# An Algorithm for the Computation of Normal Forms and Invariant Manifolds

L. Vallier *
LMC-IMAG, 46 Av. F. Viallet f38031 Grenoble Cedex.

1993

**Abstract.** This paper deals with an algorithm to compute normal forms and invariant manifolds of ordinary differential equations. This algorithm based on transformation theory, give us a usefool tool in the study of such equations, in the neighborhood of singular points This tool involves a lot of computations on homogeneous polynomials. Then in addition, a tree data structure is described to represent homogeneous polynomials in an efficient way, and we give the cost of the algorithm.

## Introduction

This paper deals with normal forms and invariant manifolds, since these two theoritical tools are often used together, in order to have differential equations easier to be studied. According to the theory, some effective algorithms are introduced. In the first part, we recall an important theorem of transformation theory which will allows us, following Hale, to give an algorithmic definition of normal forms relative to a projection,in the second part. That definition leads to an algorithm which provides linear systems to solve. These linear systems may be singular and arise from a linear differential operator. This operator will be described, and its spectrum will be defined in an effective manner. The cases where the linear systems are always regular, correspond to the case of computation of invariant manifolds in normal form. Then some modifications are needed to obtain an algorithm which computes the invariant manifolds on the initial coordinates, and the different cases are summed up by a general algorithm. The computations yields a lot of operations on the homogeneous polynomials,and in the third section, we define a tree data structure which allows to encode homogeneous polynomials over lexicographic basis, very efficiently . Then, we calculate the cost of the algorithm at the end of this section.

*Partially supported by the *PRC Mathématiques et Informatique* . SUBMISSION TO ISSAC'93.


**ACM-ISSAC '93-7/93/Kiev, Ukraine**
**© 1993 ACM 0-89791-604-2/93/0007/0225...$1.50**

## 1 Transformation Theory

In this section we give a tool to perform a change of variables to a differential equation.

Let the following differential equation defined in a neighborhood of $z = 0$ :

$$\dot{z} = Az + f(z), z \in \mathbb{C}^n \tag{1.1}$$

where $A$ is a $n \times n$ constant matrix and f a $\mathbb{C}^\infty$ function defined by the formal power series :

$$f(z) = \sum_{m \geq 2} f_m(z)$$

where the $f_m(z)$ are homogeneous polynomial in $z$ of degree $m$.
The transformation relative to the continuous projection $\pi$ is defined by:

$$z = w + h(\pi w) \tag{1.2}$$

with

$$h(\pi w) = \sum_{m \geq 2} h_m(\pi w)$$

where the $h_m(\pi w)$ are homogeneous polynomial in $\pi w$ of degree $m$. This change of variables applied to equation (1.1) leads to the differential equation for w:

$$\dot{w} = Aw + g(w)w \in \mathbb{C}^n \tag{1.3}$$

with

$$g(w) = \sum_{m \geq 2} g_m(w)$$

where the $g_m(w)$ are homogeneous in $w$ of degree $m$. Then it is easy to verify that transformation (1.2) is equivalent to the following one:

$$z = \epsilon x, w = \epsilon y, x = u(y, \epsilon) \stackrel{def}{=} y + \sum_{m \geq 1} h_{m+1}(\pi y)\frac{\epsilon^m}{m!} \tag{1.4}$$

Let us introduce the notation:

$$\begin{aligned} fog &= \frac{\partial f}{\partial x} g \\ f \times g &= \frac{\partial f}{\partial x} g - \frac{\partial g}{\partial x} f \end{aligned}$$

Furthemore, if we define the function $U(\pi y, \epsilon) = \sum_{m \geq 0} U_m(\pi y) \frac{\epsilon^m}{m!}$ such that $u(y, \epsilon)$ is the unique solution of the equation:

$$\frac{\partial u}{\partial \epsilon} = U(\pi u, \epsilon) \tag{1.5}$$

where $U_m(\pi y)$ are homogeneous in $\pi y$ of degree m+2, then the following theorem holds [6]:

**Theorem 1** *Suppose the notation as above. then if we define the sequences $f_i^{(m)}$, $U_i^{(m)}$, $i, m = 0, 1, 2, \ldots$ by the recursive relations*

$$\begin{aligned} f_i^{(m)} &= f_{i+1}^{(m-1)} + \sum_{j=0}^{i} C_i^j f_{i-j}^{(m-1)} \times U_j \pi, \quad (1.6) \\ & \quad i = 0, 1, \ldots; m = 1, 2, \ldots \\ f_i^{(0)} &= i! f_{i+1}, \; i = 0, 1, 2, \ldots \end{aligned}$$

*and*

$$\begin{aligned} U_i^{(m)} &= U_{i+1}^{(m-1)} + \sum_{j=0}^{i} C_i^j U_{i-j}^{(m-1)} o U_j, \quad (1.7) \\ & \quad i = 0, 1, \ldots; m = 1, 2, \ldots \\ U_i^{(0)} &= U_i, \; i = 0, 1, 2, \ldots \end{aligned}$$

*then*

$$g_m = \frac{1}{(m-1)!} f_0^{(m-1)}, \; m = 1, 2, \ldots \quad (1.8)$$

$$h_m = \frac{1}{(m-2)!} U_0^{(m-2)}, \; m = 2, 3, \ldots \quad (1.9)$$

notice that we put $f_1(x) = Ax$. Therefore the coefficients $g_m$ are computed recursively from $f_m$ and $U_m$, without the need of the function $h$. But the coefficients $h_m$ can be computed recursively from the $U_m$ if the transformation is needed. Let remark that computations (1.6) and (1.7) can be performed according to the following so-called Lie triangles.

$$\begin{array}{cccccc} f_0^{(0)} & & & & & \\ f_1^{(0)} & f_0^{(1)} & & & & \\ f_2^{(0)} & f_1^{(1)} & f_0^{(2)} & & & \\ f_3^{(0)} & f_2^{(1)} & f_1^{(2)} & f_0^{(3)} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \end{array} \tag{1.10}$$

$f_i^{(m)}$ depends on left-hand terms, from the same row to the top of the triangle 1.10. That is to say, $f_i^{(m)}$ is computed from the $f_s^{(r)}$, with $r, s$ such that $0 \leq r + s \leq m + i$ and $r < m$.

$$\begin{array}{cccccc} U_0^{(0)} & & & & & \\ U_1^{(0)} & U_0^{(1)} & & & & \\ U_2^{(0)} & U_1^{(1)} & U_0^{(2)} & & & \\ U_3^{(0)} & U_2^{(1)} & U_1^{(2)} & U_0^{(3)} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \end{array} \tag{1.11}$$

The same remarks can be applied to $U_i^{(m)}$ which depends on $U_s^{(r)}$, with $r, s$ such that $0 \leq r + s \leq m + i$ and $r < m$.

# 2 Computation of Normal Forms and Invariant Manifolds

## 2.1 Normal Forms

In this section the definition of normal forms relative to a projection $\pi$ is given, which leads to a normal form computation algorithm. Before going further, we would like to point out special feature of formulas (1.6). The $(k+1)$-th row of the triangle 1.10 is given by:

$$f_{k-i}^{(i)} = f_{k-i+1}^{(i-1)} + \sum_{j=0}^{k-i} C_{k-i}^j f_{k-i-j}^{(i-1)} \times U_j \pi, \; i = 1, \ldots, k$$

$$f_k^{(0)} = k! f_{k+1}$$

if we define the sequence $\tilde{f}^{(i)}$, $i = 1, \ldots, k$ by :

$$\begin{aligned} \tilde{f}^{(1)} &= f_k^{(0)} + \textstyle\sum_{j=0}^{k-2} C_{k-1}^j f_{k-1-j}^{(0)} \times U_j \pi \\ \tilde{f}^{(i)} &= \tilde{f}^{(i-1)} + \textstyle\sum_{j=0}^{k-i} C_{k-i}^j f_{k-i-j}^{(i-1)} \times U_j \pi, \; i = 2, \ldots, k \end{aligned} \tag{2.1}$$

it follows

$$f_{k-1}^{(1)} = \tilde{f}^{(1)} + f_0 \times U_{k-1}$$

and thus

$$f_0^{(k)} = \tilde{f}^{(k)} + f_0 \times U_{k-1} \tag{2.2}$$

furthemore, $\tilde{f}^{(k)}$ depends only on $f_j^{(i)}$, $0 \leq i+j \leq k$ and $U_0, U_1, \ldots, U_{k-2}$. Thus, at this step of the computation, if $U_0, U_1, \ldots, U_{k-2}$ are known $\tilde{f}^{(k)}$ is known too. Let us recall that $f_0^{(0)} = A$ then according to (1.8) we can write:

$$LU_{k-1} = \tilde{f}^{(k)} - k! g_{k+1} \tag{2.3}$$

where $L$ is the linear operator defined by:

$$(Lv)(x) = \frac{\partial v(\pi x)}{\partial x} Ax - Av(\pi x).$$

Let us assume that $A\pi = \pi A$ then $\pi A\pi = \pi A$. This yields

$$\frac{\partial v(\pi x)}{\partial x} Ax = \frac{\partial v(\pi x)}{\partial \pi x} \pi Ax = \frac{\partial v(\pi x)}{\partial \pi x} \pi A \pi x,$$

if $\bar{v}$ denote $vo\pi$ and $B = \pi A$, $y = \pi x$ then

$$(L\bar{v})(y) = \frac{\partial \bar{v}(y)}{\partial y} By - A\bar{v}(y).$$

Let $m = dim\pi\mathbb{C}^n$ then $\mathbb{C}^m$ with $\pi\mathbb{C}^n$ can be identified. Let $V(k,m,n)$ be the space of $n$-functions which are homogeneous polynomials in $m$-vector of degree $k$. Therefore, for each $k$:

$$L(V(k,m,n)) \subseteq V(k,m,n)$$

Let us notice that $B$ is a $m \times m$ matrix. We define the decomposition

$$V(k,m,n) = V^r(k,m,n) \bigoplus S(k,m,n)$$

, where $V^r(k,m,n)$ stands for the range of $L$. Let $\pi^r_{(k,m,n)}$ be the projection of $V(k,m,n)$ onto $V^r(k,m,n)$ along $S(k,m,n)$. Now a definition of a normal form can be given:

**Definition 1** *Let $f$ be as in (1.1), and assume $A\pi = \pi A$, then equation (1.3) is a normal form for equation (1.1) relative to the projection $\pi$, if $g_k$, $U_k$ satisfy :*

$$\begin{aligned} LU_{k-1} &= \pi^r_{(k+1,m,n)} \tilde{f}^{(k)} \\ k! g_{k+1} &= (I - \pi^r_{(k+1,m,n)}) \tilde{f}^{(k)} \end{aligned} \tag{2.4}$$

*for $k \geq 1$, and $\tilde{f}^{(k)}$ defined as in (2.1)*

**Remark 1** *if $\pi = I$ then $B = A$, and the condition is nothing but a nonresonnance condition on the eigenvalues of $A$. In this case, definition (1) becomes the classical Poincare normal form definition. For existence, properties and applications see [1], [9] and [4], [8] for an other algorithmical approach.*

This definition yields the algorithm 2.1, which computes $g_{k+1}$, $U_{k-1}$ and $h_{k+1}$, $i = 1, \ldots K-1$. That is to say, algorithm outputs approximation of the normal form to the order $K$.

**Remark 2** *if we want to compute classical normal form we only have to put $\pi = I$ and apply algorithm 2.1.*

---

**Algorithm 2.1** *normal form*

*for k in 1 to K-1 do*
  $f_k^{(0)} := k! f_{k+1}$
  *compute $\tilde{f}^{(k)}$ using formulas (2.1)*
  *determine $V^r_{(k+1,m,n)}$ and $\pi^r_{(k+1,m,n)}$*
  $b := \pi^r_{(k+1,m,n)} \tilde{f}^{(k)}$
  *determine the solution $U^{(0)}_{k-1}$ of : $LX = b$*
  $g_{k+1} := (\tilde{f}^{(k)} - b)/k!$
  *for i in 1 to k do*
    $f^{(i)}_{k-i} := \tilde{f}^{(i)} - b$
  *compute $h_{k+1}$ using formulas (1.7) and (1.9)*

---

## 2.2 Spectrum of the operator L

Since, the algorithm 2.1 yields linear systems of the form $LX = b$ , we are going to describe in an effective way the structure of L, identifying it with a matrix on $V(k,m,n)$.

For further discussion, we introduce the notation, $s \in \mathbb{C}^m$

$$\begin{aligned} s &= (s_1, \ldots, s_m) \\ q &= (q_1, \ldots, q_m),\ q_j \in \mathbb{N} \\ |\, q \,| &= q_1 + \cdots + q_m \\ s^q &= s_1^{q_1} s_2^{q_2} \ldots s_m^{q_m} \end{aligned} \tag{2.5}$$

If d denote the number of different $k$-degree monomials in $m$-vector s then let

$$\mathcal{P}(k,m) = \{p^1, \ldots, p^d\}$$

be the set of m-indices with module $k$, ordered by the usual lexicographic order:

$$p \leq q \Leftrightarrow \begin{cases} p_i = q_i & i = 1, \ldots, m \\ \quad or & \\ p_i = q_i & i = 1, \ldots, r < m \\ \quad and & \\ p_{r+1} < q_{r+1} & \end{cases}$$

Let $e^n_l$ be the $l$-th element of the $\mathbb{C}^n$-canonical basis, then

$$\mathcal{B}(k,m,n) = \{s^{p^i}.e^n_l, 1 \leq i \leq d, p^i \in \mathcal{P}(k,m), 1 \leq l \leq m\}$$

stands for the ordered basis of $V(k,m,n)$. Now, the operator $L$ on $V(k,m,n)$ can be identified to its matrix representation $L_k$ over $\mathcal{B}(k,m,n)$. In order to determine $L_k$, $Lu, u \in \mathcal{B}(k,m,n)$ needs to be computed:

$$L(x^{p^i}.e_l^n) = S(x^{p^i}.e_l^n) - Ax^{p^i}.e_l^n$$

with

$$\begin{aligned} S(x^{p^i}.e_l^n) &= \frac{\partial x^{p^i}.e_l^n}{\partial x} Bx \\ &= (\sum_{s=1}^m \sum_{r=1}^m p_s^i B_{sr} x_r \frac{x^{p^i}}{x_s}) e_l^n \end{aligned}$$

where

$$\frac{x^{p^i}}{x_s} = \begin{cases} x_1^{p^{i_1}} x_2^{p^{i_2}} \dots x_s^{p^{i_s}-1} \dots x_m^{p^{i_m}} & if\ p_s^i > 0 \\ x^0 & otherwise \end{cases}$$

furthemore

$$Ax^{p^i}.e_l^n = \sum_{j=1}^n A_{jl} x^{p^i}.e_j^n$$

these computations lead to the matrix :

$$L_k = \begin{pmatrix} S_k - A_{11}I_d & -A_{12}I_d & \cdots & -A_{1n}I_d \\ -A_{21}I_d & S_k - A_{22}I_d & \cdots & -A_{2n}I_d \\ \vdots & \vdots & \ddots & \vdots \\ -A_{n1}I_d & -A_{n2}I_d & \cdots & S_k - A_{nn}I_d \end{pmatrix}$$

where $S_k$ is the $d \times d$ matrix arrising from the operator $S$.

**Remark 3** *If $A$ is diagonal then $L_k$ is bloc-diagonal, if $A$ is upper triangular then $L_k$ is upper-bloc triangular.*

The end of the section is devoted to the statement of a non-singularity criterion for the operator L, which will allow us to produce an algorithm of computation of invarariant manifold. The first step consist of the computation of the spectrum $\sigma(L)$ of $L$, which will give us the NullSpace of L.

Let P be a $n \times n$ nonsingular matrix, Q a $m \times m$ nonsingular matrix. Let $y = Qz$ and $g(z) = P^{-1}v(Qz)$. It follows

$$L(v(y) = Lv(Qz) = P\frac{\partial g(z)}{\partial z} Q^{-1}BQz - APg(z)$$

we put

$$\bar{L}g(z) = P^{-1}Lv(y) = \frac{\partial g(z)}{\partial z} Q^{-1}BQz - P^{-1}APg(z)$$

it is clear that the operator $\bar{L}$ has the same spectrum as $L$. Also, the matrices $P$ and $Q$ can be choosen such that $P^{-1}AP$ and $Q^{-1}BQ$ are upper triangular.

Then in order to determine $\sigma(L)$ $A$ and $B$ are assumed to be upper triangular. Let $S_k(i,j)$ denote the element on th $i$-th row and $j$-th column of $S_k$. Since $B$ is upper triangular, $B_{sr} = 0$ if $s > r$ then

$$S_k(i,j) = \sum_{I(i,j)} p_s^j B_{sr},$$

where

$$I(i,j) = \{(s,r)/\ p_s^j > 0, x_r \frac{x^{p^j}}{x_s} = x^{p^i}, 1 \le s \le r \le m\}$$

then for $(s,r) \in I(i,j)$

$$\begin{aligned} x_r \frac{x^{p^j}}{x_s} &= x_1^{p^{j_1}} x_2^{p^{j_2}} \dots x_s^{p^{j_s}-1} \dots x_r^{p^{j_r+1}} \dots x_m^{p^{j_m}} \\ &= x^{p^j},\ if\ r = s \\ &< x^{p^j},\ otherwise \end{aligned}$$

therefore $S_k$ is lower triangular, furthemore $S_k(i,i) = \sum_{s=1}^m p_s^i B_{ss}$, thus

$$\sigma(L_k) = \{\lambda_{ij} = \sum_{s=1}^m p_s^i B_{ss} - A_{ii},\ i = 1, \dots, n; j = 1, \dots, d\}$$

**Remark 4** *According to the assumption that matrices $A$ and $B$ are upper triangular, the diagonal elements of these matrices are their eigenvalues.*

Then, the following statement holds:

**Theorem 2** *If $\lambda_i, \mu_j$ denote eigenvalues of $A$ and $B$ respectively. $L_k$ is non singular if and only if*

$$\sum_{s=1}^m q\mu_s - \lambda_i \neq 0,\ 1 \le i \le n; |\ q\ | = k$$

Consequently, if the above condition is satisfied, for every $k$, $L$ is non singular, and we obtain, according to the definition 1:

$$\begin{aligned} U_{k-1} &= L_{k+1}^{-1} \tilde{f}^{(k)} \\ g_{k+1} &= 0,\ k \ge 1 \end{aligned} \tag{2.6}$$

## 2.3 Computation of Invariant Manifolds

We are now in position to precise the relationship beetween normal forms computations and invariant manifolds.

Consider the differential equation $\dot{z} = Az + f(z)$, $z \in \mathbb{C}^n$ which can be writen as follows:

$$\begin{cases} \dot{z}_1 &= Bz_1 + f_1(z_1, z_2) \\ \dot{z}_2 &= Cz_2 + f_2(z_1, z_2) \end{cases} \tag{2.7}$$

where $z = (z_1, z_2), f(z) = (f_1(z), f_2(z))$ and $A = \begin{pmatrix} B & 0 \\ 0 & C \end{pmatrix}$ are as in equation (1.1). We suppose $z_1 = \pi z$, then $A\pi = \pi A = B$ and the following system can be seen as the normal form of system (2.7), defined by definition (1)

$$\begin{cases} \dot{w}_1 &= Bw_1 + g_1(w_1, w_2) \\ \dot{w}_2 &= Cw_2 + g_2(w_1, w_2) \end{cases} \tag{2.8}$$

If we put $w = (w_1, w_2)$, $w_1 = \pi w$ and $v = (v^1, v^2)$ then

$$Lv(w_1) = (L^1(v^1(w_1)), L^2(v^2(w_1)))$$

where

$$L^1(v^1(w_1)) = \frac{\partial v^1(w_1)}{\partial w_1} Bw_1 - Bv^1(w_1)$$

$$L^2(v^2(w_1)) = \frac{\partial v^2(w_1)}{\partial w_1} Bw_1 - Cv^2(w_1)$$

Furthemore, let assume that the eigenvalues of $B, C$ satisfy theorem 2 for each order. It follows, that operator $L^2$ is nonsingular, therefore the equations (2.6) yields: $g_2(w_1, 0) = 0$. This means that the set $w_2 = 0$ is an invariant manifold, and the vector field on this manifold is given by $Bw_1 + g_1(w_1, 0)$ which is in normal form. Therefore, the algorithm 2.1 can be applied to perform these computations. But, if the manifold is needed in the initial coordinates system, then the algorithm has to be modified. Suppose the set $(z_2 = h(z_1))$ is invariant on (2.7). This is equivalent to $\dot{z}_2 = \frac{\partial h(z_1)}{\partial z_1} \dot{z}_1$

$$\Leftrightarrow \frac{\partial h(z_1)}{\partial z_1} Bz_1 - Cz_1 = f_2(z_1, h(z_1)) - \frac{\partial h(z_1)}{\partial z_1} f_1(z_1, h(z_1)) \quad (2.9)$$

Now, let $z_1 = w_1$; $z_2 = w_2 + \bar{h}(w_1)$ be the transformation which brings (2.7) to (2.8), with the set $w_2 = 0$ invariant on (2.8). This yields, after some computations to :

$$g_1(z_1, z_2) = f_1(z_1, z_2 - h(z_1))$$

$$g_2(w_1, w_2) = -(\frac{\partial \bar{h}(w_1)}{\partial w_1} Bw_1 - Cw_1)$$

$$+ f_2(w_1, w_2 + \bar{h}(w_1)) - \frac{\partial \bar{h}(w_1)}{\partial w_1} f_1(w_1, w_2 + \bar{h}(w_1))$$

but, $w_2 = 0$ invariant implies $g_2(w_1, 0) = 0$ and leads to the same equation as (2.9) in $h$. Then, $h$ and $\bar{h}$ can be identified. If $u((z_1, z_2), \epsilon)$ is defined from

$$\begin{pmatrix} z_1 \\ z_2 + h(z_1) \end{pmatrix}$$

as in (1.4) and consider its generator $U$

$$U(\pi u, \epsilon) = U(z_1, \epsilon) = \sum_{k \geq 0} U_m(z_1) \frac{\epsilon^k}{k!}$$

with $U_k$ homogeneous of degree $k + 2$. The partial differential equation (1.5) on functions $u$ and $U$ leads to:

$$U_k(z_1) = \begin{pmatrix} 0 \\ h_{k+2}(z_1) \end{pmatrix}$$

This relation together with definition (1) yields:

$$\begin{aligned} L^2 h_{k+1}(z_1) &= (I - \pi)\tilde{f}^{(k)}(z_1) \\ g_{k+1}(z_1) &= 0 \end{aligned} \quad (2.10)$$

where $L^2$ is defined above, and $\tilde{f}^{(k)}(z_1)$ is computed by formulas (2.1). Furthemore, $\pi \tilde{f}^{(k)}(z_1)$ is the vector field on the invariant manifold $z_2 = h(z_1)$, where $h(z_1)$ is computed by the previous linear system. Refer [3] to find the similar result in the case of a center manifold computation. Rather than giving a particular algorithm derived from the general one 2.1, we prefer state the algorithm 2.2 which sums up all cases. First, let us notice that if $dim \pi \mathbb{C}^n = m$ then :

$$L^2 : V(k, m, n-m) \longrightarrow V(k, m, n-m)$$

$$v(z_1) \longmapsto \frac{\partial v(z_1)}{\partial z_1} \pi A - (I - \pi) A v(z_1)$$

$z_1 \in \pi \mathbb{C}^n$, and $(I - \pi)$ is the projection of $\mathbb{C}^n$ onto the last $(n - m)$ coordinates of $\mathbb{C}^n$.

---

**Algorithm 2.2** *main algorithm (A,f,K,n,m,InvOnly)*

*define $\pi$ to be the projection on the m-first coordinates of $\mathbb{C}^n$*
*if InvOnly = true then do*
  $\bar{\pi} := (I - \pi)$
  $p := n - m$
  *else do*
    $\bar{\pi} := I$
    $p := n$
$B := \pi A$; $C := \bar{\pi} A$
*for k in 1 to K-1 do*
  *compute $\mathcal{B}(k+1, n, n)$ and $\mathcal{B}(k+1, m, p)$*
  $f_k^{(0)} := k! f_{k+1}$ *encoded over $\mathcal{B}(k+1, n, n)$*
  *compute matrix $L_{k+1}$ over $\mathcal{B}(k+1, m, p)$*
  *compute $\tilde{f}^{(k)}$ using formulas (2.1)*
  *determine $V^r_{(k+1,m,p)}$ and $\pi^r_{(k+1,m,p)}$*
  $b := \pi^r_{(k+1,m,p)} \tilde{f}^{(k)}$
  *compute $U^{(0)}_{k-1}$ solution of : $L_{k+1} X = b$*
  $g_{k+1} := (\tilde{f}^{(k)} - b)/k!$
  *for i in 1 to k do*
    $f^{(i)}_{k-i} := \tilde{f}^{(i)} - b$
  *if InvOnly = false then do*
    *compute $h_{k+1}$ using (1.7) and (1.9)*

---

If we want to compute the normal form, we apply the main algorithm with in inputs $(A, f, K, n, n, false)$. To compute $n - m$-dimentionnal invariant manifold in

normal form the inputs are $(A, f, K, n, m, false)$, and not in normal form $(A, f, K, n, m, true)$.
Before introducing the programming aspects, let us show some particular kinds of differential systems whose linear part verifies theorem 2 at each order, and therefore admit invariant manifolds. These cases are of great interest and many authors state existence properties and applications of such invariant manifolds.See [7], [9],[2], and [5] for an constructive proof which leads to another kind of algorithm.

| | | |
|---|---|---|
| $\sigma(B) = 0$ | $\sigma(C) \neq 0$ | center manifold |
| $\sigma(B) < 0$ | $\sigma(C) \geq 0$ | stable manifold |
| $\sigma(B) > 0$ | $\sigma(C) \leq 0$ | unstable manifold |
| $\sigma(B) \leq 0$ | $\sigma(C) > 0$ | center stable manifold |
| $\sigma(B) \geq 0$ | $\sigma(C) < 0$ | center unstable manifold |

# 3 Programming Aspects

First at all, we point attention on the fact that if we do computations on a space $V(k, m, n)$ with its elements incoded over the basis $\mathcal{B}(k, m, n)$ then it is easy to write and solve linear systems which arise from above algorithm. The disavantage is that product and derivation over such basis have to be defined. Let us take an example: let $u$, $v$ be polynomial components of some $n$-functions in $V(k, m, n)$. These polynomials have the following representation:

$$u \equiv (u_1, u_2, \ldots, u_d),\ v \equiv (v_1, v_2, \ldots, v_d)$$

where $d$ is the number of terms of $\mathcal{P}(k, m) = \{p^1, \ldots, p^d\}$ and let $d'$ be the number of terms of $\mathcal{P}(2k, m) = \{q^1, \ldots, q^{d'}\}$, then $w = u.v \equiv (w_1, \ldots, w_{d'})$. and

$$w_k = \sum_{\{1 \leq i,j \leq d / p^i + p^j = q^k\}} u_i v_j.$$

Therefore, for each monomial product the position of $p^i + p^j$ in $\mathcal{P}(2k, m)$ has to be found, which yields an important cost in $\mathcal{O}(d')$. The cost of the product is here $\mathcal{O}(d'd^2)$. Also, it seems necessary to organize the basis in a way which allows faster search of elements.

## 3.1 Polynomial Representation

We are going to construct a tree data structure which will allow us to search a basis element in $O(m)$ steps.

Let us define the set of nonnegative integers lists

$$\mathcal{L}(k, m) = \{l = (l_1, l_2, \ldots, l_m) / k \geq l1 \cdots \geq l_m = 0\}$$

ordered by

$$l \leq h \iff \begin{cases} l_s = h_s,\ s = 1, \ldots, m-1 \\ or \\ l_s = h_s\ s = 1, \ldots, r < m-1 \\ l_{r+1} > h_{r+1} \end{cases}$$

We define the map

$$\begin{aligned} &\phi_k : \mathcal{P}(k, m) \longrightarrow \mathcal{L}(k, m) \\ &p \longmapsto (k - p_1, \ldots, k - \textstyle\sum_{j=1}^{r} p_j, \ldots, 0) \end{aligned} \tag{3.1}$$

**Remark 5** *last element of $\phi_k(p)$ is 0, since by definition $\mid p \mid = k$*

let $l = (l_1, l_2, \ldots, l_m)$, $l_m = 0$ then if we define $p = (k - l_1, l_1 - l_2, \ldots, l_{m-2} - l_{m-1}, l_{m_1})$, it is easy to verify that $\phi_k(p) = l$. Furthemore, let us take $p, q$ such that $\phi_k(p) = \phi_k(q)$. This implies

$$k - \sum_{j=1}^{r} p_j = k - \sum_{j=1}^{r} q_j,\ r = 1, \ldots, m$$

and by induction on $r$ this implies that $p = q$. Therefore $\phi_k$ is a bijection and we can define:

$$\begin{aligned} &\phi_k^{-1} : \mathcal{L}(k, m) \longrightarrow \mathcal{P}(k, m) \\ &l \longmapsto (k - l_1, l_1 - l_2, \ldots, l_{m-2} - l_{m-1}, l_{m-1}) \end{aligned} \tag{3.2}$$

Consider $l < h$ then it exists $r$ such that $l_{r+1} > h_{r+1}$ and the $r$ first indices of $l$ and $h$ are equal. It is easy to see that the $r$ first indices of $\phi_k^{-1}(l)$ and $\phi_k^{-1}(h)$ are equal. Then, the $r+1$-th verify $l_r - l_{r+1} < h_r - h_{r+1}$ and therefore $\phi_k^{-1}(l) < \phi_k^{-1}(h)$.
Consequentely, if $p^i, l^i$ are the $i$-th element of the ordered basis $\mathcal{P}(k, m)$, $\mathcal{L}(k, m)$ respectively then

$$\phi_k(p^i) = l^i \tag{3.3}$$

Let $\mathcal{S}$ denote the binary relation "is son of" over $\mathcal{L}(k, m)$ defined by

$$l\,\mathcal{S}\,h \iff \begin{cases} l_s = h_s,\ s = 1, \ldots, r < m-1 \\ l_{r+1} > h_{r+1} = 0,\ l_{r+2} = 0 \end{cases}$$

Let $l = (l_1, l_2, \ldots, l_r, 0, \ldots, 0)$ then $h = (l_1, l_2, \ldots, l_{r-1}, 0, \ldots, 0)$ is the unique list such that $l\mathcal{S}h$, and a direct application of the definition provides the set of its sons:

$$\mathcal{S}on(l) = \{(l_1, l_2, \ldots, l_r, i, 0, \ldots, 0),\ i = 1, \ldots, l_r\}$$

moreover, $\mathcal{S}on(l)$ possess $l_r$ elements. We say that $l = (l_1.l_2, \ldots, l_r, 0, \ldots, 0)$ is $i$-th son of $h$ iff $l$ is son of $h$ and $l_r = i$. and we denote this by $l\ \mathcal{S}^i\ h$

**Remark 6** *Let notice that if $l$ is son of $h$ then $l < h$. Moreover, if $l$ is $i$-th son of $k$, $h$ is $j$-th son of $k$ and $i \leq j$ then $h \leq l < k$. Therefore, $Son(l)$ is an ordered set, and the $i$-th element of this set is the $i$-th son of $l$.*

**Example 1** $\mathcal{L}(3,3) = \{(0,0),(1,0),(1,1),(2,0),(2,1),(2,2),(3,0),(3,1),(3,2),(3,3)\}$

In order to find out a better data structure for the basis $\mathcal{P}(k,m)$, a tree is defined from $\mathcal{L}(k,m)$. But, rather than defining a tree whose nodes are lists, we prefer building an equivalent tree whose nodes are the positions of elements in the ordered basis $\mathcal{L}(k,m)$.

**Definition 2**

$$tree(l) == \begin{cases} (\text{ position of } l \text{ }) \text{ if } Son(l) = \emptyset \\ (\text{ position of } l \text{ } tree(h) \text{ for } h \text{ in } Son(l) \text{ }) \end{cases}$$

Let us recall that one list admits at most one "father". Therefore the data structure defined above does not contain any circuit and is a tree.

$$h \in tree(l) \Longleftrightarrow \begin{cases} h = l \\ or \\ \exists l' \in Son(l) \text{ such h} \in \text{ tree( } l') \end{cases} \tag{3.4}$$

The first element of $\mathcal{P}(k,m)$ is $p^1 = (k,0,\ldots,0)$, then the first element of $\mathcal{L}(k,m)$ is $l^1 = (0,\ldots,0)$. Finally, the data structure is defined by:

**Definition 3** $T(k,m) = tree(l^1)$

Let us verify that each node of the tree $T(k,m)$ belongs to $\mathcal{L}(k,m)$ and conversely, that each element of $\mathcal{L}(k,m)$ is a node of $T(k,m)$. Definition (2) with (3.4) yield that each node of $T(k,m)$ is an element of $\mathcal{L}(k,m)$. Conversely, consider $l = (l_1,\ldots,l_r,0,\ldots,0)$, $1 \leq r \leq m-1$ element of $\mathcal{L}(k,m)$. $l$ is son of $l^{(1)} = (l_1,\ldots,l_{r-1},0,\ldots,0)$ which implies that $l \in tree(l^{(1)})$. $l^{(1)}$ is son of $l^{(2)} = (l_1,\ldots,l_{r-2},0,\ldots,0)$, then $l^{(1)} \in tree(l^{(2)})$ and therefore $l \in tree(l^{(2)})$. Repeating this process we obtain $l \in tree(l^{(s)})$ where $l^{(s)} = (l_1,\ldots,l_{r-s},0,\ldots,0)$, $1 \leq s \leq r$, which implies that $l$ is a node of $T(k,m)$.
This constructive proof means that if $l = (l_1,l_2,\ldots,l_r,0,\ldots,0)$ then

$$l = \mathcal{S}^{l_r}(\mathcal{S}^{l_{r-1}}(\cdots(\mathcal{S}^{l_1}(l^1)\cdots)) \tag{3.5}$$

which means that entries of l represent the path which leads to the node *position of* $l$ through the tree $T(k,m)$. Let $elt_k^m$ denote the algorithmical function which performs that search. Then the following statement holds:

**Theorem 3** *Let $p$ be an element of $\mathcal{P}(k,m)$ then the position of $p$ in $\mathcal{P}(k,m)$ is computed by the function $elt_k^m(\phi_k(p))$ in at most $(m-1)$ steps*

The cost is given by the lenght of $\phi_k(p)$ which is less or equal to $(m-1)$.

**Example 2** $T(3,3) = (1(2(3))(4(5)(6))(7(8)(9)(10)))$
$elt_3^3((3,1)) = 8$. *3 leads to the third son of* 1: 7, *and* 1 *leads to the first son of* 7: 8.

Now, let us consider the basis $\mathcal{P}(k,m)$, and the tree data structure $T(k,m)$. We claim that this structure can be used to localize the elements of $\mathcal{P}(k',m)$, if $k' \leq k$. We put:

$$\begin{array}{ll} p = (p_1,p_2,\ldots,p_m), & p \in \mathcal{P}(k,m) = \{p^1,\ldots p^d\} \\ q = (q_1,p_2,\ldots,p_m) & ,q \in \mathcal{P}(k',m) = \{q^1,\ldots p^{d'}\} \end{array}$$

Then $q_1 = k' - k + p_1$ and it follows $k' - q_1 = k - p_1$, which implies $\phi_k(p) = \phi_{k'}(q)$. Notice that if $p$ is the $i$-th element of $\mathcal{P}(k,m)$ then relation (3.3) yields that $q$ is the $i$-th element of $\mathcal{P}(k',m)$. Moreover, if $\mathcal{L}(k,m) = \{h^1,\ldots,h^d\}$ then

$$\mathcal{L}(k',m) = \{h^i,\ i = 1,\ldots,d'\} \tag{3.6}$$

and therefore

$$elt_k^m(l) = elt_{k'}^m(l), \text{ where } l = \phi_k(p) = \phi_{k'}(q) \tag{3.7}$$

**Example 3**
$\mathcal{L}(2,3) = \{(0,0),(1,0),(1,1),(2,0),(2,1),(2,2)\}$
$T(2,3) = (1\,(2\,(3))\,(4\,(5)\,(6)))$.
$elt_2^3((2,1)) = 5 = elt_3^3((2,1))$

This structure is used to define the product and the derivation of monomials in the following manner: let $p \in \mathcal{P}(k_1,m)$, and $q \in \mathcal{P}(k_2,m)$ and suppose $k \geq k_1 + k_2$. We put $\phi_{k_1}(p) = l$, $\phi_{k_2}(q) = h$, then, applying the definition 3.2, it is easy to see that $p+q = \phi_{k_1+k_2}^{-1}(l+h)$. Then, the position of $p+q$ in $\mathcal{P}(k_1+k_2,m)$ is given by $elt_{k_1+k_2}^m(l+h)$. The property (3.7) states that this position is obtained by $elt_k^m(l+h)$. Thus, the following theorem holds:

**Theorem 4** *With the previous notations, the position of $p+q$ in $\mathcal{P}(k_1+k_2,m)$ is computed by $elt_k^m(l+h)$ in at most $2(m-1)$ steps. Furthemore, if $p \geq q$ then the position of $p-q$ in $\mathcal{P}(k_1-k_2,m)$ is computed by $elt_k^m(l-h)$.*

The cost comes from the componentwise sum $l+h$ which needs at most $(m-1)$ steps, and from the search in the tree, in at most $(m-1)$ steps. The second statement of the theorem 4 determines the monomial: $\partial^{k_2}x^p/\partial x^q$. Furthemore, it is easy to calculate the exponent factor of the derivation, provided the total degree of the polynomial is known. Thus, the representation has to contain the total degree of the homogeneous polynomial:

$$u \equiv [k_1,(u_1,u_2,\ldots,u_{d_1})],\ v \equiv [k_2,(v_1,v_2,\ldots,v_{d_2})$$

If $u$ is homogeneous in $n$ variables $x_1,x_2,\cdots,x_n$ of degree $k_1$ and $v$ is homogeneous in the $m$-first variables

of degree $k_2$. Then $d_1 = C^{k_1}_{k_1+n-1}$ and $d_2 = C^{k_2}_{k_2+m-1}$ We suppose that the cost of the coefficients product is $\mathcal{O}(1)$. Then, the cost of the product $u.v$ is:

$$\mathcal{O}(nC^{k_1}_{k_1+n-1}C^{k_2}_{k_2+m-1}) \tag{3.8}$$

and the cost of the derivation of $u$ is given by:

$$\mathcal{O}(nC^{k_1}_{k_1+n-1}) \tag{3.9}$$

## 3.2 Cost of the Algorithm

We are going to evaluate the cost of the algorithm 2.2 when we compute the approximation of the normal form up to an order $k$. We start with the calculation of the cost of the operator $fog$, where $f, g$ belong to $V(k_1, n, n)$ and $V(k_2, m, n)$, respectively. Let $o_{[(k_1,n),(k_2,m)]}$ denote that cost. Then, the cost of $f \times g$ denoted by $\times_{[(k_1,n),(k_2,m)]}$ is given by

$$\times_{[(k_1,n),(k_2,m)]} = o_{[(k_1,n),(k_2,m)]} + o_{[(k_2,m),(k_1,n)]}$$

Let us suppose $p$ is an homogeneous polynomial in $n$ variables of degree $k_1$, and $q$ is homogeneous in $m$ variables of degree $k_2$. Let $C((k_1, n), (k_2, m))$ denote the cost of the product $p.q$ :

$$C((k_1, n), (k_2, m)) = nC^{k_1}_{k_1+n-1}.C^{k_2}_{k_2+m-1}$$

It is clear that the cost of $fog$ is given by the cost of the product $Df.g$ which is $n.C((k_1 - 1, n), (k_2, m))$, then:

$$o_{[(k_1,n),(k_2,m)]} = n^2.C^{k_1-1}_{k_1+n-2}.C^{k_2}_{k_2+m-1}$$

and, $\times_{[(k_1,n),(k_2,m)]}$ is equal to:

$$n^2.(C^{k_1-1}_{k_1+n-2}.C^{k_2}_{k_2+m-1} + C^{k_1}_{k_1+n-1}.C^{k_2-1}_{k_2+m-2}) \tag{3.10}$$

Moreover, the cost of the column $i$ in the triangle 1.10, when computations are performed to the order $k$, is

$$i \times (\sum_{j=1}^{k-i+1} \times_{[(i,n),(j,m)]}) \tag{3.11}$$

The summation of (3.11) on $i$, from 1 to $k$, provides the cost of the complete computation to the order k. Using (3.10), we obtain :

$$n^2(\sum_{i=1}^{k} i\{C^{i-1}_{n+i-2} \sum_{j=1}^{k-i+1} C^{j}_{m+j-1} + C^{i}_{n+i-1} \sum_{j=1}^{k-i+1} C^{j-1}_{m+j-2}\}).$$

which becomes

$$n^2(\sum_{i=1}^{k} i\{C^{i-1}_{n+i-2}(C^{k-i+1}_{m+k-i+1} - 1) + C^{i}_{n+i-1}C^{k-i}_{m+k-i}\}) \tag{3.12}$$

In order to simplify the above expression we introduce the two formulas:

$$\begin{aligned} rC^{r}_{p+r-1} &= pC^{r-1}_{p+r-1} \\ \textstyle\sum_{i=0}^{r} C^{i}_{p+i}C^{r-i}_{q+r-i} &= C^{r}_{p+q+r+1} \end{aligned} \tag{3.13}$$

and finally, we obtain

$$n^2(2(n-1)C^{k-1}_{n+m+k} + C^{k}_{n+m+k} - (m+1)C^{k-1}_{n+k-2} - C^{2}_{k+1}) \tag{3.14}$$

In order to obtain asymptotic behaviour of 3.14, we use Stierling formula which yields $C^{s}_{r+s} \sim s^r/r!$ when $s \longrightarrow \infty$. Then, when $k \longrightarrow \infty$ the cost of the computation is equivalent to

$$\frac{2n^3}{(n+m+1)!}k^{n+m+1} \tag{3.15}$$

# Conclusion

Firstly, we point out the fact that the same algorithm provides two usefull tools in the study of differential equations. Basically, this algorithm provides a linear system to be solved, at each step. Therefore, we suppose it is quite natural to represent the homogeneous polynomials in a vectorial form. This is why the data structure described in the last section is important. It will allow us to implement differentiation and multiplication of homogeneous polynomials efficiently. Finally, the cost of the algorithm given in the last section should be helpfull to compare this tool with other methods.

# References


[1] V. Arnold. *Chapitres supplementaires de la theorie des equations differentielles ordinaires.* Mir, 1980.

[2] J. Carr. *Applications of Center Manifold Theory.* Springer-Verlag, 1981.

[3] E. Ponce E. Freire, E. Gamero and L. Franquelo. An algorithm for symbolic computation of center manifolds. *Lecture Notes in Computer Sciences*, pages 218–230, 1989.

[4] G Chen L. Stolovitch, J. Della Dora. Nilpotent normal forms via carlemen linearization for systems of ode. *Proceedings ISSAC'91*, pages 281–288, 1991.

[5] Lawrence Perko. *Differential Equations and Dynamical Systems.* Springer-Verlag, 1991.

[6] Jack K. Hale Shui-Nee Chow. *Methods of Bifurcation Theory.* Springer-Verlag, 1982.

[7] Jan Sijbrand. Properties of center manifolds *Transactions of the American Mathematical Society*. 289(2):431–469, 1985.

[8] Laurent Stolovitch. Sur les formes normales de systemes nilpotents. *C.R. Acad. Sci. Paris*, 314(1992):355–358, 1992.

[9] A. Vanderbawhede. Center manifolds, normal forms and elementary bifurcations. *Prepublication. Universite de Nice*, 1985.