generalizes the last column to multidimensional systems defined by a full row rank matrix R with entries in the Weyl algebra A_n. Finally, $d(\widetilde{N})$ denotes the Krull dimension of the *characteristic variety* of $\widetilde{N}$ (see [17, 18]).

5 The package OreModules

The main difficulty in the computation of $\mathrm{ext}^i(\widetilde{N}, D)$ is to be able to construct a free resolution for the left D-module $\widetilde{N} = D^{1\times q}/D^{1\times p}\,\theta(R)$ (see point 1 in the previous section), namely an *exact sequence* of the form (i.e., the kernel of any map of the sequence is equal to the image of the next one)

$$\ldots \xrightarrow{.\widetilde{R}_4} D^{1\times q_3} \xrightarrow{.\widetilde{R}_3} D^{1\times q_2} \xrightarrow{.\widetilde{R}_2} D^{1\times q_1} \xrightarrow{.\widetilde{R}_1} D^{1\times q_0} \longrightarrow \widetilde{N} \longrightarrow 0$$

where $.\widetilde{R}_i : D^{1\times q_i} \to D^{1\times q_{i-1}}$ is defined by $.\widetilde{R}_i(\lambda) = \lambda\,\widetilde{R}_i$ and with the notations $\widetilde{R}_1 = \theta(R)$, $q_0 = q$, $q_1 = p$. Generally, the left D-module $S_i(\widetilde{N}) \triangleq \ker .\widetilde{R}_{i-1} = \{\lambda \in D^{1\times q_{i-1}} \mid \lambda\,\widetilde{R}_{i-1} = 0\}$ is called the ith *syzygy module* of $\widetilde{N}$. If D is a *noetherian ring* [21], which is the case for a large class of Ore algebras (e.g., A_n, S_h, D_h and H_h) [5, 6], then $S_i(\widetilde{N})$ is a finitely generated left D-module.

Let us notice that the computation of the matrix $\widetilde{R}_i$ is in fact an elimination problem. Indeed, for $\lambda \in S_i(\widetilde{N})$, we have $\widetilde{R}_{i-1}\,y = u \Rightarrow \lambda\,u = (\lambda\,\widetilde{R}_{i-1})\,y = 0$, which shows that, by using left D-linear combinations of the equations of $\widetilde{R}_{i-1}\,y = u$, we must eliminate y from the inhomogeneous system $\widetilde{R}_{i-1}\,y = u$. Hence, we need to find a family of generators of the *compatibility conditions* of $\widetilde{R}_{i-1}\,y = u$ or, equivalently, a family $\{\lambda_j\}_{1\le j\le q_i}$ of elements of $D^{1\times q_{i-1}}$ satisfying $S_i(\widetilde{N}) = D\,\lambda_1 + \ldots + D\,\lambda_{q_i}$. Then, if we define $\widetilde{R}_i = (\lambda_1^T : \ldots : \lambda_{q_i}^T)^T$, then we obtain $S_i(\widetilde{N}) = D\,\lambda_1 + \ldots + D\,\lambda_{q_i} = D^{1\times q_i}\,\widetilde{R}_i$.

This difficult problem has largely been studied for the linear systems of PDEs since the 19th century [15], but it has only recently received some computational answers based on the concept of *Gröbner bases* for the Weyl algebras. More recently, an extension of the theory of Gröbner bases to some non-commutative polynomial rings and, in particular to the classes of Ore algebras [4], has made manipulations of (one-sided) ideals and modules effective. For more details concerning Gröbner bases for commutative and non-commutative polynomials, see [1, 4]. Moreover, the library *Mgfun* [3] of the computer algebra system Maple has recently been developed for the symbolic manipulation of a large class of special functions and combinatorial sequences. In particular, it has already offered the implementation of Gröbner bases for some classes of Ore algebras (see [3] for more information).

Hence, the concept of Gröbner bases for the classes of Ore algebras that are encountered in control theory was the missing point in order to effectively check the module properties, and thus, to analyze the structural properties of the corresponding multidimensional linear systems. For this purpose, using the library *Mgfun* of Maple, the authors of this paper have recently been developing the package OreModules. The second release of OreModules as well as a library of examples are freely available at `http://wwwb.math.rwth-aachen.de/OreModules`. This second release of OreModules mainly focuses on the following problems:

- Compute free resolutions, extensions functors, adjoint, dual and bidual of a finitely presented left D-module M over some classes of Ore algebras D,
- Recognize the algebraic properties of a finitely presented left D-module M,
- Recognize the existence of the autonomous elements in the corresponding system [14, 15, 16, 22] and, if so, compute a family of generators for them,
- Check whether or not a multidimensional linear system is controllable in the sense of [8, 10, 14, 13, 15, 16, 17, 22, 23].
- Check whether or not a multidimensional linear system is parametrizable in the sense of [8, 10, 15, 16].

- Check whether or not a multidimensional linear system is flat and, if so, compute an injective parametrization of the system and the flat outputs [8, 10, 15, 16].
- Check whether or not a multidimensional system with constant coefficients is π-free and, if so, compute the ideal of all the π-polynomials.

The list of the functions of OreModules is the following:

Main functions for the treatment of linear systems over Ore algebras D	
`Parametrization`	Find a parametrization of the system in terms of functions
`MinimalParametrization(s)`	Find a (some) minimal parametrization(s) of the system
`AutonomousElements`	Find generating set of autonomous elements of the system (i.e., solve the system of equations for the torsion elements) in case of Weyl algebras $D = A_n$ (i.e., PDEs)
`LeftInverse(Rat)`	Left inverse for matrices over D
`LocalLeftInverse`	Given a $0 \neq \pi \in k[x_1, \ldots, x_n]$, compute a left inverse for matrices over $k[x_1, \ldots, x_n, \pi^{-1}]$
`RightInverse(Rat)`	Right inverse for matrices over D
`GeneralizedInverse(Rat)`	Compute a generalized inverse matrix over D
`PiPolynomial`	Given a system matrix R over a commutative polynomial ring D and a variable $x_i \in D$, compute the ideal of all the π-polynomials in x_i for the given system
`FirstIntegral`	In the case of ordinary differential equations, find first integrals of motion
`LQEquations`	Euler-Lagrange equations for linear quadratic problems of optimal control (ordinary differential equations)
Module theory over Ore algebras D	
`TorsionElements`	Compute the torsion submodule of a left f.p. D-module
`Exti(Rat)`	Given a f.p. left D-module M and j, compute $\mathrm{ext}^j_D(M, D)$
`Extn(Rat)`	Given a f.p. left D-module M and m, compute $\mathrm{ext}^i_D(M, D)$ for $0 \leq i \leq m$
`Quotient(Rat)`	Compute the quotient module of two left D-modules defined as images of two matrices
`SyzygyModule(Rat)`	Compute the first syzygy module of a f.p. left D-module
`Resolution(Rat)`	Given i, compute the first ith terms of a free resolution of a f.p. left D-module
`FreeResolution(Rat)`	Compute a free resolution of a f.p. left D-module
`OreRank`	Compute the rank of a f.p. left D-module
Some low-level functions of OreModules	
`DefineOreAlgebra`	Set up an Ore algebra D in OreModules
`Involution`	Apply an involution to a matrix over D
`Factorize`	Factorize if possible one matrix over D by respect to a second one having the same number of columns
`Mult`	Multiply two or more matrices over D
`ApplyMatrix`	Apply (matrices of) operators in D to (vectors of) functions

To conclude, OreModules is the first implementation in Maple of homological methods in this generality with regard to applications in control theory.

6 Examples obtained using OreModules

OreModules comes with a library of examples which demonstrate the above features by means of applications like two pendula mounted on a cart, differential algebraic systems, electric transmission line, wind tunnel model, two reflector antenna, Einstein equations from theoretic physics, Lie-Poisson structures from differential geometry... We only give four examples but we refer the reader to [5, 6] and to `http://wwwb.math.rwth-aachen.de/OreModules` for more examples. All examples were run on a Pentium III, 1 GHz with 1 GB RAM using Maple 8 (*OreModules* is available for Maple V release 5, Maple 6, Maple 8, and Maple 9).

Example 6. We study a bipendulum [15], i.e., a system composed of a bar, where two pendula are fixed, one of length l_1 and one of length l_2. The appropriate Ore algebra for this example is the Weyl algebra $Alg = A_1$, where D is the differential operator w.r.t. time t:

```
> Alg := DefineOreAlgebra(diff=[D,t], polynom={t}, comm={g, l1,l2}):
```

Note that we have to declare all constants appearing in the system equations (the gravitational constant g, and the lengths l_1, l_2) as variables that commute with D and t. Next we enter the system matrix:

```
> R := evalm([[D^2+g/l1, 0, -g/l1], [0, D^2+g/l2, -g/l2]]);
```

$$R := \begin{bmatrix} \mathrm{D}^2 + \frac{g}{l1} & 0 & -\frac{g}{l1} \\ 0 & \mathrm{D}^2 + \frac{g}{l2} & -\frac{g}{l2} \end{bmatrix}$$

We compute the formal adjoint of R:

```
> R_adj := Involution(R, Alg);
```

$$R\_adj := \begin{bmatrix} \mathrm{D}^2 + \frac{g}{l1} & 0 \\ 0 & \mathrm{D}^2 + \frac{g}{l2} \\ -\frac{g}{l1} & -\frac{g}{l2} \end{bmatrix}$$

By computing $\mathrm{ext}^1_{A_1}(A_1^{1\times 2}/A^{1\times 3}\, R_{adj})$, we check controllability and, equivalently, parametrizability of the bipendulum:

```
> Ext := Exti(R_adj, Alg, 1);
```

$$Ext := \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \mathrm{D}^2\, l1 + g & 0 & -g \\ 0 & \mathrm{D}^2\, l2 + g & -g \end{bmatrix}, \begin{bmatrix} l2\, \mathrm{D}^2\, g + g^2 \\ g^2 + \mathrm{D}^2\, l1\, g \\ l2\, \mathrm{D}^2\, g + l2\, l1\, \mathrm{D}^4 + \mathrm{D}^2\, l1\, g + g^2 \end{bmatrix}\right]$$

From the output, we can see that the system is *generically* controllable because $Ext[1]$ is the identity matrix which means that there are no torsion elements in the left A_1-module M which is associated with the system. The interpretation of this structural fact is that the system has no autonomous elements *in the generic case.* There may be configurations of the constants g, l_1, l_2, in which the bipendulum is not controllable. We will actually find the only configuration where it is not controllable below. Since the bipendulum is generically a time-invariant controllable system, it is also generically a flat system. A *flat output* of the system can be computed as a left-inverse of the parametrization $Ext[3]$:

```
> LeftInverse(Ext[3], Alg);
```

$$\left[\begin{array}{ccc} \dfrac{l1}{g^2\,(l1-l2)} & -\dfrac{l2}{g^2\,(l1-l2)} & 0 \end{array}\right]$$

We remark that this flat output is defined only if $l_1 - l_2 \neq 0$. Moreover, $l_1 = l_2$ describes the only case in which the bipendulum may be uncontrollable. Let us finish the generic case by writing down the parametrization $Ext[3]$ in a more familiar way with a free parameter ξ_1:

```
> Parametrization(R, Alg);
```

$$\left[\begin{array}{c} g\,(l2\,(\frac{d^2}{dt^2}\,\xi_1(t)) + g\,\xi_1(t)) \\ g\,(l1\,(\frac{d^2}{dt^2}\,\xi_1(t)) + g\,\xi_1(t)) \\ l2\,l1\,(\frac{d^4}{dt^4}\,\xi_1(t)) + g\,(l1+l2)\,(\frac{d^2}{dt^2}\,\xi_1(t)) + g^2\,\xi_1(t) \end{array}\right]$$

We now turn to the case where the lengths of the pendula are equal:

```
> R_mod := subs(l2=l1, evalm(R));
```

$$R\_mod := \left[\begin{array}{ccc} \mathrm{D}^2 + \dfrac{g}{l1} & 0 & -\dfrac{g}{l1} \\ 0 & \mathrm{D}^2 + \dfrac{g}{l1} & -\dfrac{g}{l1} \end{array}\right]$$

```
> Ext_mod := Exti(Involution(R_mod, Alg), Alg, 1);
```

$$Ext\_mod := \left[\left[\begin{array}{cc} \mathrm{D}^2\,l1 + g & 0 \\ 0 & 1 \end{array}\right], \left[\begin{array}{ccc} 1 & -1 & 0 \\ 0 & \mathrm{D}^2\,l1+g & -g \end{array}\right], \left[\begin{array}{c} g \\ g \\ \mathrm{D}^2\,l1+g \end{array}\right]\right]$$

The computation of $\mathrm{ext}^1_{A_1}(A_1^{1\times 2}/A^{1\times 3}\,R_{mod})$ gives the torsion submodule $t(M)$ of M: it is generated by the row r of $Ext\_mod[2]$ which corresponds to the row with entry $l_1\,D^2+g$ in $Ext\_mod[1]$. This means that $(l_1\,D^2+g)\,r = 0$ in M, and the difference of the positions of the pendula (relative to the bar) is an autonomous element of the system. We can conclude that the bipendulum is controllable if and only if $l_1 \neq l_2$.

Let us point out that we can directly obtain the torsion elements of M as follows:

```
> TorsionElements(R_mod,[x1(t),x2(t),u(t)],Alg);
```

$$[\left[\begin{array}{c} g\,\theta_1(t) + l1\,(\frac{d^2}{dt^2}\,\theta_1(t)) = 0 \end{array}\right], \left[\begin{array}{c} \theta_1(t) = \mathrm{x1}(t) - \mathrm{x2}(t) \end{array}\right]]$$

We can also explicitly integrate this torsion element of M:

```
> AutonomousElements(R_mod,[x1(t),x2(t),u(t)],Alg)[2];
```

$$\left[\begin{array}{c} \theta_1 = \_C1\,\sin(\dfrac{\sqrt{g}\,t}{\sqrt{l1}}) + \_C2\,\cos(\dfrac{\sqrt{g}\,t}{\sqrt{l1}}) \end{array}\right]$$

The fact that there exists an autonomous element in the system is equivalent to the existence of a first integral of motion in the system. Indeed, let us recall that we have a one-to-one correspondence between the torsion elements and the first integrals of motion. For more details, see [16]. We can compute this first integral of motion by using the command FirstIntegral:

```
> V := FirstIntegral(R_mod, [x1(t),x2(t),u(t)], Alg);
```

$$V := -(-(\tfrac{d}{dt}\,\mathrm{x1}(t))\,\_C1\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l1}})\,\sqrt{l1} - (\tfrac{d}{dt}\,\mathrm{x1}(t))\,\_C2\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l1}})\,\sqrt{l1}$$
$$+ \sqrt{g}\,\mathrm{x1}(t)\,\_C1\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l1}}) - \sqrt{g}\,\mathrm{x1}(t)\,\_C2\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l1}})$$
$$+ (\tfrac{d}{dt}\,\mathrm{x2}(t))\,\_C1\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l1}})\,\sqrt{l1} + (\tfrac{d}{dt}\,\mathrm{x2}(t))\,\_C2\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l1}})\,\sqrt{l1}$$
$$- \sqrt{g}\,\mathrm{x2}(t)\,\_C1\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l1}}) + \sqrt{g}\,\mathrm{x2}(t)\,\_C2\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l1}}))\,/\sqrt{l1}$$

We let the reader check by himself that we have $\dot{V}(t) = 0$. For the explicit computations, see the complete Maple worksheet available at `http://wwwb.math.rwth-aachen.de/OreModules`.

Finally, even if we have some autonomous elements in the system, we can parametrize all solutions of the system in terms of one arbitrary function ξ_1 and two arbitrary constants C_1 and C_2 (these constants can easily be computed in terms of the initial conditions of the system):

```
>  P := Parametrization(R_mod, Alg);
```

$$P := \begin{bmatrix} g\,\xi_1(t) \\ -\_C1\,\sin(\frac{\sqrt{g}\,t}{\sqrt{l1}}) - \_C2\,\cos(\frac{\sqrt{g}\,t}{\sqrt{l1}}) + g\,\xi_1(t) \\ l1\,(\frac{d^2}{dt^2}\,\xi_1(t)) + g\,\xi_1(t) \end{bmatrix}$$

We can easily check that P gives a parametrization of some solutions of the system as we have:

```
>  simplify(ApplyMatrix(R_mod,P,Alg));
```

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We can prove that we parametrize all the C^∞-solutions of the system. For more details, see [19].

Example 7. Let us consider the differential time-delay system of a vibrating string with an interior mass [11]. We define the Ore algebra *Alg*, where D is the differential operator w.r.t. t and σ_1 and σ_2 are two non-commensurate time-delay operators. Note that the parameters η_1, η_2, which are composed of the tensions, densities and the mass [11], have to be declared in the definition of *Alg*:

```
>  Alg := DefineOreAlgebra(diff=[D,t], dual_shift=[sigma1,y1],
>  dual_shift=[sigma2,y2], polynom={t,y1,y2}, comm={eta1,eta2}):
```

We only study the case of position control on both boundaries [11]. For the case of a single control, we refer to `http://wwwb.math.rwth-aachen.de/OreModules`. We enter the system matrix R:

```
>  R := evalm([[1, 1, -1, -1, 0, 0],[D+eta1, D-eta1, -eta2, eta2, 0, 0],
>  [sigma1^2, 1, 0, 0, -sigma1, 0], [0, 0, 1, sigma2^2, 0, -sigma2]]);
```

$$R := \begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 \\ \mathrm{D}+\eta 1 & \mathrm{D}-\eta 1 & -\eta 2 & \eta 2 & 0 & 0 \\ \sigma 1^2 & 1 & 0 & 0 & -\sigma 1 & 0 \\ 0 & 0 & 1 & \sigma 2^2 & 0 & -\sigma 2 \end{bmatrix}$$

We define the formal adjoint *R_adj* of R:

```
>  R_adj := Involution(R, Alg):
```

We check controllability of the system by applying EXTI to *R_adj*:

```
>  st := time():  Ext1 := Exti(R_adj, Alg, 1):  time()-st; Ext1[1];
```

$$1.191$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Since $Ext1[1]$ is the identity matrix, we can see that the Alg-module M which is associated with the system is torsion-free. This means that the vibrating string with interior mass is controllable and, equivalently, parametrizable. A parametrization of the system is given in $Ext1[3]$:

```
>  Ext1[3];
```

$$\begin{bmatrix} 2\,\sigma2\,\eta2\,,\ -\sigma2\,\sigma1\,\eta2\,,\ -\eta2\,\sigma1+\sigma1\,\eta1-\sigma1\,\mathrm{D} \\ 0\,,\ \sigma2\,\sigma1\,\eta2\,,\ \eta2\,\sigma1+\sigma1\,\mathrm{D}+\sigma1\,\eta1 \\ \sigma2\,\mathrm{D}+\sigma2\,\eta2+\sigma2\,\eta1\,,\ -\sigma2\,\sigma1\,\eta1\,,\ 0 \\ -\sigma2\,\mathrm{D}+\sigma2\,\eta2-\sigma2\,\eta1\,,\ \sigma2\,\sigma1\,\eta1\,,\ 2\,\sigma1\,\eta1 \\ 2\,\sigma2\,\sigma1\,\eta2\,,\ \sigma2\,\eta2-\sigma2\,\eta2\,\sigma1^2\,,\ -\eta2\,\sigma1^2+\eta2+\eta1\,\sigma1^2-\sigma1^2\,\mathrm{D}+\mathrm{D}+\eta1 \\ \mathrm{D}-\mathrm{D}\,\sigma2^2+\eta2\,\sigma2^2-\eta1\,\sigma2^2+\eta2+\eta1\,,\ -\sigma1\,\eta1+\sigma1\,\eta1\,\sigma2^2\,,\ 2\,\sigma2\,\sigma1\,\eta1 \end{bmatrix}$$

So, the system can be parametrized by means of three free functions. We want to check now whether this parametrization is a *minimal* one [16]. In order to do that, let us compute the rank of M.

```
>  OreRank(R, Alg);
```

$$2$$

Hence, we know that there exist some parametrizations of the system with only two arbitrary functions [6, 16]. We find some minimal parametrizations of the system as follows:

```
>  P := MinimalParametrizations(R, Alg);
```

$$P := \left[\begin{bmatrix} 2\,\sigma2\,\eta2 & -\sigma2\,\sigma1\,\eta2 \\ 0 & \sigma2\,\sigma1\,\eta2 \\ \sigma2\,\mathrm{D}+\sigma2\,\eta2+\sigma2\,\eta1 & -\sigma2\,\sigma1\,\eta1 \\ -\sigma2\,\mathrm{D}+\sigma2\,\eta2-\sigma2\,\eta1 & \sigma2\,\sigma1\,\eta1 \\ 2\,\sigma2\,\sigma1\,\eta2 & \sigma2\,\eta2-\sigma2\,\eta2\,\sigma1^2 \\ \mathrm{D}-\mathrm{D}\,\sigma2^2+\eta2\,\sigma2^2-\eta1\,\sigma2^2+\eta2+\eta1 & -\sigma1\,\eta1+\sigma1\,\eta1\,\sigma2^2 \end{bmatrix},\right.$$

$$\begin{bmatrix} 2\,\sigma2\,\eta2\,,\ -\eta2\,\sigma1+\sigma1\,\eta1-\sigma1\,\mathrm{D} \\ 0\,,\ \eta2\,\sigma1+\sigma1\,\mathrm{D}+\sigma1\,\eta1 \\ \sigma2\,\mathrm{D}+\sigma2\,\eta2+\sigma2\,\eta1\,,\ 0 \\ -\sigma2\,\mathrm{D}+\sigma2\,\eta2-\sigma2\,\eta1\,,\ 2\,\sigma1\,\eta1 \\ 2\,\sigma2\,\sigma1\,\eta2\,,\ -\eta2\,\sigma1^2+\eta2+\eta1\,\sigma1^2-\sigma1^2\,\mathrm{D}+\mathrm{D}+\eta1 \\ \mathrm{D}-\mathrm{D}\,\sigma2^2+\eta2\,\sigma2^2-\eta1\,\sigma2^2+\eta2+\eta1\,,\ 2\,\sigma2\,\sigma1\,\eta1 \end{bmatrix},$$

$$\left.\begin{bmatrix} -\sigma2\,\sigma1\,\eta2 & -\eta2\,\sigma1+\sigma1\,\eta1-\sigma1\,\mathrm{D} \\ \sigma2\,\sigma1\,\eta2 & \eta2\,\sigma1+\sigma1\,\mathrm{D}+\sigma1\,\eta1 \\ -\sigma2\,\sigma1\,\eta1 & 0 \\ \sigma2\,\sigma1\,\eta1 & 2\,\sigma1\,\eta1 \\ \sigma2\,\eta2-\sigma2\,\eta2\,\sigma1^2 & -\eta2\,\sigma1^2+\eta2+\eta1\,\sigma1^2-\sigma1^2\,\mathrm{D}+\mathrm{D}+\eta1 \\ -\sigma1\,\eta1+\sigma1\,\eta1\,\sigma2^2 & 2\,\sigma2\,\sigma1\,\eta1 \end{bmatrix}\right]$$

Let us continue the study of the module properties of M. Since R has full row rank (this fact can be easily checked by computing FreeResolution(R, Alg)), we know that M is projective if and only if R admits a right-inverse (see [6, 16] for more details).

```
>  RightInverse(R, Alg);
```

$$[]$$

Hence, M is not projective, which implies that M is not free, i.e., the vibrating string with interior mass is not a flat system [11]. Another way to verify this is to compute the ext^2 and ext^3 of R_adj:

```
>  Exti(R_adj, Alg, 2);
```

$$\left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \sigma 2\, \eta 2 & 0 & \sigma 1\, \eta 1 \\ \eta 2 + \eta 1 + \mathrm{D} & -\sigma 1\, \eta 1 & 0 \\ 0 & \sigma 2\, \eta 2 & \eta 2 + \eta 1 + \mathrm{D} \end{bmatrix}, \begin{bmatrix} -\sigma 1\, \eta 1 \\ -\mathrm{D} - \eta 2 - \eta 1 \\ \sigma 2\, \eta 2 \end{bmatrix}\right]$$

```
>  Exti(R_adj, Alg, 3);
```

$$\left[\begin{bmatrix} \sigma 2 \\ \sigma 1 \\ \eta 2 + \eta 1 + \mathrm{D} \end{bmatrix}, \begin{bmatrix} 1 \end{bmatrix}, \mathrm{SURJ}(1)\right]$$

We see that ext^2 of R_adj is zero, but ext^3 of R_adj is different from zero. Therefore, M is a reflexive but not a projective Alg-module (we remember that M is projective if and only if ext^i of R_adj is zero for $i = 1, 2, 3$). Let us find a polynomial π in the variable σ_1 such that the system is π-free [8, 10, 11].

```
>  PiPolynomial(R, Alg, [sigma1]);
```

$$[\sigma 1]$$

Let us find a polynomial π in the variable σ_2 such that the system is π-free [10].

```
>  PiPolynomial(R, Alg, [sigma2]);
```

$$[\sigma 2]$$

Hence, if we invert σ_1 or σ_2, i.e., we allow ourselves to have time-advance operators, then, by definition of the π-polynomial, the system becomes flat. A flat output for this system can be computed from a left-inverse of the minimal parametrization P, where we allow σ_1 or σ_2 to appear in the denominators.

Let us compute the annihilator of the cokernel of the minimal parametrization $P[1]$. We know from the theory that M is a torsion Alg-module.

```
>  Ann1 := AnnExti(linalg[transpose](P[1]), Alg, 1);
```

$$Ann1 := [\sigma 2]$$

Let us compute a left-inverse of the minimal parametrization $P[1]$ by allowing σ_2 to appear in the denominators.

```
>  L1 := LocalLeftInverse(P[1], Ann1, Alg);
```

$$L1 := \begin{bmatrix} 0 & 0 & \dfrac{1}{2\,\sigma 2\,\eta 2} & \dfrac{1}{2\,\sigma 2\,\eta 2} & 0 & 0 \\ 0 & \dfrac{\sigma 1}{\sigma 2\,\eta 2} & -\dfrac{\sigma 1}{\sigma 2\,\eta 2} & -\dfrac{\sigma 1}{\sigma 2\,\eta 2} & \dfrac{1}{\sigma 2\,\eta 2} & 0 \end{bmatrix}$$

We easily check that $L1$ is a left-inverse of $P[1]$.

```
> simplify(evalm(L1 &* P[1]));
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Thus, if we can invert σ_2, we obtain that a flat output of the system is defined by

$$(\xi_1 : \xi_2)^T = L1\,(\phi_1 : \psi_1 : \phi_2 : \psi_2 : u : v)^T,$$

where ϕ_1, ψ_1, ϕ_2, ψ_2, u, v are the system variables [11]. Let us point out that any multiplication of $(\xi_1 : \xi_2)^T$ by a $\mathbb{R}(\eta_1, \eta_2)[\frac{d}{dt}, \sigma_1, \sigma_2, \sigma_2^{-1}]^{2\times 2}$ unimodular matrix gives a new flat output of the system (e.g., $(\xi_1' = 2\,\eta_2\,\sigma_2\,\xi_1 = \phi_2 + \psi_2,\ \xi_2' = \eta_2\,\sigma_2(\xi_2 + 2\,\sigma_1\,\xi_1) = \sigma_1\,\psi_1 + u)$ [11]).

We can repeat the same procedure for $P[2]$ and $P[3]$.

```
> Ann2 := AnnExti(linalg[transpose](P[2]), Alg, 1);
> Ann3 := AnnExti(linalg[transpose](P[3]), Alg, 1);
```

$$Ann2 := [\eta 2 + \eta 1 + \mathrm{D}]$$
$$Ann3 := [\sigma 1]$$

The annihilator of $P[3]$ only contains σ_1. Let us compute a flat output by allowing the time-advance operator σ_1^{-1} to appear in the basis. Let us remark that this fact is not a problem for the main application of flatness, which is the motion planning problem. See [10] for more details.

```
> L3 := LocalLeftInverse(P[3],Ann3,Alg);
```

$$L3 := \begin{bmatrix} 0 & 0 & 0 & \dfrac{\sigma 2}{\sigma 1\,\eta 1} & 0 & -\dfrac{1}{\sigma 1\,\eta 1} \\ 0 & 0 & \dfrac{1}{2\,\sigma 1\,\eta 1} & \dfrac{1}{2\,\sigma 1\,\eta 1} & 0 & 0 \end{bmatrix}$$

$L3$ is a left-inverse of $P[3]$ in the polynomial ring $\mathbb{R}(\eta_1, \eta_2)[\frac{d}{dt}, \sigma_1, \sigma_2, \sigma_1^{-1}]$, as we can check:

```
> simplify(evalm(L3 &* P[3]));
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Therefore, if we use the time-advance operator σ_1^{-1}, we obtain the following flat output of the system $(\xi_1 : \xi_2)^T = L3\,(\phi_1 : \psi_1 : \phi_2 : \psi_2 : u : v)^T$ or, using trivial $\mathbb{R}(\eta_1, \eta_2)[\frac{d}{dt}, \sigma_1, \sigma_2, \sigma_1^{-1}]$-linear combinations of ξ_1 and ξ_2, we obtain that $(\xi_1' = \sigma_2\,\psi_2 - v,\ \xi_2' = \phi_2 + \psi_2)$ is also a flat output of the system over $\mathbb{R}(\eta_1, \eta_2)[\frac{d}{dt}, \sigma_1, \sigma_2, \sigma_1^{-1}]$.

Example 8. We define the Weyl algebra $Alg = A_4$, where $D[i]$ acts as a differential operator w.r.t. $x[i]$, $i = 1, \ldots, 4$, and where $x[1]$, $x[2]$, $x[3]$ are the spatial variables and $x[4]$ is the time variable.

```
> Alg := DefineOreAlgebra(diff=[D[1],x[1]], diff=[D[2],x[2]],
> diff=[D[3],x[3]], diff=[D[4],x[4]], polynom={x[1],x[2],x[3],x[4]}):
```

We enter the system matrix of the first set of Maxwell equations, which is a matrix with entries in Alg. The first three rows stand for the sum of the time derivative of the magnetic field and the rotation of the electric field. The last row of the matrix is the divergence of the magnetic field.

```
> R := evalm([[D[4], 0, 0, 0, -D[3], D[2]],
> [0, D[4], 0, D[3], 0, -D[1]],
> [0, 0, D[4],-D[2], D[1], 0],
> [D[1], D[2], D[3], 0, 0, 0]]);
```

$$R := \begin{bmatrix} D_4 & 0 & 0 & 0 & -D_3 & D_2 \\ 0 & D_4 & 0 & D_3 & 0 & -D_1 \\ 0 & 0 & D_4 & -D_2 & D_1 & 0 \\ D_1 & D_2 & D_3 & 0 & 0 & 0 \end{bmatrix}$$

To check whether the system of Maxwell equations is parametrizable, we compute ext^1 of the formal adjoint of R:

```
> st := time(): Ext1 := Exti(Involution(R, Alg), Alg, 1); time()-st;
```

$$Ext1 := \left[\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} D_4 & 0 & 0 & 0 & -D_3 & D_2 \\ D_1 & D_2 & D_3 & 0 & 0 & 0 \\ 0 & -D_4 & 0 & -D_3 & 0 & D_1 \\ 0 & 0 & D_4 & -D_2 & D_1 & 0 \end{bmatrix}, \begin{bmatrix} D_3 & D_2 & 0 & 0 \\ 0 & -D_1 & D_3 & 0 \\ -D_1 & 0 & -D_2 & 0 \\ 0 & 0 & -D_4 & -D_1 \\ D_4 & 0 & 0 & -D_2 \\ 0 & -D_4 & 0 & -D_3 \end{bmatrix} \right]$$

$$0.440$$

Since $Ext1[1]$ is the identity matrix, we can see that the module M which is associated with R is torsion-free. Equivalently, the first set of Maxwell equations is parametrizable and we find a parametrization of the system in $Ext1[3]$. In what follows, we shall see that this parametrization is not minimal. We compute a free resolution of M first:

```
> FreeResolution(R, Alg);
```

$$\mathrm{table}([1 = \begin{bmatrix} D_4 & 0 & 0 & 0 & -D_3 & D_2 \\ 0 & D_4 & 0 & D_3 & 0 & -D_1 \\ 0 & 0 & D_4 & -D_2 & D_1 & 0 \\ D_1 & D_2 & D_3 & 0 & 0 & 0 \end{bmatrix}, 2 = \begin{bmatrix} D_1 & D_2 & D_3 & -D_4 \end{bmatrix}, 3 = \mathrm{INJ}(1)])$$

In particular, by summing alternately the number of columns of all the entries in this free resolution, we find that the rank of M is $6 - 4 + 1 = 3$. This result can also be obtained using OreRank:

```
> OreRank(R, Alg);
```

$$3$$

Hence, a minimal parametrization of the system only involves three potentials. Let us compute some minimal parametrizations of the system using MinimalParametrizations:

```
> P := MinimalParametrizations(R, Alg);
```

$$P := \left[\begin{bmatrix} D_3 & D_2 & 0 \\ 0 & -D_1 & D_3 \\ -D_1 & 0 & -D_2 \\ 0 & 0 & -D_4 \\ D_4 & 0 & 0 \\ 0 & -D_4 & 0 \end{bmatrix}, \begin{bmatrix} D_3 & D_2 & 0 \\ 0 & -D_1 & 0 \\ -D_1 & 0 & 0 \\ 0 & 0 & -D_1 \\ D_4 & 0 & -D_2 \\ 0 & -D_4 & -D_3 \end{bmatrix}, \begin{bmatrix} D_3 & 0 & 0 \\ 0 & D_3 & 0 \\ -D_1 & -D_2 & 0 \\ 0 & -D_4 & -D_1 \\ D_4 & 0 & -D_2 \\ 0 & 0 & -D_3 \end{bmatrix}, \begin{bmatrix} D_2 & 0 & 0 \\ -D_1 & D_3 & 0 \\ 0 & -D_2 & 0 \\ 0 & -D_4 & -D_1 \\ 0 & 0 & -D_2 \\ -D_4 & 0 & -D_3 \end{bmatrix} \right]$$

As a last example, we write the first of these minimal parametrizations in a more familiar way using the free parameters ξ_1, ξ_2, ξ_3 (see [6] for more details):

```
> ApplyMatrix(P[1], [xi[1](x[1],x[2],x[3],x[4]), xi[2](x[1],x[2],x[3],x[4]),
> xi[3](x[1],x[2],x[3],x[4])], Alg);
```

$$\left[\begin{array}{c} (\frac{\partial}{\partial x_3}\,\xi_1(x_1,\,x_2,\,x_3,\,x_4)) + (\frac{\partial}{\partial x_2}\,\xi_2(x_1,\,x_2,\,x_3,\,x_4)) \\ -(\frac{\partial}{\partial x_1}\,\xi_2(x_1,\,x_2,\,x_3,\,x_4)) + (\frac{\partial}{\partial x_3}\,\xi_3(x_1,\,x_2,\,x_3,\,x_4)) \\ -(\frac{\partial}{\partial x_1}\,\xi_1(x_1,\,x_2,\,x_3,\,x_4)) - (\frac{\partial}{\partial x_2}\,\xi_3(x_1,\,x_2,\,x_3,\,x_4)) \\ -(\frac{\partial}{\partial x_4}\,\xi_3(x_1,\,x_2,\,x_3,\,x_4)) \\ \frac{\partial}{\partial x_4}\,\xi_1(x_1,\,x_2,\,x_3,\,x_4) \\ -(\frac{\partial}{\partial x_4}\,\xi_2(x_1,\,x_2,\,x_3,\,x_4)) \end{array}\right]$$

Example 9. This example deals with the problem of parametrizing a system of partial differential equations with polynomial coefficients that appears in the study of the Lie algebra $SU(2)$ [2]. We define Alg as the Weyl algebra, i.e., $D[i]$ acts as differential operator w.r.t. $x[i]$, $i = 1, 2, 3$.

```
> Alg := DefineOreAlgebra(diff=[D[1],x[1]], diff=[D[2],x[2]], diff=[D[3],x[3]],
> polynom={x[1],x[2],x[3]}):
```

We enter the system matrix R:

```
> R := evalm([[x[3]*D[1]-x[1]*D[3], x[3]*D[2]-x[2]*D[3], -1],
> [-1, x[1]*D[2]-x[2]*D[1], x[1]*D[3]-x[3]*D[1]],
> [x[2]*D[1]-x[1]*D[2], -1, x[2]*D[3]-x[3]*D[2]]]);
```

$$R := \left[\begin{array}{ccc} x_3\,\mathrm{D}_1 - x_1\,\mathrm{D}_3 & x_3\,\mathrm{D}_2 - x_2\,\mathrm{D}_3 & -1 \\ -1 & x_1\,\mathrm{D}_2 - x_2\,\mathrm{D}_1 & x_1\,\mathrm{D}_3 - x_3\,\mathrm{D}_1 \\ x_2\,\mathrm{D}_1 - x_1\,\mathrm{D}_2 & -1 & x_2\,\mathrm{D}_3 - x_3\,\mathrm{D}_2 \end{array}\right]$$

Next, we define the formal adjoint R_{adj} of R.

```
> R_adj := Involution(R, Alg):
```

Applying Exti to R_{adj}, we check whether or not the system defined by R is parametrizable:

```
> st := time(): Ext1 := Exti(R_adj, Alg, 1): time() - st; Ext1[1]; Ext1[2];
```

$$0.500$$

$$\left[\begin{array}{ccc} x_2\,\mathrm{D}_3 - x_3\,\mathrm{D}_2 & 0 & 0 \\ x_1\,\mathrm{D}_3 - x_3\,\mathrm{D}_1 & 0 & 0 \\ x_1\,\mathrm{D}_2 - x_2\,\mathrm{D}_1 & 0 & 0 \\ 0 & x_2\,\mathrm{D}_3 - x_3\,\mathrm{D}_2 & 0 \\ 0 & x_1\,\mathrm{D}_3 - x_3\,\mathrm{D}_1 & 0 \\ 0 & x_1\,\mathrm{D}_2 - x_2\,\mathrm{D}_1 & 0 \\ 0 & 0 & 1 \end{array}\right]$$

$$\left[\begin{array}{ccc} x_1 & x_2 & x_3 \\ \mathrm{D}_1 & \mathrm{D}_2 & \mathrm{D}_3 \\ -1 & x_1\,\mathrm{D}_2 - x_2\,\mathrm{D}_1 & x_1\,\mathrm{D}_3 - x_3\,\mathrm{D}_1 \end{array}\right]$$

We obtain a non-trivial torsion submodule $t(M)$ of the Alg-module M which is associated with R. So, we conclude that the system of partial differential equations given by R is *not* parametrizable.

By using TorsionElements, we can obtain a generating set of the torsion submodule $t(M)$ written in terms of the unknowns F, G, H of the system. The first matrix gives the relations that the torsion elements θ_i satisfy, $i = 1, 2$, the second matrix defines θ_i in terms of F, G, H:

```
> TorsionElements(R, [F(x[1],x[2],x[3]),G(x[1],x[2],x[3]),H(x[1],x[2],x[3])],Alg);
```

$$\left[\left[\begin{array}{c} -x_3\,(\frac{\partial}{\partial x_2}\,\%2) + x_2\,(\frac{\partial}{\partial x_3}\,\%2) = 0 \\ -x_3\,(\frac{\partial}{\partial x_1}\,\%2) + x_1\,(\frac{\partial}{\partial x_3}\,\%2) = 0 \\ -x_2\,(\frac{\partial}{\partial x_1}\,\%2) + x_1\,(\frac{\partial}{\partial x_2}\,\%2) = 0 \\ -x_3\,(\frac{\partial}{\partial x_2}\,\%1) + x_2\,(\frac{\partial}{\partial x_3}\,\%1) = 0 \\ -x_3\,(\frac{\partial}{\partial x_1}\,\%1) + x_1\,(\frac{\partial}{\partial x_3}\,\%1) = 0 \\ -x_2\,(\frac{\partial}{\partial x_1}\,\%1) + x_1\,(\frac{\partial}{\partial x_2}\,\%1) = 0 \end{array}\right], \left[\begin{array}{c} \%2 = x_1\,\mathrm{F}(x_1,\,x_2,\,x_3) + x_2\,\mathrm{G}(x_1,\,x_2,\,x_3) + x_3\,\mathrm{H}(x_1,\,x_2,\,x_3) \\ \%1 = (\frac{\partial}{\partial x_1}\,\mathrm{F}(x_1,\,x_2,\,x_3)) + (\frac{\partial}{\partial x_2}\,\mathrm{G}(x_1,\,x_2,\,x_3)) + (\frac{\partial}{\partial x_3}\,\mathrm{H}(x_1,\,x_2,\,x_3)) \end{array}\right]\right]$$

$$\%1 := \theta_2(x_1,\,x_2,\,x_3)$$
$$\%2 := \theta_1(x_1,\,x_2,\,x_3)$$

$Ext1[3]$ provides a parametrization of the torsion-free part $M/t(M)$ of M:

```
>  Ext1[3];
```

$$\left[\begin{array}{c} x_3\,\mathrm{D}_2 - x_2\,\mathrm{D}_3 \\ x_1\,\mathrm{D}_3 - x_3\,\mathrm{D}_1 \\ x_2\,\mathrm{D}_1 - x_1\,\mathrm{D}_2 \end{array}\right]$$

Let us point out that we find again the same parametrization as in [2] (up to a mistake made in [2] concerning the existence of the torsion elements):

```
>  ApplyMatrix(Ext1[3], [xi(x[1],x[2],x[3])], Alg);
```

$$\left[\begin{array}{c} x_3\,(\frac{\partial}{\partial x_2}\,\xi(x_1,\,x_2,\,x_3)) - x_2\,(\frac{\partial}{\partial x_3}\,\xi(x_1,\,x_2,\,x_3)) \\ -x_3\,(\frac{\partial}{\partial x_1}\,\xi(x_1,\,x_2,\,x_3)) + x_1\,(\frac{\partial}{\partial x_3}\,\xi(x_1,\,x_2,\,x_3)) \\ x_2\,(\frac{\partial}{\partial x_1}\,\xi(x_1,\,x_2,\,x_3)) - x_1\,(\frac{\partial}{\partial x_2}\,\xi(x_1,\,x_2,\,x_3)) \end{array}\right]$$

7 Conclusion

We hope to have convinced the reader of the main interest of the Maple package OreModules for the study the structural properties of multidimensional linear systems over Ore algebras. We hope that OreModules will become in the future a platform for the implementation of different algorithms obtained in the literature of multidimensional linear systems (see e.g., [7, 13, 14, 15, 16, 17, 18, 20, 22, 23] and the references therein). In particular, a library of examples is under development and further developments of OreModules will be the object of forthcoming publications.

References

[1] Becker, T., Weispfenning, V. *Gröbner Bases. A Computational Approach to Commutative Algebra*, Springer, 1993.

[2] Bender, C. M., Dunne, G. V., Mead, L. R. "Underdetermined systems of partial differential equations", *J. Mathematical Physics*, 41 (2000), 6388-6398.

[3] Chyzak, F. "*Mgfun* Project", `http://algo.inria.fr/chyzak/mgfun.html`.

[4] Chyzak, F., Salvy, B. "Non-commutative elimination in Ore algebras proves multivariate identities", *J. Symbolic Computation*, 26 (1998), 187-227.

[5] Chyzak, F., Quadrat, A., Robertz, D. "Linear control systems over Ore algebras: Effective algorithms for the computation of parametrizations", CDRom of the *Workshop on Time-Delay Systems* (TDS03), IFAC Workshop, INRIA Rocquencourt (France) (08-10/09/03).

[6] Chyzak, F., Quadrat, A., Robertz, D. "Effective algorithms for parametrizing linear control systems over Ore algebras", *INRIA Report 5181*, available at `http://www.inria.fr/rrrt/rr-5181.html`, submitted for publication.

[7] Cotroneo, T. *Algorithms in Behavioral Systems Theory*, PhD thesis, University of Groningen (The Netherlands), 18/05/01.

[8] Fliess, M., Mounier, H. "Controllability and observability of linear delay systems: an algebraic approach", *ESAIM COCV*, 3 (1998), 301-314.

[9] McConnell, J. C., Robson, J. C. *Noncommutative Noetherian Rings*, American Mathematical Society, 2000.

[10] Mounier, H. *Propriétés structurelles des systèmes linéaires à retards: aspects théoriques et pratiques*, PhD Thesis, University of Orsay (France), 24/10/95.

[11] Mounier, H., Rudolph, J., Fliess, M., Rouchon, P. "Tracking control of a vibrating string with an interior mass viewed as delay system", *ESAIM COCV*, 3 (1998), 315-321.

[12] Oberst, U. "Multidimensional constant linear systems", *Acta Appl. Math.*, 20 (1990), 1-175.

[13] Pillai, H. K., Shankar, S. "A behavioral approach to control of distributed systems", *SIAM J. Control and Optimization*, 37 (1998), 388-408.

[14] Polderman, J. W., Willems, J. C. *Introduction to Mathematical Systems Theory. A Behavioral Approach*, TAM 26, Springer, 1998.

[15] Pommaret, J.-F. *Partial Differential Control Theory*, Kluwer, 2001.

[16] Pommaret, J.-F., Quadrat, A. "Localization and parametrization of linear multidimensional control systems", *Systems and Control Letters*, 37 (1999), 247-260.

[17] Pommaret, J.-F., Quadrat, A. "Algebraic analysis of linear multidimensional control systems", *IMA J. Control and Optimization*, 16 (1999), 275-297.

[18] Quadrat, A. *Analyse algébrique des systèmes de contrôle linéaires multidimensionnels*, PhD thesis, Ecole Nationale des Ponts et Chaussées (France), 23/09/99.

[19] Quadrat, A., Robertz, D. "On Monge problem for uncontrollable linear systems", in preparation.

[20] Rocha, P. *Structure and Representation of 2-D systems*, PhD thesis, University of Groningen (The Netherlands), 1990.

[21] Rotman, J. J. *An Introduction to Homological Algebra*, Academic Press, 1979.

[22] Wood, J. "Modules and behaviours in nD systems theory", *Multidimensional Dimensional Systems and Signal Processing*, 11 (2000), 11-48.

[23] Zerz, E. *Topics in Multidimensional Linear Systems Theory*, Lecture Notes in Control and Information Sciences 256, Springer, London (2000).