HAL

archives-ouvertes.fr

Generating series for drawing output of dynamical systems

Farida Benmakrouha, Christiane Hespel, Edouard Monnier

▶ To cite this version:

Generating series for drawing the output of dynamical systems

Farida Benmakrouha, Christiane Hespel, and Edouard Monnier

INSA-IRISA,
20 avenue des Buttes de Coesmes, 35043 Rennes cedex, France
{farida.benmakrouha,christiane.hespel,
edouard.monnier}@insa-rennes.fr

Abstract. We provide the drawing of the output of dynamical system (Σ), particularly when the output is rough or near instability points.
(Σ) being analytical in a neighborhood of the initial state $q(0)$ and described by its state equations, its output $y(t)$ in a neighborhood of $t = 0$ is obtained by "evaluating" its generating series.
Our algorithm consists in juxtaposing local approximating outputs on successive time intervals $[t_i, t_{i+1}]_{0 \leq i \leq n-1}$, to draw $y(t)$ everywhere as far as possible.
At every point t_{i+1} we calculate at order k an approximated value of each component q_r of the state; on every interval $[t_i, t_{i+1}]_{0 \leq i \leq n-1}$ we calculate an approximated output. These computings are obtained from the symbolic expressions of the generating series of q_r and y, truncated at order k, specified for $t = t_i$ and "evaluated".
A Maple package is built, providing a suitable result for oscillating outputs or near instability points when a Runge-Kutta method is wrong.

Keywords: Curve drawing, dynamical system, symbolic algorithm, generating series, oscillating output

1 Introduction

The usual methods for drawing the output of dynamical systems consist in providing the drawing of a piecewise linear function or at best, a smooth sketch. These methods are based on an iterative construction of isolated points (Runge-Kutta).
The problem is that this drawing disregards or smooths the oscillations which may be interesting in physics or biology. For instance, the detection of climate cycles in climatology, the study of the synchronization of oscillations in biology and particularly the pharmacodynamy describing the effect of a drug on the organism, would be processed by the knowledge of an exact drawing of the model. Our idea is to obtain a drawing which preserves the oscillations as far as possible. Rather than calculate numerous successive approximate points $y(t_i)_{i \in I}$, it can be interesting to provide some few successive local curves $\{y(t)\}_{t \in [t_i, t_{i+1}]_{0 \leq 1 \leq n-1}}$.

In a previous paper [2], we provided a method for drawing the solution curve of differential equation. We had to compute the new values of the output and its derivatives at every initial point $(t_i)_{0 \leq 1 \leq n-1}$. In the case when we consider a dynamical system, we have to calculate, for every interval $[t_i, t_{i+1}]$, the new state $q(t_i)$ instead of computing the new output $y(t_i)$ and its derivatives.

And then, we introduce the computing of the generating series $(G_{q_r,t})_{1 \leq r \leq N}$ associated with the rth component of the state $q = (q_1, \cdots, q_N)$ instead of only computing the generating series $G_{y,t}$ associated with the output.

The computing of these local curves can be kept partly generic since a generic expression of the generating series $G_{y,t}, (G_{q_r,t})_{1 \leq r \leq N}$ of the system can be provided in terms of (q_r, t). The expression of the local curves $\{y(t)\}_{t \in [t_i, t_{i+1}]}$ is only a specification for $t = t_i$ at order k of the formula given in section 3.

2   Preliminaries

2.1   Affine system, Generating series

We consider the nonlinear analytical system affine in the input:

$$(\Sigma) \qquad \begin{cases} \dot{q} = f_0(q) + \sum_{j=1}^m f_j(q)u_j(t) \\ y(t) = g(q(t)) \end{cases} \tag{1}$$

- $(f_j)_{0 \leq j \leq m}$ being some analytical vector fields in a neighborhood of $q(0)$
- g being the observation function analytical in a neighborhood of $q(0)$

Its initial state is $q(0)$ at $t = 0$. The generating series $G_{y,0}$, in noncommutative variables, is built on the alphabet $Z = \{z_0, z_1, \cdots, z_m\}$, z_0 coding the drift and z_j coding the input $u_j(t)$. Generally $G_{y,0}$ is expressed as a formal sum $G_{y,0} = \sum_{w \in Z^*} \langle G_{y,0}|w \rangle w$ where $\langle G_{y,0}|z_{j_0} \cdots z_{j_l} \rangle = f_{j_0} \cdots f_{j_l} g(q)|_{q(0)}$ depends on $q(0)$.

2.2   Fliess's formula and iterated integrals

The output $y(t)$ is given by the Fliess's equation ([4]):

$$y(t) = \sum_{w \in Z^*} \langle G_{y,0}|w \rangle \int_0^t \delta(w) \tag{2}$$

where $G_{y,0}$ is the generating series of (Σ) at $t = 0$:

$$\begin{aligned} G_{y,0} &= \sum_{w \in Z^*} \langle G_{y,0}|w \rangle w \\ &= g(q)|_{q(0)} + \\ &\quad \sum_{l \geq 0} \sum_{j_i=0}^m f_{j_0} \cdots f_{j_l} g(q)|_{q(0)} z_{j_0} \cdots z_{j_l} \end{aligned} \tag{3}$$

and $\int_0^t \delta(w)$ is the iterated integral associated with the word $w \in Z^*$.
The iterated integral $\int_0^t \delta(w)$ of the word w for the input u is defined by

$$\begin{cases} \int_0^t \delta(\epsilon) & = 1 \\ \int_0^t \delta(vz_i) = \int_0^t \left(\int_0^\tau \delta(v)\right) u_i(\tau)d\tau \\ \qquad \forall z_i \in Z \quad \forall v \in Z^*. \end{cases} \tag{4}$$

where ϵ is the empty word, $u_0 \equiv 1$ is the drift and $u_{i \in [1..m]}$ is the ith input.
We define the Chen's series as follows ([3])

$$C_u(t) = \sum_{w \in Z^*} \int_0^t \delta(w) \tag{5}$$

We set

$$\xi_{i,1}(t) = \int_0^t u_i(\tau)d\tau \tag{6}$$

From the previous definitions, we obtain the following expression:

$$y(t) = \sum_{w \in Z^*} \langle G_{y,0}|w\rangle\langle C_u(t)|w\rangle \tag{7}$$

By applying the Fliess formula to every component $q_r(t)_{1 \leq r \leq N}$ of the state $q(t)$ instead of $y(t)$, we obtain

$$q_r(t) = \sum_{w \in Z^*} \langle G_{q_r,0}|w\rangle\langle C_u(t)|w\rangle \tag{8}$$

by setting

$$\begin{aligned} G_{q_r,0} &= \sum_{w \in Z^*} \langle G_{q_r,0}|w\rangle w \\ &= proj_r(q)|_{q(0)} + \\ &\quad \sum_{l \geq 0} \sum_{j_i=0}^m f_{j_0} \cdots f_{j_l} proj_r(q)|_{q(0)} \\ &\quad z_{j_0} \cdots z_{j_l} \end{aligned} \tag{9}$$

where $proj_r(q) = q_r$.

3   Main results

3.1   Approximate values of the output $y(t)$ and of the state $(q_r(t))_{1 \leq r \leq N}$ in a neighborhood of $t = 0$

The Fliess's formula can be written

$$y(t) = \langle G_{y,0}|\epsilon\rangle + \sum_{w \in Z^*-\{\epsilon\}} \langle G_{y,0}|w\rangle\langle C_u(t)|w\rangle \tag{10}$$

An approximate function $y_k(t)$ de $y(t)$ up to order k in a neighborhood of $t = 0$ is obtained by expanding this expression up to the same order k. Then we have

$$|y(t) - y_k(t)| = O(t^{k+1}) \tag{11}$$

For instance, at order $k = 1$, $y(t)$ has the following approximate expression for a single input with drift

$$y_1(t) = \langle G_{y,0}|\epsilon \rangle + \langle G_{y,0}|z_0 \rangle t + \langle G_{y,0}|z_1 \rangle \xi_1(t) \tag{12}$$

where $\xi_k(t)$ denotes the kth primitive of $u(t)$.
This computing can be generalized $\forall r \in [1 \cdots N]$ components of the state q.

$$q_r(t) = \langle G_{q_r,0}|\epsilon \rangle + \sum_{w \in Z^* - \{\epsilon\}} \langle G_{q_r,0}|w \rangle \langle C_u(t)|w \rangle \tag{13}$$

by expanding these expressions up to order k.

3.2   Generalization at time $t = t_i$

For a single input with drift, the system (Σ) can be written at $t = t_i$:

$$\begin{cases} \dot{q}(t_i + h) = f_0(q(t_i + h)) + \\ \qquad\qquad f_1(q(t_i + h))u(t_i + h) \\ y(t_i + h) = g(q(t_i + h)) \end{cases} \tag{14}$$

By setting

$$\begin{cases} U_i(h) = u(t_i + h) \\ Y_i(h) = y(t_i + h) \\ Q_i(h) = q(t_i + h) \end{cases} \tag{15}$$

we obtain the following system

$$(\Sigma_i) \qquad \begin{cases} \dot{Q}_i(h) = f_0(Q_i(h)) + f_1(Q_i(h)U_i(h) \\ Y_i(h) = g(Q_i(h)) \end{cases} \tag{16}$$

And G_{y,t_i}, G_{q_r,t_i} are the generating series of (Σ_i).
By setting $\psi_{i,k}(h) = \xi_k(t_i + h)$, then $\psi_{i,k}(h)$ is the kth primitive of $u(t_i + h)$ or the kth primitive of $U_i(h)$.

We have the equalities

$$\xi_1(t_i + h) = \int_{t_i}^{t_i+h} u(\tau)d\tau = \int_0^h U_i(t)dt = \psi_{i,1}(h) \tag{17}$$

And then, we can prove recursively that the Chen's integral $\int_{t_i}^{t_i+h} \delta(w)$ can be computed as an integral $\int_0^t \delta(W)$ by considering $U_i(t)$ instead of $u(t_i + t)$.

4   Application: For drawing the output

We present an application to the drawing of the output of dynamical system. We consider the following dynamical system

$$\begin{cases} \dot{q}(t) = f_0(q(t)) + f_1(q(t))u(t) \\ y(t) = g(q(t)) \end{cases} \tag{18}$$

with initial conditions

$$q(0) = (q_1(0), \cdots, q_N(0))$$

We propose a curve drawing of the output $y(t)$ of this system in $[0, T] = \bigcup [t_i, t_{i+1}]_{0 \leq i \leq n-1}$ according to the following algorithm:

Firstly, we compute a generic approximated expression of the generating series $G_{y,t}, (G_{q_r,t})_{1 \leq r \leq N}$:

$$\begin{cases} G_{y,t} = \sum_{l \leq k} f_{j_0} \cdots f_{j_l} g(q)|_{q(t)} z_{j_0} \cdots z_{j_l} \quad = \sum_{|w| \leq k} \langle G_{y,t} | w \rangle w \\ G_{q_r,t} = \sum_{l \leq k} f_{j_0} \cdots f_{j_l} proj_r(q)|_{q(t)} z_{j_0} \cdots z_{j_l} = \sum_{|w| \leq k} \langle G_{q_r,t} | w \rangle w \end{cases} \tag{19}$$

We compute a generic approximated expressions of the output $y(t + h)$ and $q_r(t + h)$:

$$\begin{cases} y(t + h) = \sum_{|w| \leq k} \langle G_{y,t} | w \rangle \int_0^t \delta(W) \\ q_r(t + h) = \sum_{|w| \leq k} \langle G_{q_r,t} | w \rangle \int_0^t \delta(W) \end{cases} \tag{20}$$

– Initial point $t_0 = 0$:
$q_1(0), \cdots, q_N(0)$ are given.
The vector fields f_0, f_1 evaluated in t_0 provide $\langle G_{y,0} | w \rangle, (\langle G_{q_r,0} | w \rangle)_{1 \leq r \leq N}$ for $|w| \leq k$ and $y(0) = g(q(0))$.

– Step i:
Knowing $(q_r(t_{i-1}))_{1 \leq r \leq N}, y(t_{i-1})$ and $(\langle G_{q_r,t_{i-1}} | w \rangle)_{1 \leq r \leq N}, \langle G_{y,t_{i-1}} | w \rangle$ for $|w| \leq k$, we compute

$$\begin{cases} y(t_i) = y(t_{i-1} + h) \\ \quad = \sum_{|w| \leq k} \langle G_{y,t_{i-1}} | w \rangle \int_0^h \delta(W) \\ q_r(t_i) = q_r(t_{i-1} + h) \\ \quad = \sum_{|w| \leq k} \langle G_{q_r,t_{i-1}} | w \rangle \int_0^h \delta(W) \end{cases} \tag{21}$$

And we present $G_{y,t_i}, (G_{q_r,t_i})_{1 \leq r \leq N}$ by evaluating their generic expressions at $q(t_i)$. We draw the local curve of the function $t_{i-1} + dt \to y(t_{i-1} + dt)$ on the interval $[t_{i-1}, t_i]$.

– Final point $t = T = t_n$: stop at $i = n$.

4.1 Genericity of the method

The computing of the coefficients

$$\begin{cases} \langle G_{y,t_i}|z_{j_0}\cdots z_{j_l}\rangle = f_{j_0}\cdots f_{j_l}g(q)|_{q(t_i)} \\ \langle G_{q_r,t_i}|z_{j_0}\cdots z_{j_l}\rangle = f_{j_0}\cdots f_{j_l}proj_r(q)|_{q(t_i)} \end{cases} \tag{22}$$

is generic.

The computing of the expressions of

$$\begin{cases} Y_i(h) & = y(t_i+h) = y(t_i)+ \\ & \quad \sum_{|w|\le k}\langle G_{y,t_i}|w\rangle\langle C_{U_i}(h)|w\rangle \\ q_r(t_i+h) = q_r(t_i)+ \\ & \quad \sum_{|w|\le k}\langle G_{q_r,t_i}|w\rangle\langle C_{U_i}(h)|w\rangle \end{cases} \tag{23}$$

is generic too.

We use the previous algorithm by specifying t_i at every step in the previous expressions.

4.2 Example 1: Electric differential equation

$$y^{(1)}(t) + k_1 y(t) + k_2 y^2(t) = u(t)$$
$$y(0) = y_0, \tag{24}$$

It can be written as a first order differential system

$$\begin{cases} q^{(1)}(t) = -k_1 q(t) - k_2 q^2(t) + u(t) \\ \qquad = a(q)(t) + u(t) \\ y(t) \quad = q(t), q(0) = y_0 \end{cases} \tag{25}$$

The vector fields are

$$f_0(q) = -(k_1 q + k_2 q^2)\tfrac{d}{dq}$$
$$= a(q)\tfrac{d}{dq}$$
$$f_1(q) = \tfrac{d}{dq}$$

1. Generic expression of $G_{y,t_i} = G_{q,t_i}$
   Let us remark that

   $$\langle G_{y,t_i}|wz_1\rangle = 0 \ \ \forall w \in Z^+$$

   For instance, for order $k = 2$

   $$\begin{aligned} \langle G_{y,t_i}|\epsilon\rangle \quad &= q|_{q(t_i)} \\ \langle G_{y,t_i}|z_0\rangle \quad &= a(q)|_{q(t_i)} \\ \langle G_{y,t_i}|z_1\rangle \quad &= 1 \\ \langle G_{y,t_i}|z_0^2\rangle \quad &= a(q)\tfrac{d}{dq}a(q)|_{q(t_i)} \\ \langle G_{y,t_i}|z_1 z_0\rangle &= \tfrac{d}{dq}a(q)|_{q(t_i)} \end{aligned} \tag{26}$$

2. Generic expression of $Y_i(h)$
   for order $k = 2$

$$
\begin{aligned}
Y_i(h) &= y(t_i + h) \\
&= y(t_i) + \\
&\quad \langle G_{y,t_i} | z_0 \rangle h + \langle G_{y,t_i} | z_1 \rangle \psi_{i,1}(h) + \\
&\quad \langle G_{y,t_i} | z_0^2 \rangle h^2/2 + \langle G_{y,t_i} | z_1 z_0 \rangle \psi_{i,2}(h)
\end{aligned}
\tag{27}
$$

3. And we use the previous algorithm of section 4 by specifying t_i at every step. So we obtain the drawing of $y(t)$ (see the next section).

4.3   Example 2: Dynamical system with polynomial generating series [7]

$$
\begin{aligned}
\dot{q}_1 &= q_2 + q_3 u(t) \\
\dot{q}_2 &= 1 \\
\dot{q}_3 &= u(t) \\
y(t) &= q_1(t) \\
y(0) &= q_1(0) = q_{1,0}, q_2(0) = q_{2,0}, q_3(0) = q_{3,0}
\end{aligned}
\tag{28}
$$

It can be written as a dynamical system

$$
(\Sigma) \qquad
\begin{cases}
\dot{q} = f_0(q) + f_1(q) u(t) \\
y(t) = g(q(t))
\end{cases}
\tag{29}
$$

for

$$
\begin{aligned}
f_0 &= q_2 \frac{\partial}{\partial q_1} + \frac{\partial}{\partial q_2} \\
f_1 &= q_3 \frac{\partial}{\partial q_1} + \frac{\partial}{\partial q_3} \\
y(t) &= proj_1(q(t))
\end{aligned}
$$

1. Generic expression of $G_{y,t_i} = G_{q_1,t_i}$
   This series is polynomial:

$$
G_{q_1,t_i} = q_1(t_i) + q_2(t_i) z_0 + q_3(t_i) z_1 + z_0^2 + z_1^2
\tag{30}
$$

2. Generic expression of G_{q_2,t_i}
   This series is polynomial:

$$
G_{q_2,t_i} = q_2(t_i) + z_0
\tag{31}
$$

3. Generic expression of G_{q_3,t_i}
   This series is polynomial:

$$
G_{q_3,t_i} = q_3(t_i) + z_1
\tag{32}
$$

4. Generic expression of $Y_i(h) = q_1(t_i + h)$

$$
\begin{aligned}
Y_i(h) &= y(t_i + h) \\
&= y(t_i) + \\
&\quad \langle G_{y,t_i} | z_0 \rangle h + \langle G_{y,t_i} | z_1 \rangle \psi_{i,1}(h) + \\
&\quad \langle G_{y,t_i} | z_0^2 \rangle h^2/2 + \\
&\quad \langle G_{y,t_i} | z_1^2 \rangle 1/2 (\psi_{i,1}(h))^2
\end{aligned}
\tag{33}
$$

5. Generic expression of $q_2(t_i + h)$

$$q_2(t_i + h) = q_2(t_i) + h \tag{34}$$

6. Generic expression of $q_3(t_i + h)$

$$q_3(t_i + h) = q_3(t_i) + \psi_{i,1}(h) \tag{35}$$

The drawing of $y(t)$ is exact for order ≥ 2 since the 3 generating series are polynomials of degree ≤ 2.

4.4   Example 3: Dynamical system [1]

$$
\begin{aligned}
\dot{q}_1 &= sin(q_2) \\
\dot{q}_2 &= sin(q_3) \\
\dot{q}_3 &= q_4^3 + u(t) \\
\dot{q}_4 &= -q_4^3 - q_1^{10} \\
y(t) &= q_2(t) \\
y(0) &= q_2(0) = q_{2,0}, q_1(0) = q_{1,0}, \\
& \quad q_3(0) = q_{3,0}, q_4(0) = q_{4,0}
\end{aligned}
\tag{36}
$$

It can be written as a dynamical system

$$(\Sigma) \qquad \begin{cases} \dot{q} = f_0(q) + f_1(q)u(t) \\ y(t) = g(q(t)) \end{cases} \tag{37}$$

for

$$
\begin{aligned}
f_0 &= sin(q_2)\frac{\partial}{\partial q_1} + sin(q_3)\frac{\partial}{\partial q_2} + \\
& \quad q_4^3 \frac{\partial}{\partial q_3} + (-q_4^3 - q_1^{10})\frac{\partial}{\partial q_4} \\
f_1 &= \frac{\partial}{\partial q_3} \\
y(t) &= proj_2(q(t))
\end{aligned}
$$

1. Generic expression of $G_{y,t_i} = G_{q_2,t_i}$ up to order $k = 2$

$$
\begin{aligned}
G_{q_2,t_i} = q_2(t_i) &+ sin(q_3(t_i))z_0 + \\
& (q_4(t_i))^3 cos(q_3(t_i))z_0^2 + \\
& cos(q_3(t_i))z_1 z_0.
\end{aligned}
\tag{38}
$$

2. Generic expression of G_{q_1,t_i} up to order $k = 2$

$$
\begin{aligned}
G_{q_1,t_i} = q_1(t_i) &+ sin(q_2(t_i))z_0 + \\
& sin(q_3(t_i))cos(q_2(t_i))z_0^2.
\end{aligned}
\tag{39}
$$

3. Generic expression of G_{q_3,t_i} up to $k = 2$

$$
\begin{aligned}
G_{q_3,t_i} = q_3(t_i) &+ (q_4(t_i))^3 z_0 + z_1 + \\
& 3(-(q_4(t_i))^3 - (q_1(t_i))^{10})(q_4(t_i))^2 z_0^2.
\end{aligned}
\tag{40}
$$

4. Generic expression of G_{q_4,t_i} up to $k = 2$

$$\begin{aligned} G_{q_4,t_i} = q_4(t_i) + (&-(q_4(t_i)^3 - (q_1(t_i))^{10})z_0 + \\ &(3((q_4(t_i))^3 + (q_1(t_i)^{10}))(q_4(t_i))^2 \\ &-10(q_1(t_i))^9 sin(q_2(t_i)))z_0^2. \end{aligned}$$

(41)

5. Generic expression of $Y_i(h) = q_2(t_i + h)$ up to $k = 2$

$$\begin{aligned} Y_i(h) &= y(t_i + h) \\ &= y(t_i) + \langle G_{y,t_i}|z_0\rangle h + \\ &\langle G_{y,t_i}|z_0^2\rangle h^2/2 + \langle G_{y,t_i}|z_1 z_0\rangle(\psi_{i,2}(h)) \end{aligned}$$

(42)

6. Generic expression of $q_1(t_i + h)$ up to $k = 2$

$$\begin{aligned} q_1(t_i + h) = q_1(t_i) + &\langle G_{q_1,t_i}|z_0\rangle h + \\ &\langle G_{q_1,t_i}|z_0^2\rangle h^2/2. \end{aligned}$$

(43)

7. Generic expression of $q_3(t_i + h)$ up to $k = 2$

$$\begin{aligned} q_3(t_i + h) = q_3(t_i) + &\langle G_{q_3,t_i}|z_0\rangle h + \\ &\langle G_{q_3,t_i}|z_1\rangle(\psi_{i,1}(h)) + \\ &\langle G_{q_3,t_i}|z_0^2\rangle h^2/2. \end{aligned}$$

(44)

8. Generic expression of $q_4(t_i + h)$ up to $k = 2$

$$\begin{aligned} q_4(t_i + h) = q_4(t_i) + &\langle G_{q_4,t_i}|z_0\rangle h + \\ &\langle G_{q_4,t_i}|z_0^2\rangle h^2/2. \end{aligned}$$

(45)

4.5　Maple package: some demonstrations

In this section, we produce a demonstration in the following cases

- For stable system (electric equation with positive parameters) for oscillating input $u(t) = sin(100t)$, step=0.01 (Runge-Kutta vs our method). The drawings are similar by both methods. See Fig.1

- For stable system (electric equation, linear equation) with oscillating output (Runge-Kutta vs our method vs Exact Solution)

  1. Electric equation with positive parameters for oscillating input $u(t) = sin(100t)$, step= 0.5 (Runge-Kutta vs our method). The oscillations of the output are not described by Runge-Kutta method when our method diplays a lot of oscillations. See Fig.2.

  2. Linear equation for oscillating input $u(t) = t^2 sin(100t)$, step= 0.05 (Runge-Kutta vs our method vs Exact Solution). The drawing of the exact solution is similar to the drawing of our method. See Fig. 3,4,5.
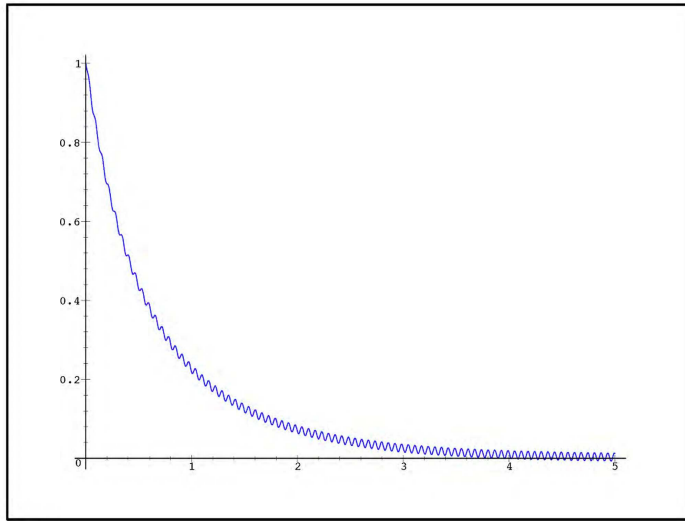
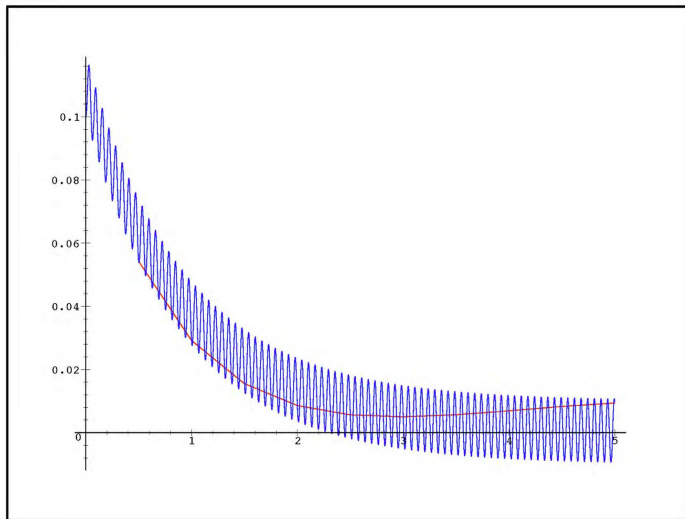**Fig. 1.** Stable system, oscillating input, small step, by Runge-Kutta or our method

Fig. 2. Stable system, oscillating input, large step, by Runge-Kutta (without oscillation) and our method
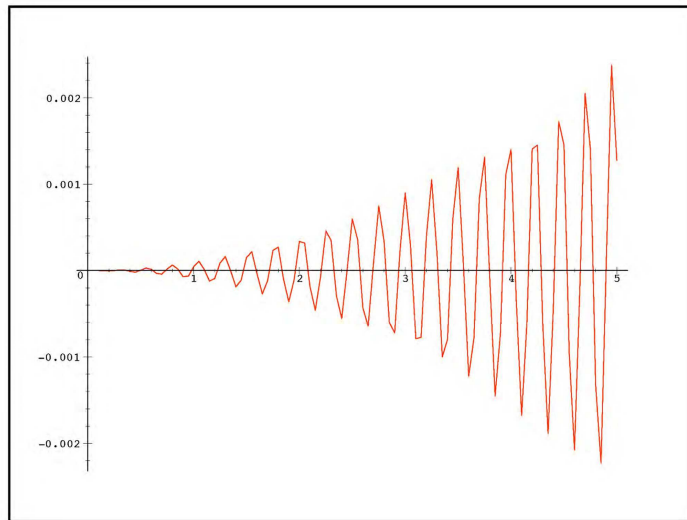
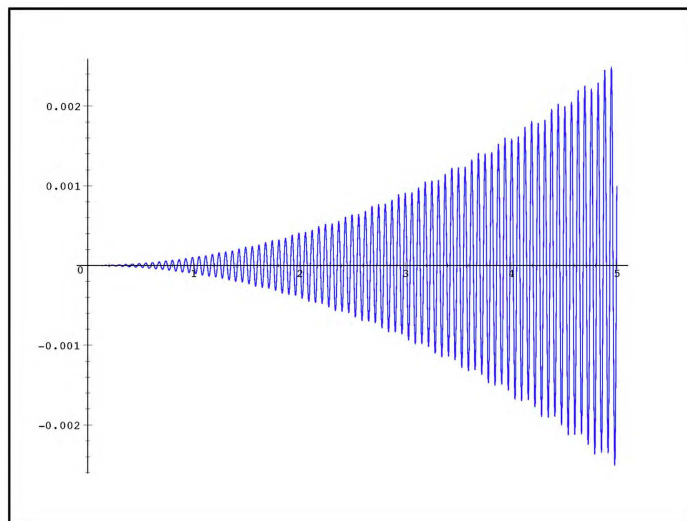**Fig. 3.** Linear equation, oscillating input, small step by Runge-Kutta

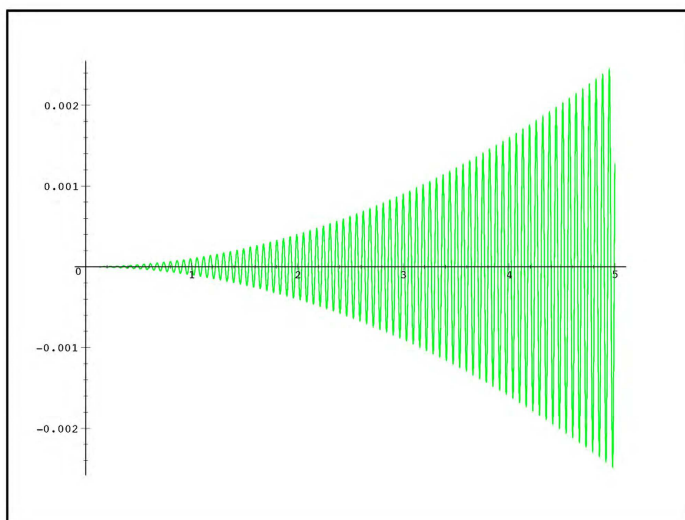**Fig. 4.** Linear equation, oscillating input, small step by our method

Fig. 5. Linear equation, oscillating input, small step by exact method

   3. Dynamical system (example 2) for oscillating input $u(t) = t^2 sin(10t)$, step= 0.5 (Our method at order 2 vs Exact Solution). The drawing of the exact solution is exactly the drawing of our method at order 2 (identical curves). See Fig.6.

– For unstable system (electric equation with negative parameters), $u(t) = sin(100t)$, step = 0.01., Runge-Kutta method notifies an error when our method displays a suitable curve. The Runge-Kutta method does not apply to this case when the drawing of our method displays an infinite branch. See Fig. 7.

5   Some practical applications

In biology, a lot of phenomenons are oscillating and sometimes cyclic. For instance, the oscillations of the hormonal secretions present some periodic activities. In the cellular cycle, the synchronization of the oscillations is basic.
An interesting application of our method in pharmacodynamy consists in modeling the effect of a drug on the organism in terms of time, in predicting the behaviour of the organism and in regulating the parameters of the organism by distributing suitably the drug in terms of time. This work is based on the associated drawings (in term of time) of the values of the drug and of the values of
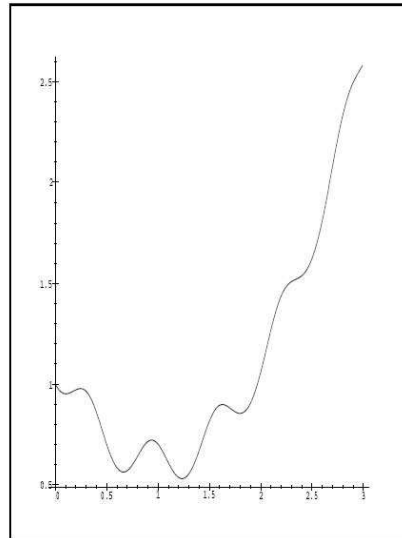
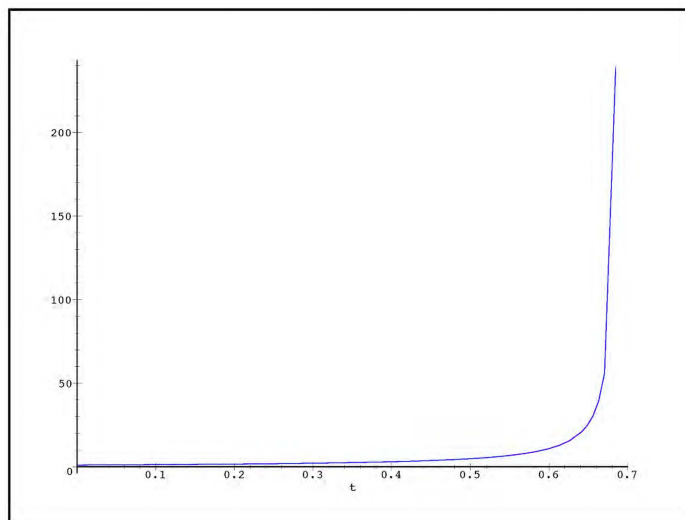**Fig. 6.** Dynamical system, oscillating input, by exact method and our method

Fig. 7. Unstable system, oscillating input, small step, by our method

the studied parameter.

In diabetology, by infusing insulin to diabetic patients, and by measuring the corresponding glycaemia, we obtain two associated curves (insulin delivery and glycaemia rate in the blood, in terms of time). With a sufficient set of these pairs, we can provide a modeling in the form of a dynamical system [11]. And then the prediction, regulation [8] can be processed by the knowledge of an accurate drawing of the model.

The detection of climate cycles in climatology is studied and several models are provided. The prediction of these cycles would be perhaps processed by the knowledge of an accurate drawing of a suitable model.

6   Conclusion

We develop a method for drawing the output of a dynamical system, based on the symbolic computing.

The symbolic computing allows us to profit from the genericity: We propose that one uses the formal expression of the generating series $G_{y,t}, G_{q_r,t}$, of the output $y(t)$ and the components of the state $(q_r)_{1 \leq r \leq N}$. Then we replace these expressions by their values at every step.

The symbolic computing allows us to profit from the precision: We can choose any order k for approximating the output. The error is on the order of $k + 1$.

And then an interest of this method consists in choosing the precision, not only by the size of the time interval h but by the order of the approximation.

The quality of any approximation depends on the order, the size of the interval but also depends on the roughness of the curve and the stability of the system. From a lot of examples, we express the following conclusions:

For stable systems with smooth outputs, our method and a Runge-Kutta method provide similar results.

For unstable systems, our methods allows us to obtain a suitable result near the instability points, when the Runge-Kutta methods give an error message.

For stable systems with rough or oscillating outputs, our method provides a suitable result when a Runge-Kutta method is wrong. Many applications in physics or biology are then available. The detection of climate cycles in climatology, the study of the synchronization of oscillations in biology, some problems of pharmacodynamy (modeling, prediction, regulation) would be partly processed by the knowledge of an accurate drawing of the model.

Acknowledgments

References

1. O. Akhrif, G.L. Blankenship. Algebraic computations for design of nonlinear control systems. Lecture Notes in Control and Information Sciences, Alge-

braic Computing in Control, G.Jacob, F.Lamnabhi-Lagarrigue Eds., vol.165, Springer Verlag, pp. 129-155, 1991.

2. F. Benmakrouha , C. Hespel , E. Monnier . Drawing solution curve of a differential equation. in "3rd International Conference on Complex Systems and Applications (ICCSA 2009)", Le Havre, 2009.

3. K.T. Chen . Iterated path integrals. Bull. Amer. Math. Soc. 83, pp. 831-879, 1977.

4. M. Fliess. Fonctionnelles causales non linéaires et indéterminés non commutatives. Bull. Soc. Math. France 109, pp. 3-40, 1981.

5. M. Fliess , C. Reutenauer. Théorie de Picard-Vessiot des systèmes réguliers. Colloque Nat.CNRS-RCP 567, Outils et Modèles Mathématiques pour l'Automatique,l'Analyse des Systèmes et le Traitement du Signal, Belle-Ile, Sept. 1982.

6. M. Fliess, M. Lamnabhi , F. Lamnabhi-Lagarrigue. An algebraic approach to nonlinear functional expansions. IEEE Trans. Circuits and Systems CAS-30, pp. 554-570, 1983.

7. M.V. Foursov, C. Hespel. On formal invertibility of the input/output behavior of dynamical systems. Publication interne IRISA, N° 1439, 2002.

8. M.V. Foursov, C. Hespel, On Algebraic Modeling and regulation of the behavior of diabetics, in "Innovative Technologies for Insulin Delivery and Glucose Sensing", Aix en Provence, August 2003.

9. C. Hespel. Iterated derivatives of a nonlinear dynamic system and Faà di Bruno formula. Mathematics and Computers in Simulation, vol.42, pp. 641-657, 1996.

10. C. Hespel.   Une étude des séries formelles non commutatives pour l'Approximation et l'Identification des systèmes dynamiques. Thèse d'état, Université de Lille 1, 1998.

11. C. Hespel, J.P. Hespel, E. Monnier, G. Jacob, M.V. Foursov, F. Benmakrouha, Algebraic Identification: application to insulin infusion, in "ISGIID'2000", Evian, Sept. 2000.