

THOMAS WOLF¹

Merging solutions of non-linear algebraic systems²

¹Department of Mathematics, Brock University, St.Catharines, Canada and ZIB Fellow

²Submitted in 2004

Merging solutions of non-linear algebraic systems

Thomas Wolf

Department of Mathematics, Brock University
500 Glenridge Avenue, St.Catharines, Ontario, Canada L2S 3A1
email: twolf@brocku.ca

July 9, 2010

Abstract

In solving large polynomial algebraic systems that are too big for standard Gröbner basis techniques one way to make progress is to introduce case distinctions. This divide and conquer technique can be beneficial if the algorithms and computer programs know how to take advantage of inequalities. A further hurdle is the form of the resulting general solutions which often have unnecessarily many branches. In this paper we discuss a procedure to merge solutions by dropping inequalities which are associated with them and, if necessary, by re-parametrizing solutions.

In the appendix the usefulness of the procedure is demonstrated in the classification of quadratic Hamiltonians with a Lie-Poisson bracket $e(3)$. This application required the solution of algebraic systems with over 200 unknowns, 450 equations and between 5000 and 9000 terms.

Keywords: ——— computer algebra, polynomial systems, Lie-Poisson brackets, quadratic Hamiltonians

Codes: ——— PACS numbers: 02.70.Wz Symbolic computation (computer algebra)

MSC2000 classification: 37J35: Completely integrable systems, topological structure of phase space, integration methods

1 Introduction

If algebraic systems to be solved get very large then a way to convert memory needs into increased execution time is to introduce artificial case distinctions. These could be tried for the only reason that some variable occurs very often and one expects large simplifications in assuming to zero or non-zero. On the other hand Gaussian elimination could require case distinctions. For example, if a, b, c are unknowns and one of the equations of a polynomial system is $0 = ab + c^3$ then the case distinction $a = 0, a \neq 0$ leads in both cases to a useful simplification of the remaining system, in one case using the substitutions $a = c = 0$ and in the other case the substitution $b = -c^3/a$ and the inequality $a \neq 0$. Less artificial are case distinctions resulting from factorizable equations where one distinguishes between the two cases that one factor is zero or non-zero and therefore can be dropped. For the large polynomial systems described in section 1.2 both circumstances arise. In these bilinear systems, unknowns occur often only with first degree and equations can be factorized.

The problem to be addressed in this contribution concerns the form of the general solution. Introducing artificial case distinctions will usually result in a general solution consisting of many

position. The aim of the paper is to describe an effective procedure that merges special solutions of polynomial algebraic systems by dropping inequalities that are attached to solutions and, if necessary, by a re-parametrization of solutions. The ability to merge solutions fixes the main drawback of the introduction of case distinctions during the solution of very large non-linear algebraic systems as those described in section 1.2 below which are too big for a traditional Gröbner basis computation and primary or triangular decomposition.

In a ground laying and very detailed paper [1] William Sit described algorithms to solve parametric linear systems. Applications like the one in section 1.2 fall into this class. The purpose of his work was exactly to avoid the generation of unnecessarily many solutions and avoid the need for merging. Unfortunately for polynomial systems with several hundred unknowns and thousands of terms, although sparse and over-determined, the computational tools used in [1], like the computation of Gröbner bases can not be performed. For problems of this size the computation of a general solution is mainly geared by complexity issues, like the length of equations to be manipulated and less by algorithmic issues.

For example, in applications we found that if any case distinctions are to be done later in the computation then it is best to do them at the very beginning leading to short conditions which can be used to shorten other conditions and so on instead of introducing case distinctions when already many eliminations have been made using long expressions which could have been much shorter if the case distinction would have been introduced earlier. Unnecessary early growth of expressions introducing higher non-linearity is not completely reversed by setting some variables to zero later in the computation.

In short, the strategy we follow currently is to give the solution process as much freedom as possible and to face the task of cleaning up the general solution by merging special branches afterwards.

Outline:...

1.1 Notation

The formulation of the algorithm and the notation we use is tailored to its use which is the merging of special solutions which together compose the general solution of an algebraic system as they are produced by the package CRACK ([2]).

For a polynomial algebraic system for a set X of unknowns $X = \{x_j\}$ we will call a solution S_i a list $S_i = \{E_i, A_i, F_i, U_i\}$ where

- E_i is a set $E_i = \{e_{i1}, \dots, e_{in_i}\}$ of unsolved polynomial equations $0 = e_{ij}$, which is a Gröbner basis according to some lexicographical ordering.¹
- A_i is a list $A_i = \{x_j = R_{ij}(x_k), \dots\}$ of assignments of unknowns x_j by rational expressions R_{ij} . None of the x_j on left hand sides of assignment A_i is supposed to appear in any expression of E_i (otherwise the substitution for x_j is to be carried out in E_i). Assignments A_i will also be used in form of a list of vanishing expressions $Z_i = \{x_j - R_{ij}(x_k), \dots\}$.
- F_i is a list of free unknowns appearing only on the right hand sides R_{ij} of assignments A_i or undetermined unknowns that can appear as well in the unsolved conditions in E_i , and
- U_i is a list of inequalities associated with the solution S_i .

¹The condition of being a Gröbner basis wrt. some lexicographical ordering is only necessary for the quick preliminary merging condition (5).

(and by that $A_i = \hat{A}_i + \tilde{A}_i$ and $Z_i = \hat{Z}_i + \tilde{Z}_i$) through $\hat{X}_1 = \hat{X}_2 = X_1 \cap X_2$ and therefore $\tilde{X}_1 \cap \tilde{X}_2 = \emptyset$ and further

$$\tilde{X}_1 \subset F_2, \quad \tilde{X}_2 \subset F_1. \quad (1)$$

An overline – will mark a re-parametrized solution $\overline{S}_2 = \{\overline{E}_2, \overline{A}_2, \overline{F}_2, \overline{U}_2\}$. With $\#()$ we denote the number of elements of a set, e.g. $\#(F_i)$ is the number of unknowns that are free or undetermined in solution S_i . With $\text{num}()$ we will denote the numerator of a rational expression. The ideal of a set of polynomials (i.e. the set of all linear combinations of elements of E_i with polynomial coefficients) is denoted by I , for example the ideal of E_i is $I(E_i)$. With the notation $A_i(H)$ we will indicate that the list of substitutions A_i is carried out in a list H of expressions, for example, $A_1(E_2)$. With $:=$ we will denote assignments like in the Pascal programming language, e.g. $Z_2 := A_1(Z_2)$ stands for performing substitutions A_1 in Z_2 and calling the new list Z_2 .

1.2 A non-trivial Application

The package CRACK has been used for a variety of classifications of integrable systems (see [2]). The harder problems become the more the program must rely on introducing case distinctions and the more solutions have to be merged afterwards. In a recent investigation [3] the merging algorithm was especially useful, merging 90% of the solutions. The aim was to find quadratic Hamiltonians of the form

$$H = (M, AM) + (M, B\gamma) + (\gamma, C\gamma) + (P, M) + (Q, \gamma) \quad (2)$$

where $M = (M_1, M_2, M_3)$, $\gamma = (\gamma_1, \gamma_2, \gamma_3)$ are the dynamic variables, A, C are constant symmetric matrices, B is a general constant matrix and P, Q are constant vectors, such that this Hamiltonian admits polynomial first integrals of 3rd or 4th degree. The equations of motion are given by

$$\frac{dM_i}{dt} = \{H, M_i\}, \quad \frac{d\gamma_i}{dt} = \{H, \gamma_i\}$$

where Poisson brackets for any two functions F, G of the variables $Y_i = (M_1, M_2, M_3, \gamma_1, \gamma_2, \gamma_3)$ are defined by

$$\{F, G\} = \sum_{i,j} \left(\frac{\partial F}{\partial Y_i} \frac{\partial G}{\partial Y_j} - \frac{\partial F}{\partial Y_j} \frac{\partial G}{\partial Y_i} \right) \{Y_i, Y_j\}, \quad (3)$$

$$\{M_i, M_j\} = \sum_k \varepsilon_{ijk} M_k, \quad \{M_i, \gamma_j\} = \sum_k \varepsilon_{ijk} \gamma_k, \quad \{\gamma_i, \gamma_j\} = 0 \quad (4)$$

with ε_{ijk} being the totally anti-symmetric tensor. These linear Poisson brackets are related to the Lie algebra $e(3)$. Hamiltonians (2) with Poisson brackets (4) describe the motion of a rigid body around a fixed point under gravity (Euler-Poinsot model) and the motion of a rigid body in ideal fluid (Kirchhoff model).

For the first integral I an ansatz for a general 4th degree polynomial in the dynamical variables M_i, γ_j is made with constant unknown coefficients. For I to be a first integral the Poisson bracket $\{H, I\}$, as defined above, has to vanish identically in the M_i and γ_j . The set of resulting algebraic conditions is linear in the components of A, B, C, P, Q in (2) and linear in the 200 coefficients of the ansatz for I . But as each term of the conditions has two factors, one from each set, the problem is non-linear (bi-linear). In our approach we assumed the matrix A to have two identical eigenvalues. In a second investigation we restrict the dynamical variables to obey $\gamma_1 M_1 + \gamma_2 M_2 + \gamma_3 M_3 = 0$ (a special motion in a plane of symmetry leading to less restrictive and therefore harder algebraic systems) and finally we run a quantum version where M_i, γ_j are non-commuting operators and the

M_i, γ_j like in (4).

Our complete classification revealed new (complex) Hamiltonians and a generalization of known classes of Hamiltonians of Kowalewski and Sokolov [4].

Lie algebra:	$e(3)$	$e(3), J_2 = 0$	$e(3)$ quantum
# of unknowns (H, I, total)	17,200,217	17,176,193	17,200,217
# of equations	451	396	451
total # of terms	5469	5243	9681
average # of terms/equ.	12.1	13.2	21.5
time to solve	18h 53min	\approx 15h	11h 43min
solutions before merging	61	68	53
solutions after merging	6	6	6
HOW LONG DOES MERGING TAKE?			

Table 1. An overview of the solved algebraic systems

Times are measured on a 1.7GHz Pentium 4 running in a 120 MByte REDUCE session under Linux. The first and third case had been solved automatically whereas the second case needed some human interaction. On the web page <http://lie.math.brocku.ca/twof/b1/e3/over.html> the conditions, the unknowns and assumed inequalities are shown as well as the solutions before and after merging. The task to merge these solutions could not be done by hand as each solution contains expressions for about 200 unknowns, different solutions may be parametrized differently and with typically 60 solutions in each of the 3 investigations over a thousand merging attempts had to be done to obtain finally only 6 different solutions in each case.

2 Merging Solutions

2.1 Preliminary Considerations

When is a solution S_1 contained in a solution S_2 after dropping inequalities or re-parametrising S_2 ? If assignments A_1 are regular in solution S_2 (i.e. $A_1(E_2)$ and $A_1(Z_2)$ do not produce singularities through zero denominators), then S_2 generalizes S_1 if and only if all resulting conditions are consequences of equations E_1 . To perform substitutions, to take a numerator and test membership of the resulting polynomials $\text{num}(A_1(E_2))$ and $\text{num}(A_1(Z_2))$ in the ideal $I(E_1)$ does not pose a problem and can be done efficiently.

This leaves us with the case that $A_1(Z_2)$ creates zero denominators. In that case we have at least the following necessary conditions for merger:

1. If

$$\#(F_1) - \#(E_1) > \#(F_2) - \#(E_2) \quad (5)$$

then S_1 can not be a special case of S_2 . For this condition to be correct we assumed above that E_i is a Gröbner Bases according to some lexicographical ordering.

2. Substitutions $A_1(E_2)$ are always regular as E_2 is a list of polynomials and any variable on a left hand side of an assignment in A_1 does not occur on a rhs of A_1 . Therefore

$$\text{num}(A_1(E_2)) \subset I(E_1) \quad (6)$$

is an effective necessary condition to be tested.

numerator of Z_2 before performing each of the substitutions A_1 and testing

$$\text{num}(A_1(\text{num}(Z_2))) \subset I(E_1). \quad (7)$$

All three tests are only necessary and not sufficient as can be seen from the following 2 examples, both arising from the application of section 1.2 and both giving singularities when computing $A_1(Z_2)$.

Example 1:

Solutions S_1, S_2 involve among other variables the unknowns x_1, \dots, x_5 (which we re-named to make the example better readable). We have

$$\begin{aligned} S_1 : \quad & x_1 = x_2 = x_3 = 0, & x_4, x_5 & \text{ are free} \\ S_2 : \quad & x_2 = -(x_3x_5)/x_1, \quad x_4 = (x_3x_5)/x_1, & x_1, x_3, x_5 & \text{ are free.} \end{aligned}$$

Both solutions satisfy conditions (5), (6) but solution S_2 satisfies the condition $x_4 + x_2 = 0$ which would be a restriction on the generality of solution S_1 . This observation is unaffected by whatever re-parametrization one would choose for solution S_2 , so S_1 can not be a special case of S_2 .

Example 2:

Solutions S_1, S_2 involve among other variables the unknowns x_1, \dots, x_5 . We have

$$S_1 : \quad x_5 = x_6 = 0, \quad (8)$$

$$x_7 = (x_4x_2)/(2x_1), \quad (9)$$

$$x_1 \neq 0, \quad x_1, x_2, x_3, x_4 \text{ are free}$$

$$S_2 : \quad x_2 = (x_6^2 - 4x_5x_7)/x_6, \quad (10)$$

$$x_4 = (x_5 - x_1)x_6/(2x_5), \quad (11)$$

$$x_5 - x_1 \neq 0, \quad x_5 \neq 0, \quad x_6 \neq 0, \quad x_1, x_3, x_5, x_6, x_7 \text{ are free.}$$

It is possible to include solution S_1 into S_2 by re-parametrizing S_2 , i.e. instead of parametrizing x_2, x_4 in terms of x_1, x_5, x_6, x_7 to have x_6, x_7 parametrized by x_1, x_2, x_4, x_5 by inverting (11) and (10):

$$x_6 = \frac{2x_4x_5}{x_5 - x_1} \quad (12)$$

$$x_7 = \frac{x_4^2x_5}{(x_5 - x_1)^2} - \frac{x_2x_4}{2(x_5 - x_1)}. \quad (13)$$

The assignments (8) do not cause problems anymore when done to the new form of solution S_2 . Equation (12) becomes an identity and (13) reduces to (9).

From now on we are concerned only with situations where the tests (5),(6) are passed but where $A_1(Z_2)$ creates singularities.

2.2 The Strategy

In full analogy to the 2nd example, the algorithm described in the following subsection will, if possible, find re-parametrization of the assignments A_2 of solution S_2 to assignments $\overline{A_2}$ such that $A_1(\overline{Z_2})$ is singularity free and $A_1(\overline{A_2}(E_2))$ is singularity free. Then a simple membership test can establish whether the numerators of all these expressions are in the ideal of E_1 . If that is the case then S_1 can be dropped as it is included in solution $\overline{S_2} = \{\overline{E_2}, \overline{A_2}, \overline{F_2}, \overline{U_2}\}$ where $\overline{A_2}$ are the new parametrized assignments, $\overline{E_2} = \text{num}(\overline{A_2}(E_2))$, $\overline{F_2}$ are the new parameters and $\overline{U_2}$ is the subset of expressions of $\overline{A_2}(U_2)$ which are not in $I(E_1)$, i.e. do not vanish due to E_1 .

In the following the *steps of the algorithm* are distinguished from comments using an italic font.

- The computation starts with performing *preliminary necessary tests 1. and 2.* as described in section 2.1. If one of them fails then a merger is not possible.

- We recall from section 1.1 that \hat{A}_1, \hat{A}_2 share the same x_j on left hand sides of their assignments. The next step is to *perform substitutions* $Z_2 := \hat{A}_1(Z_2)$.

Do we loose generality in performing these substitutions before considering parametrizations? The substituted x_j occurred only on left hand sides in \hat{A}_1, \hat{A}_2 (i.e. only in \hat{Z}_i , not \tilde{Z}_i) and nowhere else in the remaining objects of interest: $E_1, E_2, \hat{A}_1, \hat{A}_2, Z_2, \tilde{Z}_2$. The expressions in the new Z_2 would have to vanish whatever the re-parametrization would be. We therefore do not loose generality. Instead, because the number of free parameters in a solution is usually considerably smaller than the number of unknowns, we have $\#(\hat{A}_i) \gg \#(\tilde{A}_i)$. Therefore, eliminating most of the x_j from further consideration by doing $\hat{A}_1(Z_2)$ narrows and simplifies the choice for re-parametrizations below.

- From section 1.1 we recall that $\tilde{A}_1 = A_1 \setminus \hat{A}_1$. *On a copy of Z_2 test whether the remaining substitutions $\tilde{A}_1(Z_2)$ are regular.* If so, S_1, S_2 can be merged iff $\text{num}(\tilde{A}_1(Z_2)) \subset I(E_1)$.

- If execution reaches this point then substitutions $\tilde{A}_1(Z_2)$ are singular. *Preliminary necessary test 3. of section 2.1 is done next* doing each substitution of A_1 only on the numerator of the result of the previous substitution.

- If execution reaches this point then a re-parametrization of S_2 is necessary to decide whether merging is possible.

What form should the re-parametrized solution $\overline{S_2}$ take to make obvious that S_1 is contained? We will require that all x_j on left hand sides of $\overline{A_2}$ (i.e. $\overline{X_2}$) are also on left hand sides of A_1 , i.e. $\overline{X_2} \subset X_1$ and that $A_1(\overline{Z_2})$ is regular and further $\text{num}(A_1(\overline{Z_2})) \subset I(E_1)$.

How can S_2 be re-parametrized to reach this? Assignments \hat{A}_2 can not be changed as the x_j on left hand sides of \hat{A}_2 are assigned a value in A_1 too. All that can be done is to re-write \tilde{A}_2 , i.e. to solve the system of equations \tilde{Z}_2 (i.e. the equations when setting all expressions in \tilde{Z}_2 to zero) for other variables than \tilde{X}_2 . \tilde{Z}_2 has to be solved for a subset of \tilde{X}_1 with $\#(\tilde{Z}_2)$ many variables. In a complete investigation \tilde{Z}_2 would be solved for any such subset, for example by the methods described in [1]². As only one regular re-parametrization is needed, in the program CRACK a less complete but efficient and in all applications so far succesful search is performed. For example, in the application of section 1.2 it found all possible mergings. In addition, at this stage of the computation the program gives notice that the 3 preliminary tests of section 2.1 have been passed so that if no suitable re-parametrization is found below then the user could check manually.

- *The x_j in the list \tilde{X}_1 are sorted so that those x_j come first which produce singularities when substituted through \tilde{A}_1 in Z_2 .* The purpose is to try solving \tilde{Z}_2 for such variables first to avoid singularities in the later substitution $\tilde{A}_1(Z_2)$.

A side issue:

Successful re-parametrizations are not unique. As CRACK has been used to solve bi-linear systems with the strategy to keep one of the two linearly occuring sets of variables linear during computation, this aim is also pursued in a re-parametrizations. As a consequence, variables $x_j \in \tilde{X}_1$ which occur linearly in Z_2 get the highest priority in re-sorting \tilde{X}_1 .

- *Initialize $\overline{E_2} := E_2$, $\overline{A_2} := \hat{A}_2$, $\overline{F_2} := \tilde{X}_2 \cup F_2$ and $\overline{U_2} := U_2$. Solve each individual equation $0 = \tilde{z}_{2i}$, $\tilde{z}_{2i} \in \tilde{Z}_2$ for one variable $x_j \in \tilde{X}_1$ to get its solution in the form $x_j = \overline{R}_{2j}$. Do this by*

²In the application of section 1.2 the 200 unknown coefficients in the ansatz for the first integral appear linear in the beginning, during the computation and in each solution. That means, at all times the system can be seen as a parametric linear system as considered in [1].

in \tilde{z}_{2i} , that has a coefficient that is non-zero in in-equalities U_2 and that gives a regular substitution $x_j = \overline{R_{2j}}$ when applied to Z_2 . If no variable x_j and consequently no such solution $x_j = \overline{R_{2j}}$ is found then the method stops without finding a merger. If for the equation $0 = \tilde{z}_{2i}$ under consideration a variable x_j is found, then do

$$Z_2 := \text{num}((x_j = \overline{R_{2j}})(Z_2)) \quad (14)$$

$$\tilde{Z}_2 := \text{num}((x_j = \overline{R_{2j}})(\tilde{Z}_2)) \quad (15)$$

$$\tilde{X}_1 := \tilde{X}_1 \setminus \{x_j\} \quad (16)$$

$$\tilde{Z}_2 := \tilde{Z}_2 \cup \{\tilde{A}_1(x_j - \overline{R_{2j}})\} \quad (17)$$

$$\tilde{A}_1 := \tilde{A}_1 \setminus \{x_j = R_{1j}\} \quad (18)$$

$$\overline{E}_2 := \text{num}((x_j = \overline{R_{2j}})(\overline{E}_2)) \quad (19)$$

$$\overline{A}_2 := \overline{A}_2 \cup \{x_j = \overline{R_{2j}}\} \quad (20)$$

$$\overline{F}_2 := \overline{F}_2 \setminus \{x_j\} \quad (21)$$

$$\overline{U}_2 := \text{num}((x_j = \overline{R_{2j}})(\overline{U}_2)) \quad (22)$$

Comments:

- The substitution $x_j = \overline{R_{2j}}$ in Z_2 as shown in line (14) had already been computed when searching for an x_j giving a regular substitution. (14) is just assigning the computed value to Z_2 .

- By doing substitutions in \tilde{Z}_2 in line (15) it may happen that expressions become factorizable. In that case we drop all factors which do not contain the single x_j which was on the left hand side of the original assignment $x_j = R_{2j}$ in \tilde{A}_2 which became the original $\tilde{z}_{2j} = x_j - R_{2j}$.

- In (17) x_j is substituted by the value it has in solution S_1 . The new difference is another expressions added to \tilde{Z}_2 which must vanish modulo E_1 for merger.

- If finally all equations $0 = \tilde{z}_{2j}$ are solved then lines (19),(20),(21),(22) will provide the complete list of remaining unsolved equations \overline{E}_2 , assignments \overline{A}_2 , list of free or unsolved variables \overline{F}_2 and inequalities \overline{U}_2 .

- If the execution reaches this point, solution S_2 has been successfully re-parametrized to \overline{S}_2 . To include solution S_1 it has to check that the remaining substitutions \tilde{A}_1 (which make S_1 more special than S_2) do not generate singularities: *Test whether substitutions $Z_2 := \tilde{A}_1(Z_2)$ are regular. Otherwise no merger is found.*

- *Final test: Merger is possible iff $\text{num}(Z_2) \subset I(E_1)$. In that case from \overline{U}_2 expressions are dropped that are zero modulo \overline{E}_2 .*

References

- [1] Sit, William, Y.: An Algorithm for Solving Parametric Linear Systems, J. Symb. Comput. (1992) **13**, 353–394.
- [2] Wolf, T.: Applications of CRACK in the Classification of Integrable Systems, preprint, accepted for publication in the CRM Proceedings, 15 pages (2002).
- [3] Wolf, T., Efimovskaya, O. V.: Classification of integrable quadratic Hamiltonians on $e(3)$, *Regular and Chaotic Dynamics*, **8**, no 2 (2003), p 155-162.
- [4] Sokolov V.V. A new integrable case for the Kirchhoff equation, *Teoret. and Mat. Fiz.*, 2001, **129**, no.1, 31–37, translation in *Theoret. and Math. Phys.*, 2001, **129**, no.1, 1335–1340.