International Journal of Bifurcation and Chaos, Vol. 23, No. 9 (2013) 1350154 (21 pages)

DOI: 10.1142/S021812741350154X

AN APPLICATION OF REGULAR CHAIN THEORY TO THE STUDY OF LIMIT CYCLES

CHANGBO CHEN
Department of Computer Science,
University of Western Ontario, Middlesex College,
London, ON N6A5B7, Canada
cchen252@csd.uwo.ca

ROBERT M. CORLESS
Department of Applied Mathematics,
University of Western Ontario, Middlesex College,
London, ON N6A5B7, Canada
rcorless@uwo.ca

MARC MORENO MAZA
Department of Computer Science,
University of Western Ontario, Middlesex College,
London, ON N6A5B7, Canada
moreno@csd.uwo.ca

PEI YU* and YIMING ZHANG†
Department of Applied Mathematics,
University of Western Ontario, Middlesex College,
London, ON N6A5B7, Canada
**pyu@uwo.ca*
†yzhan688@uwo.ca

Received February 12, 2013

In this paper, the theory of regular chains and a triangular decomposition method relying on modular computations are presented in order to symbolically solve multivariate polynomial systems. Based on the focus values for dynamic systems obtained by using normal form theory, this method is applied to compute the limit cycles bifurcating from Hopf critical points. In particular, a quadratic planar polynomial system is used to demonstrate the solving process and to show how to obtain center conditions. The modular computations based on regular chains are applied to a cubic planar polynomial system to show the computation efficiency of this method, and to obtain all real solutions of nine limit cycles around a singular point. To the authors' best knowledge, this is the first article to simultaneously provide a complete, rigorous proof for the existence of nine limit cycles in a cubic system and all real solutions for these limit cycles.

Keywords: Regular chain; modular algorithm; triangularization; Hilbert's 16th problem; limit cycle; focus value; Maple.

*Author for correspondence

1. Introduction

In the field of dynamical systems, an interesting topic is the study of the number of limit cycles of a given system. For example, Hilbert's 16th problem asks for an upper bound of the number of limit cycles for the system

$$\dot{x} = F(x, y), \quad \dot{y} = G(x, y), \tag{1}$$

where $F(x, y)$ and $G(x, y)$ are degree k polynomials of variables x and y, with real coefficients. The second part of Hilbert's 16th problem is to find the upper bound, called Hilbert number $H(n)$, on the number of limit cycles that system (1) can have. This problem has not been completely solved even for quadratic systems (the case $n = 2$). Although the existence of four limit cycles was proved 30 years ago for quadratic systems [Chen & Wang, 1979; Shi, 1980], despite $H(2) = 4$ being still open. For cubic polynomial systems, many results have been obtained on the low bound of the Hilbert number. So far, the best result for cubic systems is $H(3) \geq 13$ [Li *et al.*, 2009; Li & Liu, 2010; Yang *et al.*, 2010]. This number is believed to be below the maximal number which can be obtained for generic cubic systems. Some recent developments on Hilbert's 16th problem may be found in the review articles [Li, 2003; Leonov, 2008] and the references therein.

In the case of finding small-amplitude limit cycles bifurcating from an elementary center or a focus point based on focus value computation, the problem has been completely solved only for generic quadratic systems [Bautin, 1952], which can have three limit cycles in the vicinity of such a singular point. For cubic systems, James and Lloyd obtained [1991] a formal construction, via symbolic computation, of a special cubic system with eight limit cycles. In 2009, Yu and Corless [2009] showed the existence of nine limit cycles with the help of a numerical method for another special cubic system.

Very recently, Lloyd and Pearson [2012] claimed to be the first to obtain a formal construction, via symbolic computation, of a new cubic system with nine limit cycles. A key step of their derivation is to show that two bivariate polynomials R_1 and R_2 have real solutions. They found that the resultant of R_1 and R_2 had a real solution and then concluded that R_1 and R_2 would have a real common solution. This is not always true. In fact, the existence of a real solution of the resultant of two bivariate polynomials does not necessarily imply the existence of a common real solution for the original two polynomial equations. For example, given $R_1 = y^2 + x + 1$ and $R_2 = y^2 + 2x + 1$ with $x < y$, the resultant of R_1 and R_2 in y is x^2, which has a real solution $x = 0$. However the two equations $R_1 = R_2 = 0$ actually do not have common real solutions. In addition, a similar flawed conclusion was made by the authors when they claimed that the existence of real solutions for $R_1 = R_2 = 0$ implied the existence of real solutions for trivariate polynomial systems $\Psi_1 = \Psi_2 = \Psi_3 = 0$. Therefore, the proof given by Lloyd and Pearson [2012] is not complete.

In the present paper, we formally prove that a specific cubic dynamical system has nine limit cycles. Our strategy is as follows. Given a cubic dynamical system, we deduce the fact that this system has (at least) nine limit cycles by testing whether a given semi-algebraic set is empty or not. This test is based on a symbolic procedure capable of producing an exact representation for each real solution of any system of polynomial equations and inequalities. Once one such real solution has been found, then this procedure can be halted, and non-emptiness has been formally established. Therefore, our approach does not have the flaws of [Lloyd & Pearson, 2012].

The symbolic computation of small limit cycles involves finding the common roots of a nonlinear polynomial system consisting of n focus values $v_0(\gamma_1, \ldots, \gamma_m), \ldots, v_{n-1}(\gamma_1, \ldots, \gamma_m)$, where the variables $\gamma_1, \ldots, \gamma_m$ are the parameters of the original system. With the help of algorithmic and software tools from symbolic computation, we are able to compute nine limit cycles symbolically, using the same system as that used by Yu and Corless [2009]. Unlike the methods used in previous studies which usually depend on good choices of free parameters and the values of dependent parameters, the new method introduces a systematic procedure to symbolically find the maximum number of limit cycles for a given system. It also provides a symbolic proof on the existence of the computed number of limit cycles. In addition, center conditions may be obtained as a by-product.

Symbolic methods for studying and solving nonlinear polynomial systems are of great interest due to their wide range of applications, for example, in theoretical physics, dynamical systems, biochemistry, to name a few. They are very powerful tools that surpass numerical methods by giving exact solutions, whether the number of solutions is finite or not, and by identifying which solutions have real coordinates.

There are two popular families of symbolic methods, based on different algebraic concepts: Gröbner bases [Becker, 1993; Buchberger & Winkler, 1998; Buchberger, 2006], and regular chains [Kalkbrener, 1991; Yang & Zhang, 1991; Moreno Maza, 1999; Aubry *et al.*, 1999; Chen *et al.*, 2007]. Gröbner bases methods have gained much attention during the past four decades due to their simpler algebraic structure: the input polynomial system, say F, is replaced by another polynomial system, say G, such that both F and G have the same solution set and geometrical information (dimension, number of solutions) can easily be read from G.

Methods based on regular chains are relatively new, and have many advantages compared to Gröbner bases methods. For example, they tend to produce much smaller output [Dahan *et al.*, 2012; Chen & Moreno Maza, 2011] in terms of number of monomials and size of coefficients. In addition, regular chain methods can proceed in an incremental manner, that is, by solving one equation after another, against the previously solved equations. This allows for more efficient implementation and makes the processing of inequality constraints much easier. These advantages will be further explained later in this paper.

Given a multivariate polynomial system F in a polynomial ring, for example $\mathbb{Q}[\mathbf{x}]$ over $\mathbb{Q}$, regular chain methods compute the algebraic variety (or zero set — the set of common complex solutions) of F in the form of a list of finitely many polynomial sets. Each of these sets is a polynomial system of triangular shape and with remarkable algebraic properties; for these reasons, it is called a *regular chain*. The algebraic variety of the input system F is given by the union of the common complex roots of the output regular chains. The notion of a regular chain was introduced independently by Kalkbrener [1991] and, by Yang and Zhang [1991] as an enhancement for the notion of a triangular set. Indeed, the regular chain is a special type of triangular set which avoids possible degenerate cases that lead to empty solution [Chen & Moreno Maza, 2011].

One of the main successes of the Computer Algebra community in the last 30 years is the discovery of algorithms, called *modular methods*, that allow to keep the swell of the intermediate expressions under control. Even better: with these methods, almost all intermediate (polynomial or matrix) coefficients fit in a machine word, making these methods competitive in terms of running time with numerical methods. Modular methods have been well developed for solving problems in linear algebra and for computing greatest common divisors (GCDs) of polynomials [Von Zur Gathen & Gerhard, 2003]. They extend the range of accessible problems that can be solved using exact algorithms. In the area of polynomial system solving, the development of those methods is quite recent. They have been applied to Gröbner bases [Trinks, 1984; Arnold, 2003] and primitive element representations [Giusti *et al.*, 1995; Giusti *et al.*, 2001]. Thanks to sharp size estimates [Dahan *et al.*, 2012], the application of modular methods to polynomial system solvers based on regular chains has been very successful in both practice and theory, see [Dahan *et al.*, 2005], opening the door to using fast polynomial arithmetic [Li *et al.*, 2011] and parallelism [Moreno Maza & Pan, 2012] in the implementation of those solvers. The modular method of [Dahan *et al.*, 2005] is available in the `RegularChains` package in Maple.

The rest of the paper is organized as follows. The advantages of incremental solving are further explained in the next section. The theory of regular chains and a modular method for solving polynomial systems by means of regular chains are presented in the third section, together with a number of examples and related Maple commands. The relationship of limit cycles and focus values is presented in the fourth section, with an example of focus value computation using a perturbation method. Then, in the fifth section, the regular chains method is applied to a generic quadratic system to show three small-amplitude limit cycles around the origin and to obtain center conditions. Moreover, with a modular method based on regular chain theory, a special cubic system is presented to show nine small-amplitude limit cycles in the vicinity of the origin.

2. Incremental Solving

The nature of the algebraic problem posed by this application to the study of dynamic systems and, more precisely, the study of limit cycles require that the supporting algebraic tools provide the following specifications and properties.

2.1. *Incremental solving of polynomial systems*

Given a polynomial system of equations, $f_1 = \cdots = f_m = 0$, one would like to solve one equation after another against the previously solved equations. To be more precise, we first choose a format for the solutions. Here we consider regular chains. Thus, we can assume that the common solutions of $f_1, \ldots, f_j$, for $1 \leq j < m$, are given by finitely many regular chains $T_1, \ldots, T_e$. Then the common solutions of $f_1, \ldots, f_{j+1}$ are obtained by taking the union of the regular chains computed by executing a procedure called Intersect and applied to f_{j+1} and $T_1, \ldots, T_e$ successively.

The advantages of this approach are numerous. First of all, from a theoretical point of view, if $\{f_1, \ldots, f_m\}$ is a regular sequence, then incremental solving is known to be a very effective process [Lecerf, 2003; Sommese *et al.*, 2008; Chen & Moreno Maza, 2014; Faugère, 2002].

There are also practical reasons. For instance, information (such as dimension, existence of real solutions) may be extracted before completing the solving of the entire system $f_1 = \cdots = f_m = 0$.

2.2. *Incremental processing of inequality constraints*

Given a component of the solution set of a system of polynomial equations, one would like to extract from that component the points that satisfy an inequality constraint, either of the type $f \neq 0$ or of the type $f > 0$. For example, in the application to limit cycles, one requires the first several focus values vanish, $v_0 = \cdots = v_{n-1} = 0$, but the last one $v_n \neq 0$. Regular chains provide this facility [Chen *et al.*, 2011; Chen & Moreno Maza, 2014]. That is, for a component encoded by one or several regular chains, one can extract the points of that component that satisfy a given inequality constraint. Moreover, the output of this refinement process is again given by a special flavor of regular chains, called regular semi-algebraic systems [Chen *et al.*, 2010]. Therefore, incremental solving can also be used with inequality constraints.

2.3. *Practical efficiency*

With respect to other algebraic tools for describing solution sets of polynomial systems, regular chains have an advantage in terms of size [Dahan, 2009]. In addition, there are sharp size estimates about the representation of the solutions of polynomial systems when this representation is done with regular chains. This is essential in order to design efficient algorithms to compute these representations.

Moreover, these efficient algorithms are able to take advantage of modular techniques. We use a standard example to introduce the principle of those techniques. Consider a square matrix A with integer entries and for which its determinant d is to be computed exactly. It is well-known that using multiprecision rational arithmetic will only solve examples of moderate size due to intermediate expression swell. Let B be a bound on the absolute value of d and let $p_1, \ldots, p_s$ be prime numbers such that their product exceeds $2B$ and each of these primes is of machine word size. One computes the determinant d_i of A modulo the prime number p_i. Then, the determinant d is obtained by applying the Chinese remainder theorem (CRT) to the residues $d_1, \ldots, d_s$

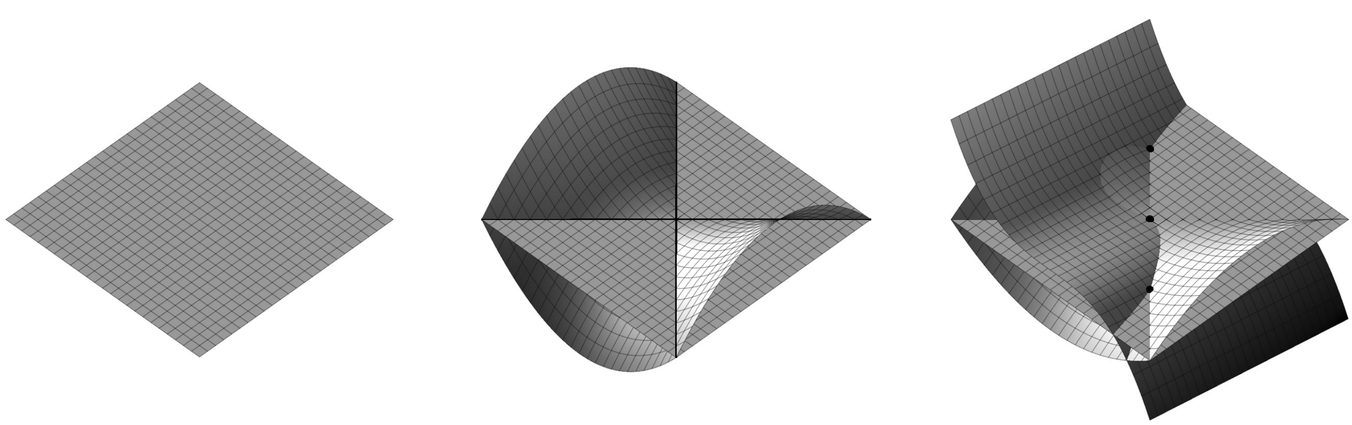

Fig. 1. The incremental solving of (2).

and the moduli $p_1, \ldots, p_s$. This approach not only avoids intermediate expression swell, but it allows for using efficient algorithms over finite fields and efficient implementation techniques in fixed single precision. Last but not least, the complexity of this modular computation process is less than that of the direct approach for computing the determinant of A via Gaussian Elimination (or LU decomposition, etc.) [Gathen & Gerhard, 1999].

The following example is introduced to demonstrate the idea of incremental solving. Given the system

$$F = \begin{cases} x, \\ x + y^2 - z^2, \\ y - z^3, \end{cases} \tag{2}$$

we wish to find the real common roots. The incremental solving algorithm processes one additional equation at a time. So it takes the first equation $x = 0$ and find the real roots, in this case the whole y–z plane (left graph of Fig. 1). In the second step, the next equation $x + y^2 - z^2$ is taken into consideration to obtain the common roots $x = 0, y = \pm z$ (middle graph of Fig. 1). At the last step, $y - z^3$ is added to compute the final answer $\{x = 0, y = 0, z = 0\}, \{x = 0, y = 1, z = 1\}, \{x = 0, y = -1, z = -1\}$ (right graph of Fig. 1).

3. The Regular Chains Method

Similarly to a linear system which can be transformed to a triangular system by Gaussian elimination, a nonlinear polynomial system can be transformed into one or finitely many systems, such that each of them is in a triangular shape. Such a system is called a *triangular set*, in that the main (or leading) variables of different polynomials are distinct. The notion of a triangular set was introduced in [Ritt, 1932; Wu, 1987], with the purpose of representing and computing the set of common zeros of a given polynomial system. Since a triangular set is already in triangular form, it is ready to be solved by evaluating the unknowns one after another using a back-substitution process, as for triangular linear systems. For example, the system

$$F = \begin{cases} x_4^2 - 2x_3 + x_1, \\ x_3^3 + 2x_2, \\ x_2^2 x_1 - 2x_1 + 3, \\ 2x_1^2 + x_1, \end{cases} \tag{3}$$

with ordered variables $x_1 < x_2 < x_2 < x_4$, is a triangular set since the polynomials in it have distinct main variables, which are here x_4, x_3, x_2, x_1, respectively.

The backward solving process of a triangular set could sometimes lead to an empty solution set. In the above example, one solution of the last equation is $x_1 = 0$, which leads to no solution for x_2. To avoid such degenerate cases, the notion of a regular chain was introduced. A regular chain is a type of triangular set which guarantees the success of the backward solving process. Regular chains are constructed by the insight that every algebraic variety is uniquely represented by some generic points of their irreducible components [Aubry *et al.*, 1999]. These generic points are given by certain polynomial sets, called regular chains. The common complex roots of any given multivariate polynomial system can be described by some finite union of regular chains. Such a family of regular chains is called a *triangular decomposition* of the input system.

3.1. *Some definitions and examples for triangular decomposition*

Before demonstrating the regular chains method, some definitions are given, followed by illustrative examples. Throughout this section, let $\mathbb{Q}$ denote the rational number field and $\mathbb{C}$ the complex number field. Let $\mathbb{Q}[\mathbf{x}]$ denote the ring of polynomials over $\mathbb{Q}$, with ordered variables $\mathbf{x} = x_1 < \cdots < x_n$. Let p be a polynomial of the polynomial ring $\mathbb{Q}[\mathbf{x}]$ and let $F \subset \mathbb{Q}[\mathbf{x}]$ be a finite subset. We denote by $V(F)$ the algebraic variety defined by F, that is, the set of points in $\mathbb{C}^n$ which are common solutions of the polynomials of F.

Definition 1. If the polynomial $p \in \mathbb{Q}[\mathbf{x}]$ is not a constant, then the greatest variable appearing in p is called the *main variable* (or *leading variable*) of p, denoted by $\mathrm{mvar}(p)$. Furthermore, the leading coefficient and leading monomial of p, regarded as a univariate polynomial in $\mathrm{mvar}(p)$, are called the *initial* and the *rank* of p, denoted by $\mathrm{init}(p)$ and $\mathrm{rank}(p)$, respectively.

Example 1. Let $p := (x_1 + 1)x_2^2 + 1 \in \mathbb{Q}[x_1, x_2]$, where $x_1 < x_2$. Then, $\mathrm{mvar}(p) = x_2$, $\mathrm{init}(p) = x_1 + 1$ and $\mathrm{rank}(p) = x_2^2$.

Definition 2. Let $T \subset \mathbb{Q}[\mathbf{x}]$ be a *triangular set*, that is, a set of nonconstant polynomials with pairwise distinct main variables. The *quasi-component* of T, denoted by $W(T)$, is the set of points in $\mathbb{C}^n$ which vanish all polynomials in T, but none of the initials of polynomials in T. The minimal algebraic variety containing $W(T)$, denoted by $\overline{W(T)}$, is called the *Zariski closure* of $W(T)$. Note that $\overline{W(T)}$ is a subset of $V(T)$, but may not equal $V(T)$.

Example 2. Consider the polynomial ring $\mathbb{Q}[x, y, z]$, where $x < y < z$. Then, the set $T := \{y - x, yz^2 - x\}$ is a triangular set. The quasi-component $W(T)$ is $\{(x, y, z) \in \mathbb{C}^3 \mid x \neq 0, y = x, z^2 - 1 = 0\}$. The Zariski closure $\overline{W(T)}$ is $\{(x, y, z) \in \mathbb{C}^3 \mid y = x, z^2 - 1 = 0\}$. The variety $V(T)$ is $\{x = 0, y = 0\} \cup W(T)$.

Definition 3. Let T be a triangular set. A polynomial p is said to be *zero* modulo T if $W(T) \subseteq V(p)$ holds. A polynomial p is said to be *regular* modulo T if the dimension of the variety $V(p) \cap \overline{W(T)}$ is strictly less than that of $\overline{W(T)}$.[1]

Example 3. Let $T := \{y - x, yz^2 - x\}$. The polynomial $y - x$ is zero modulo T since we have $W(T) \subseteq V(p)$. On the other hand, the polynomial $z - x$ is regular modulo T since $V(p) \cap \overline{W(T)}$ is the set of points $\{(x, y, z) \in \mathbb{C}^3 \mid x^2 - 1 = 0, y = x, z^2 - 1 = 0\}$, whose dimension is zero, that is, less than the dimension of $\overline{W(T)}$.

Definition 4. A triangular set $T \subset \mathbb{Q}[\mathbf{x}]$ is a *regular chain* if one of the following two conditions holds:

(i) T is empty or consists of a single polynomial;
(ii) $T \backslash \{T_{\max}\}$ is a regular chain, where $T_{\max}$ is the polynomial in T with largest main variable, and the initial of $T_{\max}$ is regular modulo $T \backslash \{T_{\max}\}$.

Example 4. The triangular set $T := \{y - x, yz^2 - x\}$ is a regular chain since $\{y - x\}$ is a regular chain and y is regular modulo $\{y - x\}$.

Definition 5. Let $F \subset \mathbb{Q}[\mathbf{x}]$ be finite, and $\mathfrak{T} := \{T_1, \ldots, T_e\}$ be a finite set of regular chains of $\mathbb{Q}[\mathbf{x}]$. We call $\mathfrak{T}$ a *triangular decomposition* of $V(F)$ if we have $V(F) = \bigcup_{i=1}^{e} W(T_i)$. We denote by Triangularize a function for computing such decompositions.

Example 5. Let $F := \{y - x, yz^2 - x\}$, $T_1 := \{y - x, z^2 - 1\}$ and $T_2 := \{x, y\}$. Then, $\{T_1, T_2\}$ is a triangular decomposition of $V(F)$.

[1] The dimension of the empty set is defined as -1.

The corresponding MAPLE program is as follows:

```
with(RegularChains):
F:=[y-x,y*z^2-x];
R:=PolynomialRing([z,y,x]);
dec:=Triangularize(F,R,output=lazard);
map(Equations, dec, R);
```

which returns,

```
[[z-1, y-x], [z+1, y-x], [y, x]]
```

Definition 6. Let T be a regular chain, and p be a polynomial of $\mathbb{Q}[x]$. Let $\mathfrak{T} := \{T_1, \ldots, T_e\}$ be a finite set of regular chains of $\mathbb{Q}[\mathbf{x}]$. We call $\mathfrak{T}$ a *regular split* of T w.r.t. p if (1) $\overline{W(T)} = \bigcup_{i=1}^{e} \overline{W(T_i)}$ and (2) the polynomial p is either zero or regular modulo T_i, for $i = 1, \ldots, e$. We denote by Regularize a function for computing such decompositions.

Example 6. Let $p := z - 1$ and $T := \{y - x, yz^2 - x\}$. Let $T_1 := \{y - x, z + 1\}$ and $T_2 := \{y - x, z - 1\}$. Then $\{T_1, T_2\}$ is a regular split of T w.r.t. p.

The MAPLE program for this example is given by,

```
with(ChainTools):
p:=z-1;
T := Chain([y-x, y*z^2-x], Empty(R), R);
reg, sing := op(Regularize(p, T, R));
map(Equations, reg, R);
map(Equations, sing, R);
```

which returns,

```
[[z+1, y-x]]
[[z-1, y-x]]
```

3.2. *Triangular decomposition algorithm*

In this section, we illustrate how to obtain a triangular decomposition of an input polynomial system.

Given an input set of polynomials $F = [P_1, \ldots, P_m] \subset \mathbb{Q}[\mathbf{x}]$, we would like to compute a triangular decomposition of $V(F)$, that is, regular chains $T_1, \ldots, T_e \subset \mathbb{Q}[\mathbf{x}]$ such that we have $V(F) = W(T_1) \cup \cdots \cup W(T_e)$. The algorithm presented here works in an incremental manner, that is, by solving one input equation after another, against the solutions of the previously solved equations. The core routine of this algorithm is denoted as Intersect. It takes a regular chain T and a polynomial P as input, and returns regular chains $T_1, \ldots, T_e$, such that we have

$$V(P) \cap W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq V(p) \cap \overline{W(T)}. \quad (4)$$

We choose a polynomial p_1 with minimum rank from F and remove it from F. Then, it is intersected with the empty regular chain, and obtain the regular chain T as p_1 itself. Next, the polynomial p_2 with minimum rank from the remaining F is chosen and removed. Then, p_2 and the regular chain T are the input for Intersect, which returns a list of regular chains $T_1, \ldots, T_e$ that satisfy (4). Further, p_3 with the minimum rank from the remaining input F is intersected with each $T_i, i \in 1, \ldots, e$, and will give more regular chains which also satisfy (4). The algorithm will go on until F is empty. A more detailed description of the algorithm can be found in [Chen & Moreno Maza, 2011].

In order to illustrate this triangular decomposition process, we compute the triangular decomposition of $V(F)$ for the following example. Let $F = [p_1, p_2, p_3]$, where

$$\begin{aligned} p_1 &:= z + y + x^2 - 1, \\ p_2 &:= z + y^2 + x - 1, \\ p_3 &:= z^2 + y + x - 1, \end{aligned} \quad (5)$$

with a order $x < y < z$.

Firstly, p_1 is picked and removed from F as the lowest rank polynomial within the three polynomials, and then is a regular chain $T_0 = p_1$ by definition.

Secondly, p_2 with the lowest rank is chosen from the remaining two polynomials. Now p_2 and T_0 are the input of Intersect, which computes $V(z + y + x^2 - 1, z + y^2 + x - 1)$. The procedure Intersect works as follows. By computing the resultant of $z+y+x^2-1$ and $z+y^2+x-1$, z is eliminated and we obtain a bivariate polynomial $(y-x)(y+x-1)$. Then $T_1 := \{(y-x)(y+x-1), z+y+x^2-1\}$ is a regular chain,[2] with $V(z + y + x^2 - 1, z + y^2 + x - 1) = W(T_1)$. Since the GCD of $z + y + x^2 - 1$ and $z + y^2 + x - 1$ modulo $(y-x)(y+x-1)$ is $z+y+x^2-1$, which is obtained by MAPLE's command RegularGcd. Note that $(y-x)(y+x-1)$ has two factors. By factorizing it,[3] we obtain two regular chains $T_{11} := \{y - x, z + y + x^2 - 1\}$ and $T_{12} := \{y + x - 1, z + y + x^2 - 1\}$ such that we have $V(z + y + x^2 - 1, z + y^2 + x - 1) = W(T_{11}) \cup W(T_{12})$.

In the third step, the variety $V(p_1, p_2, p_3)$ is finally computed. This is equivalent to computing the union of $V(p_3) \cap W(T_{11})$ and $V(p_3) \cap W(T_{12})$.

Let us consider how to compute $V(p_3) \cap W(T_{11})$. To this end, we first compute the resultant of $z^2 + y + x - 1$ and $z + y + x^2 - 1$ and obtain $\mathsf{resultant}(z^2 + y + x - 1, z + y + x^2 - 1, z) = (y + x^2 + x - 1)(y + x^2 - x)$. We then compute the resultant of $(y + x^2 + x - 1)(y + x^2 - x)$ and $y - x$, and obtain $\mathsf{resultant}((y + x^2 + x - 1)(y + x^2 - x), y - x, x) = (x^2 + 2x - 1)x^2$. Since the GCD of $(y + x^2 + x - 1)(y + x^2 - x)$ and $y - x$ mod $(x^2 + 2x - 1)x^2$ is $y - x$, and the GCD of $z^2 + y + x - 1$ and $z + y + x^2 - 1$ mod $\{(x^2 + 2x - 1)x^2, y - x\}$ is $z + y + x^2 - 1$, we know that $V(p_3) \cap W(T_{11})$ is the union of zero sets of $\{x^2 + 2x - 1, y - x, z + y + x^2 - 1\}$ and $\{x, y - x, z + y + x^2 - 1\}$, which could be further simplified as $\{x^2 + 2x - 1, y - x, z - x\}$ and $\{x, y, z - 1\}$.

Similarly, $V(p_3) \cap W(T_{12})$ can be decomposed into a union of zero sets of two regular chains $\{x, y - 1, z\}$ and $\{x - 1, y, z\}$.

To summarize, we have the following triangular decomposition to represent the zero set of F:

$$\begin{cases} z - x = 0 \\ y - x = 0 \\ x^2 + 2x - 1 = 0 \end{cases}, \quad \begin{cases} z = 0 \\ y = 0 \\ x - 1 = 0 \end{cases}, \quad \begin{cases} z = 0 \\ y - 1 = 0, \\ x = 0 \end{cases} \quad \begin{cases} z - 1 = 0 \\ y = 0 \\ x = 0 \end{cases}. \quad (6)$$

3.3. *A method based on modular techniques for computing triangular decomposition*

For challenging input polynomial systems, the method described in the previous section may require vast amounts of computing resources (time

[2]For this particular regular chain, one can check that $W(T_1) = V(T_1)$. But this does not always hold unless the regular chain is zero-dimensional.

[3]Irreducible factorization over $\mathbb{Q}$ is not necessary for computing triangular decomposition. However, factorization often helps to improve the practical efficiency of polynomial system solvers based on triangular decomposition.

and space). This situation can be improved in a spectacular manner by means of so-called *modular techniques*, which, broadly speaking, means computing by homomorphic images instead of computing directly in the original polynomial ring. We present below such an improvement for the case of input zero-dimensional systems whose coefficients are in $\mathbb{Q}$.

Let $F = \{p_1, \ldots, p_n\} \subset \mathbb{Q}[\mathbf{x}]$. Recall that $\mathbf{x}$ stands for n ordered variables $x_1 < \cdots < x_n$. We assume that the variety $V(F)$ is finite and that the Jacobian matrix of F is invertible at any point of $V(F)$. This latter assumption allows the use of Hensel lifting techniques. The algorithm proposed in [Dahan *et al.*, 2005] computes a triangular decomposition of $V(F)$ via the following two-step process:

(1) For some prime number $\wp$, compute a triangular decomposition of $V(F \bmod \wp)$,
(2) Apply Hensel lifting to recover a triangular decomposition of $V(F)$ from that of $V(F \bmod \wp)$.

Some precautions need to be taken before the algorithm produces correct answers. In fact, extraneous factorizations or recombinations could occur when working modulo some "unlucky" prime numbers. Since the same input system F could admit different triangular decompositions, it is possible that a regular chain obtained modulo $\wp$ does not match the modular image of any regular chains in a triangular decomposition $T_1, \ldots, T_e$ of $V(F)$. In [Dahan *et al.*, 2005], the following example is considered. Let $F = [p_1, p_2]$ where $p_1 := 326x_1 - 10x_2^6 + 51x_2^5 + 17x_2^4 + 306x_2^2 + 102x_2 + 34$, $p_2 := x_2^7 + 6x_2^4 + 2x_2^3 + 12$, with $x_1 < x_2$. We have the following triangular decomposition of $V(F)$, that is, over $\mathbb{Q}$:

$$T_1 = \begin{cases} x_1 - 1 = 0, \\ x_2^3 + 6 = 0, \end{cases} \quad T_2 = \begin{cases} x_1^2 + 2 = 0, \\ x_2^2 + x_1 = 0. \end{cases} \tag{7}$$

Computing the regular chains that describe $V(F \bmod 7)$ yields

$$\begin{aligned} t_1 &= \begin{cases} x_2^2 + 6x_2x_1^2 + 2x_2 + x_1 = 0, \\ x_1^3 + 6x_1^2 + 5x_1 + 2 = 0, \end{cases} \\ t_2 &= \begin{cases} x_2 + 6 = 0, \\ x_1 + 6 = 0, \end{cases} \end{aligned} \tag{8}$$

which are not the images of T_1, T_2 modulo 7. In order to overcome this difficulty, the notion of *equiprojectable decomposition* was introduced in [Dahan *et al.*, 2005].

For a given ordering of the coordinates, the equiprojectable decomposition of a zero-dimensional (that is, with finitely many points) variety V is a canonical decomposition of V into components, each of which being the zero set of a regular chain. This notion can be defined as follows. Consider the projection $\pi := V \subset \mathbb{A}^n(\bar{k}) \to \mathbb{A}^{n-1}(\bar{k})$ that forgets the last coordinate, say x. We define $N(\alpha) := \#\pi^{-1}(\pi(\alpha))$, $\alpha \in V$, that is, the number of points that share the same coordinate with α in the x-axis.

The variety V is split into $V_1, \ldots, V_d$ such that each $V_i, i = 1, \ldots, d$, consists of the point $\beta \in V$ such that $N(\beta) = i$. Then, a similar decomposition process is applied to each V_i by considering the second last coordinate. Continuing in this manner yields a partition of $C_1 \cup \cdots \cup C_d = V$, which is a equiprojectable decomposition. The key point is that each equiprojectable component C_j is the zero set of a regular chain T_j, which can be made unique by requiring that each of its initials is equal to one. Together, those regular chains $T_1, \ldots, T_d$ form now a canonical triangular decomposition of V.

In the last example, the triangular decomposition, t_1, t_2 of $V(F \bmod 7)$, is not an equiprojectable decomposition, as shown in the left graph of Fig. 2, since for the points which share the same x_1 coordinate, only the left and middle columns have the same number of points (which is two), while the right column has three points. So the decomposition is rearranged such that the left and middle columns are represented by one regular chain t_2', and the last column is another regular chain t_1' (the right graph of Fig. 2). One can use the MAPLE's procedure EquiprojectableDecomposition to compute the regular chains t_1', t_2' from t_1, t_2, and thus to obtain the equiprojectable decomposition of the input system.

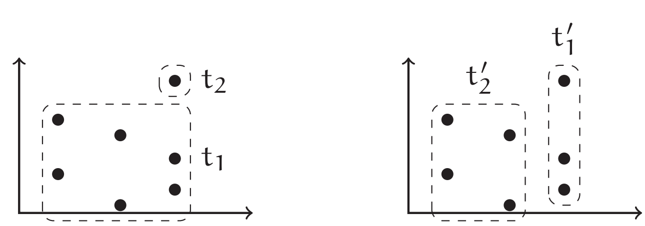

Fig. 2. Equiprojectable decomposition.

$$t_1' = \begin{cases} x_1 - 1 = 0, \\ x_2^3 + 6 = 0, \end{cases} \quad t_2' = \begin{cases} x_1^2 + 2 = 0, \\ x_2^2 + x_1 = 0. \end{cases} \qquad (9)$$

It is obvious that t_1', t_2' are equal to T_1, T_2 mod 7.

Now the modular triangular decomposition will only be lifted after the equiprojectable decomposition is applied. Another key feature of this approach based on modular techniques is the size of the prime number $\wp$. The following theorem provides an approach for selecting good primes so as to avoid unlucky reductions.

Definition 7. The height of a nonzero number $a \in \mathbb{Z}$, is $H(a) := \log(|a|)$. For a rational number $P/Q \in \mathbb{Q}$, $\mathrm{GCD}(P, Q) = 1$, the height is $\max(H(P), H(Q))$. Finally, the height of a polynomial system $F \in \mathbb{Z}[x_1, \ldots, x_m]$ is the maximum height of a nonzero coefficient in a polynomial of F.

Theorem 1 (Theorem 1 in [Dahan *et al.*, 2005]). *Let $F = [p_1, \ldots, p_m] \subset \mathbb{Q}[\mathbf{x}]$ where each polynomial has degree at most d and height at most h, let $T = T_1, \ldots, T_e$ be the equiprojectable decomposition of $V(F)$. There exists an $A \in \mathbb{N} - \{0\}$, with $H(A) \le a(m, d, h)$, and, for $m \ge 2$,*

$$a(m,d,h) = 2m^2 d^{2m+1}(3h + 7\log(m+1) + 5m\log d + 10),$$

such that, if a prime number $\wp$ does not divide A, then $\wp$ cancels none of the denominators of the coefficients of T, and the regular chains $T_1, \ldots, T_e$ reduced mod $\wp$ define the equiprojectable decomposition of $V(F \bmod \wp)$.

Therefore, the set of unlucky primes is finite. Moreover, one can always find a large enough $\wp$ that guarantees the success of the modular algorithm sketched above.

Once the equiprojectable decomposition using some good prime $\wp$ is computed, the result is ready to be lifted in the sense of Hensel lifting. According to Hensel's lemma [Eisenbud, 1995], a simple root r of a polynomial f mod $\wp^k$ can be lifted to root s of f mod $\wp^{k+m}$, which also holds in the multivariate case. Using this lemma, given a polynomial system F, its modular triangular decomposition $t = t_1, \ldots, t_e$ over $V(F \bmod \wp)$ is lifted to $t^k = t_1^k, \ldots, t_e^k$, which is the triangular decomposition of $V(F \bmod \wp^{2k})$ [Schost, 2003]. Then, rational reconstruction is used to recover the regular chains with coefficients in $\mathbb{Q}$.

Here, a probabilistic method is implemented which uses two primes $\wp_1, \wp_2$ that satisfy the condition of Theorem 1. The use of a probabilistic algorithm is a very common technique to compute values modulo primes, and then reconstruct the result to integers or rationals. It is very useful when the deterministic bound is not available or, like in our case, very high. The algorithm usually terminates when the result does not change for several primes. The output could be incorrect, but the probability of such failure is very small and controllable. In Maple many procedures are implemented using, probabilistic algorithms including the commands Determinant, LinearSolve, CharacteristicPolynomial, Eigenvalues, resultant etc.

In our case, the algorithm works as follows.

(1) Compute the equiprojectable triangularizations T and U for $\wp_1$ and $\wp_2$, respectively.
(2) Lift T to $T^k = T_1^k, \ldots, T_e^k$ in $Z(F \bmod \wp_1^{2k})$, where k starts from 1.
(3) T^k is taken as the input of the rational reconstruction to obtain $N^k = N_1^k, \ldots, N_e^k$ over $\mathbb{Q}$.
(4) The algorithm terminates if N^k mod $\wp_2$ equals U, and N^k is returned as the triangular decomposition of F over $\mathbb{Q}$.
(5) Otherwise, k is incremented by 1 and computations resume from Step 2.

Assume that N is the correct equiprojectable triangular decomposition of the input system F. The algorithm fails when N^k mod $\wp_2$ equals U (the modular image of N^k w.r.t $\wp_2$), but $N^k \neq N$. It is also possible that either one of $\wp_1, \wp_2$ divides A or both, so N^k modulo $\wp_2$ may never agree with N modulo $\wp_2$. However, the choices of $\wp_1, \wp_2$ that lead to those bad cases are finite and controllable. See Theorem 2 in [Dahan *et al.*, 2005] for details. In Maple, the Triangularize command offers this modular method. With the option "probability"="prob", the algorithm applies the probabilistic approach using the input probability of success "prob", which controls the size of the prime numbers $\wp_1, \wp_2$.

3.4. *Isolating real roots of a regular chain*

In this section, we briefly review how to obtain the real roots of a regular chain. Let T be a regular chain of $\mathbb{Q}[x_1 < \cdots < x_n]$. A Cartesian product of n intervals is called a box of $\mathbb{Q}[x_1 < \cdots < x_n]$. Let L be a list of boxes. We say L isolates the real roots of T if

- The boxes in L are pairwise disjoint;
- Each real root of T belongs to one element of L;
- Every element of L contains a real root of T.

Example 7. Let $T := \{x^2 - 2, y^2 - x\}$. Then, the Maple output of a real root isolation of T is as follows:

```
 {      19                {             -19
 { y = [--, 5/4]          { y = [-5/4, ---]
 {      16                {              16
[{                      , {                     ]
 {      181  91           {        181  91
 { x = [---, --]          {  x = [---, --]
 {      128  64           {        128  64
```

There are several existing algorithms and implementations [Lu *et al.*, 2005; Xia & Zhang, 2006; Cheng *et al.*, 2007; Boulier *et al.*, 2009] for isolating the real roots of regular chains. However, they all rely on Maple's univariate real root isolation routine, which is not efficient enough for our particular problem. Instead, we adapt a hybrid routine. The univariate polynomial in the regular chain T is isolated by a parallel and cache optimal Collins–Akritas algorithm implemented in Cilk++ [Chen *et al.*, 2012]. The obtained intervals are used to isolate the rest of the polynomials in T by a sleeve-polynomials-like algorithm [Cheng *et al.*, 2007], implemented in Maple.

4. Limit Cycle and Focus Value

In system (1), suppose that $F(x, y)$ and $G(x, y)$ contain m parameters $\gamma_1, \ldots, \gamma_m$, and there is a Hopf critical point at the origin, then the normal form of the system can be written in polar form up to the $(2n + 1)$th order as [Yu, 1998].

$$\frac{dr}{dt} = r(v_0 + v_1 r^2 + v_2 r^4 + \cdots + v_n r^{2n}), \tag{10}$$

$$r\frac{d\theta}{dt} = r\left(1 + \frac{d\phi}{dt}\right) = r(1 + \omega + t_1 r^2 + t_2 r^4 + \cdots + t_n r^{2n}), \tag{11}$$

where each v_k, $k = 0, 1, \ldots, n$ is the kth order focus value of the origin. Note that there are only r^{2k} $(k = 0, 1, \ldots, n)$ terms, since the odd power terms vanish. Each of the focus values v_k is a polynomial of the parameters γ_j $(j = 1, 2, \ldots, m)$ of the original system.

The small-amplitude limit cycles near the origin can be determined from the equation,

$$\frac{dr}{dt} = 0 = r(v_0 + v_1 r^2 + v_2 r^4 + \cdots + v_n r^{2n}), \tag{12}$$

then the right-hand side of Eq. (10) needs to be manipulated such that there are n (and at most n) positive real roots for r^2.

Assuming the first $n+1$ focus values $v_0, v_1, \ldots, v_{n-1}, v_n$ are computed, we will find a combination of parameters such that the first n focus values $v_0, v_1, \ldots, v_{n-1}$ all vanish except v_n. This can generate at most n limit cycles. Then, proper perturbations on the zeros of the n focus values yield n limit cycles. More precisely, a theorem on the relationship between the number of limit cycles and the focus values has been established in [Yu & Chen, 2008], which is given here for convenience.

Theorem 2. *Suppose the origin is an elementary center of (1). If the first n focus values associated with the origin depend on n parameters $\{\gamma_j\}, j = 1, 2, \ldots, n$ such that*

$$v_0 = v_1 = \cdots = v_{n-1} = 0, \quad v_n \neq 0, \tag{13}$$

then there are at most n small-amplitude limit cycles in the vicinity of the origin. Further suppose that $v_k(\Gamma), k = 0, 1, \ldots, n-1, \Gamma = \{\gamma_1, \ldots, \gamma_n\}$, has some positive real solution $\Gamma = C, C = \{c_1, \ldots, c_n\}$ such that $v_k(C) = 0$ and the following condition holds,

$$\det\left[\frac{\partial(v_0, v_1, \ldots, v_{n-1})}{\partial(\gamma_1, \gamma_2, \ldots, \gamma_n)}\right]\Bigg|_{\Gamma=C} \neq 0, \tag{14}$$

then there are exactly n small-amplitude limit cycles around the origin.

Accordingly, in order to compute n small limit cycles near the origin, one needs to find the common roots of a multivariate polynomial system:

$$v_0(\gamma_1, \ldots, \gamma_n) = \cdots = v_{n-1}(\gamma_1, \ldots, \gamma_n) = 0, \tag{15}$$

where the variables $\gamma_1, \ldots, \gamma_n$ are parameters of the original system. Once the common roots of $v_0, \ldots, v_{n-1}$ are computed, the next focus value v_n will be evaluated at these roots. If some of the common roots does not make v_n vanish, then this set of roots will lead to n limit cycles, given their Jacobian to be nonzero. Otherwise, the common roots leading to $v_n = 0$ will be the candidate conditions for the origin to be a center.

There are many commonly used methods to compute focus values, including the perturbation method based on multiple time scales [Yu, 1998, 2001, 2002, 2006; Nayfeh, 1973, 1993], the singular point method [Liu & Li, 1990; Liu & Huang, 2005; Chen & Liu, 2004; Chen *et al.*, 2008], and Poincare–Takens method [Yu & Chen, 2008]. In this article, we apply the perturbation method to compute the focus values.

5. Application to Limit Cycle Computation

In this section, we apply the results presented in previous sections to compute limit cycles bifurcating from an isolated singular point (the origin of the system). Without loss of generality, suppose system (1) has at most n limit cycles. Then the first $n+1$ focus values need to be computed. $v_0, \ldots, v_{n-1}$ are taken as the input for the triangular decomposition and v_n is used to verify if the output regular chains represent limit cycles. Two examples are given in this section. In the first example, we use the general quadratic system (16) to illustrate how to use the regular chains method to find the limit cycle conditions and center conditions, respectively. It is actually a simple case where small limit cycles have already been thoroughly studied [Yu & Corless, 2009] using variable elimination method. The regular chains method computes all the possible common complex roots of the input system, and provides a systematical procedure of analyzing the properties of the outputs. If a regular chain T makes v_n vanish, then it is a candidate of center condition; if v_n does not vanish on T then it is a limit cycle condition. This can be checked by calling the built-in MAPLE procedure Regularize.

In the second example, we follow the work of [Yu & Corless, 2009] on a special cubic system that yields nine limit cycles with the help of numerical computation. Unlike the case of quadratic system, the existence of nine limit cycles for this cubic system has not been confirmed by purely symbolic algorithm. Due to the large input focus value system, the modular method based on regular chain theory is applied.

5.1. *Generic quadratic system*

Consider the general quadratic system [Yu & Corless, 2009], which is the system (23) truncated at third-order terms,

$$\begin{aligned}\dot{x} &= \alpha x + y + x^2 + (b+2d)xy + cy^2,\\ \dot{y} &= -x + \alpha y + dx^2 + (e-2)xy - dy^2,\end{aligned} \tag{16}$$

where α, b, c, d and e are independent parameters. It has been proved [Bautin, 1952] that this system has three small-amplitude limit cycles near the origin. α is set to zero to make the zero-order focus value $v_0 = 0$, then the remaining focus values up to v_4 are obtained using the perturbations method,

$$\begin{aligned}
v_1 &= -\left(\frac{1}{8}\right) b(c+1)\\
v_2 &= -\left(\frac{1}{288}\right)(c+1)(20bc^2 + 19bce - 18bc + 30dce + 18b + 5b^3 + 3be + 56d^2b - 6de^2\\
&\quad - be^2 + 34b^2d + 30de)\\
v_3 &= -\left(\frac{1}{663552}\right)(c+1)(112800dec^2 - 33564bec^2 + 68944b^2dc^2 + 1054be^3c + 10224dc^2e^2 + 151200dce\\
&\quad + 4746be^2c - 52320de^2c + 238080d^3ec - 1400b^2de^2 + 7776dce^3 + 26409be^2c^2 + 104160dc^3e\\
&\quad + 71500bc^3e + 98304bd^2c + 1764bce + 130176bd^2e - 15568bd^2e^2 + 22510b^3ec + 36288b^2de\\
&\quad + 250112bd^2c^2 - 82464b^2dc + 267136bd^2ec + 126464b^2dce + 87156b + 88344bc^2 - 1071be^2\\
&\quad - 30132be + 292608d^2b - 99792de^2 + 142560de + 118800b^2d - 82128bc^3 - 35526b^3c\\
&\quad - 37248d^3e^2 + 27640b^4d + 127536b^3d^2 - 94b^3e^2 + 222208bd^4 - 1968de^4 - 83be^4 + 270208b^2d^3\\
&\quad + 4756bc^4 + 7985b^3c^2 + 1110be^3 + 7014b^3e + 238080d^3e + 24096de^3 + 40176bc + 4473b^3 + 2293b^5),
\end{aligned} \tag{17}$$

$$
\begin{aligned}
v_4 = -\left(\frac{1}{238878720}\right)(c+1)(&258892800d^4be + 82198656b^2c^2d^3 - 204901296b^2c^4d - 56338704b^3c^2d^2 \\
&+ 831702b^3ce^3 + 263761920bc^2d^4 - 119804160bc^4d^2 + 8476608b^3c^2e^2 - 29882016b^4c^2d \\
&+ 18389145b^3c^3e + 3850887b^5ce - 17649bce^5 + 31704606bc^4e^2 - 10436580bc^5e + 7987025bc^3e^3 \\
&+ 742995bc^2e^4 + 344856de^2b^4 + 157049280dec^2 - 7783989eb^3c^2 - 12255624bec^2 - 59918688b^2dc^2 \\
&- 2618973be^3c - 83645568dc^2e^2 + 150426720dce - 3031152e^2bc^3 - 179620608b^2cd^3 + 208343040bd^4c \\
&+ 7007904be^2c - 129060864de^2c + 881619e^3bc^2 - 21854976d^4be^2 + 845184d^2b^3e^2 + 228864000d^5ec \\
&+ 307564800d^3ec^3 - 116280de^4b^2 - 3741696d^3b^2e^2 - 1332000d^2be^4 - 3115008d^3e^2c^2 \\
&+ 19222272d^3e^3c + 31738560de^2c^4 + 65987040dec^5 + 534120de^5c + 10858872de^3c^3 - 911400de^4c^2 \\
&+ 522720000d^3ec - 115056768dcb^4 - 26398440b^2de^2 + 46163304dce^3 + 68285280edc^4 + 3137580be^2c^2 \\
&+ 111913920dc^3e - 301641120b^3cd^2 - 105235956ebc^4 - 1264968bc^3e + 296421120bd^2c - 1935048e^2b^3c \\
&- 427123200dc^3b^2 - 19519380bce + 98286048bd^2e - 96552720bd^2e^2 - 16922112e^2dc^3 \\
&- 325651200bd^2c^3 + 20525499b^3ec - 145720320e^2d^3c + 604638e^4bc + 21063528e^3dc^2 - 7734480e^4dc \\
&+ 35263080b^2de + 218522880bd^2c^2 - 2090880b^2dc + 493843200ed^3c^2 + 341404704bd^2ec \\
&+ 146193336b^2dce - 62052000e^2bd^2c + 342006432ebd^2c^2 + 65831736eb^2dc^2 - 18987024e^2b^2dc \\
&+ 158803248b^3cd^2e + 338098944b^2cd^3e + 38615568b^4cde + 5083512b^2cde^3 + 59250288bc^2d^2e^2 \\
&+ 368805984bc^3d^2e + 393851904bcd^4e + 13093680bcd^2e^3 + 37870296b^2c^2de^2 + 130825704b^2c^3de \\
&+ 4543992b^2de^3 + 162174720b^2d^3e + 69325104b^3d^2e + 17444112b^4de + 21348144d^2be^3 + 59923800b \\
&+ 66397320bc^2 + 15739110be^2 - 48688452be + 299427840d^2b - 123591744de^2 + 102993120de \\
&+ 98507664b^2d + 66713760bc^3 - 27441504b^3c - 201636864d^3e^2 - 23445216b^4d + 16298352b^3d^2 \\
&- 4639752b^3e^2 + 371957760bd^4 - 15410088de^4 + 685611be^4 + 263984256b^2d^3 + 148406760bc^4 \\
&- 30238380b^3c^2 - 3321567be^3 + 134933385b^3e + 336441600d^3e + 61004664de^3 - 59586960bc^5 \\
&- 100956048b^3c^3 - 13151334b^5c - 3304704d^3e^4 + 23561376d^2b^5 + 92370176d^3b^4 + 209773824d^4b^3 \\
&- 103128de^6 + 3281784db^6 + 142458880d^6b + 262901760d^5b^2 - 4431e^4b^3 + 6355e^6b + 30825e^2b^5 \\
&- 29515776d^5e^2 - 41260494b^3c^4 - 3141747b^5c^2 - 75978440bc^6 + 624246e^3b^3 - 103137e^5b \\
&+ 1883415eb^5 + 44268288d^3e^3 + 1981416de^5 + 228864000d^5e + 49163760bc - 5071734b^3 \\
&- 4189203b^5 + 193675b^7). \qquad (18)
\end{aligned}
$$

The existence of three small-amplitude limit cycles requires that the focus values v_0, v_1, v_2 vanish, while $v_3 \neq 0$ [Yu & Chen, 2008]. Since v_0 is already zero, the triangular decomposition of v_1 and v_2 gives the following regular chains.

$$
c+1=0, \quad \begin{cases} d=0, \\ b=0, \end{cases} \quad \begin{cases} e=0, \\ b=0, \end{cases} \quad \begin{cases} e-5c-5=0, \\ b=0. \end{cases} \qquad (19)
$$

Note that these regular chains represent the common roots of v_1 and v_2. They are candidates of center conditions or the conditions for the existence of three limit cycles, depending on whether v_3 vanishes on them or not. In this case, it is easy to check by directly substituting each regular chain into v_3. However, in a more general case with a large input system, regular chains obtained by triangular decomposition are not simple. It cannot be substituted into higher-order focus values. Therefore, two different methods are introduced to verify the properties of the regular chains. The first method involves the triangular decomposition using one or few more higher-order focus values, while the second method uses the Regularize procedure to check whether the input regular chains make the next focus value vanish implicitly.

In the first method, another triangular decomposition using all three focus values v_1, v_2 and v_3 is conducted. The newly generated regular chains are then compared with the ones obtained using only v_1 and v_2. The triangular decomposition of v_1, v_2 and v_3 gives the new regular chains,

$$c+1=0, \quad \begin{cases} d=0, \\ b=0, \end{cases} \quad \begin{cases} e=0, \\ b=0, \end{cases}$$

$$\begin{cases} d^2+2c^2+c=0, \\ e-5c-5=0, \\ b=0. \end{cases} \tag{20}$$

Comparing with the regular chains in (19) generated from v_1 and v_2, the first three regular chains $\{c+1=0\}, \{d=0, b=0\}, \{e=0, b=0\}$ are identical. This indicates that on these three regular chains v_3 vanishes as well, therefore they are center conditions. Now consider the fourth regular chain, d^2+2c^2+c must also be zero in order to make v_3 vanish on $\{e-5c-5=0, b=0\}$. Therefore $\{e-5c-5=0, b=0, d^2+2c^2+c \neq 0\}$ is a condition for the existence of three limit cycles, while $\{e-5c-5=0, b=0, d^2+2c^2+c=0\}$ is a possible center condition.

To further verify the result, one can conduct the triangular decomposition with one additional focus value v_4, which yields,

$$c+1=0, \quad \begin{cases} d=0, \\ b=0, \end{cases} \quad \begin{cases} e=0, \\ b=0, \end{cases}$$

$$\begin{cases} d^2+2c^2+c=0, \\ e-5c-5=0, \\ b=0. \end{cases} \tag{21}$$

These are exactly the same regular chains as that given in (20). So v_4 vanishes on the regular chain $\{e-5c-5=0, b=0, d^2+2c^2+c=0\}$, which confirms that it is a center condition.

The advantage of this method is easy to see how the results are verified. However, the triangular decomposition computation with additional higher-order focus values could be very heavy, and sometimes impossible to compute. Therefore, we introduce another method which is less illustrative but computationally efficient.

The second method uses the built-in MAPLE procedure Regularize. Recall from Example 6, Regularize takes a polynomial p and a regular chain T as input, in this case the polynomial is v_3, and T is chosen from (19). It returns two lists. The first one consists of the regular chain T_r such that p is regular modulo T_r. The second list consists of the regular chain T_z such that p is zero (or singular) modulo T_z. If the first list is empty, then p is zero modulo the input regular chain T, implying that T will make v_3 vanish. If the second list is empty, then p is regular modulo T, which implies that this regular chain will make $p \neq 0$.

After the triangular decomposition of v_1 and v_2 the regular chains in (19) are then used to regularize v_3. The Regularize process shows that for the first three regular chains in (19), the first output list is empty, implying that the first three regular chains make v_3 vanish. For the last regular chain, the second output of the Regularize procedure is empty, indicating that the last regular chain makes $v_3 \neq 0$. One can also use Regularize on v_4 with respect to each regular chain in (19) as well to further verify, which gives exactly the same result as that obtained using the first method. Compared to the first method, the Regularize procedure takes much less time in computation. We shall apply the Regularize method in the next subsection to compute nine limit cycles for a special cubic system.

5.2. *A special cubic system*

A general normalized cubic system with a fixed point at the origin has the form:

$$\begin{aligned}\dot{x} &= a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 \\ &\quad + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3, \\ \dot{y} &= b_{10}x + b_{01}y + b_{20}x^2 + b_{11}xy + b_{02}y^2 \\ &\quad + b_{30}x^3 + b_{21}x^2y + b_{12}xy^2 + b_{03}y^3,\end{aligned} \tag{22}$$

where a_{ij}'s and b_{ij}'s are parameters. According to [Yu & Corless, 2009], the system can be simplified into

$$\begin{aligned}\dot{x} &= \alpha x + y + ax^2 + (b+2d)xy + cy^2 \\ &\quad + fx^3 + gx^2y + (h-3p)xy^2 + ky^3, \\ \dot{y} &= -x + \alpha y + dx^2 + (e-2a)xy - dy^2 + \ell x^3 \\ &\quad + (m-h-3f)x^2y + (n-g)xy^2 + py^3,\end{aligned} \tag{23}$$

where a can be an arbitrary nonzero constant, usually set to $a = 1$ by a proper scaling.

It has been proved [Liu & Li, 1989] that $\alpha = b = d = e = h = n = m = 0$ is a center condition for the origin. In order to find nine limit cycles we need $v_0 = \cdots = v_8 = 0$, but $v_9 \neq 0$. We follow the set-up of [Yu & Corless, 2009] and set the following five parameters to be zero:

$$\alpha = b = d = e = h = 0. \tag{24}$$

By the perturbation method, eight focus values are computed, with v_1 given by

$$\frac{m}{8}, \tag{25}$$

which obviously indicates that $m = 0$ to ensure $v_1 = 0$. With this new condition, the second focus value v_2 becomes

$$-\frac{1}{8fn} + \frac{1}{8pn}. \tag{26}$$

Note that nf is a factor in v_3 and all higher-order focus values. This indicates that either $n = 0$, leading to the center condition [Liu & Li, 1989], or a new candidate condition for center: $\alpha = b = d = e = h = m = f = p = 0$. So, in the following, we assume $nf \neq 0$. Thus, the only choice of making $v_2 = 0$ for the existence of limit cycles is

$$p = f. \tag{27}$$

Under this condition, v_3 has the following form:

$$-\frac{1}{192fn(3n + 15\ell - 30c + 45 - 35c^2 + 15k)}. \tag{28}$$

Since $nf \neq 0$, an easy choice of making v_3 vanish is

$$n = -5\ell + 10c - 15 + \frac{35}{3}c^2 - 5k. \tag{29}$$

Now there are five free parameters,

$$c, k, \ell, f, g, \tag{30}$$

remaining in the five focus values $v_4, v_5, \ldots, v_8$. Using the above results and removing the common factor nf and a constant factor in the resulting focus values we obtain

$$\begin{aligned}v_4 &= 648 - 162c - 516c^2 + 72\ell + 81k + 45g - 30gc - 434c^3 + 60c\ell + 54ck - 168c^4 + 56c^2\ell - 24k^2 \\ &\quad - 6gk - 7c^2g - 6g\ell - 30k\ell - 6\ell^2 + 21kc^2,\end{aligned} \tag{31}$$

$$\begin{aligned}v_5 &= 231336 - 265836c^3k + 37350kc^2\ell + 6174c^2g\ell - 4428gk\ell + 1764c^2gk - 66204gc - 184098c^2\ell \\ &\quad + 40392gk - 133182c^2g + 25002g\ell + 74610k\ell - 361344kc^2 + 270c^2\ell^2 - 14448c^4\ell - 101871kc^4 \\ &\quad - 1944k\ell^2 - 7506k^2\ell + 24165k^2c^2 - 13587c^4g - 1575g^2c^2 - 540g^2\ell - 864f^2k \\ &\quad - 13860f^2c^2 - 864f^2\ell - 540g^2k - 3618gk^2 - 810g\ell^2 - 34296c^3\ell - 156888ck - 135828c\ell \\ &\quad - 4590g^2c + 360c\ell^2 + 40104ck\ell + 6912gc\ell - 3348gck - 41580c^3g - 11880f^2c + 38394ck^2 \\ &\quad - 497556c^2 + 655080c^3 + 548132c^4 + 60525k^2 + 16110\ell^2 + 187306c^5 - 270\ell^3 + 54152c^6 \\ &\quad - 5832k^3 + 6885g^2 + 17820f^2 - 607122c + 115398\ell + 363339k + 131625g,\end{aligned} \tag{32}$$

$$\begin{aligned}v_6 &= 323074872gkc^2\ell + 46434531132c^3k - 10614656412kc^2\ell - 4323518316c^2g\ell + 1747144728gk\ell \\ &\quad - 8537169420c^2gk + 477367776k^2c^2\ell - 242856468kc^2\ell^2 + 512185086kc^4\ell - 4214700gc^2\ell^2 \\ &\quad - 103297626gc^4\ell - 762314922gkc^4 - 34795656gk\ell^2 - 93514176gk^2\ell + 234557856gk^2c^2\end{aligned}$$

$$
\begin{aligned}
&+191130624f^2c^2\ell - 80777088f^2k\ell - 26967924g^2kc^2 - 33543720g^2k\ell + 29212704g^2c^2\ell \\
&- 428849856f^2kc^2 - 189314496f^2gc^2 - 18942336f^2gk - 18942336f^2g\ell + 3496808634c^5k \\
&- 12158345106gc + 494477136gck\ell - 5648392872c^2\ell + 6530829606gk - 8063653761c^2g \\
&+ 2727654102g\ell + 5077228878k\ell - 14369006205kc^2 - 2308784724c^2\ell^2 + 11211047880c^4\ell \\
&+ 26955499191kc^4 + 847752156k\ell^2 + 2178967392k^2\ell - 10546897392k^2c^2 + 11692092699c^4g \\
&- 1454945976g^2c^2 + 324725760g^2\ell + 1155995712f^2k - 3206863872f^2c^2 + 492687360f^2\ell \\
&+ 522334332g^2k + 1571957280gk^2 + 409782564g\ell^2 + 1168019685kc^6 - 20942712k\ell^3 \\
&+ 499013568k^3c^2 - 53343360k^3\ell - 22915872k^2\ell^2 - 1728185544k^2c^4 + 11043864c^2\ell^3 \\
&- 527082024c^6\ell + 144765594c^4\ell^2 - 26181792g^2k^2 - 7361928g^2\ell^2 - 29683332g^2c^4 \\
&- 66407040gk^3 + 512530473gc^6 - 7688520g\ell^3 - 11975040f^2\ell^2 - 287005824f^2c^4 \\
&- 68802048f^2k^2 - 15098076g^3c^2 - 3143448g^3k - 3143448g^3\ell + 617404032ck^3 \\
&- 449534652c^5\ell + 17856001944c^3\ell - 4222690272c^3k^2 - 181543032g^2c^3 + 2704428702gc^5 \\
&- 1408703616f^2c^3 - 31447268118ck - 38578680g^3c - 12575398716c\ell - 909902808g^2c \\
&+ 1301328c^3\ell^2 + 16674336c\ell^3 - 1886860656c\ell^2 - 10011607128ck\ell - 4634256888gck \\
&+ 233217792f^2c\ell + 337929408gck^2 + 1314575496c^3k\ell - 247758264ck\ell^2 + 365912640ck^2\ell \\
&- 561043368gc^3\ell + 150984gc\ell^2 - 2608877592gc^3k - 208987776f^2ck + 40376880g^2c\ell \\
&- 91362168g^2ck - 251475840f^2gc + 14327069940c^3g + 377213760f^2g - 1438591104f^2c \\
&- 7726593888ck^2 + 2289369096 + 11186921988c^2 + 49162023090c^3 - 4045402440c^4 \\
&+ 7440988536k^2 + 1963517274\ell^2 - 46874362782c^5 + 176926680\ell^3 - 19564392796c^6 \\
&+ 1527553728k^3 + 1314588204g^2 + 3474845568f^2 - 369870578c^7 - 47900160k^4 + 176215256c^8 \\
&- 3470040\ell^4 + 57868020g^3 - 17873296866c + 4874228136\ell + 5523913665k + 3624801597g. \qquad (33)
\end{aligned}
$$

The other two polynomials,

$$v_7 = v_7(c, f, g, k, \ell), \quad v_8 = v_8(c, f, g, k, \ell), \qquad (34)$$

with degrees 10 and 12, are too large to be presented here. These five focus values are input to the triangular decomposition algorithm. To simplify the computing process, a better order was generated before the triangular decomposition (by using the built-in MAPLE procedure SuggestVariableOrder),

$$f > g > \ell > k > c. \qquad (35)$$

According to the size of the input system, a sufficiently large prime,

$$\wp := 304166505300000047, \qquad (36)$$

with 2^{58} bits, is chosen to conduct the modular triangular decomposition. Note that the prime chosen here guarantees the success of modular algorithm.

The program was successfully executed to generate seven regular chains. In order to be lifted, they are mapped into two equiprojectable regular chains. The first one is omitted since it contains $f = 0$. The second regular chain is

$$
\begin{cases}
f^2 + Q_1(c) + 109048982804251206, \\
g + Q_2(c) + 213759544982554218, \\
\ell + Q_3(c) + 212357665370487176, \\
k + Q_4(c) + 235643319065695752, \\
Q_5(c) + 249698644301675923,
\end{cases} \qquad (37)
$$

where $Q_1(c), Q_2(c), \ldots, Q_5(c)$ are polynomials in c with order $425, 425, 425, 425$ and 426, respectively.

This regular chain is lifted using the same prime given in (36) to obtain,

$$T = \begin{cases} R_1(c)f^2 + S_1(c) + P_1, \\ R_2(c)g + S_2(c) + P_2, \\ R_3(c)\ell + S_3(c) + P_3, \\ R_4(c)k + S_4(c) + P_4, \\ S_5(c) + P_5, \end{cases} \tag{38}$$

where $R_1(c), \ldots, R_4(c), S_1(c), \ldots, S_5(c)$ are polynomials in c, with order 426 in $S_5(c)$ and 425 in the rest; $P_1, \ldots, P_5$ are big constant terms, and approximately equal to

$$\begin{cases} P_1 \approx 0.9531642255 \cdot 10^{2755}, \\ P_2 \approx 0.6286620222 \cdot 10^{1432}, \\ P_3 \approx 0.6286809511 \cdot 10^{1432}, \\ P_4 \approx -0.2811943803 \cdot 10^{1428}, \\ P_5 \approx -0.1285851059 \cdot 10^{517}. \end{cases} \tag{39}$$

Since these constants are long, only their first 10 digits and their size are presented. In order to check if v_9 vanishes or not on the common roots of T, one can follow the quadratic example, and use Regularize procedure. However, since T is very large, we check this by the following steps instead. Firstly, we compute $T_p = T \bmod \wp$, and check if T_p is a regular chain which turns out to be true. Secondly, we take $v_9 \bmod \wp$ and T_p as the input for Regularize, and find out that $v_9 \bmod \wp$ does not vanish on T_p. According to the specialization property of resultants [Mishra, 1993] (or Theorem 4 in [Chen & Moreno Maza, 2014]), this is a sufficient condition for $v_9 \neq 0$ on T. Therefore, we have found the conditions such that $v_1 = v_2 = \cdots = v_8 = 0$ but $v_9 \neq 0$, indicating that there exist at most nine limit cycles. Note that one requirement during the lifting procedure is for the Jacobian to be nonzero, which satisfies the condition of Theorem 2. This implies that all the positive real roots of the second regular chain lead to nine limit cycles.

By isolating the real roots of the obtained regular chain, we found that it has 78 real roots. The computer outputs of the intervals for the first several ones are shown below:

```
[f = [-11/32, -41/128], g = [-93359084781/1073741824, -186718169557/2147483648],
 l = [1244408533/67108864, 39821073059/2147483648],
 k = [64099524509/68719476736, 128199049023/137438953472],
 c = [-121790475331111530718965725230856466924457099433735144066985116204186867199659306166505138577217\
      441/341757925747345613183203472987128338336432723577064443191526657251555156124902488003673933909\
      85216, -38059523540972353349676789134642645913892843573042232520932848813808395999893533177032855\
      80538045/1067993517960455041197510853084776057301352261178326384973520803911109862890320275011481\
      043468288]]

[f = [41/128, 11/32], g = [-93359084781/1073741824, -186718169557/2147483648],
 l = [1244408533/67108864, 39821073059/2147483648],
 k = [64099524509/68719476736,128199049023/137438953472],
 c = [-121790475331111530718965725230856466924457099433735144066985116204186867199659306166505138577217\
      441/341757925747345613183203472987128338336432723577064443191526657251555156124902488003673933909\
      85216, -38059523540972353349676789134642645913892843573042232520932848813808395999893533177032855\
      80538045/1067993517960455041197510853084776057301352261178326384973520803911109862890320275011481\
      043468288]]

[f = [-19/4, -35/8], g = [-5239003/262144, -83824045/4194304],
 l = [292265139/16777216, 1169060569/67108864],
 k = [-247962889/134217728, -991851547/536870912],
 c = [-115680680925314261355705483664130489454918902845080667457220728245158022295965108754679631530185\
      579177248617/366959778558411441857731343248333910527450398266924979798014214301907660174157569291\
      20296849762010984873984, -14460085115664282669463185458016311181864862855635083432152591030644752\
      786995638594334953941273197397156077/458699723198014302322164179060417388159312997833656224747517\
      7678773845752176969616140037106220251373109248]]
```

...

The total time used for the modular triangular decomposition is 1622615.24 sec (almost 19 days), on a computer with Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83 GHz and 8 G of memory. Isolating the real roots of the regular chain takes about nine hours in Maple on one node of a cluster. The node has four processors, each of which is a 12-core AMD Opteron(tm) 6168 @ 0.8 GHz processor, and total memory of 250 GB.

To illustrate the critical focus values, we take one solution with 1000 significant figures (only the first 50 decimals are printed for convenience):

$$
\begin{aligned}
&\alpha = b = d = e = h = m = 0,\\
&p = f,\\
&n = -5\ell + 10c - 15 + \frac{35}{3}c^2 - 5k,\\
&c = -3.56364742865242710744648501223601521780672396036150\ldots,\\
&f = -0.33257083410940510824128708562052896225706851485676\ldots,\\
&g = -86.94742320093437741980569581134408309860036604648 6\ldots,\\
&l = 18.54313214259950665162503242771432751681531446660 4\ldots,\\
&k = 0.93277084686805751726888595860136166253862306463035\ldots.,
\end{aligned}
$$

which yields the following approximations for critical focus values:

$$
\begin{aligned}
&v_0 = 0\,, \quad v_4 = -0.2628637706 \cdot 10^{-1088}\,, \quad v_9 = 0.9410263940 \cdot 10^{19}\,.\\
&v_1 = 0\,, \quad v_5 = -0.3957953881 \cdot 10^{-1078}\,,\\
&v_2 = 0\,, \quad v_6 = -0.5385553132 \cdot 10^{-1076}\,,\\
&v_3 = 0\,, \quad v_7 = -0.5135260069 \cdot 10^{-1074}\,,\\
& \quad v_8 = -0.4251758871 \cdot 10^{-1072}\,,
\end{aligned}
$$

and the determinant of the Jacobian matrix is $-0.4633625957 \cdot 10^{1259}$. This clearly indicates the existence of nine limit cycles. By increasing the precision used to 2000 digits, the size of $v_4, \ldots, v_8$ is reduced to $\mathcal{O}(10^{-2000})$. These numbers are zero in actuality. By constructing isolating intervals for the real root earlier, this was proved. The numerical computation here merely illustrates the proof.

6. Conclusion

Quantitative analysis of polynomial dynamical systems, such as determining the number of small-amplitude limit cycles around the origin, naturally leads to solve systems of multivariate polynomial equations and inequalities. Proving formally that such a semi-algebraic system is consistent, and, if it is, computing all its solutions or a sample of them, are goals that make the use of symbolic and exact methods desirable.

In this paper, we have demonstrated that the theory of regular chains possesses powerful algorithmic tools to achieve those goals. We have applied to large input focus value systems an algorithm for computing triangular decompositions of polynomial systems via modular techniques. From these calculations, we have obtained conditions for the existence of limit cycles and potential center conditions. One example, in particular, exhibiting nine limit cycles shows the computational power and efficiency of these tools from regular chain theory.

These tools, available in the RegularChains library in MAPLE can be applied to solve other polynomial systems arising from real physical or engineering systems.

Acknowledgments

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC), and ADF from the University of Western Ontario.

References

Arnold, E. [2003] "Modular algorithms for computing Gröbner bases," *J. Symb. Comput.* **35**, 403–419.

Aubry, P., Lazard, D. & Moreno Maza, M. [1999] "On the theories of triangular sets," *J. Symb. Comput.* **28**, 105–124.

Bautin, N. [1952] "On the number of limit cycles appearing with variation of the coefficients from an equilibrium state of the type of a focus or a center," *Matematicheskii Sbornik* **72**, 181–196.

Becker, T. [1993] "Gröbner basics," *Physics World*, p. 20.

Boulier, F., Chen, C., Lemaire, F. & Moreno Maza, M. [2009] "Real root isolation of regular chains," *Proc. ASCM'09*, pp. 15–29.

Buchberger, B. & Winkler, F. [1998] *Gröbner Bases and Applications*, Vol. 251 (Cambridge University Press).

Buchberger, B. [2006] "Bruno Buchberger Ph.D. thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal," *J. Symb. Comput.* **41**, 475–511.

Chen, L. S. & Wang, M. S. [1979] "The relative position, and the number, of limit cycles of a quadratic differential system," *Acta. Math. Sin.* **22**, 751–758.

Chen, H. & Liu, Y. [2004] "Linear recursion formulas of quantities of singular point and applications," *Appl. Math. Comput.* **148**, 163–171.

Chen, C., Golubitsky, O., Lemaire, F., Moreno Maza, M. & Pan, W. [2007] "Comprehensive triangular decomposition," *Computer Algebra in Scientific Computing* (Springer), pp. 73–101.

Chen, H., Liu, Y. & Yu, P. [2008] "Center and isochronous center at infinity in a class of planar systems," *Dyn. Contin., Discr. Impul. Syst. Series B: Appl. Algor.* **15**, 57–73.

Chen, C., Davenport, J. H., May, J., Moreno Maza, M., Xia, B. & Xiao, R. [2010] "Triangular decomposition of semi-algebraic systems," *Proc. ISSAC' 2010*, pp. 187–194.

Chen, C. & Moreno Maza, M. [2011] "Algorithms for computing triangular decompositions of polynomial systems," *Proc. 36th Int. Symp. Symbolic and Algebraic Computation*, ISSAC'11 (ACM, NY, USA), pp. 83–90.

Chen, C., Davenport, J. H., Moreno Maza, M., Xia, B. & Xiao, R. [2011] "Computing with semi-algebraic sets represented by triangular decomposition," *Proc. ISSAC'11*, pp. 75–82.

Chen, C., Moreno Maza, M. & Xie, Y. [2012] "Cache complexity and multicore implementation for univariate real root isolation," *J. Phys.: Conf. Series* **341**, 012026.

Chen, C. & Moreno Maza, M. [2014] "An incremental algorithm for computing cylindrical algebraic decomposition," *Lecture Notes in Artifical Intelligence* (Springer).

Cheng, J., Gao, X. & Yap, C. [2007] "Complete numerical isolation of real zeros in zero-dimensional triangular systems," *ISSAC '07: Proc. 2007 Int. Symp. Symbolic and Algebraic Computation*, pp. 92–99.

Dahan, X., Moreno Maza, M., Schost, E., Wu, W. & Xie, Y. [2005] "Lifting techniques for triangular decompositions," *Proc. 2005 Int. Symp. Symbolic and Algebraic Computation* (ACM), pp. 108–115.

Dahan, X. [2009] "Size of coefficients of lexicographical Gröbner bases: The zero-dimensional, radical and bivariate case," *ISSAC*, pp. 119–126.

Dahan, X., Kadri, A. & Schost, É. [2012] "Bit-size estimates for triangular sets in positive dimension," *J. Compl.* **28**, 109–135.

Eisenbud, D. [1995] *Commutative Algebra*, Graduate Texts in Mathematics, Vol. 50 (Springer-Verlag, NY).

Faugère, J. C. [2002] "A new efficient algorithm for computing Gröbner bases without reduction to zero (f5)," *Proc. 2002 Int. Symp. Symbolic and Algebraic Computation*, ISSAC'02 (ACM, NY, USA), pp. 75–83.

Gathen, J. & Gerhard, J. [1999] *Modern Computer Algebra* (Cambridge University Press).

Giusti, M., Heintz, J., Morais, J. & Pardo, L. [1995] "When polynomial equation systems can be solved fast?" *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pp. 205–231.

Giusti, M., Lecerf, G. & Salvy, B. [2001] "A Gröbner free alternative for polynomial system solving," *J. Compl.* **17**, 154–211.

James, E. & Lloyd, N. [1991] "A cubic system with eight small-amplitude limit cycles," *IMA J. Appl. Math.* **47**, 163–171.

Kalkbrener, M. [1991] "Three contributions to elimination theory," PhD thesis, Johannes Kepler University, Linz.

Lecerf, G. [2003] "Computing the equidimensional decomposition of an algebraic closed set by means of lifting fibers," *J. Compl.* **19**, 564–596.

Leonov, G. A. [2008] "Hilbert's 16th problem for quadratic systems: New methods based on a transformation to the Lienard equation," *Int. J. Bifurcation and Chaos* **18**, 877–884.

Li, J. [2003] "Hilbert's 16th problem and bifurcations of planar polynomial vector fields," *Int. J. Bifurcation and Chaos* **13**, 47–106.

Li, C., Liu, C. & Yang, J. [2009] "A cubic system with thirteen limit cycles," *J. Diff. Eqs.* **246**, 3609–3619.

Li, J. & Liu, Y. [2010] "New results on the study of Z_q-equivariant planar polynomial vector fields," *Qualit. Th. Dyn. Syst.* **9**, 167–219.

Li, X., Moreno Maza, M., Rasheed, R. & Schost, É. [2011] "The modpn library: Bringing fast polynomial arithmetic into maple," *J. Symb. Comput.* **46**, 841–858.

Liu, Y. & Li, J. [1989] "On the singularity values of complex autonomous differential systems," *Sci. China* (*Ser. A*) **3**, 245–255.

Liu, Y. & Li, J. [1990] "Theory of values of singular point in complex autonomous differential system," *Sci. China (Ser. A)* **33**, 10–23.

Liu, Y. & Huang, W. [2005] "A cubic system with twelve small amplitude limit cycles," *Bull. des Sci. Math.* **129**, 83–98.

Lloyd, N. & Pearson, J. [2012] "A cubic differential system with nine limit cycles," *J. Appl. Anal. Comput.* **2**, 293–304.

Lu, Z., He, B., Luo, Y. & Pan, L. [2005] "An algorithm of real root isolation for polynomial systems," *Proc. Symbolic Numeric Computation 2005*, eds. Wang, D. & Zhi, L. (Birkhäuser, Basel), pp. 94–107.

Mishra, B. [1993] *Algorithmic Algebra* (Springer-Verlag, NY).

Moreno Maza, M. [1999] "On triangular decompositions of algebraic varieties," Tech. Rep., TR 4/99, NAG Ltd, Oxford, UK.

Moreno Maza, M. & Pan, W. [2012] "Solving bivariate polynomial systems on a GPU," *J. Phys.: Conf. Series* (IOP Publishing), p. 012022.

Nayfeh, A. [1973] *Perturbation Methods* (Wiley Online Library).

Nayfeh, A. [1993] *The Method of Normal Forms* (Wiley-VCH).

Ritt, J. F. [1932] *Differential Equations from an Algebraic Standpoint*, Vol. 14 (American Mathematical Society, NY).

Schost, É. [2003] "Complexity results for triangular sets," *J. Symb. Comput.* **36**, 555–594.

Shi, S. [1980] "A concrete example of the existence of four limit cycles for plane quadratic systems," *Sci. Sin.* **23**, 153–158.

Sommese, A., Verschelde, J. & Wampler, C. [2008] "Solving polynomial systems equation by equation," *Algorithms in Algebraic Geometry*, pp. 133–152.

Trinks, W. [1984] "On improving approximate results of Buchberger's algorithm by Newton's method," *ACM SIGSAM Bull.* **18**, 7–11.

Von Zur Gathen, J. & Gerhard, J. [2003] *Modern Computer Algebra* (Cambridge University Press).

Wu, W. T. [1987] "A zero structure theorem for polynomial equations solving," *MM Res. Preprints* **1**, 2–12.

Xia, B. & Zhang, T. [2006] "Real solution isolation using interval arithmetic," *Comput. Math. Appl.* **52**, 853–860.

Yang, L. & Zhang, J. [1991] "Searching dependency between algebraic equations: An algorithm applied to automated reasoning," Tech. Rep. IC/89/263, International Atomic Energy Agency, Miramare, Trieste, Italy.

Yang, J., Han, M., Li, J. & Yu, P. [2010] "Existence conditions of thirteen limit cycles in a cubic system," *Int. J. Bifurcation and Chaos* **20**, 2569–2577.

Yu, P. [1998] "Computation of normal forms via a perturbation technique," *J. Sound Vibr.* **211**, 19–38.

Yu, P. [2001] "Symbolic computation of normal forms for resonant double Hopf bifurcations using a perturbation technique," *J. Sound Vibr.* **247**, 615–632.

Yu, P. [2002] "Analysis on double Hopf bifurcation using computer algebra with the aid of multiple scales," *Nonlin. Dyn.* **27**, 19–53.

Yu, P. [2006] "Computation of limit cycles — the second part of Hilbert's 16th problem," *Fields Instit. Commun.* **49**, 151–177.

Yu, P. & Chen, G. [2008] "Computation of focus values with applications," *Nonlin. Dyn.* **51**, 409–427.

Yu, P. & Corless, R. [2009] "Symbolic computation of limit cycles associated with Hilbert's 16th problem," *Commun. Nonlin. Sci. Numer. Simul.* **14**, 4041–4056.

Appendix A

Maple Input for the Quadratic Example

```
read "focusvalues_quadric": # Read in the focus values
eqs := [v2, v3];
vars := SuggestVariableOrder(eqs); # Suggest a best order for the variables
R := PolynomialRing(vars);  # Construct the polynomial ring
dec := Triangularize(eqs, R, output=lazard);
# Compute the triangular decomposition
Info(dec, R);
# Display the output which contains four regular chains,
# [[c+1], [d, b], [e-5*c-5, b], [e, b]];

# Now we check if $v4$ vanishes on each of the regular chains
#Method1: using Regularize.
```

```
Regularize(v4, dec[1], R);
# [[], [regular_chain]]
# This output shows that v4 vanishes on zeros of dec[1];
# This is equivalent to say that dec[1] is a center condition.

Regularize(v4, dec[2], R);
# Same as above

Regularize(v4, dec[4], R);
# Same as above

Regularize(v4, dec[3], R);
# The output is [[regular_chain], []],
# which says that v4 does not vanish on all the zeros of dec[3]

# Method2:
dec2 := Triangularize([v2, v3, v4], R, output=lazard);
Info(dec2, R);
# [[c+1], [d, b], [e, b], [d^2+2*c^2+c, e-5*c-5, b]]
# According the result from dec (v2, v3 only),
# [c+1], [d, b], [e, b] are center conditions, since v4 vanishes on them.
# d^2+2*c^2+c must be zero in order to make v4 vanishes at [e-5*c-5, b].
# Thus, [e-5*c-5=0, b=0], but d^2+2*c^2+c<>0 is condition for limit cycle.

dec3 := Triangularize([v2, v3, v4, v5], R, output=lazard);
Info(dec3, R);
# [[c+1], [d, b], [e, b], [d^2+2*c^2+c, e-5*c-5, b]]
# By dec2, all the components from dec2 makes v5 vanishes,
# which means [d^2+2*c^2+c, e-5*c-5, b] is a new center condition.
```

Appendix B

Maple Input for the Cubic Example

```
read "focusvalues_cubic";
with(RegularChains);
F:= [F1, F2, F3, F4, F5];
R:= PolynomialRing[vars];  # Construct the polynomial ring
vars:= SuggestVariableOrder(F);  # Suggest a best order for the variables
p := 304166505300000047;  # Pick a large enough prime
Rp := PolynomialRing(vars, p);  # Construct the polynomial ring mod p
dec := Triangularize(F, Rp);  # Compute the triangular decomposition modulo p
map(NumberOfSolutions, dec, Rp);

# Check the number of solutions of each output regular chain
# [474, 214, 112, 34, 18, 1, 1]

ndec := [seq(op(NormalizeRegularChain(rc, Rp, 'normalized'='strongly')), rc=dec)];
# Normalize each regular chain
edec := [op(EquiprojectableDecomposition(ndec, Rp))];
# Compute the equiprojectable decomposition, which contains two regular chains
# edec[1], edec[2]
```

```
with(MatrixTools);
jm1 := JacobianMatrix(F, edec[1], Rp);  # Jacobian of edec[1]
MatrixTools:-MatrixInverse(jm1, edec[1], Rp);
# Check if the Jacobian is invertable, which returns false
jm2 := JacobianMatrix(F, edec[2], Rp);  # Jacobian of edec[2]
MatrixTools:-MatrixInverse(jm2, edec[2], Rp);

# The Jacobian of edec[1] is zero
Equation(edec[1],Rp); # Show the equations in edec[1], which contains f=0
# This is a known center condition

# The Jacobian of edec[2] is non-zero
Lift(F, R, edec[2], 10, p); # Lift the edec[2]
eqn0 := Equations(dec, Rp); # Extract the equations from edec[2]
#check if the five equations is  initial is 0 mod p
expand(Initial(eqn0[1], R)) mod p;
expand(Initial(eqn0[2], R)) mod p;
expand(Initial(eqn0[3], R)) mod p;
expand(Initial(eqn0[4], R)) mod p;
expand(Initial(eqn0[5], R)) mod p;

#check if still a regular chain mod p;
eqp := map(x->expand(x) mod p, eq0);

rc := Empty(Rp);
rc := Chain(eqp[5..-1], rc, Rp); # Reconstruct the regular chain mod p
Regularize(Initial(eqp[4], Rp), rc, Rp);
# [[regular_chain], []]
rc := Chain(ListTools:-Reverse(eqp[4..-1]), Empty(Rp), Rp);
Regularize(Initial(eqp[3], Rp), rc, Rp);
# [[regular_chain], []]
rc := Chain(ListTools:-Reverse(eqp[3..-1]), Empty(Rp), Rp);
Regularize(Initial(eqp[2], Rp), rc, Rp);
# [[regular_chain], []]
rc := Chain(ListTools:-Reverse(eqp[2..-1]), Empty(Rp), Rp);
Regularize(Initial(eqp[1], Rp), rc, Rp);
# [[regular_chain], []]
rc := Chain(ListTools:-Reverse(eqp), Empty(Rp), Rp);
# It turns out that it is still a regular chains mod p

read "v9":  # Read the next focus value v9
Regularize(v9, rc, Rp);  # Check if the regular chain makes v9 vanish
#[[regular_chain], []]
# v9 does not vanish on the regular chain, so the eq0 deals to limit cycles.
```