REDUCE and the Bifurcation of Limit Cycles

N. G. LLOYD AND J. M. PEARSON

Department of Mathematics, The University College of Wales, Aberystwyth, Dyfed SY23 3BZ, UK

(Received 8 November 1988)

A technique is described which has been used extensively to investigate the bifurcation of limit cycles in polynomial differential systems. Its implementation requires a Computer Algebra System, in this case REDUCE. Concentration is on the computational aspects of the work, and a brief resume is given of some of the results which have been obtained.

1. Introduction

We describe how the availability of symbolic manipulation procedures has recently led to significant progress in one of the famous problems in the theory of nonlinear ordinary differential equations.

Consider systems of the form

\[ \dot{x} = P(x, y), \quad \dot{y} = Q(x, y), \]  

(1.1)

in which \( P \) and \( Q \) are polynomials. Let \( \pi(P, Q) \) be the number of limit cycles of (1.1) and define

\[ H_n = \sup \{ \pi(P, Q) \mid \partial P, \partial Q \leq n \}. \]

The question of interest is the maximum possible number of limit cycles, and estimates are sought for \( H_n \) in terms of \( n \). This is part of the sixteenth of the celebrated list of twenty-three problems posed by Hilbert in 1900 and is the one on which least progress has been made. Since it is, in fact, a whole area of the subject, it is rather misleading to think of it as a single problem; its history and present status are described in detail in the survey paper (Lloyd, 1988a).

Recent research has proceeded by considering various classes of systems which are of independent interest. The maximum number of limit cycles is sought and their possible configurations investigated. Much of the recent progress has been achieved by consideration of various kinds of bifurcation. The work of the group working at The University College of Wales, Aberystwyth, which we describe here, has concentrated on limit cycles which bifurcate out of a critical point—so-called small-amplitude limit cycles.

We have made extensive use of the REDUCE computer algebra system, and the availability of such procedures has not only made these developments possible but has done much to stimulate interest in the problem. Symbolic manipulation techniques are also being used in the investigation of homoclinic bifurcation, which is becoming increasingly significant [see, for instance, Guckenheimer, Rand & Schlomiuk (1989); Joyal & Rousseau (1989) and Li & Rousseau (1989)].
Very briefly, the position is that remarkably little is known about the $H_n$: it has not even been established that they are finite, and it is only very recently that it has been proved that a given polynomial system cannot have infinitely many limit cycles (Ecalle et al. 1987a, b). The first major contribution was that of Bautin (1952), who proved that $H_2 \geq 3$. He did this by showing that, for quadratic systems, three limit cycles can bifurcate out of a critical point, and his ideas have been very influential in the development of the subject. Soon afterwards, Landis & Petrovskii published two papers (1955, 1957), in one of which it was suggested that $H_2 = 3$ and in the other precise bounds were given for $H_n$ with $n \geq 3$. However, the proofs of these results were soon withdrawn (Landis & Petrovskii, 1959), but nevertheless it appears to have been widely believed that $H_2 = 3$. It was not until 1979 that the first examples of quadratic systems with at least four limit cycles appeared [see Shi (1980), Chen & Wang (1979) and Blows & Lloyd (1984a), for instance]. These developments stimulated renewed interest in Hilbert's problem and the work which we describe in this paper is part of that response. We have concentrated on the generation of limit cycles by bifurcation from a critical point and have considered a variety of classes of systems, some of which we shall describe in section 5. Cubic systems and systems of Liénard type are two particular cases which we have studied extensively.

In section 2 we explain the basic idea of the bifurcation of limit cycles out of critical points, and in section 3 we explain the algorithmic procedure for computing the focal values which enables us to describe differential systems with several limit cycles. The implementation of this algorithm is described in section 4. In section 5 we summarise some of the results which we have obtained. Throughout we concentrate on the computational aspects; the detailed mathematical results are presented in a series of papers [Blows & Lloyd (1984a, b), Lloyd (1988b), Lloyd & Lynch (1988), Lloyd, Blows & Kalenge (1988), Basarab-Horwath & Lloyd (1988), Lynch (1989)] as well as in Lloyd (1988a).

2. Small-Amplitude Limit Cycles

We consider systems in which the origin is a critical point of focus type, and show how to bifurcate limit cycles out of it. Thus we investigate systems of the form

$$\dot{x} = \lambda x + y + p(x, y), \quad \dot{y} = -x + \lambda y + q(x, y),$$

and write

$$p(x, y) = p_2(x, y) + \ldots + p_n(x, y),$$
$$q(x, y) = q_2(x, y) + \ldots + q_n(x, y),$$

where $p_k$ and $q_k$ are homogeneous polynomials of degree $k$. The linear part is in canonical form, and the stability of the origin is determined by the sign of $\lambda$. If $\lambda = 0$, the origin is a centre for the linearised system, and is said to be a fine focus of the nonlinear system. The idea is to perturb the coefficients arising in the $p_k$ and $q_k$, so that limit cycles bifurcate out of the origin. Such limit cycles are said to be of small amplitude.

To maximise the number of limit cycles which can bifurcate, we start with a fine focus which is as close to being a centre for the nonlinear system as possible. This is formally encapsulated in the concept of the order of the fine focus, which we proceed to explain.

There is a function $V$ defined in a neighbourhood of the origin such that its rate of change along orbits is of the form

$$V = \eta_2 r^2 + \eta_4 r^4 + \ldots \quad (r^2 = x^2 + y^2).$$
The coefficients in $\eta_{2k}$ are the focal values and they are polynomials in the coefficients in $p$ and $q$. The origin is said to be a fine focus of order $k$ if $\eta_2 = \eta_4 = \ldots = \eta_{2k} = 0$ but $\eta_{2k+2} \neq 0$. It can be shown that, at most, $k$ limit cycles bifurcate out of a fine focus of order $k$ [see Blows & Lloyd (1984a)]. Clearly, the stability of the origin is determined by the sign of the first non-vanishing focal value, and the origin is a nonlinear centre if and only if all the focal values are zero.

Suppose that the origin is a fine focus of order $k$. The first step is to perturb the coefficients in $p$ and $q$ so that $\eta_{2k} \neq 0$, $\eta_{2i} = 0$ for $i < k$ and $\eta_{2k} \eta_{2k+2} < 0$; if this can be achieved, the stability of the origin is reversed, and a limit cycle ($F_1$, say) bifurcates. Next, further perturbations are introduced so that $\eta_{2k-2} \eta_{2k} < 0$ and $\eta_{2i} = 0$ for $1 \leq i \leq k-2$. The stability of the origin is again reversed, and another limit cycle appears. Provided that $\eta_{2k-2}$ is small enough, $F_1$ persists, and there are therefore two limit cycles. Proceeding in this way, $k$ limit cycles bifurcate provided perturbations can be so arranged that $\eta_{2i} \eta_{2i+2} < 0$ for $1 \leq i \leq k$. Though in many cases the full complement of $k$ limit cycles is attained, this is not necessarily so, and we shall give an example later in which it is not.

Since it is the first non-zero focal value that is of significance, what we really need are the so-called Liapunov quantities $L(0), L(1), \ldots$. These are the non-zero expressions obtained by calculating each $\eta_{2k}$ under the conditions $\eta_2 = \ldots = \eta_{2k-2} = 0$. In general, $L(k)$ is derived from $\eta_{2k+2}$, but it can happen that a reduced focal value is necessarily zero, in which case it does not contribute a Liapunov quantity. In our reduction of $\eta_{2k}$ we use substitutions from the relations $\eta_2 = \eta_4 = \ldots = \eta_{2k-2} = 0$ which involve rational functions of the coefficients arising in $p$ and $q$. This contrasts with the formal calculation of a basis for the ideal of polynomials generated by the focal values: in that case all substitutions are polynomial. Much valuable work has been done on the use of Computer Algebra to determine a basis of an ideal of polynomials; in particular, Wang (1988) has applied Buchberger's Gröbner basis method to the calculation of Liapunov quantities.

For a given class $\mathcal{C}$ of systems, the aim is to maximise the number of limit cycles which can bifurcate from the origin. Thus, it is necessary to find $k_{\text{max}}(\mathcal{C})$, the maximum possible order of a fine focus for systems in $\mathcal{C}$; $k_{\text{max}}$ is characterised by the fact that the origin is a centre if $\eta_{2k} = 0$ for $k \leq 1 + k_{\text{max}}$, but not if any of these focal values is non-zero. In practice, one proceeds with the computation of focal values until it appears that $k_{\text{max}}$ has been reached; it is then necessary to prove independently that the origin is a centre. This is often difficult, and developing criteria for the existence of a centre is a significant and substantive problem in its own right.

To summarise, for any given class of systems, there are four phases to the procedure.

1. Calculation of the focal values.
2. Reduction of the focal values to obtain the Liapunov quantities.
3. Establishing the value of $k_{\text{max}}$ by proving that the origin is a centre if $\eta_{2k} = 0$ for $k \leq k_{\text{max}} + 1$.
4. Commencing with a fine focus of maximal order, finding a sequence of perturbations each of which reverses the stability of the origin.

In the calculation of the focal values, very large expressions arise, and it soon becomes apparent that one cannot even contemplate completing the computation by hand in any but the simplest situations. It is here that Computer Algebra systems have proved so valuable. Calculation of the focal values now becomes a realistic possibility, and much of the recent progress has been achieved because of the availability of such systems. Indeed, the first two steps noted in the previous paragraph can be automated.
3. The Calculation of Liapunov Quantities

To compute the focal values, we follow a classical procedure [see Nemytskii & Stepanov (1960)] and write

\[ V(x, y) = V_2(x, y) + V_3(x, y) + \ldots, \]

where \( V_k \) is a homogeneous polynomial of degree \( k \): let

\[ V_k = \sum_{i=0}^{k} V_{k-l} x^{k-l} y^l. \]

For convenience, we say that \( V_{k,i} \) is an even or odd coefficient according to whether \( i \) is even or odd. Let \( D_k \) denote the terms of degree \( k \) in \( \dot{V} \); we seek the \( \eta_k \) and \( V_{ij} \) so that \( D_k = 0 \) if \( k \) is odd and \( D_k = \eta_k (x^2 + y^2)^{k/2} \) if \( k \) is even.

By direct substitution

\[ D_k = y(V_k)_x - x(V_k)_y + R_k(x, y), \tag{3.1} \]

where

\[ R_k(x, y) = (V_{k-l})_x p_2 + (V_{k-l})_y q_2 + \ldots + xp_{k-1} + yq_{k-1}, \]

and the subscripts \( x \) and \( y \) denote partial differentiation with respect to \( x \) and \( y \), respectively. Note that \( R_k \) is determined by the \( V_i, p_i \) and \( q_i \) with \( i < k \); in particular, \( R_k \) is independent of \( V_k \).

Suppose first that \( k \) is odd (\( k = 2m + 1 \), say). Since \( D_k = 0 \) if and only if the coefficient of each monomial \( x^i y^{k-i} \) in \( D_k \) is zero, we have \( 2m + 2 \) linear equations to determine the \( 2m + 2 \) unknowns \( V_{i,j} \) in terms of the \( V_{i,j} \) with \( i+j < k \) and the coefficients arising in the original differential equations. It is easily confirmed that these \( 2m + 2 \) equations decouple into two sets of \( m + 1 \) linear equations, one set determining the odd coefficients of \( V_k \) and the other determining the even coefficients. The corresponding matrices of coefficients are, respectively,

\[
\begin{pmatrix}
  k & -2 & \cdots & 0 \\
  k-2 & -4 & \cdots & 0 \\
  \vdots & \ddots & \ddots & \vdots \\
  0 & \cdots & 3 & -(k-1) \\
 -1 & k-1 & -3 & 0 \\
 0 & k-3 & -5 & 0 \\
 \end{pmatrix}
\begin{pmatrix}
  k-1 & -3 & 0 \\
  k-3 & -5 & 0 \\
  \vdots & \ddots & \ddots \\
  0 & 2 & -k \\
\end{pmatrix}
\]

Both matrices are non-singular, and so \( V_{i,k-i} \) (\( 0 \leq i \leq k \)) are uniquely determined.

When \( k \) is even (\( k = 2m \), say), the situation is rather different. The requirement that \( D_k = \eta_{2m} (x^2 + y^2)^m \) gives us \( 2m + 1 \) linear equations for \( \eta_{2m} \) and the \( 2m + 1 \) coefficients of \( V_k \), again in terms of the \( V_i, p_i \) and \( q_i \) with \( i < k \). These also decouple into two sets: \( m + 1 \) equations for \( \eta_{2m} \) and the \( m \) odd coefficients of \( V_k \), and \( m \) equations for the \( m + 1 \) even
coefficients. The coefficient matrix for the first set is

\[
\begin{pmatrix}
1 & 1 \\
\binom{m}{1} & -(k-1) & 3 & 0 \\
\binom{m}{2} & 0 & -(k-3) & 5 \\
\vdots & \ddots & \ddots & \ddots \\
\binom{m}{m-1} & 0 & -3 & (k-1) \\
1 & & 0 & -1 \\
\end{pmatrix}
\]

This is non-singular, so that \( \eta_{2m} \) and the odd coefficients of \( V_k \) are uniquely determined. To obtain unique values for the even coefficients of \( V_k \), we introduce a supplementary relation, in the choice of which we have considerable discretion. We have opted for the equation

if \( m \) is even, and

\[ V_{m,m} = 0, \]

\[ V_{m+1,m} + V_{m,m+1} = 0, \]

if \( m \) is odd. It can then be checked easily that the even coefficients of \( V_k \) are determined uniquely.

Thus, the calculation of the focal values is a recursive procedure, each iteration of which consists of

- solving the two sets of linear equations for the \( V_{i,j} \) with \( i+j = k \)

if \( k \) is odd, and

- solving the two sets of linear equations for \( \eta_k \) and the \( V_{i,j} \) with \( i+j = k \)

if \( k \) is even.

It is easy to see that \( \eta_2 = \lambda \) and \( V_2 = \frac{1}{2}(x^2 + y^2) \). Thereafter, however, the calculations soon become extremely onerous, and in most cases it is not possible to compute much beyond \( V_3 \) by hand. A computer algebra system therefore becomes a necessary tool.

The program which we have developed to compute focal values and reduce them has been named FINDETA, and we shall describe the details of its implementation in the next section. The reduction of the focal values can be approached in various ways. In some contexts it is best to compute all the focal values up to the desired order and then reduce them. In other cases, it is better to reduce the focal values one at a time, as they are calculated. FINDETA allows the possibility of a middle way: some focal values are calculated, these are reduced, and the program restarted to compute subsequent focal values. This "restart" facility is very useful, for the unreduced focal values can be very large, but premature reduction is often counter-productive.
4. Implementation

When we first became involved in computations relating to small-amplitude limit cycles, we used a program for manipulating polynomials devised by two colleagues (Long & Danicic, 1976). This was written in ALGOL 68 and was used to obtain the results pertaining to quadratic and symmetric cubic systems described in Blows & Lloyd (1984a). As our investigations developed, a more sophisticated approach became necessary, and we have used the various versions of REDUCE, initially on a VAX 11/750. Our current implementation of the algorithm, and that which we describe here, uses REDUCE 3.3 on the Amdahl 5890/30 at the Manchester Computing Centre, which we access via the JANET network.

The procedure which we have developed has an interactive version (FINDETA) and a batch version (BATCHETA). The program's operation is controlled by a REXX command file running under the VM/CMS operating system, and the three-part file identifier of VM/CMS is exploited to give the user a simple method of distinguishing between the various files relating to a particular system of differential equations. Initially, the user is required to provide information in two files. In <filename> SYSTEM A the user enters the coefficients in the differential equations themselves together with the range of k for which \( V_k \) is to be computed. The other input file, <filename> SUBS A, is optional and consists of a sequence of statements that control the reduction of the focal values as described at the end of section 3.

The program is organised so that in the kth "round", the polynomial \( V_{k+1} \) is computed. When the nominated terminal value of k is reached, three files are produced: <filename> ETA A, <filename> SAVETA A and <filename> RESTART A. Their contents will be described later.

In the SYSTEM file, the coefficients of the monomials in \( P \) and \( Q \) are assigned to arrays \( XX \) and \( YY \), the coefficient of \( x^ty^j \) in \( P \) being stored in \( XX(i, j) \) and its coefficient in \( Q \) stored in \( YY(i, j) \). We see from equation (3.1) that the kth round of the program requires \( P_t \), \( q_t \), \( (V_t)_x \) and \( (V_t)_y \) for \( 1 \leq k \). Since the \( p_t \) and \( q_t \) are homogeneous polynomials, they can be recorded as polynomials in \( y \) alone—that is, they are "dehomogenised" by setting \( x = 1 \). We do this to reduce the demands made on the available storage space. We define arrays \( PP, QQ \):

\[
PP(k) = \sum_{i=0}^{k} XX(k-i, i)y^i, \quad QQ(k) = \sum_{i=0}^{k} YY(k-i, i)y^i \quad (k \geq 1).
\]

The partial derivatives of the \( V_k \) are stored in arrays \( VX \) and \( VY \):

\[
VX(k) = (V_k)_x, \quad VY(k) = (V_k)_y \quad (k \geq 2),
\]

again with \( x = 1 \). Since it is the partial derivatives of \( V_k \) that are required subsequently, it is \( (V_k)_x \) and \( (V_k)_y \) that are stored, rather than \( V_k \) itself.

On the first run, the initial value of \( k \) in the SYSTEM file is 2. Let the terminal value be \( K \). The program then runs as far as round \( K \). The focal values \( \eta_2, \ldots, \eta_{K+1} \) (if \( K \) is odd) or \( \eta_2, \ldots, \eta_K \) (if \( K \) is even) are stored in the ETA file. The RESTART file contains \( PP, QQ \) and the partial derivatives of the polynomials \( V_2, \ldots, V_{k+1} \). In this way, the calculations can be restarted at \( k = K + 1 \). In practice, the program is first run from \( k = 2 \) to \( k = 2r - 1 \), say; substitutions from \( \eta_2, \eta_6, \ldots, \eta_{2r} \) are decided upon and entered in the SUBS file. The program is called again, but now the initial value of \( k \) in the SYSTEMS file is \( 2r \). This is a valuable facility, for the appropriate substitutions cannot usually be seen in advance of
knowing the $\eta_{2i}$. It is a matter of judgement how many focal values should be calculated before entering further substitutions—as a rough guide one would not normally compute more than two or three, and often only one. When the restart option is used, the ETA file is overwritten, but the focal values which have already been computed are stored in the SAVETA file. That is, the ETA file contains output only from the latest run of FINDETA (or BATCHETA), while the accrued focal values from previous runs are retained in SAVETA. We remark that the RESTART file becomes very large as $k$ increases.

At the $k$th iteration, the following expression is first computed:

$$w = PP(1) \sum_{i=0}^{k} (k+1-i) V_{k+1-i} y^i + QQ(1) \sum_{i=1}^{k+1} i V_{k+1-i} y^{i-1} + PP(k) V X(2) + QQ(k) V Y(2) + m1 + n1,$$

where $m1 = 0$ if $k = 2$, and

$$m1 = \sum_{i=2}^{k-1} (PP(i) V X(k+2-i) + QQ(i) V Y(k+2-i)),$$

if $k > 2$, while $n1 = 0$ if $k$ is even, and if $k$ is odd

$$n1 = -\eta_{k+1}(1+y^{k+1}/2).$$

The task is to solve $w = 0$, giving $V_{0,k+1}, \ldots, V_{k+1,0}$, and, in addition $\eta_{k+1}$ if $k$ is odd. The REDUCE command COEFFN is used to extract the coefficient of each monomial in $w$ to give the two sets of linear equations which are required to be solved. The solution is then found by using the SOLVE command.

Batch and interactive modes of working are totally compatible, allowing lower $k$ values to be processed interactively and then a switch to batch working when execution time or required space become too great. We are restricted to 3 megabytes of core store when working interactively but batch mode allows up to 16 megabytes and 2500 seconds of cpu time.

After the focal values are computed, their reduction is continued using REDUCE interactively. The ETA file is input, and appropriate substitutions made; use is made of the FACTOR switch. This part of the procedure is heavily interactive. We have not sought to automate it, for experience suggests that some of the information which we require for phase (4) of the scheme at the end of section 2 would be lost if we did.

This computer algorithm is generally available to members of the group, so it was our aim to make it as efficient and user friendly as possible. Like many other computer implementations, it has evolved with changes being made in response to user requirements as well as the continuing efforts to improve its efficiency.

The implementation of the algorithm on the Amdahl is a great improvement on the VAX 11/750 versions which we used previously. For example, one investigation which took 8–9 hours on the VAX was completed in an elapsed time of under 20 minutes on the Amdahl. When we first used the Amdahl, the available version of REDUCE was version 3.2, but recently version 3.3 has been installed. Strangely, this has been less suitable for our purposes, making greater demands on both storage space and cpu time. As a result, increased use of batch mode became necessary and some examples proved to be intractable. Recently, the Manchester Computing Centre has kindly made REDUCE 3.2 available to us again, and so we have been able to make precise comparisons. In the case of one particular cubic system ([system (4.6) in Lloyd et al. (1988)]) we obtained the
following timings:

<table>
<thead>
<tr>
<th></th>
<th>REDUCE 3.2</th>
<th>REDUCE 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 2$ to $k = 15$</td>
<td>34,895 s</td>
<td>92,273 s</td>
</tr>
<tr>
<td>$k = 16$ to $k = 19$</td>
<td>154,243 s</td>
<td>390,700 s</td>
</tr>
</tbody>
</table>

In version 3.2, the whole calculation could be run interactively within the 3 megabyte limit, whereas this was not possible in version 3.3.

The reason for the relative slowness of version 3.3 in this particular application is unclear. The commands SOLVE and COEFFN appear to be implicated. They return results in lists in version 3.3 as opposed to arrays in version 3.2, but this of itself is hardly sufficient to account for the observed differences. We have also observed that the RESTART file contains approximately twice as many records in version 3.3 as it does in version 3.2, though the information stored is exactly the same. We can only conclude that the cause lies in the algorithms used within REDUCE, though we emphasise that there is no suggestion that version 3.3 is inferior to version 3.2 in general.

5. Some Results

We give a brief resumé of some of the results which have been obtained using the techniques described in this paper. Let $H$ denote the maximum number of limit cycles which can bifurcate out of a fine focus. It has long been known that $H = 3$ for quadratic systems; this was shown by Bautin (1952). In Blows & Lloyd (1984a) we proved that $H = 5$ for cubic systems in which the quadratic terms are absent (that is, systems of the form (2.1) in which $n = 3$ but $p_2 = q_2 = 0$). For general cubic systems, the focal values rapidly become very large indeed; for instance, $L(3)$ contains over 600 terms. This is too complicated to be able to handle, and it is necessary to consider particular cases. In Lloyd et al. (1988) we describe a class of cubic systems with six small-amplitude limit cycles; they are of the form

$$\dot{x} = \lambda x + y + Cy^2 + Hxy^2 + Ky^3,$$
$$\dot{y} = -x + \lambda y + D(x^2 - y^2) + Lx^3 - Hx^2y + Nxy^2.$$

In Lloyd et al. (1988) we also discuss the simultaneous bifurcation of limit cycles from several fine foci, and also from one fine focus and infinity. We describe examples of symmetric cubic systems with seven limit cycles, three of which bifurcate from each of a pair of fine foci other than the origin, and the seventh bifurcates from the origin. We also present a system with four limit cycles bifurcating from the origin and one from infinity. Recently, members of our group have found two independent classes of cubic systems with seven small-amplitude limit cycles and one with eight.

Certain quartic systems have recently been investigated by a research student, J. M. Abdulrahman, and he has come across the unusual occurrence to which we referred in section 2. For systems of the form

$$\dot{x} = y, \quad \dot{y} = -x + a_1x^4 + a_2x^3y + a_3x^2y^2 - a_4xy^3 + a_5y^4,$$

it is easily proved that $\eta_{2k} \neq 0$ only if $2k = 2 \mod 6$; hence $L(k)$ is derived from $\eta_{6k+2}$ and not from $\eta_{2k+1}$ as is usually the case. More strikingly, it emerges that $L(3)$ is necessarily zero when $L(2) = 0$, so that only three limit cycles can be generated in the usual way if initially $L(k) = 0$ for $k < 4$ but $L(4) \neq 0$. 

Much of our activity has been concerned with systems of Liénard type

\[ \dot{x} = y - F(x), \quad \dot{y} = -g(x). \]  

(5.1)

In Blows & Lloyd (1984b) and Lloyd & Lynch (1988) we prove a number of results without recourse to the calculation of focal values, and in Lynch (1989), extensive work is described on equation (5.1) when \( F \) and \( g \) are polynomials of degree no more than six. The value of \( H \) is obtained in a large number of cases.

Other types of systems which we have studied in detail are the so-called "homogeneous" systems; these are of the form (2.1) with

\[ p_2 = \ldots = p_{n-1} = q_2 = \ldots = q_{n-1} = 0. \]

It is found that there is a close relationship with the scalar non-autonomous equation

\[ \dot{r} = \alpha(\theta)r^3 + \beta(\theta)r^2. \]  

(5.2)

This is investigated thoroughly in Alwash & Lloyd (1987), for instance. Recently, a colleague, E. M. James, and a research student, N. Yasmin, have studied cubic systems of the form

\[ \dot{x} = \lambda x + y + p_2(x, y) + x(4x^2 + Bxy + Cy^2), \]

\[ \dot{y} = -x + \lambda y + q_2(x, y) + y(4x^2 + Bxy + Cy^2). \]

These are also related to equation (5.2), and it has now been shown that such systems can have five small-amplitude limit cycles but no more. The computing has not been easy, with difficulties arising both with available time and the allocated storage space.

Since the proofs of the results which we have outlined depend on the use of Computer Algebra systems, it is necessary to address the question of the logical status of our conclusions. It is clearly desirable to introduce as many checks and balances as possible. In our case, several researchers have been involved over a number of years, and each has produced a slightly different implementation of the basic algorithm. Moreover, different machines have been used. Thus our procedures have gone through a whole sequence of improvements and refinements. At each stage, great care has been exercised to ensure consistency, some systems being investigated using the different versions of the program. No inconsistencies have been found. Some of the systems which we have considered have also been investigated independently by other mathematicians; again there is complete agreement. Of particular value are those results which can be proved without recourse to computing: see Blows & Lloyd (1984b) and Lloyd & Lynch (1988), for example. Instances of these general results can be obtained using the computational approach which we have described: in these cases, we have detected no inconsistencies. In the light of all these checks, we are persuaded that our conclusions are reliable.

We gratefully acknowledge the cooperation of the Manchester Computing Centre and the financial support of the Science and Engineering Research Council under research grant GR/D 16536.

**REDUCE Code:** Both the REDUCE 3.2 and REDUCE 3.3 versions of the program are available to anyone who is interested. Please contact J. M. Pearson via electronic mail, address: EYEJMP @ UK.AC. Manchester Computing Centre CMS.

**References**


Landis, E. M., Petrovskii, I. G. (1955). On the number of limit cycles of the equation \( \frac{dy}{dx} = \frac{P(x, y)}{Q(x, y)} \) where \( P \) and \( Q \) are polynomials of the second degree. *Mat. Sb. N.S.* 37, (79), 209–230; (1958) *Amer. Math. Soc. Transl.* (2), 16, 177–221.

Landis, E. M., Petrovskii, I. G. (1957). On the number of limit cycles of the equation \( \frac{dy}{dx} = \frac{P(x, y)}{Q(x, y)} \) where \( P \) and \( Q \) are polynomials. *Mat. Sb. N.S.* 43, (85), 149–168; (1960) *Amer. Math. Soc. Transl.* (2), 14, 181–200.

Landis, E. M., Petrovskii, I. G. (1959). Corrections to the articles “On the number of limit cycles of the equation \( \frac{dy}{dx} = \frac{P(x, y)}{Q(x, y)} \) where \( P \) and \( Q \) are polynomials of the second degree” and “On the number of limit cycles of the equation \( \frac{dy}{dx} = \frac{P(x, y)}{Q(x, y)} \), where \( P \) and \( Q \) are polynomials”. *Mat. Sb. N.S.* 48, (90), 253–255.


