Drawing Solution Curve of Differential Equation

Farida Benmakrouha, Christiane Hespel, Edouard Monnier

Journal of Nonlinear Systems and Applications (2009) Copyright © 2009 Watam Press http://www.watam.org/JNSA/

DRAWING SOLUTION CURVE OF DIFFERENTIAL EQUATION *†

Farida Benmakrouha, Christiane Hespel, and Edouard Monnier

Abstract. We develop a method for drawing solution curves of differential equations. This method is based on the juxtaposition of local approximating curves on successive intervals $[t_i, t_{i+1}]_{0\leq i\leq n-1}$. The differential equation, considered as a dynamical system, is described by its state equations and its initial value at time t_0.
A generic expression of its generating series G_t truncated at any order k, of the output and its derivatives $y^{(j)}(t)$ expanded at any order k, can be calculated. The output and its derivatives $y^{(j)}(t)$ are expressed in terms of the coefficients of the series G_t and of the Chen series.
At the initial point t_i of every interval, we specify the expressions of G_t and $y^{(j)}(t)$. Then we obtain an approximated output $y(t)$ at order k in every interval $[t_i, t_{i+1}]_{0\leq i\leq n-1}$.
We have developed a Maple package corresponding to the creation of the generic expression of G_t and $y^{(j)}(t)$ at order k and to the drawing of the local curves on every interval $[t_i, t_{i+1}]_{0\leq i\leq n-1}$.
For stable systems with oscillating output, or for unstable systems near the instability points, our method provides a suitable result when a Runge-Kutta method is wrong.

Keywords. Curve drawing, differential equation, symbolic algorithm, generating series, dynamical system, oscillating output

1 Introduction

The usual methods for drawing curves of differential equations consist in an iterative construction of isolated points (Runge-Kutta). Rather than calculate numerous successive approximate points $y(t_i)_{i\in I}$, it can be interesting to provide some few successive local curves $\{y(t)\}_{t\in[t_i,t_{i+1}]_{0\leq 1\leq n-1}}$.
Moreover, the computing of these local curves can be kept partly generic since a generic expression of the generating series G_{t_i} of the system can be provided in terms of t_i. 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 the proposition of section 3.
We consider a differential equation

$$y^{(N)}(t) = \phi(t, y(t), \cdots, y^{(N-1)}(t)) \tag{1}$$

with initial conditions

$$y(0) = y_{0,0}, \cdots, y^{(N-1)}(0) = y_{0,N-1}$$

We assume that $\phi(t, y(t), \cdots, y^{(N-1)}(t))$ is polynomial in $y, \cdots, y^{(N-1)}$.
Then this differential equation can be viewed as an affine single input dynamical system.

2 Preliminaries

2.1 Affine system, Generating series

We consider the nonlinear analytical system affine in the input:

$$(\Sigma) \qquad \left\{ \begin{array}{rcl} \dot{q} & = & f_0(q) + \sum_{j=1}^{m} f_j(q)u_j(t) \\ y(t) & = & g(q(t)) \end{array} \right. \tag{2}$$

- $(f_j)_{0\leq j\leq m}$ 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_0 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_0 is expressed as a formal sum $G_0 = \sum_{w\in Z^*} \langle G_0|w\rangle w$ where $\langle G_0|z_{j_0}\cdots z_{j_l}\rangle = f_{j_0}\cdots f_{j_l} g(q)|_{q(0)}$ depends on $q(0)$.

2.2 Fliess's formula and iterated integrals

The output $y(t)$ is given by the Fliess's equation ([2]):

$$y(t) = \sum_{w\in Z^*} \langle G_0|w\rangle \int_0^t \delta(w) \tag{3}$$

where G_0 is the generating series of (Σ) at $t = 0$:

$$\begin{array}{rcl} G_0 & = & \sum_{w\in Z^*} \langle G_0|w\rangle w \\ & = & g(q)|_{q(0)} + \\ & & \sum_{l\geq 0} \sum_{j_i=0}^{m} f_{j_0}\cdots f_{j_l} g(q)|_{q(0)} z_{j_0}\cdots z_{j_l} \end{array} \tag{4}$$

and $\int_0^t \delta(w)$ is the iterated integral associated with the word $w \in Z^* = \{z_0, z_1, \cdots, z_m\}^*$.
The iterated integral $\int_0^t \delta(w)$ of the word w for the input u is defined by

$$\left\{ \begin{array}{rcl} \int_0^t \delta(\epsilon) & = & 1 \\ \int_0^t \delta(vz_i) & = & \int_0^t \left(\int_0^\tau \delta(v)\right) u_i(\tau)d\tau \\ & & \forall z_i \in Z \quad \forall v \in Z^*. \end{array} \right. \tag{5}$$

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 ([1])

$$C_u(t) = \sum_{w\in Z^*} \int_0^t \delta(w) \tag{6}$$

We set

$$\xi_{i,1}(t) = \int_0^t u_i(\tau)d\tau \tag{7}$$

From the previous definitions, we obtain the following expression:

$$y(t) = \sum_{w\in Z^*} \langle G_0|w\rangle\langle C_u(t)|w\rangle \tag{8}$$

2.3 Iterated derivatives $y^{(n)}(0)$ of the output

G_0 being the generating series of the system, the ith derivative of $y(t)$ is

$$y^{(i)}(t) = \langle G_0|C_u^{(i)}(t)\rangle \tag{9}$$

We prove the following lemma ([5]) based on the Picart-Vessiot theory ([3])

Lemma :
Let be $\sum_{0\leq j\leq m} u_j.z_j = \mathrm{A}$. Then the derivative of the Chen's series is $\frac{d}{dt}C_u = C_u.\mathrm{A}$

*F. Benmakrouha, C. Hespel, E. Monnier are with the Department of Computer Engineering, INSA-IRISA, 20 avenue des Buttes de Coesmes, 35043 Rennes cedex, France, E-mail: Benma@insa-rennes.fr, Hespel@insa-rennes.fr, Monnier@insa-rennes.fr

†Manuscript received October 5, 2009 .

From it, results the following recurrence relation:

$$C_u^{(i)} = C_u \mathrm{A_i}, \quad \mathrm{A_1} = \mathrm{A}, \quad \mathrm{A_{i+1}} = \mathrm{A}\mathrm{A_i} + D_t \mathrm{A_i} \tag{10}$$

D_t being the operator of time derivation.
Since $C_u(0) = 1$ and $C_u^{(i)}(0) = \mathrm{A_i}(0)$ then

$$y^{(i)}(0) = \sum_{w \in Z^*} \langle G_0|w\rangle \langle C_u^{(i)}(0)|w\rangle = \langle G_0|\mathrm{A_i}(0)\rangle \tag{11}$$

Let us remark that the successive derivatives $y(0), y^{(1)}(0), \cdots, y^{(k)}(0)$ are obtained from the coefficients $\langle G_0|w\rangle$ associated with the words whose length is $\leq k$.
It results that the Taylor expansion of $y(t)$ up to order k only depends on the coefficients of G_0 truncated at order k.
For instance, for a single input $u(t)$ with drift $u_0(t) \equiv 1$, the derivatives are the following

$$\begin{array}{lcl} y(0) & = & \langle G_0|\epsilon\rangle \\ y^{(1)}(0) & = & \langle G_0|z_0\rangle + \langle G_0|z_1\rangle u(0) \\ y^{(2)}(0) & = & \langle G_0|z_0^2\rangle + (\langle G_0|z_0 z_1\rangle + \langle G_0|z_1 z_0\rangle) u(0) + \\ & & \langle G_0|z_1^2\rangle u(0)^2 + \langle G_0|z_1\rangle u^{(1)}(0) \\ \cdots & = & \cdots \end{array} \tag{12}$$

This method allows us to compute recursively the successive derivatives of $y(t)$ at $t = 0$.

3 Main results

3.1 Approximate value of $y^{(n)}(t)$

The Fliess's formula can be written

$$y(t) = \langle G_0|\epsilon\rangle + \sum_{w \in Z^* - \{\epsilon\}} \langle G_0|w\rangle \langle C_u(t)|w\rangle \tag{13}$$

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

$$|y(t) - y_k(t)| = O(t^{k+1}) \tag{14}$$

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

$$y_1(t) = \langle G_0|\epsilon\rangle + \langle G_0|z_0\rangle t + \langle G_0|z_1\rangle \xi_1(t) \tag{15}$$

where $\xi_k(t)$ denotes the kth primitive of $u(t)$.
This computing can be generalized to the successive derivatives of $y(t)$.

Proposition
Given the expression of $y^{(n)}(0)$ in terms of the coefficients of G_0 and of the derivatives of order $\leq n-1$ of the input $u(t)_{t=0}$ obtained recursively according to the previous section, we can deduce the expression of $y^{(n)}(t)$ by executing in $y^{(n)}(0)$ the following transformations

1. **We substitute $u^{(i)}(t)$ to $u^{(i)}(0)$ for $0 \leq i \leq n-1$**
2. **For every occurrence of a coefficient $\langle G_0|v\rangle$ where $v \in Z^*$, we add the following corrective term**

$$\sum_{w \neq \epsilon} \langle G_0|wv\rangle \langle C_u(t)|w\rangle$$

The proof is based on the following properties

$$\left\{ \begin{array}{lcl} \frac{d}{dt}\langle C_u(t)|v z_i\rangle & = & \langle C_u(t)|v\rangle u_i(t) \\ \langle C_u(t)|\epsilon\rangle & = & 1 \end{array} \right. \tag{16}$$

For instance, for a single input with drift, we compute from

$$y^{(1)}(0) = \langle G_0|z_0\rangle + \langle G_0|z_1\rangle u(0)$$

the expression of $y^{(1)}(t)$:

$$\begin{array}{lcl} y^{(1)}(t) & = & \langle G_0|z_0\rangle + \sum_{w \neq \epsilon} \langle G_0|w z_0\rangle \langle C_u(t)|w\rangle + \\ & & (\langle G_0|z_1\rangle + \sum_{w \neq \epsilon} \langle G_0|w z_1\rangle \langle C_u(t)|w\rangle) u(t) \end{array} \tag{17}$$

By restricting the sums to the words w whose length $|w|$ satisfies $1 \leq |w| \leq k$, we obtain a function $y_k^{(n)}(t)$ approximating $y^{(n)}(t)$ up to order k. And then

$$|y_k^{(n)}(t) - y^{(n)}(t)| = O(t^{k+1}) \tag{18}$$

3.2 Generalization at time $t = t_i$

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

$$\left\{ \begin{array}{lcl} \dot{q}(t_i + h) & = & f_0(q(t_i + h)) + f_1(q(t_i + h)) u(t_i + h) \\ y(t_i + h) & = & g(q(t_i + h)) \end{array} \right. \tag{19}$$

By setting

$$\left\{ \begin{array}{lcl} U_i(h) & = & u(t_i + h) \\ Y_i(h) & = & y(t_i + h) \\ Q_i(h) & = & q(t_i + h) \end{array} \right. \tag{20}$$

we obtain the following system

$$(\Sigma_i) \qquad \left\{ \begin{array}{lcl} \dot{Q}_i(h) & = & f_0(Q_i(h)) + f_1(Q_i(h) U_i(h) \\ Y_i(h) & = & g(Q_i(h)) \end{array} \right. \tag{21}$$

And G_i is 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

$$\xi_1(t_i + h) = \int_{t_i}^{t_i + h} u(\tau) d\tau = \int_0^h U_i(t) dt = \psi_{i,1}(h) \tag{22}$$

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 to curves drawing

We present an application to the curve drawing of the solution of differential equations. We consider a differential equation

$$y^{(N)}(t) = \phi(t, y(t), \cdots, y^{(N-1)}(t)) \tag{23}$$

with initial conditions

$$y(0) = y_{0,0}, \cdots, y^{(N)}(0) = y_{0,N}$$

It can be written for $y = q_1$:

$$\left\{ \begin{array}{lcl} q_1^{(1)} & = & q_2 \\ q_2^{(1)} & = & q_3 \\ \cdots & = & \cdots \\ q_N^{(1)} & = & \phi(t, q_1, \cdots, q_N) \end{array} \right. \tag{24}$$

We assume that

$$\phi(t, q_1, \cdots, q_N) = P_0(q_1, \cdots, q_N) + \sum_{j=1}^{m} P_j(q_1, \cdots, q_N) u_j(t)$$

for $P_0, P_1, \cdots, P_N$ polynomials in commutative variables $q_1, \cdots, q_N$.

For an analytical affine single input system (Σ) then $m = 1$ and the vector fields are $f_0, \ f_1$, corresponding to P_0, P_1.

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 expression of the generating series G_t.

- Initial point $t_0 = 0$: $y(0) = q_1(0), \cdots, y^{(N-1)}(0) = q_N(0)$ are given.
  The vector fields f_0, f_1 applied to $g(q)$ evaluated in t_0 provide $\langle G_0|w\rangle$ for $|w| \leq k$
- Step i: Knowing $y(t_{i-1}) = q_1(t_{i-1}), \cdots, y^{(N-1)}(t_{i-1}) = q_N(t_{i-1})$ and $\langle G_{i-1}|w\rangle$. for $|w| \leq k$, we compute $y(t_i), \cdots, y^{(N-1)}(t_i)$ according to section 3 and $\langle G_i|w\rangle$ for $|w| \leq k$ by applying the vector fields f_0, f_1 to $g(q)$ 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

$$\langle G_i|z_{j_0} \cdots z_{j_l}\rangle = f_{j_0} \cdots f_{j_l} g(q)|_{q(t_i)}$$

is generic.
The computing of the expressions of

$$Y_i(h) = y(t_i + h) = y(t_i) + \sum_{|w| \leq k} \langle G_i|w\rangle\langle C_{U_i}(h)|w\rangle$$

and of

$$\begin{aligned} Y_i^{(1)}(h) \quad = \quad & \langle G_i|z_0\rangle + \textstyle\sum_{1\leq|w|\leq k} \langle G_i|wz_0\rangle\langle C_{U_i}(h)|w\rangle + \\ & (\langle G_i|z_1\rangle + \textstyle\sum_{1\leq|w|\leq k} \langle G_i|wz_1\rangle\langle C_{U_i}(h)|w\rangle)U_i(h) \end{aligned} \tag{25}$$

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

4.2 Example 1: Duffing equation

Its equation is the following:

$$\begin{aligned} y^{(2)}(t) + ay^{(1)}(t) + by(t) + cy^3(t) \quad &= \quad u(t) \\ y(0) = y_0, & \\ y^{(1)}(0) = y_{1,0} & \end{aligned} \tag{26}$$

It can be written as a first order differential system

$$\begin{cases} q_1^{(1)}(t) & = \quad q_2(t) \\ q_2^{(1)}(t) & = \quad -aq_2(t) - bq_1(t) - cq_1^3(t) + u(t) \\ & = \quad F(q(t)) + u(t) \\ y(t) & = \quad q_1(t) = g(q) \\ q_1(0) = y_0, & \quad\;\; q_{2,0} = y_{1,0} \end{cases} \tag{27}$$

The vector fields are

$$\begin{aligned} f_0(q_1, q_2) \quad &= \quad q_2 \frac{\partial}{\partial q_1} - (aq_2 + bq_1 + cq_1^3)\frac{\partial}{\partial q_2} \\ &= \quad q_2 \frac{\partial}{\partial q_1} + F(q)\frac{\partial}{\partial q_2} \\ f_1(q_1, q_2) \quad &= \quad \frac{\partial}{\partial q_2} \end{aligned}$$

1. We write generic equations describing the generating series G_i at $t = t_i$:

   $$\forall t_i \quad \langle G_i|z_{j_1} \cdots z_{j_l}\rangle = (f_{j_1} \cdots f_{j_l} g(q))|_{q(t_i)}$$

   Let us remark that

   $$\langle G_i|wz_1\rangle = 0 \ \ \forall w \in Z^*, \quad \langle G_i|wz_1z_0\rangle = 0 \ \ \forall w \in Z^+$$

   For instance, for order $k = 3$, we have only to compute 6 coefficients of G_i instead of 15 coefficients.

   $$\begin{aligned} \langle G_i|\epsilon\rangle \quad &= \quad q_1(t_i) \\ \langle G_i|z_0\rangle \quad &= \quad q_2(t_i) \\ \langle G_i|z_0^2\rangle \quad &= \quad F(q(t_i)) \\ \langle G_i|z_1z_0\rangle \quad &= \quad 1 \\ \langle G_i|z_0^3\rangle \quad &= \quad (q_2 \frac{\partial}{\partial q_1} F(q) + F(q)\frac{\partial}{\partial q_2} F(q))_{q(t_i)} \\ \langle G_i|z_1z_0^2\rangle \quad &= \quad -a \end{aligned} \tag{28}$$

2. We write generic approximate expression of the output $y(t_{i+1})$ and its derivative $y^{(1)}(t_{i+1})$ for every $t = t_{i+1} = t_i+h$ at order k:

   $$\begin{aligned} & y(t_{i+1}) = \langle G_i|\epsilon\rangle + \textstyle\sum_{1\leq|w|\leq k} \langle G_i|w\rangle\langle C_{U_i}(h)|w\rangle \\ & y^{(1)}(t_{i+1}) = \langle G_i|z_0\rangle + \\ & \textstyle\sum_{1\leq|w|\leq k} \langle G_i|wz_0\rangle\langle C_{U_i}(h)|w\rangle + \\ & (\langle G_i|z_1\rangle + \textstyle\sum_{1\leq|w|\leq k} \langle G_i|wz_1\rangle\langle C_{U_i}(h)|w\rangle)U_i(h) \end{aligned} \tag{29}$$

   For instance, for order $k = 3$

   $$\begin{aligned} Y_i(h) \quad &= \quad y(t_i + h) \\ &= \quad y(t_i) + \langle G_i|z_0\rangle h + \langle G_i|z_0^2\rangle h^2/2 + \\ & \quad\quad \langle G_i|z_1z_0\rangle\psi_{i,2}(h) + \langle G_i|z_0^3\rangle h^3/(3!) + \\ & \quad\quad \langle G_i|z_1z_0^2\rangle\psi_{i,3}(h) \end{aligned} \tag{30}$$

   and

   $$\begin{aligned} Y_i^{(1)}(h) \quad &= \quad y^{(1)}(t_i + h) \\ &= \quad \langle G_i|z_0\rangle + \langle G_i|z_0^2\rangle h + \\ & \quad\quad \langle G_i|z_1z_0\rangle\psi_{i,1}(h) + \langle G_i|z_0^3\rangle h^2/2 + \\ & \quad\quad \langle G_i|z_1z_0^2\rangle\psi_{i,2}(h) \end{aligned} \tag{31}$$

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

4.3 Example 2: Electric equation

$$\begin{aligned} y^{(1)}(t) + k_1y(t) + k_2y^2(t) \quad &= \quad u(t) \\ y(0) = y_0, & \end{aligned} \tag{32}$$

It can be written as a first order differential system

$$\begin{cases} q^{(1)}(t) & = \quad -k_1q(t) - k_2q^2(t) + u(t) \\ & = \quad a(q(t)) + u(t) \\ y(t) & = \quad q(t), q(0) = y_0 \end{cases} \tag{33}$$

The vector fields are

$$\begin{aligned} f_0(q) \quad &= \quad -(k_1q + k_2q^2)\frac{d}{dq} \\ &= \quad a(q(t))\frac{d}{dq} \\ f_1(q) \quad &= \quad \frac{d}{dq} \end{aligned}$$

1. Generic expression of G_i
   Let us remark that

   $$\langle G_i|wz_1\rangle = 0 \ \ \forall w \in Z^+$$

   For instance, for order $k = 2$

   $$\begin{aligned} \langle G_i|\epsilon\rangle \quad &= \quad q(t_i) \\ \langle G_i|z_0\rangle \quad &= \quad a(q(t_i)) \\ \langle G_i|z_1\rangle \quad &= \quad 1 \\ \langle G_i|z_0^2\rangle \quad &= \quad a(q(t))\frac{d}{dq}a(q(t)) \\ \langle G_i|z_1z_0\rangle \quad &= \quad \frac{d}{dq}a(q(t)) \end{aligned} \tag{34}$$

2. Generic expression of $Y_i(h), Y^{(1)}(h)$
   for order $k = 2$

   $$\begin{aligned} Y_i(h) \quad &= \quad y(t_i + h) \\ &= \quad y(t_i) + \langle G_i|z_0\rangle h + \langle G_i|z_1\rangle\psi_{i,1}(h) + \\ & \quad\quad \langle G_i|z_0^2\rangle h^2/2 + \langle G_i|z_1z_0\rangle\psi_{i,2}(h) \end{aligned} \tag{35}$$

   and

   $$\begin{aligned} Y_i^{(1)}(h) \quad &= \quad y^{(1)}(t_i + h) \\ &= \quad \langle G_i|z_0\rangle + \langle G_i|z_1\rangle U_i(h) + \\ & \quad\quad \langle G_i|z_0^2\rangle h + \langle G_i|z_1z_0\rangle\psi_{i,1}(h) \end{aligned} \tag{36}$$

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

4.4 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)

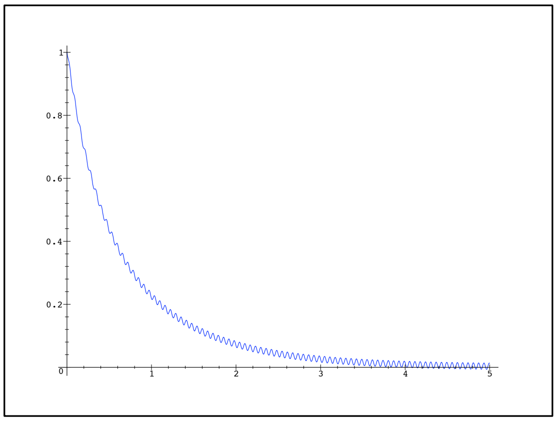

Fig.1: Stable system, oscillating input, small step, by Runge-Kutta or our method

The drawings are similar by both methods (Runge-Kutta or our method).

- 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)

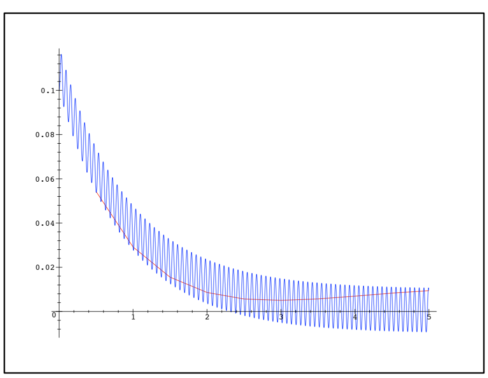

Fig.2: Stable system, oscillating input, large step, by Runge-Kutta (without oscillation) and our method

The oscillations of the output are not described by Runge-Kutta method when our method diplays a lot of oscillations.

2. Linear equation for oscillating input $u(t) = t^2 sin(100t)$, step= 0.05 (Runge-Kutta vs our method vs Exact Solution)

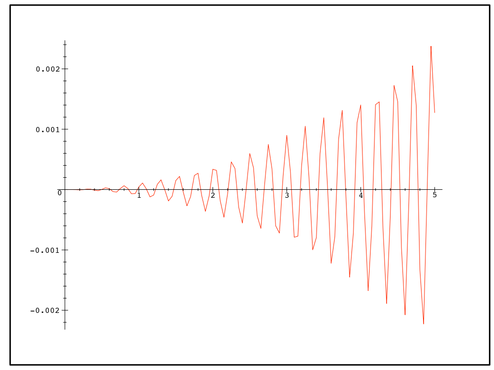

Fig.3: Linear equation, oscillating input, small step by Runge-Kutta

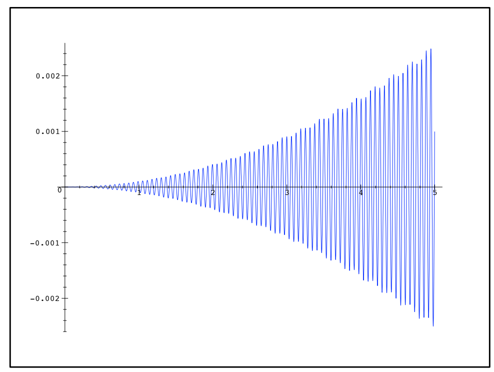

Fig.4: Linear equation, oscillating input, small step by our method

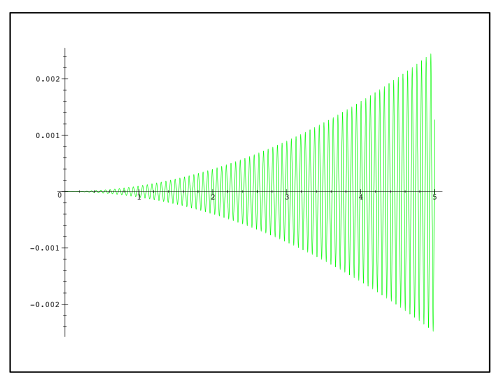

Fig.5: Linear equation, oscillating input, small step by exact method

The drawing of the exact solution is similar to the drawing of our method.

- 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.

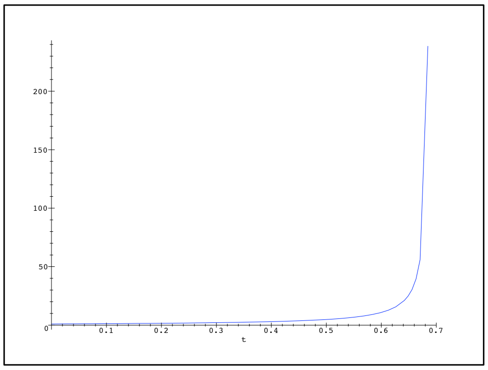

Fig.6: Unstable system, oscillating input, small step, by our method

The Runge-Kutta method does not apply to this case when the drawing of our method displays an infinite branch.

5 Conclusion

We develop a method for drawing a solution curve of a differential equation, 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_i and of the output $y(t_i)$ and its derivative $y^{(1)}(t_i)$. 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 and its derivative. 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.

References

[1] Chen K.T., *Iterated path integrals*, Bull. Amer. Math. Soc. 83, pp. 831-879, 1977.

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

[3] Fliess M., Reutenauer C., *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.

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

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

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