Solving Dynamical Nonlinear Problem with C^k Spline Functions and Grobnër Bases Techniques : An Example

Zakaria LAKHDARI, Philippe MAKANY and Marc ROUFF

Laboratoire Universitaire des Sciences Appliquées de Cherbourg - UCBN, BP 78

F-50130 Cherbourg Octeville, France

E-mail : zakaria.lakhdari@chbg.unicaen.fr

Abstract-- In this article, we describe a method to solve the Differential Equations involved in Dynamical Nonlinear Problem with C^k spline functions and their algebraic properties. Spline functions generate k time continuous and derivable approximations. These functions seem to be very efficient in several applied nonlinear differential problems. The computation of these functions use the algebraic methods known as Gröbner Bases, which are more robust , more precise and often more efficient than the purely numerical methods. An example on solving nonlinear differential equations using these methods is presented.

Keywords: Differential Equations, Dynamical Nonlinear Problem C^k spline functions, Gröbner basis, Algebra computation.

I. Introduction

The idea of computing an approximated solution of differential problem is not new. Since Fourier and his heat equations, various functions have been proposed for this task, trigonometric functions of course, exponential functions, various polynomials functions like Legendre, Chebitcheff, Laguerre or spline functions, and some discontinuous functions like Walsh functions.

In the first time, we present in this article the C^k spline functions which generate k time continuous and derivable approximations. These functions seem to be a very tool for various applied nonlinear differential problems, and particularly in nonlinear control, functional differential equations problems, delayed differential equations, etc.

In the second time, in order to obtain an approximated solutions, we solve these spline functions wih computer algebra and Gröbner bases techniques.

The plan of the paper is as follows. The main Section II is devoted to presenting the general form of the Differential Equations and their approximations by C^k spline functions, we review the necessary mathematical background. Section III contains an introduction to computer algebra and Gröbner bases techniques. In section IV we report an evaluation of this approach by presenting an example. Finally, in section V we outline the main features of the combination of C^k spline functions and algebra computation in order to solve Ordinary Differential Equations (ODE) and open issues.

II. General Form Of the ODE'S And Spline Functions

Let us consider the following nonlinear differential system of dimension N on [0,T]

$$\dot{q} = F(q)$$

The representation of C^k spline functions of q(t) in the base k is given by,

$$q(t) = \sum_{i=0}^{I} \sum_{\nu=0}^{K} q_i^{\nu} S_i^{K,\nu}(t)$$

where i is the discretization index on [0,T] and,

$$q_i^{\nu} = \frac{d^{\nu}}{dt^{\nu}} q(t)\bigg|_i$$

or given by

$$q(t) = \sum_{i=0}^{I} \sum_{\nu=0}^{K} \tilde{\mathrm{F}}(q)\bigg|_i S_i^{K,\nu}(t)$$

with $\tilde{\mathrm{F}}(q)$ the ν iteration of the vector fields defined by :

$$\tilde{\mathrm{F}}(q) = \sum_{j=1}^{N} F^j \frac{\partial}{\partial q_j}$$

Then by derivation we have,

$$\dot{q}(t) = \sum_{i=0}^{I} \sum_{\nu=0}^{K} q_i^{\nu} \dot{S}_i^{K,\nu}(t)$$

and,

$$F(q(t)) = \sum_{i=0}^{I} \sum_{\nu=0}^{K} \frac{d^{\nu} F(q)}{dq^{\nu}}\bigg|_i S_i^{K,\nu}(t)$$

Then in this kind of problems the C^k functional expansion of $q(t)$ involves only the values of q at each point of discretization, and not their derivatives, and k can be understood as the induced topology in the functional space of the solution of (1).

The algorithm used in this paper is based on minimisations of canonical functional distances represented by *H* the error of the method on [0, T] quadratic error defined as quadratic functional

$$H(q_i^{\nu}) = \int_0^T \left[\dot{q} - F(q)\right]^2 dt$$

H can be written as,

$$\begin{aligned} H(q_i^{\nu}) = & \sum_{i=0}^{I}\sum_{\nu=0}^{K}(q_i^{\nu})^2 < \dot{S}_i^{K,\nu}, \dot{S}_i^{K,\nu} > \\ & - 2.\sum_{i=0}^{I}\sum_{\nu=0}^{K} q_i^{\nu}.\left.\frac{d^{\nu}F(q)}{dq^{\nu}}\right|_i < \dot{S}_i^{K,\nu}, \dot{S}_i^{K,\nu} > \\ & + \sum_{i=0}^{I}\sum_{\nu=0}^{K}\left(\left.\frac{d^{\nu}F(q)}{dq^{\nu}}\right|_i\right)^2 < \dot{S}_i^{K,\nu}, \dot{S}_i^{K,\nu} > \end{aligned}$$

This quadratic error can be solve by two methods, a first one consists to use numerical methods as simulated annealing, a second one which use algebraic methods.

In this paper, we use Gröbner bases methods (algebraic methods) to compute these functions, in this case we use the minimisation of this quadratic error given by the following condition:

$$\frac{\partial H(q_i^{\nu})}{\partial q_i^{\nu}} = 0$$

This condition allows to obtain a polynomial system witch can be solve by Gröbner bases.

III. Computer Algebra And Gröbner

In order to give an intuitive presentation of these notions, we frequently use analogies with linear algebra well known concepts. In the following, a polynomial is a finite sum of terms, and a term is the product of a coefficient and a monomial. Refer to [5,6,7,8] for a more detailed introduction.

A. Simplification Of Polynomial System

Solving linear systems consists of studying vector spaces, and similarly, solving polynomial systems consists of studying ideals. More precisely, we define a system of polynomial equations $P_1 = 0, \ldots\ldots P_R = 0$ as a list of multivariate polynomials with rational coefficients in the algebra $Q\ [X_1, \ldots\ldots, X_N]$. To such a system, we associate ***I***, which is the ideal generated by $P_1, \ldots.., P_R$; it is the smallest ideal containing these polynomials, as well as the set of $\sum_{k=1}^{R} P_k U_k$, U_k are in $Q\ [X_1, \ldots\ldots, X_N]$. Since the P_k vanish exactly at points where all polynomials of ***I*** vanish, it is equivalent to studying the system of equations or the ideal ***I***.

For a set of linear equations, one can compute an equivalent triangular system by "canceling" the leading term of each equation. A similar method can also be done for multivariate polynomials. Of course, we have to define the leading term of a polynomial or, in other words, order the monomials. Thus, we choose an ordering on monomials compatible with the multiplication. In this paper, we only use three kinds of ordering:

- "lexicographic" order: (Lex)
  $x^{\alpha} = x^{(\alpha_1,\ldots,\alpha_N)} <_{Lex} x^{\beta} = x^{(\beta_1,\ldots,\beta_N)} \Leftrightarrow$
  $\exists i_0 \forall i = 1 \ldots i_0, \alpha_i = \beta_i$ and $\alpha_{i_0} < \beta_{i_0}$

- "degree reverse lexicographic" order: (DRL)
  $x^{(\alpha_1,\ldots,\alpha_N)} <_{DRL} x^{(\beta_1,\ldots,\beta_N)} \Leftrightarrow$
  $x^{[(\sum_k \alpha_k),\beta_N,\ldots,\beta_1]} <_{Lex} x^{[(\sum_k \beta_k),\alpha_N,\ldots,\alpha_1]}$

- "DRL by blocks" order: (DRL, DRL)

We split variables into two blocks $\alpha = (\alpha_1, \ldots, \alpha_N) =$ $[(\alpha_1, \ldots,\ \alpha_{N'-1}), (\alpha_{N'}, \ldots, \alpha_N)] = (\alpha', \alpha'')$ for some N' < N.

$$x^{(\alpha',\alpha'')} <_{DRL,DRL} x^{(\beta',\beta'')} \Leftrightarrow$$
$$(x^{\alpha'} <_{DRL} x^{\beta'}) \ \ or \ \ [(\alpha' = \beta') \ \ and \ \ x^{\alpha''} <_{DRL} x^{\beta''}]$$

Now, we can define the leading monomial (resp., term) of a polynomial as its monomial (resp., term) with highest degree.

B. Gröbner Bases

For solving systems of algebraic equations, we use Gröbner bases. Let us recall very briefly what it is, leaving details and precise definitions to [11, 8].

Given a set of polynomials, a Gröbner basis is another set of polynomials which has the same common roots in a very strong sense (the multiplicities are the same, the generated ideal is the same). A Gröbner basis may be viewed as a compiled form for a system of equations in the sense that no information is lost and, on the opposite, many properties of the solutions, such as their number or their values may easily be deduced from the Gröbner basis.

A Gröbner basis is a canonical form for a system of equations which depends only on the input equations and on a total ordering on the monomials (power products). Two orderings are especially important, the degree-reverse-lexicographical one, which leads to rather easy computations and the purely lexicographical one for which the Gröbner basis is more difficult to compute, but for which the information is more accessible.

We can now give a sketch of the Buchberger [2]-[4] algorithm, which can be seen as a constructive definition of Gröbner bases,

Gröbner (**polynomials** $f_1, \ldots, f_n$, < **a monomial ordering**)

$Pairs = \left\{[f_i, f_j], 1 \le i < j \le n\right\}$

while $Pairs \neq 0$ **do**

Choose and remove a pair $[f_i, f_j]$ in *Pairs.*

f_{n+1}=Reduce(S-pol(f_i , f_j ,<), [$f_1, \ldots, f_n$],)

if $f_{n+1} \neq 0$ **then**

n=n+1

$$Pairs = Pairs \cup \{[f_i, f_n], 1 \le i \le n\}$$

end if

end while

return $[f_1, \ldots, f_n]$

Definition 1: The output *G* of the algorithm is called a Gröbner base of *I* for the order <.

Theorem 1: *G* has the following properties :

1) G *is an equivalent set of generators of* I .

2) *A polynomial* p *belongs to* I *if and only if Reduce* (p, G) *=0.*

3) *The output of Reduce (*p, G*) does not depend on the order of the polynomials in the list. Thus, this is a canonical reduced expression modulus* I*, and the Reduce function can be used as a simplification function.*

4) *From* G*, it is easy to compute the number of complex solutions (counted with multiplicities) of the input system.*

5) *If < is lexicographic,* G *has a* "simple form" .

Solutions of an algebraic system could be of a variety of kinds that can be classified their algebraic dimension. For example

- finite number of isolated points, in which case we say that the dimension is 0;
- curves, where the dimension is 1;
- surfaces, where the dimension is 2.

If a system has different kinds of solutions (e.g., isolated points and curves), then the global dimension is the maximum dimension of each component.

Another meaningful interpretation of the dimension is that it corresponds to the remaining free degrees when all of the equations are satisfied.

C. Lexicographic Gröbner Bases

The computation time depends strongly on the monomial order that is used. In general, Gröbner bases for a lexicographic ordering are much more difficult to compute than the corresponding DRL Gröbner base. On the other hand, this computational cost is, however, worth it because the lexicographic Gröbner bases has a more or less triangular structure that is suitable for further processings. Fortunately we can compute efficiently lexicographic Gröbner bases with a different method.

D. Computer Algebra System

There is a Gröbner function in every computer algebra system (Maple, Mathematica, Axiom,...), but it must be emphasized that these implementations are very inefficient compared with recent software; even the specialized software (Magma, Singular, Macaulay, Asir) are unable to solve the most difficult systems. In this paper, we use FGB, an efficient C/C++ software developed by J.C Faugère [9], it includes a new generation of algorihms for solving polynomial systems.

IV. AN EXAMPLE

Let us consider the following Ordinary Differential equation given by :

$$\dot{q} = -q^2 \text{ with initial condition } q(0) = 1.00$$

The theoretical solution of this ODE is :

$$q(t) = \frac{1}{t+1} \; ; \qquad \dot{q}(t) = \frac{-1}{(t+1)^2} \; ; \qquad \ddot{q}(t) = \frac{2}{(t+1)^3}$$

On [0,1] interval, we obtain the exact solution of this differential equation represented by Fig. 1.

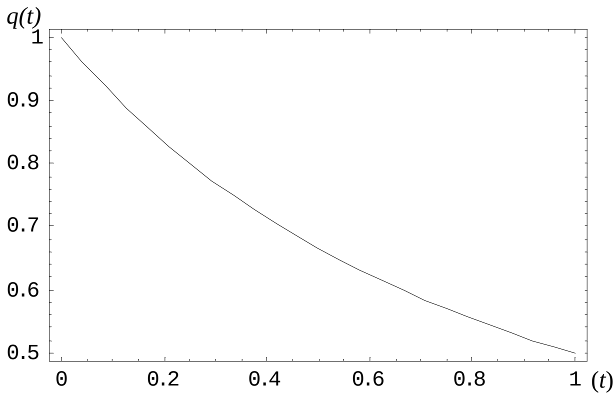

Fig. 1. Curve of the exact solution of $\dot{q} = -q^2$

An approximation of this ODE by C^k spline functions (k=2) under the computation of

$$\frac{\partial H(q_i^\gamma)}{\partial q_i^\gamma} = 0$$

gives a polynomial system (6 equations). Taking into account the initial value $q(0)$=1, this system is reduce to 5 equations given as follow(a, b, c, d, e) :

$$(((6042 + 249q[0,1]^2 + 24q[0,1]^3 + 600q[0,2] + 1320q[1,0] - 1818q[1,0]^2 - 2112q[1,1] + 652q[1,0]q[1,1] - 39q[1,1]^2 +275q[1,2] - 39q[1,0]q[1,2] + 2*q[0,1]*((3621 + 12q[0,2] +181q[1,0]^2 - 143q[1,1] + 10q[1,1]^2 - 2q[1,0]((-330 +52q[1,1] - 5q[1,2])) + 11q[1,2])))) = 0, \qquad \text{(a)}$$

$$((281 + 600q[0,1] +12q[0,1^2 + 56q[0,2] + 330q[1,0] - 149q[1,0]^2 - 275q[1,1]+ 39q[1,0]q[1,1] - q[1,1]^2 + 33q[1,2] - q[1,0]q[1,2])) = 0, \qquad \text{(b)}$$

(((-25740) + 17880q[1,0] + 41580q[1,0]² + 21720q[1,0]^3 - 492q[1,1] - 7260q[1,0]q[1,1] - 11196q[1, 0]²q[1,1] + 660q[1,1]² + 2226q[1,0]q[1,1]² - 138q[1,1]^3 + q[0,1]((1320 - 3636q[1,0] + 652q[1,1] - 39q[1,2])) - 149q[1,2] +660q[1,0]q[1,2] + 843q[1,0]²q[1,2] - 276q[1,0]q[1,1]q[1,2] + 12q[1,1]²q[1,2] + 12q[1,0]q[1,2]² - q[0,2](((-330) +298q[1,0] - 39q[1,1] + q[1, 2])) + 2q[0, 1]²((330 + 181q[1,0]- 52q[1,1] + 5q[1,2])))) = 0,

(c)

((2310 - 492q[1,0] -3630q[1,0]² - 3732q[1, 0] ^{3} + 2q[0,1]((-1056 + 326q[1,0] -39q[1,1])) + q[0,2]((-275 + 39q[1,0] - 2q[1,1])) + 5378q[1,1] + 1320q[1,0]q[1,1] + 2226q[1,0]²q[1,1] - 165q[1,1]² -414q[1,0]q[1,1]² + 24q[1,1]^3 + q[0,1]²((-143 - 104q[1,0] + 20q[1,1])) - 462q[1,2] - 138*q[1,0]²q[1,2] + 24q[1,0]q[1, 1]q[1,2])) = 0,

(d)

((q[0,1]((275 - 39q[1,0])) - q[0,2]((-33 + q[1,0])) - 149q[1,0] + 330q[1,0]² + 281q[1,0]^3 +q[0,1]²((11 + 10q[1,0])) - 462q[1,1] - 138q[1,0]²q[1,1] +12q[1,0]q[1,1]² + 44q[1,2] + 12q[1,0]²q[1,2]))) = 0,

(e)

We have 5 unknowns values q[0,1] (*derivative of q*), q[0,2] (*curvative of q*), q[1,0] (*position of q*), q[1,1] and q[1,2].

The Computation of this example using Gröbner bases gives 47 square root (3 reels, 44 complexes). Only reels square root are use to obtain the global extrema (minima).

The following Figure shows the performance results of the computation with FGB software .

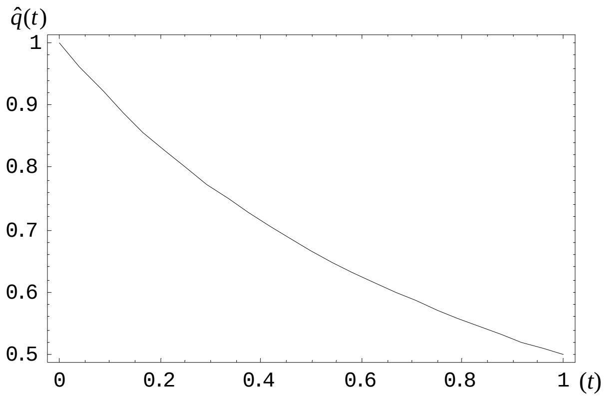

Fig. 2. Curve of the approximated solution of $\dot{q} = -q^2$

The error **e**(t) between $q(t)$ and $\hat{q}(t)$ on [0,1] interval is given by Fig. 3.

$e(t)$

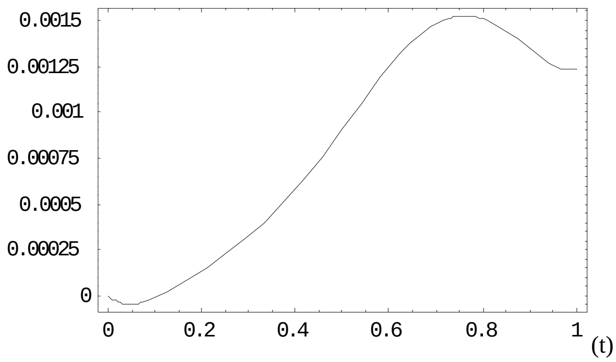

Fig. 2. Curve of the $e(t)= q(t) - \hat{q}(t)$ on [0,1]

The computation of quadratic error of this approximation is:

$$\varepsilon = \int_0^1 \left(q(t) - \hat{q}(t)\right)^2 dt = 0.00000018$$

This error shows the powerful and precise approximation using C^k spline functions and algebraic methods.

V. Conclusion

We have presented in this paper a method to solve the Ordinary Differential Equations (ODE) with C^k spline functions and algebra computation. C^k spline functions on combination with Gröbner bases computation seem to be a new powerful tool in the investigation of nonlinear differential problems.

VI. Acknowledgment

The authors would like to thank Dr. Jean Charles Faugère (Laboratoire d'Informatique de Paris 6) for his helpfull on solving these systems with FGB and for the fruitfull discussions we had.

VII. References

[1] M. Rouff, "The Computation of C^k spline functions", *Computers and Mathematics with Applications*, *23*, (1), 1992, pp.103-110.

[2] M. Rouff, "C^k spline functions in applied differential problem", *Systems Analysis Modelling and Simulations, Gordon & Breach Science Publishers, Vol. 26,* 1996, pp.197-205.

[3] M. Rouff, and M. Alaoui, "Computation of dynamical electromagnetic problems using Lagrangian formalism and multidimensional C^k spline functions", *Zeitschrift für Angewandte Mathematik und Mechanik, Academie Verlag Berlin, Vol. 76* No.1, 1996, pp. 513-14.

[4] M. Rouff, and W.L. Zhang, "C^k spline functions and linear operators", *Computers and Mathematics with Applications, Vol. 28* No. 4, 1994, pp. 51-59.

[5] B. BUCHBERGER. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. (PhD thesis, Innsbruck, 1965).

[6] B. BUCHBERGER, “An Algorithmical Criterion for the Solvability of Algebraic Systems”. *Aequationes Mathematicae 4,* 3, 1970, 374-383. (German).

[7] B. BUCHBERGER, “A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Basis”. In Proc. *EUROSAM 79,* vol. 72 of *Lect. Notes in Comp. Sci.,* Springer Verlag, 1979, pp. 3-21.

[8] J.C FAUGÈRE, *Résolution des systèmes d'équations algébriques.* (PhD thesis, Université Paris 6, Feb. 1994).

[9] J.C FAUGÈRE, “A new efficient algorithm for computing Grôbner bases (F4)”. *Journal of Pure and Applied Algebra* 139, 1-3 (June 1999), 61-88.

[10] J.C FAUGÈRE, GIANNI, P., LAZARD, D. AND MORA T. “Efficient Computation of Zero-Dimensional Gröbner Basis by Change of Ordering”. *Journal of Symbolic Computation 16,* 4 (October 1993), 329-344.

[11] D. LAZARD, “Solving zero-dimensional algebraic systems”. *Journal of Symbolic Computation 13,* 2 (February 1992), 117-132. available by anonymous ftp posso.lip6.fr.