Implementing Computer Algebra within the Dirac Constraint Formalism

V. Gerdt [a] , **A. Khvedelidze** [a,b] **and Yu. Palii** [a,c]
[a] *Laboratory of Information Technologies, JINR*
[b] *Department of Theoretical Physics, A.M.Razmadze Mathematical Institute, Tbilisi, Georgia*
[c] *Institute of Applied Physics, Moldova Academy of Sciences, Chisinau, Moldova*

Abstract

We briefly summarize a new approach elaborated to describe algorithmically the finite dimensional polynomial Hamiltonian systems with constraints. The proposed scheme is based on the intensive use of the Gröbner bases technique and allows one to compute a complete set of constraints, categorize them and to construct the generator of a local symmetry transformations for a given finite dimensional polynomial degenerate model.

1 Introduction

Nowadays due to the progress in computer algebra many classical problem of mathematics and theoretical physics requiring a huge calculational work become feasible. One of the basic mathematical procedure, *completion to involution* [1, 2, 3], of systems of differential equations represents such a complicated and highly nontrivial issue in view of its computational realization. Its practical application to theoretical physics is very actual since the problem of completion is the heart of the Dirac constraint formalism [4], which is the basic tool of all modern gauge theories.

During the last few years we have attempted to elaborate an algorithmic description for the degenerate polynomial Hamiltonian mechanical models [5]-[7] and implement it in a proper computer algebra software. The key element of our approach was the intensive use of the most universal algorithmic tool of commutative algebra, the well-known *Gröbner bases* theory [8]-[10]. Because this technique provides an effective algorithmic instrument to verify whether a polynomial vanishes on the manifold defined by a set of other polynomials, the Gröbner bases method plays the principal role in algorithmic implementation of the key issues of the Dirac constraint formalism: computation and separation of constrains.

In the present report we only briefly sketch some main elements of the *Dirac-Bergmann-Gröbner* algorithmic procedure to deal with the practically important case of finite-dimensional degenerate polynomial Lagrangian system. All details of the proposed algorithmic scheme as well as an application can be found in recent publications [5]-[7],[11]-[13].

2 The Dirac-Bergmann-Gröbner algorithm

Here we describe an algorithmic reformulation of the Dirac method to making it computationally effective for the models where both, the number of degrees of freedom as well as the number of free parameters are sufficiently large. We restrict our consideration to dynamical systems with the finitely many degrees of freedom whose Lagrangian is a polynomial in coordinates and velocities with rational (possibly parametric) coefficients

$L(q,\dot{q}) \in \mathbf{Q}[q,\dot{q}]$. We use the notions and definitions of the commutative algebra adopted from [8] and [9].

Consider an arbitrary n-dimensional mechanical system in $\mathbf{R}^n$ with the Lagrangian $L(q,\dot{q})$, defined on a tangent space as a function of the coordinates $q := q_1, q_2, \ldots, q_n$ and velocities $\dot{q} := \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n$.

The Lagrangian system is *regular* if the rank $r := rank\|H_{ij}\|$ of the corresponding Hessian function $H_{ij} := \partial^2 L/\partial\dot{q}_i\partial\dot{q}_j$ is maximal ($r = n$). In this case the Euler-Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0\,, \qquad 1 \le i \le n \tag{1}$$

rewritten explicitly as

$$H_{ij}\ddot{q}_j + \frac{\partial^2 L}{\partial q_j \partial \dot{q}_i}\,\dot{q}_j - \frac{\partial L}{\partial q_i} = 0$$

can be resolved with respect to the accelerations ($\ddot{q}$) and there are no *hidden constraints*. Otherwise, if $r < n$, the Euler-Lagrange equations (and, thus, the Lagrangian system itself) are *degenerate* or *singular*. In the last case not all differential equations (1) are of second order, namely there are $n - r$ independent equations, Lagrangian constraints, containing only coordinates and velocities. Performing the Legendre transformation

$$p_i := \frac{\partial L}{\partial \dot{q}_i}\,, \tag{2}$$

the degeneracy of the Hessian results in the existence of $n-r$ relations between coordinates and momenta, the *primary constraints*

$$\varphi_a^{(1)}(p,q) = 0\,, \qquad 1 \le a \le n - r\,. \tag{3}$$

From (3) the dynamics is constrained by the set Σ_1 and according to the Dirac prescription is governed by the *total* Hamiltonian

$$H_T := H_C + U_a\,\varphi_a^{(1)}\,, \tag{4}$$

which differs from the *canonical* Hamiltonian $H_C(p,q) = p_i\dot{q}_i - L$ by a linear combination of the primary constraints with the *Lagrange multipliers* U_a.

Algorithm to determine the primary constraints

Equations (3) define the so-called *primary constraints subset* Σ_1. These relations generate the polynomial ideal in $\mathbf{Q}[p,q,\dot{q}]$

$$I_{p,q,\dot{q}} \equiv \mathrm{Id}(\cup_{i=1}^{n}\{p_i - \partial L/\partial\dot{q}_i\}) \subset \mathbf{Q}[p,q,\dot{q}]\,. \tag{5}$$

Thereby, primary constraints (3) belong to the radical $\sqrt{I_{p,q}}$ of the elimination ideal

$$I_{p,q} = I_{p,q,\dot{q}} \cap \mathbf{Q}[p,q]\,.$$

Correspondingly, for an appropriate term ordering which eliminates $\dot{q}$, a Gröbner basis of $I_{p,q}$ (denotation: $GB(I_{p,q})$) is given by [8, 9, 10]

$$GB(I_{p,q}) = GB(I_{p,q,\dot{q}}) \cap \mathbf{Q}[p,q]\,.$$

This means that construction of the Gröbner basis for the ideal (5) with omitting elements in the basis depending on velocities and then constructing of $GB(\sqrt{I_{p,q}})$ allows us to compute the set of primary constraints. If $GB(\sqrt{I_{p,q}}) = \emptyset$, then the dynamical system is regular. Otherwise, the algebraically independent set Φ_1 of primary constraints is the subset $\Phi_1 \subset GB(\sqrt{I_{p,q}})$ such that

$$\forall \phi(p,q) \in \Phi_1 \ : \ \phi(p,q) \notin \mathrm{Id}(\Phi_1 \setminus \{\phi(p,q)\}) \,. \tag{6}$$

An algorithmic verification of (6) consist in computation of the following *normal form*:

$$NF(\phi, GB(\mathrm{Id}(\Phi_1 \setminus \{\phi\})).$$

In addition, the canonical Hamiltonian $H_c(p,q)$ is computed as

$$NF(p_i\dot{q}_i - L, GB(I_{p,q,\dot{q}})).$$

Algorithm to determine the higher constraints and to classify them

The dynamical requirement that classical trajectories remain in Σ_1 during the evolution

$$\dot{\varphi}_a^{(1)} = \{H_T, \varphi_a^{(1)}\} \stackrel{\Sigma_1}{=} 0 \tag{7}$$

provides a quarantine of the consistency. In (7) the evolutional changes are generated by the canonical *Poisson brackets* with the total Hamiltonian (4) and the abbreviation $\stackrel{\Sigma_1}{=}$ stands for *a week equality*, i.e., the right-hand side of (7) vanishes modulo the primary constraints (3).

The consistency condition (7), unless it is satisfied identically, may lead either to a contradiction or to a determination of the Lagrange multipliers U_a , or to new constraints. The former case indicates that the given Hamiltonian system is inconsistent. In the latter case when (7) is not satisfied identically and is independent of the multipliers U_a the left-hand side of (7) defines the new constraints. Otherwise, if the left-hand side depends on some Lagrange multipliers U_a the consistency condition determines these multipliers, and, therefore, the constraints set is not enlarged by new constraints. The subsequent iteration of this consistency check ends up with the complete set of constraints and/or determination of some/or all Lagrange multipliers.

The number of Lagrange multipliers U_a which can be found is determined by the rank of the so-called *Poisson bracket matrix*

$$\mathbf{M}_{\alpha\beta} :\stackrel{\Sigma}{=} \{\phi_\alpha, \phi_\beta\} \,, \tag{8}$$

where Σ denotes the subset of a phase space defined by the complete set of constraints $\Phi := (\phi_1 \,, \phi_2 \,, \ldots \,, \phi_k \,)$

$$\Sigma \ : \quad \phi_\alpha(p,q) = 0 \,, \qquad 1 \le \alpha \le k \,. \tag{9}$$

including all primary $\varphi^{(1)}$, *secondary* $\varphi^{(2)}$, *ternary* $\varphi^{(3)}$, etc., constraints,

If $rank\,(\mathbf{M}) = m$, then $s := k - m$ linear combinations of constraints ϕ_α

$$\psi_\alpha(p,q) = \sum_\beta \mathrm{c}_{\alpha\beta}(p,q)\, \phi_\beta \,, \tag{10}$$

define the *first-class constraints*, whose Poisson brackets are weakly zero

$$\{\psi_\alpha(p,q),\psi_\beta(p,q)\} \stackrel{\Sigma}{=} 0 \qquad 1 \leq \alpha\,,\beta \leq s\,. \tag{11}$$

The remaining functionally independent constraints form the subset of the so-called *second-class constraints*.

The dynamical consequences (7) of a primary constraint can also be algorithmically analyzed by computing the normal form of the Poisson brackets of the primary constraint and the total Hamiltonian modulo $GB(\sqrt{I_{p,q}})$. Here the Lagrange multipliers U_a in (4) are treated as time-dependent functions. If the non-vanishing normal form does not contain U_a, then it is nothing else than the secondary constraint. In this case the set of primary constraints is enlarged by the secondary constraint obtained and the process is iterated. At the end either the complete set Φ of constraints (9) is constructed or some inconsistency is detected. The detection holds when the intermediate Gröbner basis, whose computation is a part of the iterative procedure, becomes $\{1\}$.

Categorization of constraints

In order to separate the set $\Phi = \{\phi_1,\ldots,\phi_k\}$ into subsets of the first and second class constraints the entries of Poisson brackets matrix $\mathbf{M}$ are evaluated as normal forms of the Poisson brackets of the constraints modulo a Gröbner basis of the ideal generated by set Φ. Afterwards if the basis $E = \{\mathbf{e}^{(1)},\ldots,\mathbf{e}^{(k-m)}\}$ of the null space (kernel) of this matrix $\mathbf{M}$ is known the each basis vector $\mathbf{e}^{(s)} \in E\,, s = 1,\ldots,k-m$ generates the first-class constraint of form $\mathbf{e}^{(s)}_\alpha\phi_\alpha$. The second class constraints are build using the basis of the m-dimensional space $E_\perp$, orthogonal to the E subspace. With the aid of these vectors $\mathbf{e}^{(l)}_\perp \in E_\perp\,, l = 1,\ldots,m$ the second-class constraint are constructed as $\mathbf{e}^{(l)}_{\perp\alpha}\phi_\alpha$.

Concluding we see that the constraints separation can be performed using the linear algebra operations with the matrix $\mathbf{M}$ alone. Together with the Gröbner bases technique this implies full algorithmisation for computing the complete set of algebraically independent constraints and their classification.

Algorithmic search for the generator of local symmetry transformations

First-class constraints play a very special role in the Hamiltonian description: they provide the basis for a *generator of local symmetry transformations*. Now we will demonstrate that the construction of the generator of local symmetry can be cast into the algorithmic form also.

Follow to the Dirac conjecture [4] the generator G of local transformations is expressed as a linear combination of all first-class constraints

$$G = \sum_{\beta=1}^{k_1} \varepsilon^{(1)}_\beta\phi^{(1)}_\beta \;+\; \sum_{\gamma=k_1+1}^{s} \varepsilon^{(2)}_\gamma\phi^{(2)}_\gamma\,, \tag{12}$$

and its action on phase space coordinates (q,p) is given now with the aid of the *Dirac bracket* due to the presence of the second-class constraints [4]

$$\delta q_i = \{G,q_i\}_D, \qquad \delta p_i = \{G,p_i\}_D\,.$$

In (12) the coefficients $\varepsilon^{(1)}_\beta$ and $\varepsilon^{(2)}_\gamma$ are functions of time t and the first sum includes k_1 primary first-class constraints while the second sum contains the all remaining first-class

constraints. The total time derivative of the gauge-symmetry generator (12) must be zero

$$\frac{dG}{dt} = \frac{\partial G}{\partial t} + \{G, H_C\}_D \stackrel{\Sigma_1}{=} 0\,, \tag{13}$$

the generator of local transformation is conserved modulo the primary constraints. Since the set of first-class constraints is complete, the Dirac bracket in the right-hand side of (13) is

$$\{\phi_\mu, H_C\}_D = \rho_{\mu\nu}\phi_\nu\,. \tag{14}$$

The functions $\rho_{\mu\nu}$ can be algorithmically computed by using the Gröbner bases method. To perform this computation one can use, for example, the extended Gröbner basis algorithm [8]. Given a set of polynomials $F = \{f_1, \ldots, f_m\} \subset \mathbf{Q}[p, q]$ generating the polynomial ideal $\mathrm{Id}(F)$, this algorithm yields the explicit representation

$$g_i = h_{ij}\, f_j \tag{15}$$

of elements in the Gröbner basis $\{g_1 \ldots, g_n\}$ of this ideal in terms of the ideal generated by polynomials in F. Therefore, having computed a Gröbner basis for the ideal generated by the first-class constraints and the corresponding polynomial coefficients for the elements in the Gröbner basis as given in (15), the coefficients $\rho_{\mu\nu}$ are easily computed by reduction [10, 9, 8] of the Dirac bracket in (14) modulo the Gröbner basis expressed in terms of the first-class constraints ϕ_ν. Note that one can similarly compute the algebra of first-class constraints

$$\{\phi_\alpha, \phi_\beta\}_D = \varrho_{\alpha\beta\gamma}\phi_\gamma\,,$$

if the structure functions $\varrho_{\alpha\beta\gamma}$ are polynomials in p, q.

This finalize the list of basic issues of the Dirac constraint formalism admitting the algorithmic realization.

3 Implementation

The above described algorithms where at first implemented in *Maple* [5, 7] and exemplified for the model having important physical applications, so-called Yang-Mills mechanics with the structure group $SU(2)$.

However, in attempts to perform calculations for the higher structure groups, e.g. $SU(3)$, we faced with computational difficulties. The standard Gröbner bases routines built-in *Maple* was not efficient enough to perform computation needed. We also tried recent extensions of the Maple Gröbner bases facilities with the external packages *Gb* and *Fgb* created by J.C. Faugère [14]. Unfortunately *Gb* runs for our problems even slower than the built-in package whereas *Fgb* cannot deal with the parametric coefficients. By the last reason we cannot use yet the *Ginv* [15] software that is a C++ module of Python and implements the efficient involutive algorithms [3] for the construction of the involutive or/and Gröbner bases.

4 Concluding remarks

It should be emphasized that manipulation with the parametric coefficients is essential for the Dirac formalism due to the presence of physical parameters (e.g. masses, coupling

constants) in the initial Lagrangian, the Lagrange multipliers in the total Hamiltonian (4). Having these needs in mind we implemented the algorithms in *Mathematica* whose built-in routine `GroebnerBasis` as well as `Groebner` in *Maple* allows to compute parametric Gröbner bases but performs computation much faster.

It is worth to note that the implementation in *Ginv* of multivariate GCD computation that is necessary for computation of Gröbner bases with the parametric coefficients is in progress now in collaboration with the RWTH, Aachen.

More details on the suggested scheme to implement computer algebra for calculations in the framework of the constrained Hamiltonian systems as well as its applications to the physical interesting models one can find in our recent publications [11]-[13].

References

[1] Pommaret, J.F.: *Partial Differential Equations and Group Theory. New Perspectives for Applications*, Kluwer, Dordrecht, 1994.

[2] Gerdt, V.P., Blinkov, Yu.A.: Involutive bases of polynomial ideals. *Math. Comp. Simul.* **45**, 519–542, 1998, arXiv:math.AC/9912027.

[3] Gerdt, V.P.: Involutive algorithms for computing Gröbner bases, in: S. Cojocaru, G. Pfister and V. Ufnarovski (Eds.), *Computational Commutative and Non-Commutative algebraic geometry*, NATO Science Series, IOS Press, 199–225, 2005, arXiv:math.AC/0501111.

[4] Dirac, P.A.M.: Generalized Hamiltonian dynamics, *Canad. J. Math.* **2**, 129–148, 1950; *Lectures on Quantum Mechanics*, Belfer Graduate School of Science, Monographs Series, Yeshiva University, New York, 1964.

[5] Gerdt, V.P., Gogilidze, S.A.: Constrained Hamiltonian systems and Gröbner bases, in: V.G. Ganzha, E.W. Mayr and E.V. Vorozhtsov (Eds.), *Computer Algebra in Scientific Computing/CASC'99*, Springer-Verlag, Berlin, 138–146, 1999.

[6] Gerdt, V., Khvedelidze, A. and Mladenov, D.: On application of involutivity analysis of differential equations to constrained dynamical systems, in: A.M. Sissakian (Ed.), *Symmetries and Integrable Systems, Selected Papers of the Seminar, 2000-2005* , vol. I, Dubna, JINR, 132–150, 2006, arXiv:hep-th/0311174.

[7] Gerdt, V., Khvedelidze, A. and Palii, Yu.: Towards an algorithmisation of the Dirac constraint formalism, in: J. Calmet, W.M. Seiler and R.W. Tucker (Eds.), *Global Integrabilty of Field Theories/GIFT'2006*, Cockroft Institute, Daresbury, UK, 135–154, 2006, arXiv:math-ph/0611021.

[8] Becker, T., Weispfenning, V.: *Gröbner Bases. A Computational Approach to Commutative Algebra*. Graduate Texts in Mathematics **141**, Springer-Verlag, New York, 1993.

[9] Cox D., Little J. and O'Shea D.: *Ideals, Varieties and Algorithms*. 2nd Edition, Springer-Verlag, New York, 1996.

[10] Buchberger B. and Winkler F. (Eds.): *Gröbner Bases and Applications*, Cambridge University Press, 1998.

[11] Gerdt, V., Khvedelidze A., and Palii Yu.,: Light-cone Yang-Mills mechanics: SU(2) vs. SU(3), *to appear* in: Theor. Math. Phys., 2007.

[12] Gerdt V., Khvedelidze A., and Palii Y.: Deducing the constraints in the light-cone SU(3) Yang-Mills mechanics via Gröbner bases, in: *"Computer Algebra in Scientific Computing"*, Proceedings of the 10th International Workshop, CASC 2007, Eds., V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov, Bonn, Germany , September 16-20, 2007, LNCS 470, Springer-Verlag Berlin Heidelberg, 145-159, 2007.

[13] Гердт В., Палий Ю., Хведелидзе А., Об однородном базисе Гребнера для тензоров , *to appear* in: Программирование (2007).

[14] `http://fgbrs.lip6.fr/salsa/Software/`

[15] `http://invo.jinr.ru`

[16] Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5), *Proceedings of ISSAC 2002*, ACM Press, New York, 75–83, 2002.