Analysis of ODE using Hybrid Computation

Antoine Girard

LMC-IMAG, 51 rue des Mathématiques,
38041 Grenoble Cedex 9, France
Antoine.Girard@imag.fr

Abstract. Hybrid Computation is a new field of numerical analysis. It can be seen as the application of the philosophy of hybrid systems to the field of computation. In this paper we introduce an alternative to classical methods (e.g. Runge Kutta) for the analysis of differential equations. Our method differs mainly of classical methods by the fact that we use a discretization of the phase space instead of the time space. Our method and its variations have very good theoretical results (e.g. order up to 6 with 3 evaluations of f at each step) which are strengthened by numerical experiments.

1 Introduction

The fundamental concepts of numerical algorithms for the analysis of ordinary differential equations have not evolved a lot through the latest years. Indeed, we still have recourse to classical methods such as Runge-Kutta (see [3], [4]). In this paper, we introduce a new way to take up numerical analysis of dynamical systems, it has been called hybrid computation (see [1]). We use the word hybrid for two main reasons:

- It is based on some ideas coming from computer science and more precisely from the field of hybrid systems.
- It is at the intersection of numerical analysis and computer algebra.

Hybrid systems are usually presented ([8]) as systems with discrete and continuous dynamics. Some classical examples are automotive engines whose fuel injection can be seen as a continuous process regulated by some microprocessor (the discrete part of the system).
The idea of hybrid computation is to replace the study of a non linear differential equation by the study of an approximating piecewise differential equation. We can justify this process by the following remark: most of the models that physicists, biologists and economists are working with are approximate models. Indeed, the principal work of modeling is the approximation itself ([9]). It is a fact that we take into account in hybrid computation.
In this paper we present a method for the analysis of scalar autonomous differential equations. Given a system $\dot{x} = f(x)$, we choose f_h a simple approximant (for example piecewise linear) of the function f and we consider the system

$\dot{x}_h = f_h(x_h)$. Then, the question is in which way does the solutions of the second system approximates the solutions of the first one?
After the description of the method thanks to an example, we will define the theorical frame of our study. We will give theoretical results of convergence and stability of a numerical implementation of hybrid computation. We will also talk about the choice of the approximating system, and finally make a theoretical and experimental comparison with classical methods of integration of differential equations.

2 The Example $\dot{x} = x^2$

Let us explain the idea with a simple example. Consider the initial value problem:

$$\dot{x}(t) = (x(t))^2,\ x(0) = 1\ . \tag{1}$$

The solution of 1 is $x(t) = 1/(1-t)$. We are now going to approximate $x(t)$ by the solution $x_h(t)$ of a piecewise linear hybrid system. An important point of the definition of hybrid systems ([1], [8]) is that we have a decomposition of the phase space (that is the x-axe). Let h be a positive real, the x-axe is cut into intervals $I_i = [ih, (i+1)h[$. Note that we do not cut the time space (as is done with classical methods) but the phase space itself. The different intervals of the decomposition are the discrete states of the system. On each interval we define a vector field f_i that approximates locally the vector field $f(x) = x^2$, for example the linear interpolant of f with respect to I_i:

$$f_i(x) = a_i x + b_i$$

with

$$a_i = \frac{[(i+1)h]^2 - [ih]^2}{(i+1)h - ih} = (2i+1)h$$

and

$$b_i = (ih)^2 - a_i(ih) = ih^2[i - (2i+1)] = -i(i+1)h^2\ .$$

Let $x_h(t)$ be the solution of:

$$\dot{x}_h(t) = f_i(x_h(t)),\ \text{if } x_h(t) \in I_i,\ x_h(0) = 1\ . \tag{2}$$

For the computation of the transitions between two succesive intervals, we impose that x_h is a continuous function of the time. Consequently,

$$x_h(t) = (ih + \frac{b_i}{a_i})e^{a_i(t-t_i)} - \frac{b_i}{a_i},\ \text{if } x_h(t) \in I_i$$

where t_i is the time value such that $x_h(t_i) = ih$. Hence,

$$x_h(t) = \frac{i^2 h}{2i+1} e^{(2i+1)h(t-t_i)} + \frac{i(i+1)h}{2i+1},\ \text{if } x_h(t) \in I_i\ .$$

The continuity of the solution gives us

$$t_{i+1} - t_i = \frac{2}{(2i+1)h} \ln\left(\frac{i+1}{i}\right) .$$

Here is the big difference with classical methods; indeed the time subdivision is given by the approximating system and is not fixed a priori.
Let us take $h = 1/N$ with $N \in \mathbb{N}$. We set $x_h(0) = 1 = Nh$, so we have $t_N = 0$. We remark that the solution of 1 has a vertical asymptote at $t = 1$. If $x_h(t)$ is a good approximant of $x(t)$, $x_h(t)$ should also have a vertical asymptote. We should have:

$$\lim_{i \to \infty} t_i = \sum_{i \geq N} (t_{i+1} - t_i) < \infty .$$

Moreover, it should converge to 1 as $h \to 0$:

$$\sum_{i \geq N} (t_{i+1} - t_i) \xrightarrow[N \to \infty]{} 1 .$$

It is easy to show that we have

$$2N \int_N^{\infty} \frac{1}{(2x+1)} \ln\left(\frac{x+1}{x}\right) dx \leq \sum_{i \geq N} (t_{i+1} - t_i) \leq 2N \int_N^{\infty} \frac{1}{(2x-1)} \ln\left(\frac{x}{x-1}\right) dx .$$

It can be shown that the limit of these integrals is 1 as $N \to \infty$ which is the expected result.

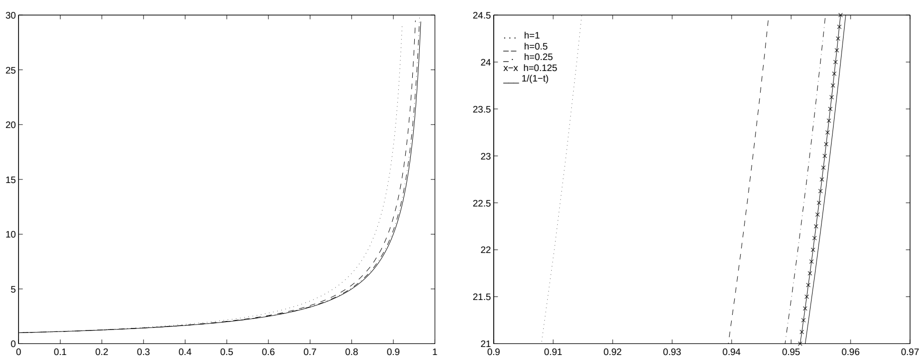

Fig. 1. Solution of (2) for $h \in \{0.125, 0.25, 0.5, 1\}$ and solution of (1) (i.e. $1/(1-t)$) (*full*)

In Fig. 1, we have plotted $x(t)$ and $x_h(t)$ for different values of h. We did not plot the approximating solutions for small values of h since the approximation was so good that one was unable to distinguish the representative curve of $x(t)$ from the one of $x_h(t)$. Experimentally we find an order 2 for our method. We will see in section 4 that this is indeed the case.

3 General formulation and algorithm

In this section, we define the theoretical frame of our study. Afterwards, we will give the algorithm for the numerical implementation of our method.

3.1 General formulation

We consider scalar autonomous differential equations. Thus, let us consider the equation:

$$\dot{x}(t) = f(x(t)),\ x(t) \in [a,b],\ x(0) = x_0\ . \tag{3}$$

We define a uniform subdivision $(\nu_i)_{i\in\{0\dots N\}}$ of size h of the interval $[a,b]$. On each interval $[\nu_i, \nu_{i+1}[$, we note f_i the linear interpolant of f with respect to ν_i and ν_{i+1}:

$$f_i(x) = a_i x + b_i$$

with

$$a_i = \frac{f(\nu_{i+1}) - f(\nu_i)}{h}$$

and

$$b_i = f(\nu_i) - a_i \nu_i\ .$$

Afterwards, we define the piecewise linear function:

$$f_h(x) = f_i(x) \text{ if } x \in [\nu_i, \nu_{i+1}[\ . \tag{4}$$

We have the following classical result (from interpolation theory):

$$\forall x \in [a,b],\ |f(x) - f_h(x)| \le \sup_{[a,b]} |f''| \frac{h^2}{2}\ . \tag{5}$$

Hybrid computation consists in replacing the study of 3 by:

$$\dot{x}_h(t) = f_h(x_h(t)),\ x_h(t) \in [a,b],\ x_h(0) = x_0\ . \tag{6}$$

Let us assume that we choose an initial condition x_0 so that we have $f_h(x_0) > 0$. Then, the solutions of the system are given by:

$$x_h(t) = \begin{cases} \nu_i + b_i(t - t_i) & \text{if } f(\nu_i) = f(\nu_{i+1}) \\ (\nu_i + \frac{b_i}{a_i})e^{a_i(t-t_i)} - \frac{b_i}{a_i} & \text{in the other cases} \end{cases} \text{ if } x_h(t) \in [\nu_i, \nu_{i+1}[\tag{7}$$

where t_i is the time value such that $x_h(t_i) = \nu_i$.
The computation of $x_h(t)$ essentially consists in computing the serie (t_i), which is given by the following formulas:

$$t_{i+1} = \begin{cases} \text{not defined} & \text{if } f(\nu_{i+1}) \le 0 \\ t_i + \frac{h}{b_i} & \text{if } f(\nu_i) = f(\nu_{i+1}) \\ t_i + \frac{\ln(f(\nu_{i+1})) - \ln(f(\nu_i))}{a_i} & \text{in the other cases.} \end{cases} \tag{8}$$

3.2 Algorithm

The computation of $x_h(t)$ can be done easily thanks to the formulas given in the previous paragraph. Indeed, $x_h(t)$ can be computed thanks to the following algorithm:

- Initialization
  We must before all, find the interval of the subdivision in which x_0 lies:
  Find i so that $x_0 \in [\nu_i, \nu_{i+1}[$.
- Main loop
  We must solve the equation (6) on the interval $[\nu_i, \nu_{i+1}[$:
  Interpolation of f to the points ν_i, ν_{i+1}.
  Computation of t_{i+1} (see equation 8). Please note that there is a little variation for the first step of the algorithm. Indeed, we do not know t_i, but t_{i+1} can be computed by using the fact that $x_h(0) = x_0$.
  If $t_{i+1} \geq t$ or t_{i+1} is not defined then $x_h(t)$ is in the interval $[\nu_i, \nu_{i+1}[$:
    compute $x_h(t)$ (see equation 7)
    return $x_h(t)$.
  else (we compute the next interval)
    $i = i + 1$.

4 Convergence and Stability

In this section we give the proof of the convergence of the method, a result about the stability of its numerical implementation and some experimental results.

4.1 Theoretical convergence

It is useful, for the the proof of the convergence to recall a fundamental theorem of the theory of ordinary differential equations.

Theorem 1 (Fundamental Inequality [2]). *If on an interval $[a, b]$ the differential equation $\dot{x} = f(x)$ satisfies a Lipschitz condition with Lipschitz constant $K \neq 0$ and if $u_1(t)$ and $u_2(t)$ are two continuous, piecewise differentiable, functions satisfying*

$$|\dot{u}_i(t) - f(u_i(t)| \leq \epsilon_i$$

for all t at which $u_1(t)$ and $u_2(t)$ are differentiable; and if

$$|u_1(0) - u_2(0)| \leq \delta$$

then for all t where u_1 and u_2 are defined,

$$|u_1(t) - u_2(t)| \leq \delta e^{K|t|} + \frac{\epsilon}{K}(e^{K|t|} - 1)$$

where $\epsilon = \epsilon_1 + \epsilon_2$.

The proof of the convergence is an immediate corollary of this theorem.

Corollary 1 (Convergence of our method). *Let the system $\dot{x} = f(x)$ be defined on an interval $[a, b]$. Assuming that f is C^2 and K-Lipschitz on $[a, b]$; let f_h be the piecewise linear function defined as in 4.*
If $x(t)$ is the solution of the initial value problem $\dot{x} = f(x)$, $x(0) = x_0$ and $x_h(t)$ the solution of $\dot{x}_h = f_h(x_h)$, $x_h(0) = x_0$ then for all t where x and x_h are defined,

$$|x(t) - x_h(t)| \leq \frac{\sup_{[a,b]} |f''|}{2K}(e^{K|t|} - 1)h^2 \ .$$

Proof. We have

$$|\dot{x}(t) - f(x(t))| = 0$$

and equation 5 gives

$$|\dot{x}_h(t) - f(x_h(t))| = |f_h(x_h(t)) - f(x_h(t))| \leq \sup_{[a,b]} |f''| \frac{h^2}{2} \ .$$

Moreover $|x_h(0) - x(0)| = 0$, consequently the fundamental inequality gives us

$$|x(t) - x_h(t)| \leq \frac{\sup_{[a,b]} |f''|}{2K}(e^{K|t|} - 1)h^2 \ .$$

4.2 Stability

Before giving a result about the stability of our method, we must answer to the following question: what is stability for hybrid computation? Since the main loop of the algorithm consists in the computation of the serie (t_i), we should consider how the numerical errors occur in the computation of the (t_i). Consequently, we will say that hybrid computation is stable if the numerical errors that we have made in the computation of $(t_i)_{i \leq p}$ occurs only linearly in the numerical evaluation of t_{p+1}.

Proposition 1. *Hybrid computation is stable.*

Proof. The equation 8 tells us that t_{i+1} depends only linearly of t_i. Consequently:

$$t_{i+1} = t_i + G(\nu_i, \nu_{i+1}, f, a_i, b_i, h)$$

Now we define τ_i to be the numerical evaluation of t_i, then

$$\tau_{i+1} = \tau_i + G(\nu_i, \nu_{i+1}, f, a_i, b_i, h) + \epsilon_i$$

and

$$|\tau_p - t_p| \leq \sum_{i=0}^{i=p-1} |\epsilon_i| \ .$$

This gives the proof of the stability of hybrid computation.

4.3 Experimental results

We made a numerical implementation of hybrid computation for the example presented in section 2. We computed the value of $x_h(0.95)$ for different values of the parameter h. We obtained the results presented in the table 1.
We checked that $x_h(t)$ approximates $x(t)$ with the precision $O(h^2)$ which was the expected result. We can also notice that the complexity of the method is linear with respect to $1/h$ which is a reasonable cost.

Table 1. Experimental results (Dec Alpha 500 MHz, C++) of convergence for the example of the section 2.

h-value	Error at $t = 0.95$	Run Time (Seconds)
10^{-1}	$2.2397\ 10^{-1}$	0
10^{-2}	$2.2221\ 10^{-3}$	0.001952
10^{-3}	$2.2219\ 10^{-5}$	0.020496
10^{-4}	$2.2219\ 10^{-7}$	0.203008
10^{-5}	$2.2304\ 10^{-9}$	2.01739

4.4 When does the method fail?

There are some cases where hybrid computation does not work. It may happen when the solution of 3 has a horizontal asymptote (i.e. there exists ξ in $[a, b]$ such that $f(\xi) = 0$).
This case is illustrated in the Fig. 2 for the system $\dot{x} = 1+\sin(x)$, with $x(0) = 1$. The solution of this initial value problem has a horizontal asymptote at $x = 3\pi/2$, but the approximants of the solution generally do not have any horizontal asymptote. This is due to the fact that $f(x) = 1+\sin(x)$ has a zero at $x = 3\pi/2$ but is strictly positive in a neighbourhood of this zero. This kind of problem is directly related to the structural stability of the system 3.
The equation $\dot{x} = 1 + \sin(x)$ is clearly unstable since for any pertubation $\epsilon > 0$, the system $\dot{x} = 1 + \sin(x) + \epsilon$ has no equilibrium.

However, hybrid computation works for almost all differential equations:

Proposition 2. *Let us consider equation 3, we assume that f is $C^2([a, b])$. Let f_h be defined as in 4. If $f(\xi) = 0$ and $f'(\xi) \neq 0$, then there exists ϵ so that; for all $h < \epsilon$, $x(t)$ and $x_h(t)$ have the same asymptotic behaviours.*

Proof. First, let us prove that f'_h is an approximant of f'. Indeed, $f(\nu_i) = f_h(\nu_i)$ for all ν_i of the subdivision. Hence, the Rolles theorem tells that

$$\exists y_i \in]\nu_i, \nu_{i+1}[\text{ so that } f'(y_i) = f'_h(y_i) .$$

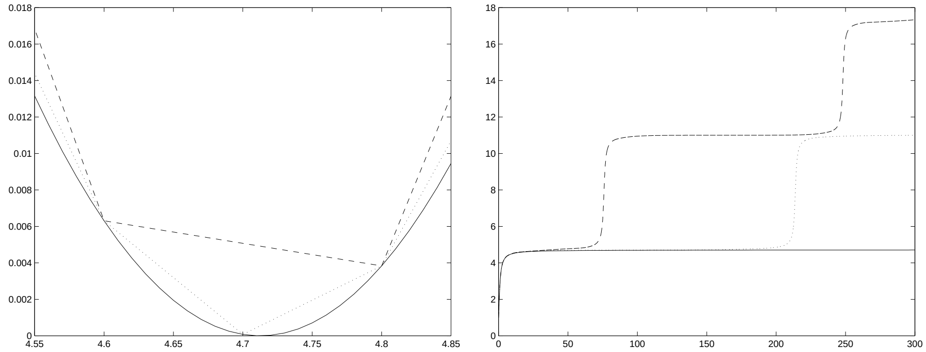

Fig. 2. Left: representative curves of $1+\sin(x)$ (solid curve) and of its piecewise linear interpolant with regard to a subdivision of size $h = 0.2$ (dashed curve) and $h = 0.1$ (dotted curve). We can see that the interpolants do not have any zeros.
Right: associated solutions.

Moreover, since f is $C^2([a,b])$:

$$\forall x \in]\nu_i, \nu_{i+1}[,\ \exists \psi_x \in]\nu_i, \nu_{i+1}[,\ f'(x) = f'(y_i) + (x - y_i) f''(\psi_x)$$

and since f_h is linear on $]\nu_i, \nu_{i+1}[$:

$$\forall x \in]\nu_i, \nu_{i+1}[,\ f_h'(x) = f_h'(y_i) = f'(y_i)\ .$$

Consequently

$$\forall x \in]\nu_i, \nu_{i+1}[,\ |f'(x) - f_h'(x)| \leq \sup_{]\nu_i, \nu_{i+1}[} |f''(x)| h$$

and

$$\forall x \in [a,b],\ |f'(x) - f_h'(x)| \leq \sup_{[a,b]} |f''(x)| h \tag{9}$$

Let us assume that $f(\xi) = 0$ and $f'(\xi) < 0$, so that $x(t)$ has a horizontal asymptote at $x = \xi$. There exist $\delta > 0$ and a neighbourhood $[\alpha, \beta]$ of x_0 so that $f(\alpha) > \delta$, $f(\beta) < -\delta$ and $f'(x) < -\delta$ on $[\alpha, \beta]$. Let ϵ be a positive real so that

$$\epsilon < \frac{\delta}{\sup_{[a,b]} |f''(x)|} \text{ and } \epsilon < \sqrt{\frac{2\delta}{\sup_{[a,b]} |f''(x)|}}\ .$$

9 tells that for all x in $[\alpha, \beta]$ where f_h' is defined, $f_h'(x) < 0$. Moreover, 5 gives $f_h(\alpha) > 0$ and $f_h(\beta) < 0$. Since f_h is continuous there exists a unique $\tilde{\xi}$ in $[\alpha, \beta]$ so that $f_h(\tilde{\xi}) = 0$. Consequently $x_h(t)$ has a horizontal asymptote at $x = \tilde{\xi}$.

In other words the proposition 2 tells that if $x(t)$ has a horizontal asymptote then there exists $\epsilon > 0$, so that for all subdivision of $[a,b]$ of size $h < \epsilon$, $x_h(t)$ has a horizontal asymptote.

5 Some possible choices for the function f_h

In the previous sections, the choice of f_h was limited to the piecewise linear interpolant of f with respect to a subdivision of $[a,b]$. However, there are many other choices for f_h that give better results. In this section we present some other methods that we have imagined. First, we will see that there exists a better choice of piecewise linear approximant. Afterwards, we will see that it is possible to take higher degrees approximant (here piecewise quadratic).

5.1 Distribution of the error

In this part we are going to prove that there exists a piecewise linear approximant of the function f whose associated solution approximates $x(t)$ with a better order (3 instead of 2). We will also show that a correction on the last step of the computation can increase the order to 4.
Following an idea of professor P.J. Laurent (personal communication), we tried to choose a better piecewise linear approximant of the function f. Indeed, we can remark that the interpolating function f_h of the section 2 is always above the function f; consequently, the solution $x_h(t)$ will overapproximate $x(t)$. Then, the idea is to distribute properly the error. A new analysis of the error will help to explain the idea. Let us consider the system 3 and assume that f and its approximant f_h have no root in $[a,b]$.

$$t = \int_0^t dt = \int_{x(0)}^{x(t)} \frac{dx}{f(x)} = \int_{x(0)}^{x_h(t)} \frac{dx}{f_h(x)} \ .$$

Hence

$$\int_{x_h(t)}^{x(t)} \frac{dx}{f(x)} = \int_{x(0)}^{x_h(t)} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \ .$$

Then we have a new evaluation of the error of approximation

$$|x(t) - x_h(t)| \leq \sup_{[a,b]} |f| \left| \int_{x(0)}^{x_h(t)} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right| \ . \tag{10}$$

We see that the best choice for the function f_h is the function which minimizes the integral of equation 10. The idea then is to choose

$$f_h(x) = f_i(x) \text{ if } x \in [\nu_i, \nu_{i+1}[\tag{11}$$

where f_i is the linear interpolant of f at the Gauss integration points of the interval $[\nu_i, \nu_{i+1}[$:

$$\mu_{i,1} = (\frac{1+\sqrt{3}}{\sqrt{3}} \nu_i + \frac{-1+\sqrt{3}}{\sqrt{3}} \nu_{i+1})/2$$
$$\mu_{i,2} = (\frac{-1+\sqrt{3}}{\sqrt{3}} \nu_i + \frac{1+\sqrt{3}}{\sqrt{3}} \nu_{i+1})/2 \ .$$

Proposition 3. *Let the system $\dot{x} = f(x)$ be defined on an interval $[a, b]$. Assuming that f is sufficiently smooth and has no zero on $[a, b]$; let f_h be the piecewise linear function defined as in 11.*
If $x(t)$ is the solution of the initial value problem $\dot{x} = f(x)$, $x(0) = x_0$ and $x_h(t)$ the solution of $\dot{x}_h = f_h(x_h)$, $x_h(0) = x_0$ then for all $t \in \mathbb{R}$,

$$|x(t) - x_h(t)| = O(h^3) .$$

Proof. The idea of the proof is to show that the integral of equation 10 is in $O(h^3)$.
First, let us compute the following integral:

$$J_i = \left| \int_{\nu_i}^{\nu_{i+1}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right| .$$

Let g be the linear interpolant of $1/f$ at the Gauss integration points of the interval $[\nu_i, \nu_{i+1}[$. Then we can can bound J_i using g:

$$J_i \leq \left| \int_{\nu_i}^{\nu_{i+1}} \left(\frac{1}{f(x)} - g(x) \right) dx \right| + \left| \int_{\nu_i}^{\nu_{i+1}} \left(g(x) - \frac{1}{f_h(x)} \right) dx \right| .$$

We have, from Gaussian integration

$$\left| \int_{\nu_i}^{\nu_{i+1}} \left(\frac{1}{f(x)} - g(x) \right) dx \right| = O(h^5) .$$

Moreover, since f_h equals f at the Gauss integration points we have:

$$\left| \int_{\nu_i}^{\nu_{i+1}} \left(g(x) - \frac{1}{f_h(x)} \right) dx \right| = O(h^5) .$$

Hence $J_i = O(h^5)$.
Let ν_{I_1}, ν_{I_n} be the points of the subdivision verifying $x_0 \in [\nu_{I_1 - 1}, \nu_{I_1}[$ and $x_h(t) \in [\nu_{I_n}, \nu_{I_n + 1}[$. Then,

$$\left| \int_{\nu_{I_1}}^{\nu_{I_n}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right| \leq \sum_{i=I_1}^{i=I_n - 1} J_i$$

and

$$\left| \int_{\nu_{I_1}}^{\nu_{I_n}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right| = O(h^4) .$$

The equation 10 gives:

$$|x(t) - x_h(t)| \leq \sup_{[a,b]} |f| \left[\left| \int_{\nu_{I_1}}^{\nu_{I_n}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right| \right. \tag{12}$$
$$\left. + \int_{x_0}^{\nu_{I_0}} \left| \frac{f(x) - f_h(x)}{f(x) f_h(x)} \right| dx + \int_{\nu_{I_n}}^{x_h(t)} \left| \frac{f(x) - f_h(x)}{f(x) f_h(x)} \right| dx \right] .$$

As f_h is an affine interpolant we have $|f(x) - f_h(x)| = O(h^2)$. Then

$$|x(t) - x_h(t)| \leq \sup_{[a,b]} |f| \left[O(h^4) + \int_{x_0}^{\nu_{I_0}} |O(h^2)|\, dx + \int_{\nu_{I_n}}^{x_h(t)} |O(h^2)|\, dx \right] \tag{13}$$

and, since $|\nu_{I_0} - x_0| \leq h$ and $|\nu_{I_n} - x_h(t)| \leq h$, we have:

$$|x_h(t) - x(t)| = O(h^3)\ . \tag{14}$$

We can see that we have a method of order three using only piecewise linear approximants.
Now, let us assume that we have $x_0 = \nu_{I_0}$, then we can remark that:

$$\forall i \in \{I_0, \ldots, I_n\},\ |x(t_i) - x_h(t_i)| = O(h^4)\ . \tag{15}$$

The fact that we have an approximant of order 4 for some values of the time encouraged us to find a method which would correct $x_h(t)$ and give a new approximant of $x(t)$ with the precision $O(h^4)$.
Since the $x_h(t)$ approximates $x(t)$ in $O(h^4)$ when $x_h(t)$ equals a point of the subdivision, the idea of the correction is to reduce the interval $[\nu_{I_n}, x_h(t)]$ so that the last integral of equation 13 is as small as possible.
Indeed, the order 3 of the method comes from the fact that $|\nu_{I_n} - x_h(t)| = O(h)$. If we want the method to be of order 4 we must insure that $|\nu_{I_n} - x_h(t)| = O(h^p)$, with $p \geq 2$.
Let $\bar{\nu}$ be the value of x_h at the time t. We define a new subdivision of [a,b]:

$$\bar{\nu}_i = \begin{cases} \nu_i & \forall i = 0 \ldots I_n \\ \bar{\nu} & \text{if } i = I_n + 1 \\ \nu_{i-1} & \forall i = I_n + 2 \ldots N \end{cases}$$

We note $\bar{f}_h$ the piecewise linear interpolant of f with regard to the Gauss points of the new subdivison. Let us consider the system:

$$\dot{\bar{x}}_h = \bar{f}_h(\bar{x}_h),\ \bar{x}_h(t_{I_n}) = \nu_{I_n}\ . \tag{16}$$

Please note that the computation of $\bar{x}_h(t)$ is simple:

- First we compute $x_h(t) = \bar{\nu}$.
- We interpolate f to the Gauss points of the interval $[\nu_{I_n}, \bar{\nu}[$ and $[\bar{\nu}, \nu_{I_n+1}[$.
- We solve the system 16 (at most two iterations).

Then, we have the following result:

Proposition 4.

$$|x(t) - \bar{x}_h(t)| = O(h^4)\ .$$

Proof. Since $\bar{x}_h(t_{I_n}) = x_h(t_{I_n})$, the equation 15 tells that $\nu_{I_n} = \bar{x}_h(t_{I_n}) = x(t_{I_n}) + O(h^4)$. Moreover we have:

$$t - t_{I_n} = \int_{t_{I_n}}^{t} dt = \int_{x(t_{I_n})}^{x(t)} \frac{dx}{f(x)} = \int_{\bar{x}_h(t_{I_n})}^{\bar{x}_h(t)} \frac{dx}{\bar{f}_h(x)}$$

which means

$$\int_{x(t)}^{\bar{x}_h(t)} \frac{dx}{\bar{f}_h(x)} + \int_{x(t_{I_n})}^{\bar{x}_h(t_{I_n})} \frac{dx}{f(x)} = \int_{\bar{x}_h(t_{I_n})}^{x(t)} \left(\frac{1}{f(x)} - \frac{1}{\bar{f}_h(x)} \right) dx \ .$$

Consequently,

$$|x(t) - \bar{x}_h(t)| \leq \sup_{[a,b]} |f| \left[\left| \int_{\nu_{I_n}+O(h^4)}^{\nu_{I_n}} \frac{dx}{\bar{f}_h(x)} \right| + \left| \int_{\nu_{I_n}}^{x(t)} \left(\frac{1}{f(x)} - \frac{1}{\bar{f}_h(x)} \right) dx \right| \right]$$

and

$$\begin{aligned} |x(t) - \bar{x}_h(t)| \leq \sup_{[a,b]} |f| \big[O(h^4) + \\ \left| \int_{\nu_{I_n}}^{\bar{\nu}} \left(\frac{1}{f(x)} - \frac{1}{\bar{f}_h(x)} \right) dx \right| + \left| \int_{\bar{\nu}}^{x(t)} \left(\frac{1}{f(x)} - \frac{1}{\bar{f}_h(x)} \right) dx \right| \big] \ . \end{aligned}$$

Hence (Gaussian integration on $[\nu_{I_n}, \bar{\nu}]$ and equation 14),

$$\begin{aligned} |x(t) - \bar{x}_h(t)| \leq \sup_{[a,b]} |f| \big[O(h^4) \\ + O(h^5) + \left| \int_{\bar{\nu}}^{\bar{\nu}+O(h^3)} \left(\frac{\bar{f}_h(x) - f(x)}{\bar{f}_h(x) f(x)} \right) dx \right| \big] \ . \end{aligned}$$

Then ($\bar{f}_h$ affine interpolant)

$$|x(t) - \bar{x}_h(t)| \leq O(h^4) + \sup_{[a,b]} |f| \left| \int_{\bar{\nu}}^{\bar{\nu}+O(h^3)} O(h^2) dx \right|$$

and

$$|x(t) - \bar{x}_h(t)| = O(h^4) \ .$$

We can remark that the corrected method is very powerfull since with only $2N + 2$ evaluations of f, the precision of the approximation is $O(h^4)$. For comparison, a Runge-Kutta method needs $4N$ evaluation of f for the same order of approximation.

5.2 Piecewise quadratic interpolation

The fundamental inequality (see 1) tells that the order of hybrid computation is the same as the order of the approximant f_h of the function f. Consequently, a natural idea is to choose a piecewise quadratic approximant. Then, given the uniform subdivision $(\nu_i)_{i\in\{0\ldots N\}}$ of $[a,b]$, we note f_i the quadratic interpolant of f with respect to the points ν_i, $\nu_i + h/2$ and ν_{i+1}. Let f_h be the piecewise quadratic approximant of f:

$$f_h(x) = f_i(x) = a_i x^2 + b_i x + c_i \text{ if } x \in [\nu_i, \nu_{i+1}[\ . \tag{17}$$

The solution of this equation is

$$x_h(t) = \begin{cases} \dfrac{\tan(\frac{t-t_i}{2}\sqrt{-\Delta_i} + \arctan(\frac{2a_i x_i + b_i}{\sqrt{-\Delta_i}}))\sqrt{-\Delta_i} - b_i}{2a_i} & \text{if } \Delta_i = b_i^2 - 4a_i c_i < 0 \\ \dfrac{(t-t_i)r_i(r_i - x_i) + x_i}{(t-t_i)(r_i - x_i) + 1} & \text{if } \Delta_i = 0 \text{ with } r_i = \frac{-b_i}{2a_i} \\ \dfrac{-r_{i,1} + \frac{x_i - r_{i,1}}{x_i - r_{i,2}} r_{i,2} e^{(r_{i,1} - r_{i,2})(t-t_i)}}{-1 + \frac{x_i - r_{i,1}}{x_i - r_{i,2}} e^{(r_{i,1} - r_{i,2})(t-t_i)}} & \text{if } \Delta_i > 0 \text{ with } r_{i,j} = \frac{-b_i + (-1)^j\sqrt{\Delta_i}}{2a_i} \end{cases}$$

where t_i is the time value at which $x_h(t_i) = x_i$. The serie (t_i) can be computed thanks to the following formulas:

$$t_{i+1} = t_i + \begin{cases} 2\dfrac{\arctan(\frac{2a_i x_{i+1} + b_i}{\sqrt{-\Delta_i}}) - \arctan(\frac{2a_i x_i + b_i}{\sqrt{-\Delta_i}})}{\sqrt{-\Delta_i}} & \text{if } \Delta_i = b_i^2 - 4a_i c_i < 0 \\ \dfrac{x_{i+1} - x_i}{(x_{i+1} - r_i)(x_i - r_i)} & \text{if } \Delta_i = 0 \text{ with } r_i = \frac{-b_i}{2a_i} \\ \dfrac{\ln(\frac{x_{i+1} - r_{i,1}}{x_{i+1} - r_{i,2}}) - \ln(\frac{x_i - r_{i,1}}{x_i - r_{i,2}})}{r_{i,1} - r_{i,2}} & \text{if } \Delta_i > 0 \text{ with } r_{i,j} = \frac{-b_i + (-1)^j\sqrt{\Delta_i}}{2a_i} \end{cases}$$

The classical result follows

$$\forall x \in [a,b], |f(x) - f_h(x)| \leq \sup_{[a,b]} |f^{(3)}| \frac{h^3}{6}\ .$$

The theorem 1 gives the convergence of the method:

Corollary 2 (Convergence of the method). *Let the system $\dot{x} = f(x)$ be defined on an interval $[a,b]$. Assuming that f is C^3 and K-Lipschitz on $[a,b]$; let f_h be the piecewise quadratic function defined in 17.*
If $x(t)$ is the solution of the initial value problem $\dot{x} = f(x)$, $x(0) = x_0$ and $x_h(t)$ the solution of $\dot{x}_h = f_h(x_h)$, $x_h(0) = x_0$ then for all t where $x(t)$ and $x_h(t)$ are defined,

$$|x(t) - x_h(t)| \leq \frac{\sup_{[a,b]} |f^{(3)}|}{6K}(e^{K|t|} - 1)h^3\ .$$

Experimental results tell us that this method is in fact of order 4. This can be explained with the evaluation of the error given in equation 10. Indeed it is easy to show that

$$\int_{\nu_i}^{\nu_{i+1}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)}\right) dx = O(h^5)\ .$$

Hence 12 gives

$$|x(t) - x_h(t)| \leq \sup_{[a,b]} |f| \Big[O(h^5) + \int_{x_0}^{\nu_{I_0}} |O(h^3)|\, dx + \int_{\nu_{I_n}}^{x_h(t)} |O(h^3)|\, dx\Big]$$

and

$$|x_h(t) - x(t)| = O(h^4) \ .$$

Moreover, the methods presented in the previous sections (interpolation to the Gauss points, and the corrected version) can be generalized with quadratic approximants.

Proposition 5. *Let the system $\dot{x} = f(x)$ be defined on an interval [a,b]. Assuming that f is sufficiently smooth and has no zero on $[a, b]$:*
let f_h be the piecewise quadratic function defined as in 17;
let f_h^G be the piecewise quadratic interpolant of f with regard to the Gauss integration points of the intervals $[\nu_i, \nu_{i+1}[$.
If $x(t)$ is the solution of the initial value problem $\dot{x} = f(x)$, $x(0) = x_0$,
$x_h(t)$ the solution of $\dot{x}_h = f_h(x_h)$, $x_h(0) = x_0$,
$x_h^G(t)$ the solution of $\dot{x}_h^G = f_h(x_h^G)$, $x_h^G(0) = x_0$
and $\bar{x}_h^G(t)$ the corrected value of $x_h^G(t)$,

$$|x(t) - x_h(t)| = O(h^4)$$
$$|x(t) - x_h^G(t)| = O(h^4)$$
$$|x(t) - \bar{x}_h^G(t)| = O(h^6) \ .$$

The proof of this proposition is similar to the proofs of the propositions 3 and 4.

The method can be generalized to higher order approximants, the only conditions are that we need algebraic formulas for the computation of $x_h(t)$ and the serie (t_i).

6 Comparison with classical methods

In this section we compare the different methods presented in this paper with some classical methods of Runge-Kutta type. In the following tables, all these methods are refered to as:

- **RKN** Runge-Kutta method with N evaluations of the function at each step
- **PLI** Hybrid computation using a Piecewise Linear Interpolant
- **GPLI** Hybrid computation using the Piecewise Linear Interpolant at the Gauss points
- **CGPLI** Correction of the value obtained with the method GPLI
- **PQI** Hybrid computation using a Piecewise Quadratic Interpolant
- **GPQI** Hybrid computation using the Piecewise Quadratic Interpolant at the Gauss points
- **CGPQI** Correction of the value obtained with the method GPQI.

Table 2. Comparison of the theoretical order and costs of the methods presented in this paper with the classical Runge-Kutta methods

Method	Number of Evaluations of f at each Iteration	Theoritical Order
RK2	2	2
PLI	2	2
GPLI	2	3
CGPLI	2	4
RK3	3	3
PQI	3	4
GPQI	3	4
CGPQI	3	6
RK4	4	4

In the table 2, we sumed up the theorical results about the convergence and the theoretical cost of all the methods.
We can see that for the same number of evaluations of the function f our methods have at least as good results as the method of Runge-Kutta. Moreover, our method is in many case much better than a Runge-Kutta method. For example, with only 3 evaluations of f by step, the method CGPQI is nevertheless of order 6. A method of Runge-Kutta would require for the same order, 7 evaluations at each step.
We made some numerical experiments to verify our theoretical results. We considered the equation

$$\dot{x} = e^x, \, x(0) = 1$$

whose solution is

$$x(t) = \ln(\frac{1}{e^{-1} - t}) \ .$$

Table 3. Experimental results for $\dot{x} = e^x$, $x(0) = 1$. Error of approximation at $t = 0.3$.

Number of Iterations	PLI	GPLI	CGPLI	RK3
10	1.48e-2	6.13e-5	3.71e-5	1.45e-3
20	3.69e-3	3.42e-6	2.33e-6	2.07e-4
40	9.22e-4	8.32e-7	1.46e-7	2.76e-5
80	2.30e-4	5.22e-8	9.16e-9	3.56e-6
160	5.75e-5	1.63e-8	5.74e-10	4.52e-7
320	1.43e-5	9.87e-10	3.58e-11	5.70e-8
640	3.59e-6	2.40e-10	2.24e-12	7.15e-9
1280	8.99e-7	1.70e-11	1.41e-13	8.96e-10

For the computation of the order of the methods we computed the error for $t = 0.3$. The results are given in the tables 3 and 4.

Table 4. Experimental results for $\dot{x} = e^x$, $x(0) = 1$. Error of approximation at $t = 0.3$.

Number of Iterations	PQI	GPQI	CGPQI	RK4
10	2.13e-5	7.58e-7	1.25e-8	1.40e-5
20	1.10e-6	3.97e-8	1.98e-10	1.55e-6
40	7.38e-8	1.62e-9	3.10e-12	1.23e-7
80	5.13e-9	1.56e-10	4.84e-14	8.66e-9
160	2.88e-10	6.43e-12	0	5.71e-10
320	2.00e-11	6.20e-13	1.33e-15	3.66e-11
640	1.11e-12	2.79e-14	8.88e-16	2.32e-12
1280	7.95e-14	4.44e-16	2.22e-15	1.46e-13

Please note, that our methods have very good results in comparison to those of the Runge-Kutta methods. Moreover, the method CGPQI has very impressive results since we have a high precision approximation of the solution with about only 100 iterations.

Table 5. Experimental results for $\dot{x} = e^x$, $x(0) = 1$. (Pentium 3 i686, 1 GHz, C++)

Method	Mean run time for 1280 iterations (ms)	Experimental Order
PLI	1.124	2.001
GPLI	1.168	3.058
CGPLI	1.223	3.997
RK3	0.677	2.968
PQI	1.597	4.210
GPQI	1.599	4.812
CGPQI	1.587	5.993
RK4	0.843	3.897

Moreover, we computed the experimental order (in table 5) of all these methods and the results are accurate to the theory. We also computed the mean run time of all these methods (with 1000 experiments). We can see that our methods are experimentally more expensive than the methods of Runge-Kutta even if the latter require more evaluation of the function f. This can be easily explain by the fact that the function $f(x) = e^x$ has not a very expensive evaluation in comparison with the computation needed for the computation of the series

(t_i). However, if the numerical evaluation of f had been very expensive then our methods would have had better results than the methods of Runge-Kutta.

7 Conclusion and future work

The very good results of hybrid computation encourage us to continue our work. Indeed there are many problems which need to be studied. For example, we shall search a method for non-autonomous systems. Actually, the methods presented in this paper only work with scalar autonomous differential equations. We worked on this problem and one solution ([6]) is to make subdivisions of the phase space as well as of the time space. Thus

$$\dot{x} = f(x,t)$$

becomes

$$\dot{x}_h = a_i x_h + b_{i,j} + c_j t \text{ if } x_h \in [\nu_i, \nu_{i+1}[\text{ and } t \in [\tau_j, \tau_{j+1}[\;.$$

We also tried to adapt the method to higher dimensional systems. There is a problem here, indeed we can not hope to use quadratic approximants (we do not have general formulas for their solutions). However, piecewise linear approximants can be used and they give very good results ([1], [5], [6]).
Planar systems can be studied this way. We replace the computation of the phase portrait of a non-linear systems by the computation of the phase portrait of its piecewise linear interpolant with regard to triangular mesh (see FIG. 3).

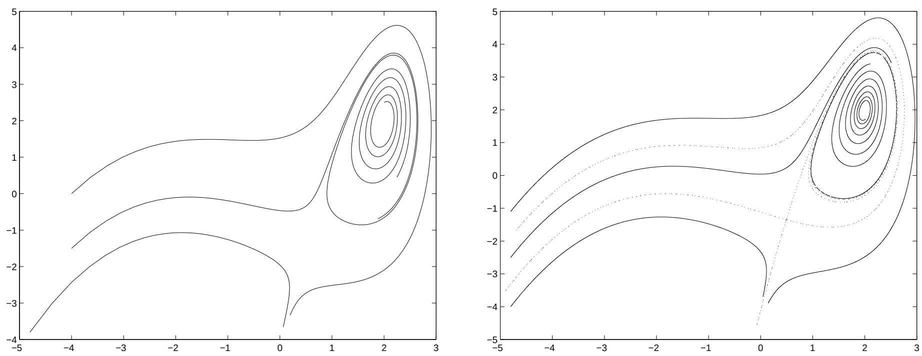

Fig. 3. Left: Phase portrait of a planar dynamical system. Right: Phase portrait of the associated piecewise linear dynamical system (with regard to a mesh of size 0.2).

We can see that the two phase portraits are very similar even if the mesh is not very refined. Indeed for the two systems there are two equilibrium points (a saddle and a source) and a limit cycle.

In this paper we have presented the application of hybrid computation to the analysis of differential equations and it has been proved that it is a very powerfull tool. However there many other fields of application of hybrid computation. Indeed, some experiments ([1]) have been done in the fields of computational geometry (generalized splines) and optimal control (Zemerlo's problem).

As a conclusion we will say that we strongly think that hybrid computation is a big field of investigation for future research and that the number of its real applications will probably grow in the next years.

Acknowledgements

I am very grateful to Jean-Guillaume Dumas for his invaluable advices and his collaboration for the numerical implementation of the methods.

References

1. J. Della Dora, A. Maignan, M. Mirica-Ruse, S. Yovine, *Hybrid Computation*, in ISSAC'01, Ontario, Canada, July 2001.
2. J. Hubbard, B. West, *Differential Equations: A Dynamical Systems Approach*, Texts in Applied Mathematics, Vol. 5, Springer-Verlag, New York, (1991).
3. J. P. Demailly, *Analyse Numérique et Equations Différentielles*, Collection Grenoble Sciences, Presses Universitaires de Grenoble, (1996).
4. J. R. Dormand, *Numerical Methods for Differential Equations*, Library of Engineering Mathematics, CRC Press, (1996).
5. A. Girard, *Etude de Systèmes Dynamiques Hybrides Affines par Morceaux*, Master's thesis, ENSIMAG, DEA de Mathématiques Appliquées, June 2001. `http://www-lmc.imag.fr/CF/page_mash/Accueil.html`.
6. A. Girard, B. Legros, *Résolution Numérique des Equations Différentielles par Approximation Affine*, Technical report, Rapport projet MASH-IMAG, June 2000
7. A. Yvart, E. Farcot, *Résolution Numérique d'Equations Différentielles Ordinaires par Approximation par Morceaux*, Technical report, Rapport projet MASH-IMAG, June 2000.
8. P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, S. Sastry, editors, *Hybrid Systems*, Lecture Notes in Computer Science, Springer-Verlag, New York, (1998).
9. L. O. Chua, A. C. Dang, *Canonical Piecewise-Linear Modeling*, IEEE Transactions on Circuits and Systems, Vol CAS-33, 5, may 1986, pp 511-525.