Approximate Solutions of ODEs Using Piecewise Linear Vector Fields

Antoine Girard

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

Abstract. In this paper, we present a method for the analysis of ordinary differential equations. The main idea is to replace the original non-linear dynamical system by an approximate piecewise linear one. The latter gives an algebraic expression of the solutions. Our method differs also from classical methods (e.g. Runge-Kutta) by the fact that we use a discretization of the phase space instead of the time space. Theoretical results of convergence exist and are proved here. An efficient symbolic-numeric algorithm for the construction of the solutions of this piecewise linear system is given. As an application, we show how we can compute an accurate approximation of periodic orbits using symbolic differentiation of the Poincar map.

1 Introduction

Computation of approximate solutions of ordinary differential equations (ODEs) is an important problem on which many things have been done (see for example [2], [8]). Most of classical methods (e.g. Runge-Kutta) use a discretization of the time space to compute approximate solutions. In [1], an alternative to these methods was proposed. The principle was to replace the initial vector field by an approximate piecewise linear one which allows much more symbolic computations.
The use of linear systems as a technique for local analysis of dynamical systems almost dates back to the birth of the subject (e.g. Hartman-Grobman theorem, [7]). In the eighties ([10]), approximate piecewise linear systems were introduced as a tool for the global qualitative analysis of dynamical systems, but had never been used (as far as we know) for the computation of approximate solutions.
This method differs from the classical ones by the fact that we use a discretization of the phase space instead of the time space. Indeed, given a simplicial subdivision of size h of the phase space, let F_h be the interpolate piecewise linear vector field of F on the subdivision. Then, the question is the following. In which way does the solutions of the system $\dot{X} = F_h(X)$ approximate the solutions of $\dot{X} = F(X)$?
In [3], a detailed study of this method applied to scalar systems have been presented. Numerical tests have shown that it is comparable to Runge-Kutta methods in terms of computational cost. Moreover, some improvements have been proposed in order to icrease its accuracy (up to 6 using approximate piecewise quadratic vector fields).
In the present paper, we focus on higher dimensional systems. After giving an overview of the method, we will see how it works with a simple example. Then, we will explain how the construction of the solutions of the approximate piecewise linear system can be practically implemented using a symbolic-numeric algorithm. Finally, after giving the proof of the convergence of approximate solutions, we will show how our method can be used in order to compute approximate periodic orbits with automatic differentiation.

2 Overview of the Method

We consider the following initial value problem:

$$\dot{X}(t) = F(X(t)),\ X(t) \in \Omega,\ X(0) = X_0 \tag{1}$$

where Ω is a bounded subset of $\mathbb{R}^n$. The idea of our method is to replace the vector field F by a piecewise approximation which can be locally and symbolically integrated. For example, we can replace F by its piecewise linear interpolant on a simplicial subdivision of Ω.
An important point of our method is that we have a decomposition of the phase space (that is Ω).
Hence, let us define a subdivision $(\Omega_i)_{i\in I}$ of Ω. For all i of I, we define F_i a vector field which approximates F on Ω_i. Then, the integration of the piecewise approximate vector field allows to construct an approximate solution of (1). It can be done using the following algorithm :

– **Initialization**
  - Set $t_0 = 0$.
  - Compute the active cell Ω_{i_0} containing X_0.
– **Main loop**
  - *First step* :
    Solve symbolically the differential equation on the active cell Ω_{i_k}

$$X'(t) = F_{i_k}(X(t)),\ X(t_k) = X_k\ .$$

    i.e. compute the local flow Φ_{i_k} in Ω_{i_k}.
  - *Second step* :
    Compute the time t_{k+1} and the point X_{k+1} where $X(t)$ exits Ω_{i_k} (using a symbolic or numerical method).
    On the time-interval $[t_k, t_{k+1}[$, $X(t) = \Phi_{i_k}(X_k, t - t_k)$.
  - *Third step* :
    If t_{k+1} exists, compute the new active simplex $\Omega_{i_{k+1}}$ containing X_{k+1}.

This algorithm will be detailed in the next sections for piecewise linear approximate vector fields. But first, in order to catch the idea of the method, let us consider a simple scalar example.

3 A One-Dimensional Example

An example may help to understand the principle. Consider the initial value problem:

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

The solution of (2) is $x(t) = 1/(1-t)$. We are now going to approximate $x(t)$ by the solution of a piecewise linear dynamical system. We follow the different steps presented in the previous section. First, a subdivision of the phase space (that is the x-axe) must be set. 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 we do with classical methods) but the phase space. Then, the next task is to choose an approximation of the function $f(x) = x^2$ on each interval, for example the linear interpolate function of f at the bounds of 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

$$\dot{x}_h(t) = f_i(x_h(t)),\ \text{if } x_h(t) \in I_i,\ x_h(0) = 1\ . \tag{3}$$

Let us assume that $h = 1/N$ with $N \in \mathbb{N}$. We can compute $x_h(t)$ the approximation of $x(t)$.
Thus , $t_0 = 0$ and the first active cell is the interval I_N since $x_h(0) = 1 = Nh$. Let t_{i-N} be the time at which $x_h(t)$ enters in I_i. Then, while $x_h(t)$ is in this interval, its expression is

$$x_h(t) = (ih + \frac{b_i}{a_i})e^{a_i(t - t_{i-N})} - \frac{b_i}{a_i}$$

which is equivalent to

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

Then, $x_h(t)$ exits I_i at the point $(i+1)h$ and at the time

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

Here is the big difference with classical methods; indeed, the time subdivision is given by the approximate system and is not fixed a priori. We can remark that the solution of (2) has a vertical asymptote at $t = 1$. A good approximation of $x(t)$ should also have a vertical asymptote. It is easy to show that $x_h(t)$ has one:

$$t_{i+1-N} - t_{i-N} \sim \frac{2}{i(2i+1)h} \Rightarrow \lim_{i\to\infty} t_{i+1-N} = \sum_{i\geq 0}(t_{i+1} - t_i) < \infty \ .$$

This is a great advantage in comparison with classical methods. Indeed, a constant step method can clearly not reproduce such behaviours. The second point is that we do not have just a sequence of points (t_i, x_i) but a symbolic expression of the approximate solution:

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

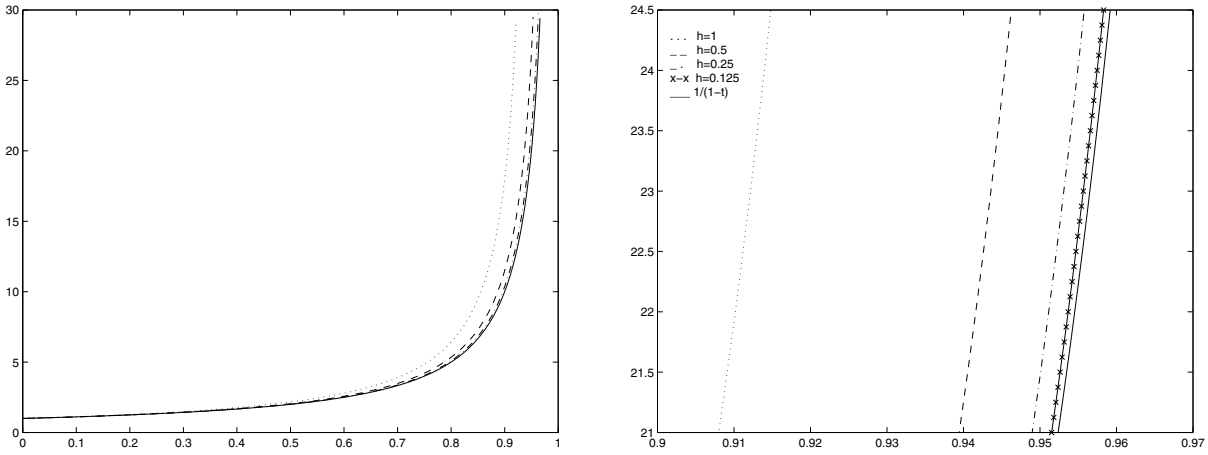

Fig. 1. Solution of (3) for $h \in \{0.125, 0.25, 0.5, 1\}$ and solution of (2) (i.e. $1/(1-t)$) (*full*)

In Fig. 1, we have plotted $x(t)$ and $x_h(t)$ for different values of h. We did not plot the approximate 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 a second order of accuracy for our method.

A detailed study of the method applied to scalar differential equations has been made in [3]. It has been shown that the solutions of the piecewise linear interpolate (PLI) vector field approximates the exact solution with the precision $O(h^2)$. Moreover, the choice of a better linear approximation (e.g. the interpolate function at the Gauss integration points (GPLI)) leads to a third order approximation. Tests on computational complexity have been made showing that the cost of our method is comparable to the cost of the Runge-Kutta scheme though a bit more expensive (see table 1).

Table 1. Experimental results for $\dot{x} = e^x$, $x(0) = 1$, [3]. (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
RK3	0.677	2.968

In the next section, we will detail the algorithm of the second section for general n-dimensional systems.

4 The Detailed Algorithm for the General n-Dimensional Case

We consider the initial value problem (1). This section deals with the efficient implementation of the algorithm of section 2 for the construction of an approximate solution $X(t)$ of the problem (1).

4.1 Choice of the Approximate Vector Field

We define a simplicial subdivision $(\Omega_i)_{i\in I}$ of Ω. For all i of I, A_i and b_i are defined so that

$$F_i(X) = A_i X + b_i$$

is the interpolate linear vector field of F at the vertices of the simplex Ω_i.

4.2 First Step : Symbolic Local Integration

If $X(t)$ is in the simplex Ω_{i_k}, then it verifies

$$\dot{X}(t) = A_{i_k} X(t) + b_{i_k} \ .$$

This equation is equivalent to

$$\begin{pmatrix} \dot{X}(t) \\ 0 \end{pmatrix} = M_{i_k} \begin{pmatrix} X(t) \\ 1 \end{pmatrix}$$

where

$$M_{i_k} = \begin{pmatrix} A_{i_k} & b_{i_k} \\ 0 & 0 \end{pmatrix} \ .$$

Let t_k be the time at which $X(t)$ entered the simplex Ω_{i_k} by the point X_k, then using results of theory of linear differential equations, it comes that

$$\begin{pmatrix} X(t) \\ 1 \end{pmatrix} = e^{(t-t_k)M_{i_k}} \begin{pmatrix} X_k \\ 1 \end{pmatrix}$$

until $X(t)$ leaves Ω_{i_k}.

4.3 Second Step : Numerical Computation of the Exit Time

The main difficulty of the method is the computation of the times (t_k) at which $X(t)$ exits a simplex. Indeed, this series is easily computable in scalar systems (see in the previous section). But for n-dimensional systems ($n \geq 2$), there does not exist general algebraic expressions (see [1]). Consequently, a numerical iterative method must be used to evaluate (t_k). This problem has been studied in [4]; in this paragraph, the main results are given.
The problem to be solved is

$$\begin{array}{l} \text{For } X'(t) = A_{i_k} X(t) + b_{i_k},\ X(t_k) = X_k \in \Omega_{i_k}; \\ \text{Find } t_{k+1} \text{ (if it exists) the time at which } X(t) \text{ leaves } \Omega_{i_k} \ . \end{array}$$

But for the simplicity of the notations, we replace it by the equivalent problem

$$\begin{array}{l} \text{For } X'(t) = AX(t) + b,\ X(0) = X_0 \in \Omega; \\ \text{Find } t^* > 0 \text{ (if it exists) the time at which } X(t) \text{ leaves } \Omega \ . \end{array} \tag{4}$$

In a first time, let us assume that t^* exists, the case when the approximate solution does not exit the simplex will be discussed at the end of the paragraph.
The formulation of the problem (4) is simple but it needs to be transformed in order to be solved. Consequently, we have to introduce some additional notations. Since Ω is a simplex of $\mathbb{R}^n$, it has $n+1$ vertices $\{Y_1, \ldots, Y_{n+1}\}$. Moreover, it is delimited by $n+1$ hyperplanes $\{F_1, \ldots, F_{n+1}\}$. Let k_j be the unitary vector orthogonal to F_j and directed towards the interior of Ω. There exist $n+1$ scalars $\{d_1, \ldots, d_{n+1}\}$ such that a point X of $\mathbb{R}^n$ is in Ω, if and only if

$$\forall j \in \{1, \ldots, n+1\},\ k_j^t X - d_j \geq 0 \ .$$

We define the functions $f_j(t)$ which represent the distance of $X(t)$ to the faces F_j:

$$\forall j \in \{1, \ldots, n+1\},\ f_j(t) = k_j^t X(t) - d_j\ .$$

The problem (4) is clearly equivalent to

$$\begin{cases} \forall j \in \{1, \ldots, n+1\},\ \forall t \in]0, t^*[,\ f_j(t) > 0 \\ \exists j \in \{1, \ldots, n+1\}, \qquad\qquad\quad f_j(t^*) = 0\ . \end{cases}$$

Consequently, the problem to solve is equivalent to the resolution of a non-linear equation. The method proposed here to find t^* is quite similar to the classical method of Newton. Indeed, note that the derivative of $f_j(t)$ is simple,

$$\forall j \in \{1, \ldots, n+1\},\ f_j'(t) = k_j^t X'(t) = k_j^t(AX(t) + b)\ .$$

However, we do not apply the classical method of Newton since it does not allow to insure that it has a global convergence. Moreover, the functions $f_j(t)$ have some interesting properties that can be very useful. For example, it is easy to compute (using linearity of the vector field and the barycentric definition of Ω) some bounds on $f_j^{(2)}(t)$ for all t where $X(t)$ is in Ω. Indeed, we can show ([4]) that

$$\forall t \in [0, t^*]\ , \begin{cases} f_j^{(2)}(t) \geq m_j\ = \min_{i=1}^{i=n+1}[k_j^t(A^2 Y_i + Ab)] \\ f_j^{(2)}(t) \leq M_j = \max_{i=1}^{i=n+1}[k_j^t(A^2 Y_i + Ab)]\ . \end{cases}$$

We consequently define the series (τ_k) which converges to the solution of the problem (4).

$$\begin{cases} \tau_0 \quad\ = 0 \\ \tau_{k+1} = \tau_k + \min_{j=1}^{j=n+1}(s_k^j) \end{cases} \tag{5}$$

where s_k^j is the smallest positive root (if it exists) of the quadratic polynomial

$$f_j(\tau_k) + s f_j'(\tau_k) + \frac{s^2}{2} m_j\ . \tag{6}$$

If this polynomial does not have any strictly positive root then set $s_k^j = +\infty$. As it was told before, the series (τ_k) converges to the value of t^*. Moreover, in [4], it is proved that it converges quadratically. Indeed,

Theorem 1. *Let (τ_k) be defined as in equation (5).*

$$\lim_{k \to \infty} \tau_k = t^*$$

Assuming j^* in $\{1, \ldots, n+1\}$ such that $f_{j^*}(t^*) = 0$ is unique and $f_{j^*}'(t^*) \neq 0$, then

$$\exists \bar{k} \geq 0 \text{ so that, } \left[\frac{M_{j^*}}{\inf_{[\tau_{\bar{k}}, t^*]} |f_{j^*}'(t)|} \right] |t^* - \tau_{\bar{k}}| < 1$$

and

$$\forall k \geq \bar{k},\ |t^* - \tau_k| \leq \left(\left[\frac{M_{j^*}}{\inf_{[\tau_{\bar{k}}, t^*]} |f_{j^*}'(t)|} \right] |t^* - \tau_{\bar{k}}| \right)^{2^{k-\bar{k}}-1} |t^* - \tau_{\bar{k}}|.$$

The proof of the theorem is simple though quite long. Let us give an idea. We can remark that the linear part of the quadratic polynomial (6) is the first order Taylor expansion of $f_j(t)$ at the point τ_k. The quadratic term allows to insure that the value of the polynomial is always smaller than $f_j(t)$. This implies that t^* is an upper bound for (τ_k). Moreover, by definition (τ_k) is increasing and therefore convergent (it is easy to show that it converges to t^*). Then, the second order convergence comes from the fact that if the quadratic term in the polynomial (6) is omited, the iteration (4) becomes an iteration of Newton's method.

Numerical experiments have been realized ([4]) to test this iterative method and the results are impressive, we generally need very few iterations to evaluate t^* with a very good accuracy.

We now consider the case when $X(t)$ does not exit the simplex Ω. Then, there necessarly exists either an equilibrium or a periodic orbit in Ω to which $X(t)$ converges. It can be shown that in this case

$$\lim_{k \to \infty} \tau_k = +\infty\ .$$

Consequently, there are two distinct cases :

- We want to construct the approximate solution $X(t)$ on a finite time interval $[0, T]$. There exist a finite k such that $\tau_k > T$. Then, the algorithm stops.
- We want the solution on the interval $[0, \infty[$. The first thing to do is to compute the possible limit of $X(t)$. Then, special techniques involving Lyapunov stability (see [9]) may be used to prove that $X(t)$ never exits Ω.

Currently only the first case has been implemented, but the second one should be considered for a later version.

4.4 Third Step : Computation of the New Active Cell

Once we have t_{k+1} and X_{k+1} then we can compute the new active simplex $\Omega_{i_{k+1}}$ using the following rule:

$$\begin{cases} X_{k+1} \text{ is in } \Omega_{i_{k+1}} \ , \\ A_{i_{k+1}} X_{k+1} + b_{i_{k+1}} \text{ is directed towards the interior of } \Omega_{i_{k+1}} \ . \end{cases} \tag{7}$$

The second condition allows to insure that the solution $X(t)$ is going to stay in the simplex $\Omega_{i_{k+1}}$ during a non-zero time interval.

4.5 Construction of the Expression of the Approximate Solution

After the computation of the series (t_k) and (X_k), the approximate solution can be evaluated at any time t using the symbolic expression of $X(t)$:

$$\begin{pmatrix} X(t) \\ 1 \end{pmatrix} = e^{(t-t_k)M_{i_k}} \begin{pmatrix} X_k \\ 1 \end{pmatrix} \text{ if } t \in [t_k, t_{k+1}] \ .$$

4.6 Example

The algorithm described in this section has been implemented in Matlab. In this paragraph, the results of the application of our method to the computation of an approximate solution of the following planar dynamical system are shown.

$$\begin{cases} \dot{x} = -x + y + (x - 1.85)^2 \\ \dot{y} = 3x + y - x^3 \ . \end{cases} \tag{8}$$

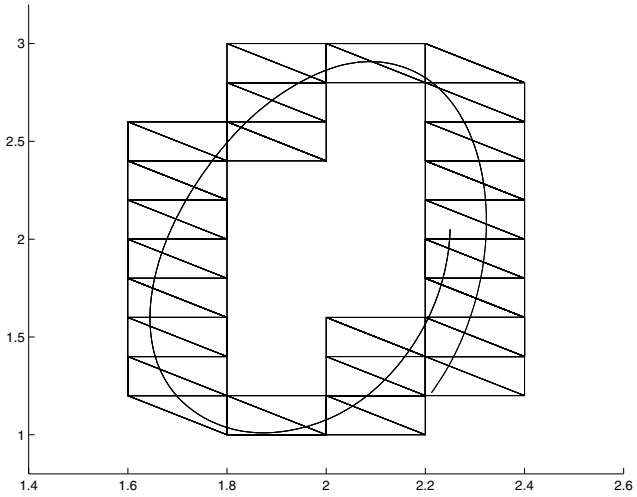

Fig. 2. A trajectory and the associated simplices

We chose an uniform triangulation of size $h = 0.2$. A solution of the approximate system is plotted in figure 2. We can see that the solution is C^1 (because the piecewise interpolate vector field is continuous).

In the next section, it is proved that as the subdivision (Ω_i) is refined, the solutions of the piecewise linear interpolate dynamical system converge to the solutions of the exact system.

5 Convergence of the Approximate Solutions

Before giving the proof of the convergence of the approximate solutions, it is useful to recall a fundamental theorem of the theory of dynamical systems.

Theorem 2 (Fundamental Inequality [8]). *If on a compact Ω 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 $\epsilon = \epsilon_1 + \epsilon_2$.

Now, let us consider the dynamical system :

$$\dot{X}(t) = F(X(t)),\ X(t) \in \Omega\ . \tag{9}$$

We define, as described in the previous section, a piecewise linear interpolate dynamical system. We choose a simplicial subdivision $(\Omega_i)_{i\in I}$ of Ω. Let h be the length of the longest edge of the subdivision. We note F_h the piecewise linear interpolate vector field of F with regard to $(\Omega_i)_{i\in I}$. The approximate dynamical system we consider is

$$\dot{X}_h(t) = F_h(X_h(t)),\ X_h(t) \in \cup_{i\in I}\Omega_i\ . \tag{10}$$

The proof of the convergence of the solutions of this system to the solutions of (9) is an immediate corollary of the theorem 2.

Corollary 1 (Convergence of approximate solutions). *Assuming that F is C^2 and K-Lipschitz on Ω; let $X(t)$ be the solution of the system (9) verifying $X(0) = X_0$ and $X_h(t)$ be the solution of the system (10) verifying $X_h(0) = X_0$. Then for all t where X and X_h are defined,*

$$\|X(t) - X_h(t)\| = O(h^2)\ .$$

Proof. From interpolation theory, we have the classical result :

$$\forall X \in \cup_{i\in I}\Omega_i,\ \|F(X) - F_h(X)\| = O(h^2)\ .$$

Hence,

$$\|\dot{X}(t) - F(X(t))\| = 0$$

and

$$\|\dot{X}_h(t) - F(X_h(t))\| = \|F_h(X_h(t)) - F(X_h(t))\| = O(h^2)\ .$$

Moreover $\|X_h(0) - X(0)\| = 0$, consequently the fundamental inequality gives

$$\|X(t) - X_h(t)\| = O(h^2)\ .$$

In [3], numerical experiments have been made. They confirm the theoretical result.

In the next section, a method for the computation of approximate periodic solutions of dynamical systems using piecewise linear vector field is given.

6 An Application: Computation of Limit Cycles with Automatic Differentiation

The computation of periodic orbits is an important point of the study of dynamical systems. In [6] and [11], algorithms for this problem have been proposed. Both of them use automatic differentiation of the Poincar map in a neighbourhood of a limit cycle. In this section we show how, using the differentiation technique and piecewise linear vector fields, we are able to compute approximate limit cycles of dynamical systems.

6.1 Notations and Assumptions

Let us assume that the approximate piecewise linear dynamical system (10) has a limit cycle Γ. Γ passes by a finite closed sequence of simplices $\{\Omega_{i_0}, \ldots, \Omega_{i_l}, \Omega_{i_0}\}$. Γ enters each simplex Ω_{i_j} by one of its faces F_{i_j} at the point $X^*_{i_j}$ and leaves it after a time $s^*_{i_{j+1}}$ by one of its faces $F_{i_{j+1}}$ at the point $X^*_{i_{j+1}}$. Please note that we have

$$\begin{cases} F_{i_0} = F_{i_{l+1}} \\ X^*_{i_0} = X^*_{i_{l+1}} \end{cases} .$$

We define the sequence $\{t^*_{i_0}, \ldots, t^*_{i_{l+1}}\}$:

$$\begin{cases} t^*_{i_0} = 0 \\ t^*_{i_j} = \sum_{k=1}^{k=j} s^*_{i_k} \end{cases} .$$

We note k_i the orthogonal vectors to the faces F_i.
We must make the two following assumptions:

Assumption 1 *$\forall i \in \{i_0, \ldots, i_l\}$, X^*_i is in the interior of F_i.*

Assumption 2 *Γ does not enter nor exit the simplices tangently:*

$$\forall j \in \{0, \ldots, l\},\ k^t_{i_{j+1}}(A_{i_j} X^*_{i_{j+1}} + b_{i_j}) \neq 0 .$$

These assumptions are reasonable; indeed, almost all the trajectories of the system (10) verify these properties.
In the next paragraph, the computation of the derivative of the Poincaré map on the face F_{i_0} under the assumptions 1 and 2 is detailed.

6.2 Algebraic Differentiation of the Poincaré Map

Since the vector field F_h is continuous, then the associated flow Φ_h is also continuous with regard to the initial condition. Then, there exists a trajectory $X_h(t)$ with initial condition X_{i_0} on the face F_{i_0} such that it passes by the same sequence of simplices as Γ, entering the simplex Ω_{i_j} by the face F_{i_j} at a point X_{i_j} and leaving after a time s_{i_j} by the face $F_{i_{j+1}}$ at the point $X_{i_{j+1}}$.
Moreover, for X_{i_0} sufficiently close to $X^*_{i_0}$ the sequence of points (X_{i_j}) verifies the assumptions 1 and 2.
The point X_{i_l} is called the successor of X_{i_0} on the Poincaré section F_{i_0}.
Note that the point $X^*_{i_0}$ is its own successor. Consequently, the computation of the limit cycle is equivalent to the computation of the fixed point of the Poincaré map.
This paragraph deals with the computational differentiation of the Poincar map. But first, let us consider the functions $\mathcal{P}_{i_j}$. Given a point X_{i_j} of the face F_{i_j}, we note, if it exists, $\mathcal{P}_{i_j}(X_{i_j}) = X_{i_{j+1}}$ the point of $F_{i_{j+1}}$ at which the trajectory entering Ω_{i_j} at the point X_{i_j}, exits the simplex.
These functions have the following properties:

Lemma 1. *$\mathcal{P}_{i_j}$ is defined in a neighbourhood of $X^*_{i_j}$. Moreover it is continuous, differentiable and its derivative at the point X_{i_j} is:*

$$P_{i_j}(X_{i_j}) = \left(I - \frac{(A_{i_j} X_{i_{j+1}} + b_{i_j}) k^t_{i_{j+1}}}{k^t_{i_{j+1}}(A_{i_j} X_{i_{j+1}} + b_{i_j})} \right) e^{s_{i_j} A_{i_j}} .$$

Proof. The proof of this lemma comes immediatly with the classical theorem of implicit functions. Indeed, let Φ_{i_j} be the local flow of the system (10) associated to the linear vector field on the simplex Ω_{i_j}. Then the point $X_{i_{j+1}}$ is defined by:

$$X_{i_{j+1}} = \Phi_{i_j}(X_{i_j}, s_{i_j}) \ .$$

Since $X_{i_{j+1}}$ is on the face $F_{i_{j+1}}$, s_{i_j} and X_{i_j} are related according to the equation

$$\mathcal{H}_{i_j}(X_{i_j}, s_{i_j}) = k^t_{i_{j+1}} \Phi_{i_j}(X_{i_j}, s_{i_j}) - d_{i_{j+1}} = 0 \ . \tag{11}$$

$\mathcal{H}_{i_j}(X_{i_j}, s_{i_j})$ is continuous and differentiable and its derivatives are:

$$\frac{\partial \mathcal{H}_{i_j}}{\partial X_{i_j}}(X_{i_j}, s_{i_j}) = k^t_{i_{j+1}} e^{s_{i_j} A_{i_j}}$$

and

$$\frac{\partial \mathcal{H}_{i_j}}{\partial s_{i_j}}(X_{i_j}, s_{i_j}) = k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j}) \ .$$

Moreover,

$$\mathcal{H}_{i_j}(X^*_{i_j}, s^*_{i_j}) = 0$$

and according to assumption 2

$$\frac{\partial \mathcal{H}_{i_j}}{\partial s_{i_j}}(X^*_{i_j}, s^*_{i_j}) \neq 0 \ .$$

Thus, the theorem of implicit functions applies; consequently there exists a function $\mathcal{S}_{i_j}$ defined on a neighbourhood of $X^*_{i_j}$ such that

$$s_{i_j} = \mathcal{S}_{i_j}(X_{i_j})$$

is a solution of equation (11). Moreover, $\mathcal{S}_{i_j}$ is differentiable and its derivative is:

$$\frac{\partial \mathcal{S}_{i_j}}{\partial X_{i_j}}(X_{i_j}) = -\frac{\frac{\partial \mathcal{H}_{i_j}}{\partial X_{i_j}}(X_{i_j}, s_{i_j})}{\frac{\partial \mathcal{H}_{i_j}}{\partial s_{i_j}}(X_{i_j}, s_{i_j})} = \frac{-k^t_{i_{j+1}} e^{s_{i_j} A_{i_j}}}{k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j})} \ . \tag{12}$$

It follows that $\mathcal{P}_{i_j}$ is defined on a neighbourhood of $X^*_{i_j}$ and

$$X_{i_{j+1}} = \mathcal{P}_{i_j}(X_{i_j}) = \Phi_{i_j}(X_{i_j}, S_{i_j}(X_{i_j})) \ .$$

Furthermore, it is differentiable and its derivative is

$$P_{i_j}(X_{i_j}) = \frac{\partial \Phi_{i_j}}{\partial X_{i_j}}(X_{i_j}, S_{i_j}(X_{i_j})) + \frac{\partial \Phi_{i_j}}{\partial s_{i_j}}(X_{i_j}, S_{i_j}(X_{i_j})) \frac{\partial \mathcal{S}_{i_j}}{\partial X_{i_j}}(X_{i_j}) \ .$$

$$P_{i_j}(X_{i_j}) = e^{s_{i_j} A_{i_j}} + (A_{i_j} X_{i_{j+1}} + b_{i_j}) \frac{-k^t_{i_{j+1}} e^{s_{i_j} A_{i_j}}}{k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j})}$$

which leads to the result presented in lemma 1.

The matrix $P_{i_j}(X_{i_j})$ has the following property:

Proposition 1.

$$\forall X \in \mathbb{R}^n, \ k^t_{i_{j+1}} P_{i_j}(X_{i_j}) X = 0 \ .$$

Proof. Let X be in $\mathbb{R}^n$:

$$k^t_{i_{j+1}} P_{i_j}(X_{i_j}) X = k^t_{i_{j+1}} \left(I - \frac{(A_{i_j} X_{i_{j+1}} + b_{i_j}) k^t_{i_{j+1}}}{k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j})} \right) e^{s_{i_j} A_{i_j}} X \ .$$

So,

$$k^t_{i_{j+1}} P_{i_j}(X_{i_j}) X = k^t_{i_{j+1}} e^{s_{i_j} A_{i_j}} X - \frac{k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j})}{k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j})} k^t_{i_{j+1}} e^{s_{i_j} A_{i_j}} X \ .$$

And the result follows immediatly.

Now, finding an expression of the Poincaré map $\mathcal{P}$ is very simple. Indeed, $\mathcal{P}$ is the composition of all the functions $\mathcal{P}_{i_j}$. Thus,

$$\mathcal{P}(X_{i_0}) = \mathcal{P}_{i_l} o \mathcal{P}_{i_{l-1}} o \dots o \mathcal{P}_{i_1} o \mathcal{P}_{i_0}(X_{i_0}) \; . \tag{13}$$

It follows from lemma 1 that

Theorem 3. *$\mathcal{P}$ is defined in a neighbourhood of $X^*_{i_0}$. Moreover it is continuous, differentiable and its derivative at the point X_{i_0} is:*

$$P(X_{i_0}) = \prod_{j=l}^{j=0} \left[\left(I - \frac{(A_{i_j} X_{i_{j+1}} + b_{i_j}) k^t_{i_{j+1}}}{k^t_{i_{j+1}} (A_{i_j} X_{i_{j+1}} + b_{i_j})} \right) e^{s_{i_j} A_{i_j}} \right] \; .$$

Proof. According to equation (13) and lemma 1, $\mathcal{P}$ is the composition of continuous differentiable functions. Consequently, it is continuous differentiable and moreover its derivative can be computed using the classical formula:

$$P(X_{i_0}) = \prod_{j=l}^{j=0} \left[P_{i_j}(X_{i_j}) \right]$$

which leads to the expected result.

We also have, as a consequence of proposition 1

Proposition 2.

$$\forall X \in \mathbb{R}^n, \; k^t_{i_{l+1}} P(X_{i_0}) X = 0 \; .$$

Proof. Let X be in $\mathbb{R}^n$

$$k^t_{i_{l+1}} P(X_{i_0}) X = k^t_{i_{l+1}} P_{i_l}(X_{i_l}) \left(\prod_{j=l-1}^{j=0} \left[P_{i_j}(X_{i_j}) \right] X \right) \; .$$

From proposition 1,

$$k^t_{i_{l+1}} P(X_{i_0}) X = 0 \; .$$

Now, let us consider a trajectory with the initial condition X_{i_0} on the face F_{i_0}. The sequences (X_{i_j}) and (s_{i_j}) can be computed using the method of paragraph 4.2. Consequently, the computation of $P(X_{i_0})$ is possible. Then, the trajectory with initial condition $X_{i_0} + dX_{i_0}$ on the face F_{i_0} comes back on F_{i_0} at a point $X_{i_l} + dX_{i_l}$ such that

$$dX_{i_l} = P(X_{i_0}) dX_{i_0} + O(\|dX_{i_0}\|^2) \; .$$

In the next paragraph, this result is applied to the computation of the limit cycles of the system (10).

6.3 Computation of Limit Cycles

The algorithm presented here for the computation of limit cycles is the method of Newton applied to the Poincaré map on the face F_{i_0}.
Let $X^k_{i_0}$ be a point of F_{i_0} approximating $X^*_{i_0}$. Let us assume that the trajectory $X^k_h(t)$ begining at $X^k_{i_0}$ passes by the same sequence of simplices as Γ; $X^k_h(t)$ is an approximation of Γ.
The sequences $X^k_{i_j}$ and $s^k_{i_j}$ and the matrix $P(X^k_{i_0})$ are computed as before. We have to make the following assumption:

Assumption 3 *Γ is not singular: $P(X^*_{i_0})$ has no eigenvalue equal to 1.*

For $X^k_{i_0}$ on F_{i_0} sufficiently near of $X^*_{i_0}$, $P(X^k_{i_0})$ has no eigenvalue equals to 1 so $(I - P(X^k_{i_0}))$ is invertible. We choose the next approximation of $X^*_{i_0}$, $X^{k+1}_{i_0}$ defined by:

$$X^{k+1}_{i_0} = [I - P(X^k_{i_0})]^{-1} (X^k_{i_l} - P(X^k_{i_0}) X^k_{i_0}) \; . \tag{14}$$

If the iteration is well defined, then since $X^k_{i_0}$ and $X^k_{i_l}$ are on the face F_{i_0}, $X^{k+1}_{i_0}$ should also be on F_{i_0}. This is indeed the case ; the equation (14) is equivalent to

$$X^{k+1}_{i_0} = X^k_{i_l} + P(X^k_{i_0})(X^{k+1}_{i_0} - X^k_{i_0}) \; . \tag{15}$$

Using proposition 2 we have

$$k^t_{i_{l+1}} P(X^k_{i_0})(X^{k+1}_{i_0} - X^k_{i_0}) = 0 \, .$$

Consequently, $X^{k+1}_{i_0}$ is on the face F_{i_0}. We can see that our iterations are well defined and moreover we have the following classical result of convergence of Newton's method:

Theorem 4. *Let $X^0_{i_0}$ be on F_{i_0} and sufficiently near $X^*_{i_0}$, then*

$$\lim_{k\to\infty} X^k_{i_0} = X^*_{i_0} \, ,$$

moreover

$$\|X^*_{i_0} - X^{k+1}_{i_0}\| = O(\|X^*_{i_0} - X^k_{i_0}\|^2) \, .$$

Proof. From theorem 3 and since $X^*_{i_0}$ is a fixed point of $\mathcal{P}$, we have

$$X^*_{i_0} = X^k_{i_l} + P(X^k_{i_0})(X^*_{i_0} - X^k_{i_0}) + O(\|X^*_{i_0} - X^k_{i_0}\|^2) \, .$$

Substracting the equation (15) to this equation, we obtain

$$X^*_{i_0} - X^{k+1}_{i_0} = P(X^k_{i_0})[X^*_{i_0} - X^{k+1}_{i_0}] + O(\|X^*_{i_0} - X^k_{i_0}\|^2) \, .$$

Consequently,

$$[I - P(X^k_{i_0})](X^*_{i_0} - X^{k+1}_{i_0}) = O(\|X^*_{i_0} - X^k_{i_0}\|^2) \, .$$

And since $I - P(X^k_{i_0})$ is invertible,

$$\|X^*_{i_0} - X^{k+1}_{i_0}\| = O(\|X^*_{i_0} - X^k_{i_0}\|^2) \, .$$

Some numerical tests have been done in [5] and the numerical experiments confirm the theoretical estimation of the convergence. Here, this algorithm was used to compute the limit cycle of the system (8), the limit cycle and the associated simplices (here triangles) are shown in figure 3.

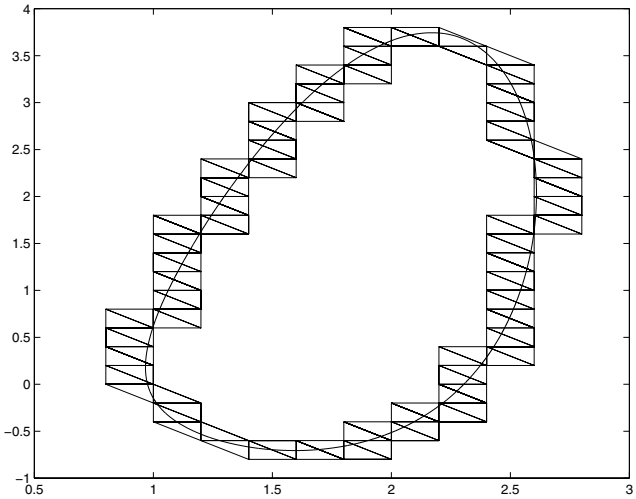

Fig. 3. Limit cycle and the associated simplices

This method works very well but has a quite important defect. Indeed, like all the methods of Newton, the convergence is guaranteed only in a neighbourhood of $X^*_{i_0}$. Consequently, we must have an idea of where the cycle is located to use the algorithm. Currently, we do not have methods to do the location work and simulations must be used to choose the first approximation of the limit cycle.

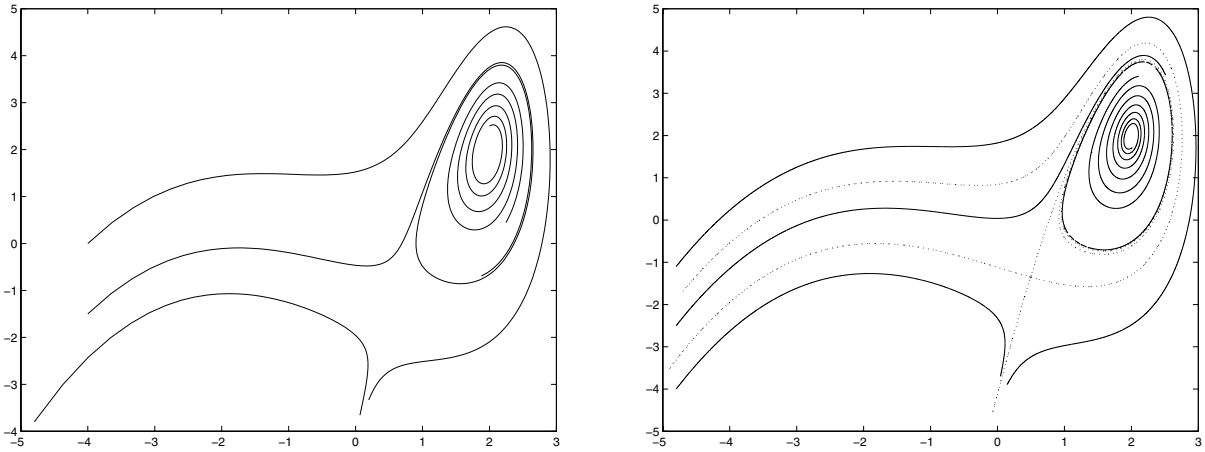

Fig. 4. Left: Phase portrait of a planar dynamical system. Right: Phase portrait of the associated piecewise linear dynamical system (with regard to a mesh of size 0.2).

7 Conclusion

In order to show that our method gives good results, we applied it to the system (8). We computed the phase portrait of the system using our method and also using a Runge-Kutta method in order to make a comparison. The results are shown in figure 4.

For our method, a regular triangular mesh of size 0.2 was used. Please note that the phase portraits are equivalent though the mesh is not very refined. The advantage of our method is here. Indeed, for many systems we do not need to have a great precision to catch its behaviour. Our method allows to compute approximate solutions of a system, and even though the mesh is not very refined, our solutions are always C^1.

There are of course some systems that our method cannot handle. These cases arise typically when the actual system (9) undergoes a bifurcation (e.g. the Jacobian of the vector field at a steady point becomes singular). We can, for example, consider the following scalar differential equation.

$$\dot{x}(t) = \alpha + \sin(x(t)),\ x(0) = 1,\ \alpha \in [0, +\infty[\ . \tag{16}$$

The solution of this equation has a horizontal asymptote for α in $[0, 1]$. For α greater than 1 the solution of (16) grows to infinity. It is clear that the solution of the associated interpolate piecewise linear equation can catch (if the mesh is refined enough) the global behaviour of the actual solution for any alpha in $[0, 1[\cup]1, \infty]$. However, for $\alpha = 1$ the approximate solution generally does not have any asymptote (see figure 5). Indeed, to catch the asymptote, we should interpolate the vector field $1 + \sin(x)$ at the point $x = 3\pi/2$ (this is generally not the case). Further researches should be done to see how such equations can be handled by our method.

Future work should focus on adaptative triangulation. Indeed, we have only used regular meshes here. But, it is clear that the mesh should be refined in some regions (e.g. neighbourhood of equilibrium points) and could be larger in some others (e.g. regions where the system is quite linear).
Another direction that we would like to explore is differential equations on surfaces. Indeed, a triangulation of the surface could be made and then the vector field could be approximated with a piecewise linear one. The applications of our method are numerous (see in [1]) and we are conviced that there will be more in the next years.

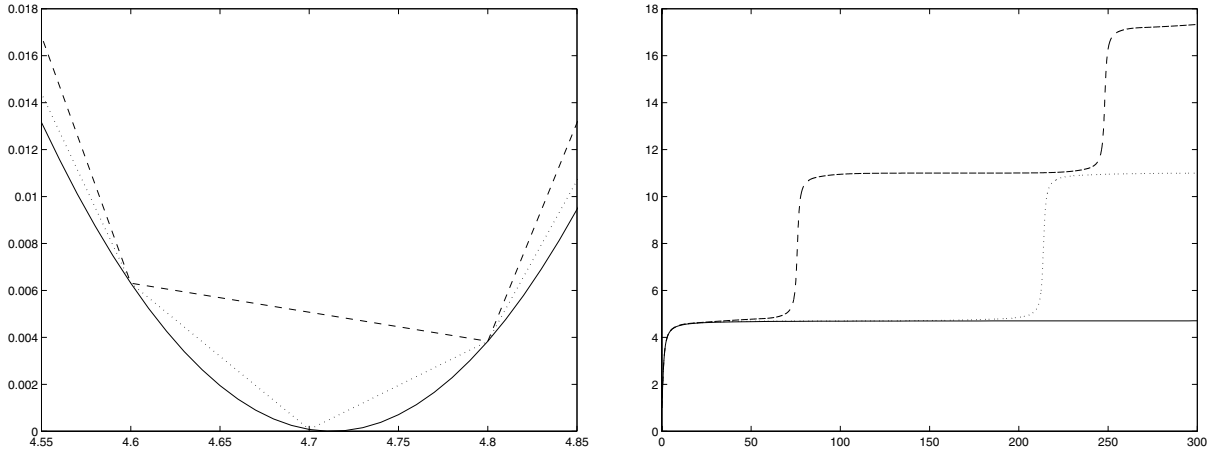

Fig. 5. Left: representative curves of $1 + \sin(x)$ (solid curve) and of its piecewise linear interpolant with regard to a subdivision of size $h = 0.2$ (dashed curve) and $h = 0.1$ (dotted curve). Right: associated solutions.

References

1. J. Della Dora, A. Maignan, M. Mirica-Ruse, S. Yovine, *Hybrid Computation*, in ISSAC'01, Ontario, Canada, July 2001.
2. J. R. Dormand, *Numerical Methods for Differential Equations*, Library of Engineering Mathematics, CRC Press, 1996.
3. A. Girard, *Analysis of Differential Equations using Hybrid Computation*, submitted to SIAM, J. Num. Anal., 2002.
4. A. Girard, *Computation of the Transitions in Piecewise Linear Hybrid Systems*, Preprint, 2002. `http://www-lmc.imag.fr/lmc-mosaic/Antoine.Girard/`.
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. J. Guckenheimer, B. Mellon, *Computing Periodic Orbits and their Bifurcations with Automatic Differentiation*, SIAM J. Sci Comput., Vol. 22, pp. 951-985, 2000.
7. P. Hartman, *Ordinary Differential Equations*, John Wiley and Sons, New York, 1964.
8. J. Hubbard, B. West, *Differential Equations: A Dynamical Systems Approach*, Texts in Applied Mathematics, Vol. 5, Springer-Verlag, New York, 1991.
9. A. Rantzer, M. Johansson, *Computation of Piecewise Quadratic Lyapunov Functions for Hybrid Systems*, in IEEE Transactions on Automatic Control, Vol 43, no 4,pp 555-559, 2001.
10. E. Sacks, *Automatic Qualitative Analysis of Dynamic Systems Using Piecewise Linear Approximations*, in Artificial Intelligence, Vol. 41, pp. 313-364, 1989.
11. D. Viswanath, *The Lindstedt-Poincaré Technique as an Algorithm for Computing Periodic Orbits*, SIAM Review, Vol. 43, no 5, PP 478-495, 2001.