Reachability Analysis via Face Lifting*

Thao Dang and Oded Maler

VERIMAG, Centre Equation, 2, av. de Vignate, 38610 Gières, France,
{Thao.Dang, Oded.Maler}@imag.fr

Abstract. In this paper we discuss the problem of calculating the reachable states of a dynamical system defined by ordinary differential equations or inclusions. We present a prototype system for approximating this set and demonstrate some experimental results.

1 Introduction

One of the main activities in verifying a discrete system consists in finding the set of system states which are reachable, via the transition relation, from a given initial set of states (control synthesis for discrete-event systems [RW89] can ultimately be reduced to some variant of reachability analysis [AMP95-b]). For small finite-state systems this is done using simple graph algorithms which manipulate set-theoretical representations of the reachable sets. For systems which are very large, or even infinite, symbolic methods are used, that is, the set of states reachable after k steps of the system is represented by some formula rather than being enumerated explicitly.

Some of this technology has been exported to certain classes of hybrid systems which deserve to be termed *piecewise-trivial dynamical systems*. These systems, such as timed automata [AD94] or PCD systems[1] [ACH+95], [AMP95-a] exhibit a trivial dynamics in the continuous phase, and all their complexity is due to the interaction between this dynamics and the discrete transitions. For such systems, given some initial polyhedral subset of the state-space, the sets of all its successors via the continuous dynamics can be calculated by straightforward linear algebraic calculation. Even with this simplicity, the reachability problem for such systems is undecidable or even worse ([HKPV95], [AM95]). A practical conclusion from the experience with this class of systems is not to look for fully-automatic decision procedures but rather for more modest goals while trying to analyze continuous systems.

In this paper we discuss the problem of extending the methodology of calculating reachable sets to systems with non-trivial continuous dynamics and no discrete dynamics at all,[2] namely systems defined by ordinary differential

* This research was supported in part by the European Community project HYBRID EC-US-043. VERIMAG is a joint laboratory of CNRS and UJF.

[1] Dynamical systems with piecewise-constant derivatives; The term *Linear Hybrid Automata* used in [ACH+95] is unfortunate and causes confusion with linear systems.

[2] Discrete transitions can later be incorporated naturally into the continuous techniques, if and when such techniques are established.

equations. We formulate the problem and describe a technique, suggested by M. Greenstreet [G96], for over-approximating reachable sets. We then introduce a variation on this technique which can be applied more easily to more than two dimensions. Finally we show the results obtained by an experimental implementation of the algorithm for both linear and non-linear systems.

2 Statement of the Problem

2.1 Deterministic Systems

Definition 1 [Dynamical System. *] A differential dynamical system is $S = (X, f)$ where $X = \mathbb{R}^n$ is the Euclidean space and $f : X \to X$ is a continuous function (vector field). A behavior of S starting from a point $x_0 \in X$ is a trajectory $\xi : \mathbb{R}_+ \to X$ satisfying $\xi[0] = x_0$ and for every t,*

$$d\xi[t]/dt = f(\xi[t]).$$

People less pedantic than the average formal methodologists would simply say:

$$\dot{x} = f(x).$$

It can also be expressed in a somewhat more operational manner:

$$\xi[t] = x_0 + \int_0^t f(\xi[\tau])d\tau.$$

The set of states reachable by the system from x_0 is defined as

$$Reach(x_0, f) = \{\xi[t] : t \geq 0\}.$$

Typically when we want to prove safety properties of such a system we would like to show that $Reach(x_0, f) \cap Q = \emptyset$ for some $Q \subseteq X$. Except for the rare case when $Reach(x_0, f)$ has a closed-form solution, such as $\{x_0 e^{At} : t \in \mathbb{R}_+\}$ for linear systems, the common way to achieve that goal is to use numerical integration to calculate an approximation of $Reach(x_0, f)$ incrementally. This means starting from $\xi[0] = x_0$ and applying some iteration

$$\xi[(n+1)\Delta] = \xi[n\Delta] + g(\xi[n\Delta])$$

where Δ is the discretization step and g is supposed to be a good approximation of the integral.

According to the strict standards of discrete verification, this approach is far from being satisfactory: first, we compute ξ only for a small subset of time points, and we might miss a visit of the system in Q at some t, $n\Delta < t < (n+1)\Delta$. Secondly, even for points of the form $t = n\Delta$, we compute only an approximation of $\xi[t]$. And finally, the calculation is not guaranteed to terminate (and if it terminates, it is not always for a good reason). Termination of the calculation of $Reach(x_0, f)$ means that the trajectory becomes periodic,[3] i.e. $\xi[t] = \xi[t']$ for some $t' > t$, which may sometimes happen numerically only because we approximate the ideal mathematical reals by a finite subset of the rationals. Nevertheless, generations of mathematicians, pure and applied, assure us that given reasonable f and Q, we can find Δ and g such that we need not worry about the first two problems. As for the third one, we should accept it as a sad fact of life, as do all engineers who use simulation methods.

To summarize, given a system (X, f), an inital state x_0 and a set of bad states Q, we have a methodoloy, or a semi-algorithm (modulo some numerological conditions) for verifying that from x_0 you never reach Q:

$R_0 := \{x_0\}$;
repeat $\quad i = 1, 2 \ldots$
$\quad R_i := R_{i-1} \cup Next(R_{i-1})$
until $\quad (R_i = R_{i-1}) \vee (R_i \cap Q \neq \emptyset) \vee$ (The user gives up)

Here, $Next(R_i)$ means just integrating numerically starting from the last element of R_i. Up to this point this is nothing but rephrasing, in a somewhat awkward manner, the common practice of simulation.

2.2 Non-deterministic Systems

In many situations we cannot be sure of the initial conditions nor of the dynamics of the system. In most cases we will have an equation of the form

$$\dot{x} = f(x, u).$$

where u is some unobserved external disturbance, about which we know only some constraints.[4] The behavior of the system resulting from interaction with any admissible input u can be characterized using *differential inclusion* [AC84] of the form

$$\dot{x} \in F(x),$$

where $F : X \to 2^X$ is roughly

$$\bigcup_u f(x, u).$$

This is the continuous analogue of a non-deterministic transition system. Such a system, when started at some initial state x_0, usually produces dense bundles of trajectories (solutions), which we denote by $L(F, x_0)$. The set of states reachable from x_0 at time t (which was simply $\{\xi[t] : t \in \mathbb{R}_+\}$ in deterministic systems) is defined as

$$Reach_t(x_0, F) = \bigcup_{\xi \in L(F, x_0)} \xi[t].$$

[3] Which is always the case in finite-state systems.

[4] Things get even more complicated in control synthesis problems whose generic form is $\dot{x} = f(x, u, v)$ where u and v are two different types of external inputs.

The set of all states visited during the interval $[0, t]$ is

$$Reach_{[0,t]}(x_0, F) = \bigcup_{\tau \in [0,t]} Reach_\tau(x_0, F)$$

and the set of all reachable states is

$$Reach(x_0, F) = Reach_{[0,\infty]}(x_0, F).$$

In order to apply the symbolic verification methodology we would like to have a diverging sequence $t_0, t_1, \ldots$ of time points and calculate a sequence $R_0, R_1 \ldots$ such that $R_0 = \{x_0\}$ and for every i, $R_i = Reach_{[0,t_i]}(x_0, F)$. As in the case of numerical integration of a single trajectory, the calculation of R_{i+1} will be based on f and R_i, and from a computational viewpoint, the main novel feature here is the calculation of differential successors of *a set of points* rather than that of a *single point*. This motivates us to attack first a slightly more restricted version of the problem: calculating the reachable states of a *deterministic* system starting from a *set* $P \subseteq X$, namely to find

$$Reach(P, f) = \bigcup_{x \in P} Reach(x, f).$$

This problem already exhibits the major computational difficulty associated with representing and simulating a set of trajectories (see figure 1 for an illustration of the above notions).

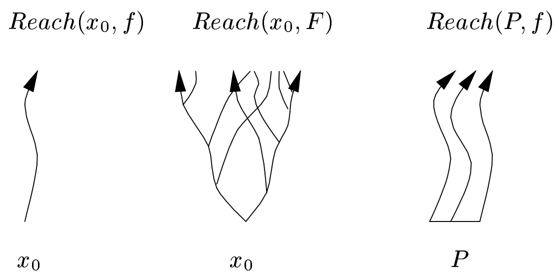


Fig. 1. Calculating reachable states for: 1) A deterministic system starting at a point, 2) A non-deterministic system starting at a point and 3) A deterministic system starting at a set.

3 The Face Lifting Approach

We assume from now on that everything takes place inside a bounded subset of X in which f is Lipschitz.

3.1 Arbitrary Polyhedra

The first ingredient of any solution is a formalism for representing subsets of X. Not being computer algebraists, we restrict ourselves to polyhedral sets. These are sets which can be written as boolean combinations of linear inequalities.[5] Polyhedral sets come in two major varieties, convex and non-convex. Those of the former type can be written as conjunctions of inequalities (intersections of half-spaces) and they are uniquely determined by their sets of vertices.

If the initial set P is convex and f preserves convexity (as in the case of linear systems), we are lucky because for every t we have

$$Reach_t(conv(x_1,\ldots,x_n),f) = conv(Reach_t(x_1,f),\ldots,Reach_t(x_n,f))$$

where *conv* denotes the convex hull. With this property it would have been sufficient to simulate a finite number of trajectories starting at the vertices. However, in the case of arbitrary differential systems, the approximation of a non-convex polyhedron by its convex hull is usually useless. Just consider what such an approximation gives when P contains a bifurcation point.

The treatment of non-convex polyhedra poses enormous problems in terms of representation, normal forms (which are important to detect the condition $R_{i+1} = R_i$), etc. In the sequel we present a technique, due to M. Greenstreet [G96], which we call *face lifting*. In the abstract sense, face lifting can be applied to systems in *any* dimension, but concretely, its practical application to 3 or more dimensions is not at all evident.

The approach is based, first of all, on the following basic observation concerning continuous trajectories: if some point $y \in Reach_t(x,f) - P$ for an interior point $x \in P$, then there exists a point $x' \in bd(P)$ (the boundary of P) and $t' < t$ such that $y \in Reach_{t'}(x',f)$. In other words,

$$Reach_{[0,t]}(P) = P \cup Reach_{[0,t]}(bd(P)).$$

Hence, when coming to calculate R_{i+1} from R_i it is sufficient to look at the boundary of the latter (the union of its *faces* in the case of polyhedral sets and, in particular, its *edges* in 2-dim).

Consider a face e of a polyhedron such that it is included in the set characterized by the linear equality $a \cdot x = b$. Let $\hat{f}_e(x)$ denote the *outward component* of $f(x)$ relative to e, that is, the projection of $f(x)$ on the normal to e, and let $\hat{f}(e)$ denote its maximum over $x \in N(e)$, where $N(e)$ is some neighborhood of e. Clearly, if $\hat{f}(e)$ is negative, the face does not contribute new reachable states which cannot be reached from other faces. Otherwise, for every Δ, one can find an ε such that all the points reachable from e in time Δ satisfy

$$a \cdot x \leq b + \Delta \cdot \hat{f}(e) + \varepsilon.$$

Geometrically speaking, this amounts to lifting the face e outward by $\Delta \cdot \hat{f}(e) + \varepsilon$ (see figure 2). (We omit some details concerning the relation between Δ,$N(e)$,

[5] If you want to impress non-logicians, you can say they are possible models of sentences in the first order theory of $(\mathbb{R},+,<)$ or something.

ε and the Lipschitz constant of f, which guarantees the desired property of the approximation). This gives the following procedure for over-approximating $\mathit{Reach}_{[0,\Delta]}(P, f)$:

Calculate $\hat{f}(e)$ for every face e of P. Based on these find the appropriate ε and push every e whose $\hat{f}(e)$ is positive by $\Delta \cdot \hat{f}(e) + \varepsilon$ to obtain P'.

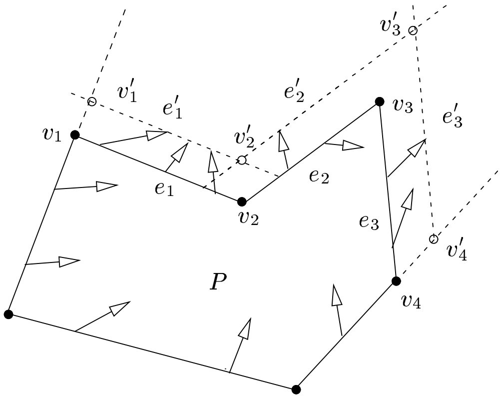


Fig. 2. A 2-dimensional example of the approach: a polyhedron P and a sample of the values of f on its edges. Only edges e_1, e_2 and e_3 have a positive outward component of f and they are pushed into e'_1, e'_2 and e'_3. The vertices $\{v_1, \ldots, v_4\}$ are replaced by $\{v'_1, \ldots, v'_4\}$..

By construction, we have $\mathit{Reach}_{[0,\Delta]}(P, f) \subseteq P'$. It can be shown that locally, you can make the difference between the reachable set and its approximation as small as you like, by taking smaller Δ. Better approximation can be achieved by cutting a face into sub-faces whenever $\hat{f}$ has a large variation over the face. However, there are cases where, in the long run, the method will produce unboundedly large over-approximations of $\mathit{Reach}(P, f)$, as shown in figure 3.

We have implemented the method for dimension 2 and obtained results similar to those obtained by other means (see section 4 for experimental results). However the extension to more than two dimensions is difficult as the special properties of the plane no more hold. In $\mathbb{R}^2$, an ordered set of vertices always defines a unique polygon[6] and the abstract operation of identifying a face can be realized by picking a pair of neighboring vertices. Similarly, the face lifting operation can ultimately be realized by replacing vertices in a list.

This is not true in more than two dimensions, where even convex polyhedra can exhibit a complicated structure with degeneracy which makes face recognition very hard. Consequently, we have tried another approach, slightly inspired

[6] In fact, if we do not insist on *connected* polygons, it defines either the polygon or its complement.

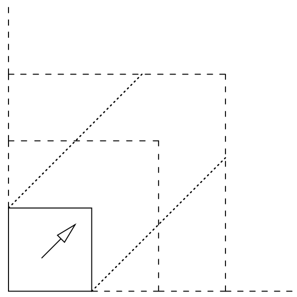


Fig. 3. A bad example: consider an axes-parallel rectangle and a constant vector field f with non-zero components in both dimensions. The reachable set lies between the two dotted diagonal lines, but the method will produce the whole upper-left orthant.

by the basic ideas underlying the numerical solution of PDEs.

3.2 Griddy and Isothetic Polyhedra

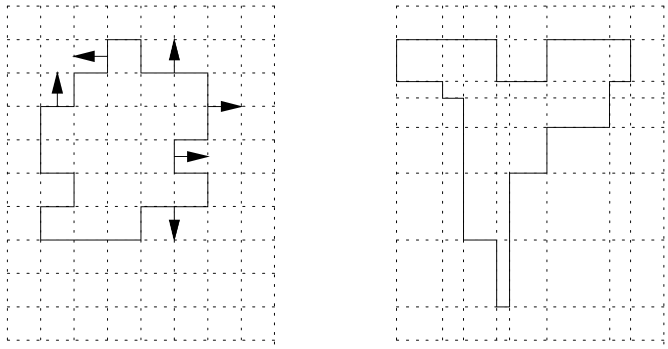
Consider the sub-class of polyhedra which can be obtained by boolean combinations of inequalities of the form $x_i \leq c$ where x_i is a component of x and c is an integer constant.[7] In other words, we partition the space into uniform hyper-rectangles and consider all polyhedra which can written as unions of those (see figure 4-a). We call these *griddy polyhedra*.

Since such polyhedra are "finitely generated" (in a bounded sub-space) they admit a very simple representation using n-dimensional $0-1$ matrices. It is also easy to determine whether an $(n-1)$-dimensional hypercube is indeed part of the face of the polyhedron, and there is a systematic simple way to enumerate all the faces and calculate $\hat{f}$, which is now always parallel to one of the axes (see figure 4-a). With such a representation we can apply, in principle, face lifting in *any* dimension.

Techniques developed for griddy polyhedra can be adapted to the more general class of *isothetic* polyhedra, generated by arbitrary axes-parallel hyper-rectangles. These can be represented by a non-uniform grid depending on the represented polyhedron. The set of grid coordinates in any dimension consists of all projections of vertices of the polyhedron (see figure 4-b) and may change during the computation. The non-uniform grid has two main advantages over the uniform one:

1. Space: a griddy polyhedron which can be decomposed into few large rectangles can be represented more succinctly. However, when this method is used to represent, say, an approximation of a circle, the grid becomes very dense and this advantage is lost.

[7] Of course, c can belong to the set of integer multiples of some rational constant as well.



(a) (b)

Fig. 4. (a) A Griddy Polygon. Some of the faces are annotated by their corresponding outward directions. (b) An isothetic polygon and its associated non-uniform grid. Face lifting can cause a refinement of the grid.

2. Expressive power and accuracy: with a fixed grid we need to push every face further to the next integer value, which sometimes creates an unnecessary over-approximation, beyond what is inherent in face lifting alone (see example in the next section). With a variable grid we can push faces as little as we want.

Both methods are not very space efficient and we are currently investigating a canonical and much more succinct representation of these polyhedra.

4 Experimental Results

We have implemented griddy face lifting in 2 and 3 dimensions using the abovementioned representation methods. For the uniform grid we use simply an n-dimensional array. For the non-uniform grid we use a linked list representation which currently consumes much more computation time.

In both methods we decompose every face into elementary hyper-rectangular elements and apply the basic operation of numerical optimization of $\hat{f}$ to every such element. This is, of course, less efficient than a coarser decomposition of the face into larger hyper-rectangles, an approach we intend to implement in the future. On the other hand, this is better in terms of accuracy. All the results described below, except for the 3-dimensional example, were obtained using the fixed grid implementation.

4.1 Linear Systems in $\mathbb{R}^2$

In figure 5 we demonstrate the behavior of the algorithm on various classes of linear systems of the form $\dot{x} = Ax$ (see [HS74] for the classification). We treat the following cases:

Type	A	Initial set
Center	$\begin{pmatrix} 0.0 & -6.0 \\ 3.0 & 0.0 \end{pmatrix}$	$[-0.25, 0.25] \times [-0.25, 0.25]$
Node	$\begin{pmatrix} -5.0 & 0.0 \\ 0.0 & -2.0 \end{pmatrix}$	$[0.2, 0.5] \times [0.2, 0.4]$
Saddle	$\begin{pmatrix} -5.0 & 0.0 \\ 0.0 & 4.0 \end{pmatrix}$	$[0.0, 0.4] \times [-0.0, 0.4]$
Sink	$\begin{pmatrix} -2.0 & -3.0 \\ 3.0 & -2.0 \end{pmatrix}$	$[-0.1, 0.3] \times [0.1, 0.3]$

Sometimes, the use of a fixed grid generates an over-approximation which covers all the space. This is evident in the case of a *center* where every edge will have a non-zero outward component in some dimension.[8] Consequently we have changed in these cases the rounding rule to obtain the desired result, that is, we push a face to the nearest grid unit and not necessarily outward. The price is in not being an over-approximation anymore. Using a variable grid is another way to solve this problem. Note that optimization of a linear $\hat{f}$ is much cheaper computationally in the linear case.

4.2 Mixing Tank

This example, taken from [SKE97], is a typical non-linear equation encountered in chemical engineering. The variables x_1, x_2 denote, respectively, the height and the concentration of liquid in a mixing tank with two inlets (with different rates and concentrations) and one outlet. The equation is

$$\dot{x}_1 = a_1 - a_2\sqrt{x_1}$$

$$\dot{x}_2 = \tfrac{1}{a_3 x_1}(1 - a_4 x_2)$$

With our choice of parameters, $(1.322, 1.652)$ is an equilibrium state of the system. In figure 6 the states reachable from an initial set $[1.12 \times 1.17] \times [1.56 \times 1.68]$ are depicted, and one can see the convergence to the equilibrium.

[8] At least, this case is not generic.

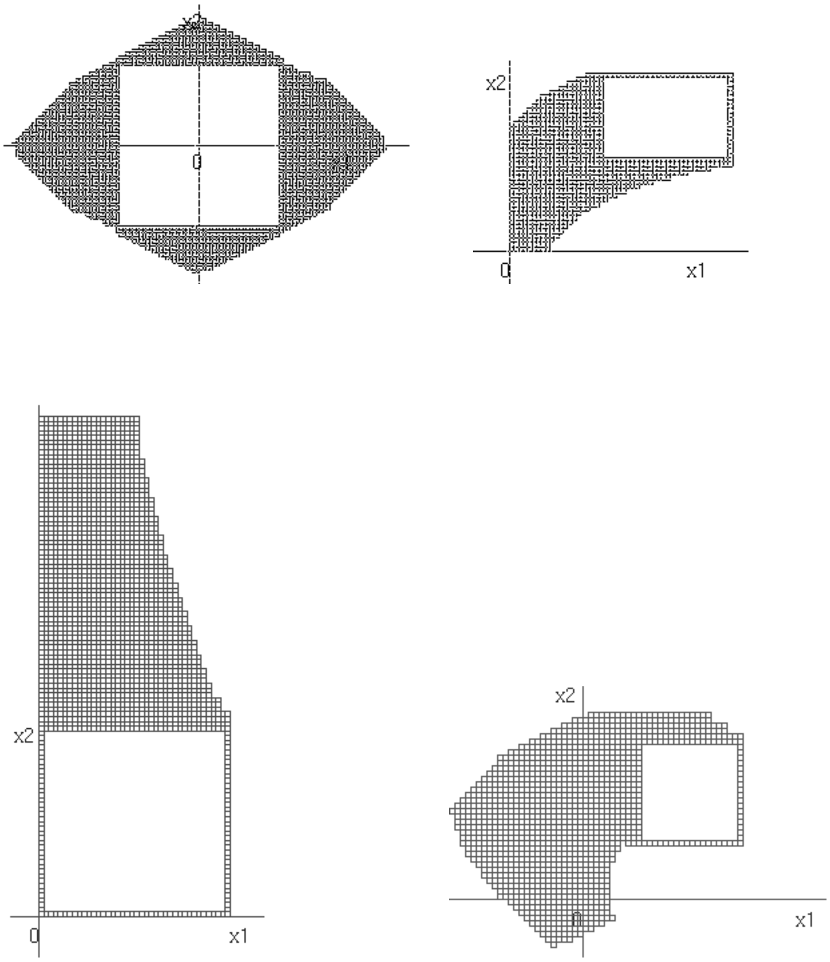


Fig. 5. Reachable sets of linear systems of type: 1) Center, 2) Node, 3) Saddle and 4) Sink. The white rectangles denote the initial sets.

4.3 Airplane Safety

The next example is taken from [LTS97]. The state variables x_1, x_2 represent, respectively, the velocity and the flight path angle. Their evolution is governed by

$$\dot{x}_1 = -\frac{a_D x_1^2}{m} - g \sin x_2 + \frac{u_1}{m}$$

$$\dot{x}_2 = \frac{a_L x_1 (1 - c x_2)}{m} - \frac{g \cos x_2}{x_1} + \frac{A_L c x_1}{m} u_2$$

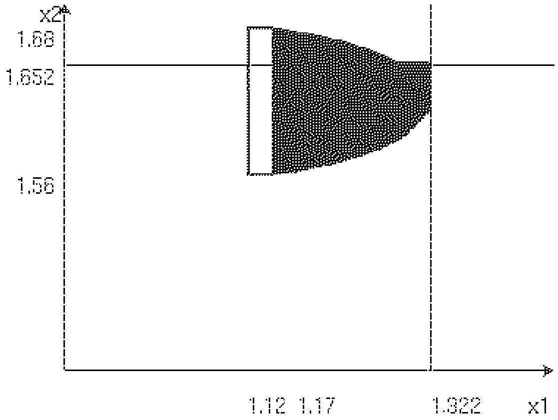


Fig. 6. Mixing Tank

The problem is to determine the safe subset of the state-space, i.e. the states from which the system does not leave the envelope P defined as the rectangle $[V_{min}, V_{max}] \times [\Theta_{min}, \Theta_{max}]$. This is equivalent to calculating the complement of the set of states reachable from $X - P$ by the reverse system. The results, depicted in figure 7 correspond to specific choices of values for parameters and for the controls $u_1 = \theta_{min}, u_2 = T_{max}$ (left) and $u_1 = \theta_{max}, u_2 = T_{min}$ (right). The results are consistent with those obtained in [LTS97].

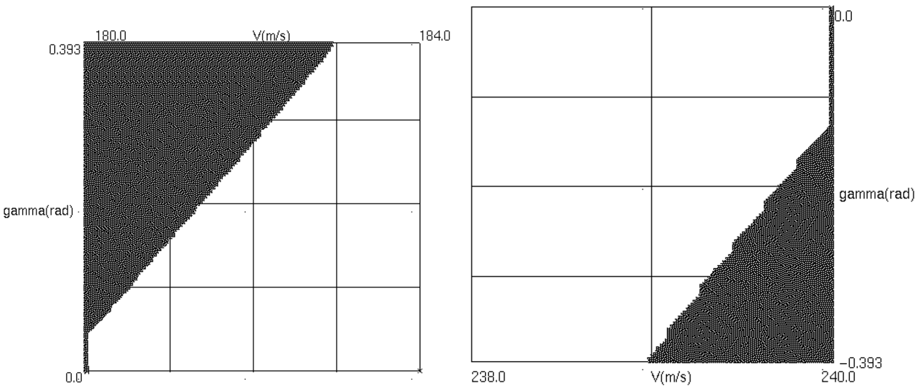


Fig. 7. Airplane Safety

4.4 Linear Systems in $\mathbb{R}^3$

In figure 8 one can see the reachable set of a $3-$dimensional system with

$$A = \begin{pmatrix} -2 & 0 & 0 \\ 1 & -2 & 0 \\ 0 & 1 & -2 \end{pmatrix}$$

starting from the initial region $[-0.025, 0.025] \times [-0.1, 0.1] \times [0.05, 0.07]$.

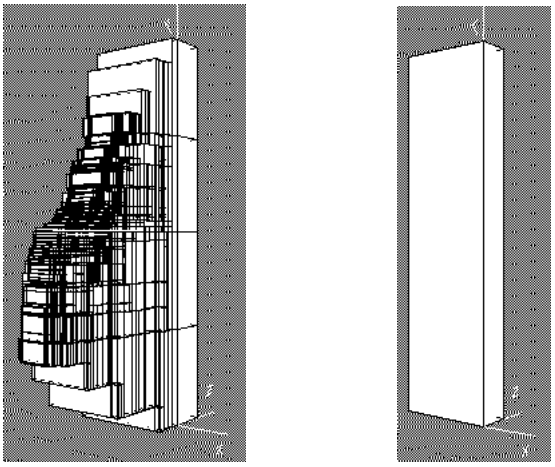


Fig. 8. Reachable states (left) starting from an initial region (right) for a 3-dimensional linear system.

5 Relation to other Work

There are various works concerning the calculation of reachable sets for differential inclusions. Many of these works are numerical analytic in nature, concerned mostly with calculation of abstract error bounds and less with the crucial questions of data-structures for high dimensional sets.

The problem of calculating $Reach(P, f)$ can be rephrased as a PDE[9]

$$\frac{\partial \varphi}{\partial t} = -\mathrm{grad}(\varphi) \cdot f$$

where $\varphi : X \times \mathbb{R}_+ \rightarrow \{0, 1\}$ is defined as $\varphi(x, t) = 1$ iff $x \in Reach_{[0,t]}(P, f)$ and in particular $\varphi(x, 0) = 1$ iff $x \in P$. Sometime a "continualized" version of φ is

[9] We owe this insight to P. Caspi [C93]. See also [TPS98] for a PDE-based approach.

used, namely a function $\varphi : X \times I\!\!R_+ \rightarrow I\!\!R$ such that $\varphi(x, 0) = 0$ exactly when x is on the boundary of P and $f(x, 0) > 0$ if x is inside P. Various methods exist for tracking the evolution of φ, see, e.g. [S96]. So far we have found no special computational nor didactic advantage in viewing the problem as a PDE instead of a direct ODE formulation, but this might change in the future.

In [PBV96] an alternative approach was suggested based on cutting the state-space into cubes, and associating with every cube a *rectangular differential inclusion* which is a differential inclusion of the form $c_i < \dot{x}_i < d_i$ for every i, with constants c_i and d_i. The reachability problem is decidable for this class of systems [PV94], and the idea here is to do *exact* calculations on an *approximate* model, where the bounds on f are calculated in a preprocessing stage. Similar to face lifting, this approach can guarantee, by refining the grid, error bounds only for a *finite* time horizon. This approach has been applied to several examples in [HW96] and in [SKE97]. Some of the ideas underlying face lifting appear already in [KM91] where the authors try to prove a homomorphism from a transistor-level differential model into an automaton. While doing so they also cut the space into a grid and try to calculate the reachability relation among cubes.

Finally, in [G96], [GM98], the authors try to extend face lifting to higher dimensions using another strategy. They restrict themselves to polyhedra which can be written as intersections of cylindrifications of two-dimensional (arbitrary) polygons. This way all the operations are performed on the two-dimensional projections of the polyhedron. There are obvious advantages and shortcomings of this approach compared to the grid-based one, and only time will tell their relative performances in practice.

Acknowledgment We thank Mark Greenstreet for introducing us to the face lifting concept, and for answering many technical questions. Part of this work was done while the second author was visiting Berkeley, benefiting from discussions with P. Varaiya, S. Sastry, C. Tomlin, G. Pappas and many others. At VERIMAG we are indebted to comments of E. Asarin, O. Bournez and P. Caspi on dynamical systems and to the help of Y. Raoul and S. Tripakis in software engineering.

References

[AD94] R. Alur and D.L. Dill, A Theory of Timed Automata, *Theoretical Computer Science* 126, 183–235, 1994.

[ACH+95] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis and S. Yovine, The Algorithmic Analysis of Hybrid Systems, *Theoretical Computer Science* 138, 3–34, 1995.

[AM95] E. Asarin and O. Maler, Achilles and the Tortoise Climbing Up the Arithmetical Hierarchy, in P.S. Thiagarajan (Ed.), Proc. FST/TCS'95, 471-483, LNCS 1026, Springer, 1995.

[AMP95-a] A. Asarin, O. Maler and A. Pnueli, Reachability Analysis of Dynamical Systems having Piecewise-Constant Derivatives, *Theoretical Computer Science* 138, 35-66, 1995.

[AMP95-b] E. Asarin, O. Maler and A. Pnueli, Symbolic Controller Synthesis for Discrete and Timed Systems, in P. Antsaklis, W. Kohn, A. Nerode and S. Sastry(Eds.), *Hybrid Systems II*, LNCS 999, Springer, 1995.

[AC84] J.-P. Aubin and A. Cellina, *Differential Inclusions: Set-valued Maps and Viability Theory*, Springer, 1984.

[C93] P. Caspi, Global Simulation via Partial Differential Equations, Unpublished note, Verimag, 1993.

[G96] M.R. Greenstreet, Verifying Safety Properties of Differential Equations, in *Proc. CAV'96*, 277-287, 1996.

[GM98] M.R. Greenstreet and I. Mitchell, Integrating Projections, these proceedings.

[HW96] T.A. Henzinger and H. Wong-Toi, Linear Phase-Portrait Approximation for Nonlinear Hybrid Systems, in R. Alur, T.A. Henzinger and E.D. Sontag (Eds.), *Hybrid Systems III*, 377-388, LNCS 1066, Springer, 1996.

[HKPV95] T.A. Henzinger, P.W. Kopke, A. Puri and P. Varaiya, What's Decidable about Hybrid Automata?, *Proc. 27th STOC*, 373–382, 1995.

[HS74] M.W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems and Linear Algebra*, Academic Press, 1974.

[KM91] R.P. Kurshan and K.L. McMillan, Analysis of Digital Circuits Through Symbolic Reduction, *IEEE Trans. on Computer-Aided Design*, 10, 1350-1371, 1991.

[LTS97] J. Lygeros, C. Tomlin and S. Sastry, Multiobjective Hybrid Controller Synthesis, in O. Maler (Ed.), *Proc. Int. Workshop on Hybrid and Real-Time Systems*, 109-123, LNCS 1201, Springer, 1997.

[PBV96] A. Puri, V. Borkar and P. Varaiya, ε-Approximation of Differential Inclusions, in R. Alur, T.A. Henzinger and E.D. Sontag (Eds.), *Hybrid Systems III*, 363-376, LNCS 1066, Springer, 1996.

[PV94] A. Puri and P. Varaiya, Decidability of Hybrid Systems with Rectangular Differential Inclusions, in D. Dill (Ed.), *Proc. CAV '94*, LNCS 1066, Springer, 1996.

[RW89] P.J. Ramadge and W.M. Wonham, The Control of Discrete Event Systems, *Proc. of the IEEE* 77, 81-98, 1989.

[S96] J.A. Sethian, *Level Set Methods : Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge, 1996.

[SKE97] O. Stursberg, S. Kowalewski ans S. Engell, Generating Timed Discrete Models of Continuous Systems, in *Proc. MATHMOD'97*, Vienna, 1997.

[TPS98] C. Tomlin, G. Pappas, and S. Sastry, Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems, *IEEE Trans. on Automatic Control*, to appear.