

# Parameter Synthesis in Nonlinear Dynamical Systems: Application to Systems Biology

Alexandre Donzé<sup>1</sup>, Gilles Clermont<sup>2</sup>,  
Axel Legay<sup>1</sup>, and Christopher J. Langmead<sup>1,3\*</sup>

<sup>1</sup> Computer Science Department, Carnegie Mellon University, Pittsburgh, PA

<sup>2</sup> Department of Critical Care Medicine, University of Pittsburgh, Pittsburgh, PA

<sup>3</sup> Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA

**Abstract.** The dynamics of biological processes are often modeled as systems of nonlinear ordinary differential equations (ODE). An important feature of nonlinear ODEs is that seemingly minor changes in initial conditions or parameters can lead to radically different behaviors. This is problematic because in general it is never possible to know/measure the precise state of any biological system due to measurement errors. The parameter synthesis problem is to identify sets of parameters (including initial conditions) for which a given system of nonlinear ODEs does not reach a given set of undesirable states. We present an efficient algorithm for solving this problem that combines sensitivity analysis with an efficient search over initial conditions. It scales to high-dimensional models and is exact if the given model is affine. We demonstrate our method on a model of the acute inflammatory response to bacterial infection, and identify initial conditions consistent with 3 biologically relevant outcomes.

**Key words:** Verification, Nonlinear Dynamical Systems, Uncertainty, Systems Biology, Acute Illness

## 1 Introduction

The fields of Systems Biology, Synthetic Biology, and Medicine produce and use a variety of formalisms for modeling the dynamics of biological systems. Regardless of its mathematical form, a model is an invaluable tool for thoroughly examining how the behavior of a system changes when the initial conditions are altered. Such studies can be used to generate verifiable predictions, and/or to address the uncertainty associated with experimental measurements obtained from real systems.

In this paper, we consider the *parameter synthesis problem* which is to identify sets of parameters for which the system does (or does not) reach a given set of states. Here, the term “parameter” refers to both the initial conditions of the model (e.g., bacterial load at time  $t = 0$ ) and dynamical parameters (e.g., the bacterium’s doubling rate). For example, in the context of medicine, we might be interested in partitioning the parameter space into two regions — those that, without medical intervention, deterministically lead to the patient’s recovery, and those that lead to the patient’s death. The parameter synthesis problem is relatively easy to solve when the system has linear dynamics, and there are a variety of methods for doing so (e.g., [6–8]). Our algorithm, in contrast, solves the parameter synthesis problem for *nonlinear dynamical systems*. That is, for systems of nonlinear ordinary differential equations (ODEs). Moreover, our approach can also be extended to nonlinear hybrid systems (i.e., those containing mixtures of discrete and continuous variables, see [11] for details). Nonlinear ODE and hybrid models are very common in the Systems Biology, Synthetic Biology, and in Medical literature but there are very few techniques for solving the parameter synthesis problem in such systems. This paper’s primary contribution is a practical algorithm that can handle systems of this complexity.

Our algorithm combines sensitivity analysis with an efficient search over parameters. The method is exact if the model has affine dynamics. For nonlinear dynamical systems, we can guarantee an arbitrarily

---

\* Corresponding Author: [cjl@cs.cmu.edu](mailto:cjl@cs.cmu.edu)

high degree of accuracy with respect to identifying the boundary delineating reachable and non-reachable sets. Moreover, our method runs in minutes, even on high-dimensional models. We demonstrate the method by examining two models of the inflammatory response to bacterial infection [20, 26]. In each case, we identify sets of initial conditions that lead to each of 3 biologically relevant outcomes.

The contributions of this paper are as follows:

- An algorithm for computing parameter synthesis in nonlinear dynamical systems. This work builds on and extends formal verification techniques that were first introduced in the context of continuous and hybrid nonlinear dynamical systems [13].
- The results of two studies on two different models of the inflammatory response to bacterial infection. The first model is a 4-equation model, the second is a 17-equation model.

This paper is organized as follows: We outline previous work in reachability for biological systems in Sec. 2. Next, we present our algorithm in Sec. 3. We demonstrate our method on two models of acute inflammation in Sec. 4. We finish by discussing our results and ideas for future work in Sec. 5.

## 2 Background

Our work falls under the category of *formal verification*, a large area of research which focus on techniques for computing provable guarantees that a system satisfies a given property. Formal verification methods can be characterized by the kind of system they consider (e.g., discrete-time vs continuous-time, finite-state vs continuous-state, linear vs non-linear dynamics, etc), and by the kind of properties they can verify (e.g., reachability – the system can be in a given state, liveness – the system will be in a given set of state infinitely often, etc). The algorithm presented in this paper is intended for verifying reachability properties under parameter uncertainty in nonlinear hybrid systems. The most closely related work in this area uses *symbolic* methods for restricted class of models (e.g., timed automata [4], linear hybrid systems [1, 19, 17]). Symbolic methods for hybrid systems have the advantage that they are exhaustive, but in general only scale to systems of small size ( $< 10$  continuous state variables). Another class of techniques invokes abstractions of the model [2]. Such methods have been applied to biological systems whose dynamics can be described by multi-affine functions. Here, examples include applications to genetic regulatory networks (e.g., [6–8]). Batt and co-workers proposed an approach to verify reachability and liveness properties written in the linear temporal logic (LTL) [24] (LTL can be used to check assumptions about the future such as equilibrium points) of genetic regulatory networks under parameter uncertainty. In that work, the authors show that one can reduce the verification of qualitative properties of genetic regulatory networks to the application of Model Checking techniques [10] on a conservative discrete abstraction. Our method is more general in the sense that we can handle arbitrary nonlinear systems but a limitation is that we cannot handle liveness properties. However, we believe that our algorithm can be extended to handle liveness properties by combining it with a recent technique proposed by Fainekos [15]. We note that there is also work in the area of analyzing piecewise (stochastic) hybrid systems (e.g., [14, 16, 18, 9]). Our method does not handle stochastic models at the present time.

Several techniques relying on numerical computations of the reachable set apply to systems with general nonlinear dynamics ([5, 28, 22]). In [5], the authors presents an hybridization technique, which consists in approximating the system with a piecewise-affine approximation to take advantage of the wider family of methods existing for this class of systems. In [21], the authors reduce the reachability problem to a partial differential equation which they solve numerically. As far as we know, none of these techniques have been

applied successfully to nonlinear systems of more than a few variables. By contrast, our method builds on techniques proposed in [12, 13] which can be applied to significantly larger models.

A more “traditional” tool used for the analysis of nonlinear ODEs is bifurcation analysis, which was applied to the biological models used in our experiments ([26, 20]). Our approach deviates from bifurcation analysis in several ways. First, it is simpler to apply since it only relies on the capacity to compute numerical simulations for the system, avoiding the need of computing equilibrium points or limit cycles. Second, it provides the capacity of analyzing transient behaviors. Finally, when it encounters an ambiguous behavior (e.g., bi-stability) for a given parameter set, it reports that the parameter has uncertain dynamics and can refine the result to make such uncertain sets as small as desired.

### 3 Algorithm

In this section, we give a mathematical description of the main algorithm used in this work.

#### 3.1 Preliminaries

The set  $\mathbb{R}^n$  and the set of  $n \times n$  matrices are equipped with the infinite norm, noted  $\|\cdot\|$ . We define the diameter of a compact set  $\mathcal{R}$  to be  $\|\mathcal{R}\| = \sup_{(x,x') \in \mathcal{R}^2} \|x - x'\|$ . The distance from  $x$  to  $\mathcal{R}$  is  $d(x, \mathcal{R}) = \inf_{y \in \mathcal{R}} \|x - y\|$ . The Hausdorff distance between two sets  $\mathcal{R}_1$  and  $\mathcal{R}_2$  is:

$$d_H(\mathcal{R}_1, \mathcal{R}_2) = \max\left(\sup_{x_1 \in \mathcal{R}_1} d(x_1, \mathcal{R}_2), \sup_{x_2 \in \mathcal{R}_2} d(x_2, \mathcal{R}_1)\right).$$

Given a matrix  $S$  and a set  $\mathcal{P}$ ,  $S\mathcal{P}$  represents the set  $\{Sp, p \in \mathcal{P}\}$ . Given two sets  $\mathcal{R}_1$  and  $\mathcal{R}_2$ ,  $\mathcal{R}_1 \oplus \mathcal{R}_2$  is the Minkowski sum of  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , i.e.,  $\mathcal{R}_1 \oplus \mathcal{R}_2 = \{x_1 + x_2, x_1 \in \mathcal{R}_1, x_2 \in \mathcal{R}_2\}$ .

#### 3.2 Simulation and Sensitivity Analysis

We consider a dynamical system  $Sys = (f, \mathcal{P})$  of the form:

$$\dot{x} = f(t, x, p), \quad p \in \mathcal{P}, \quad (1)$$

where  $x \in \mathbb{R}^n$ ,  $p$  is a *parameter vector* and  $\mathcal{P}$  is a compact subset of  $\mathbb{R}^{n_p}$ . We assume that  $f$  is continuously differentiable. Let  $\mathcal{T} \subset \mathbb{R}^+$  be a time set. For a given  $p$ , a *trajectory*  $\xi_p$  is a function of  $\mathcal{T}$  which satisfies the ODE (Eq. 1), i.e., for all  $t$  in  $\mathcal{T}$ ,  $\dot{\xi}_p(t) = f(t, \xi_p(t), p)$ . For convenience, we include the initial state in the parameter vector by assuming that if  $p = (p_1, p_2, \dots, p_{n_p})$  then  $\xi_p(0) = (p_1(0), p_2(0), \dots, p_{n_p}(0))$ . Under these conditions, we know by the Cauchy-Lipshitz theorem that the trajectory  $\xi_p$  is uniquely defined.

The purpose of sensitivity analysis techniques is to predict the influence on a trajectory of a perturbation of its parameter vector. A first order approximation of this influence can be obtained by a Taylor expansion of  $\xi_p(t)$  around  $p$ . Let  $\delta p \in \mathbb{R}^{n_p}$ . We have:

$$\xi_{p+\delta p}(t) = \xi_p(t) + \frac{\partial \xi_p}{\partial p}(t) \delta p + \mathcal{O}(\|\delta p\|^2). \quad (2)$$

The second term in the right hand side of Eq. (2) is the derivative of the trajectory with respect to  $p$ . Since  $p$  is a vector, this derivative is a matrix, which is called the *sensitivity matrix*. We denote it as:  $S_p(t) = \frac{\partial \xi_p}{\partial p}(t)$

The sensitivity matrix can be computed as the solution of a system of ODEs. Let  $\mathbf{s}_i = \frac{\partial \xi_p}{\partial p_i}(t)$  be the  $i^{\text{th}}$  column of  $S_p$ . If we apply the chain rule to its time derivative, we get:

$$\begin{cases} \dot{\mathbf{s}}_i(t) = \frac{\partial f}{\partial x}(t, x(t), p)\mathbf{s}_i(t) + \frac{\partial f}{\partial p_i}(t, x(t), p), \\ \mathbf{s}_i(0) = \frac{\partial x(0)}{\partial p_i}. \end{cases} \quad (3)$$

Here  $\frac{\partial f}{\partial x}(t, x(t), p)$  is the Jacobian matrix of  $f$  at time  $t$ . The equation above is thus an affine, time-varying ODE. In the core of our implementation, we compute  $\xi_p$  and the sensitivity matrix  $S_p$  using the CVODES numerical solver [27], which is designed to solve efficiently and accurately ODEs (like Eq. 1) and sensitivity equations (like Eq. 3).

### 3.3 Reachable Set Estimation Using Sensitivity

The reachability problem is the problem of computing the set of all the states visited by the trajectories starting from all the possible initial parameters in  $\mathcal{P}$  at a given time  $t$ .

**Definition 1 (Reachable Set).** *The reachable set induced by the set of parameters  $\mathcal{P}$  at time  $t$  is:*

$$\mathcal{R}_t(\mathcal{P}) = \bigcup_{p \in \mathcal{P}} \xi_p(t).$$

The set  $\mathcal{R}_t(\mathcal{P})$  can be approximated by using sensitivity analysis. Assume that for a given parameter  $p$  in  $\mathcal{P}$  we computed a trajectory  $\xi_p$  and the sensitivity matrix  $S_p$  associated with it. Given another parameter vector  $p'$  in  $\mathcal{P}$ , we can use this matrix to get an estimate  $\hat{\xi}_{p'}^p(t)$  of  $\xi_{p'}(t)$ . This is done by dropping higher order terms in the Taylor expansion given in Equation 2. We have:

$$\hat{\xi}_{p'}^p(t) = \xi_p(t) + S_p(t)(p' - p). \quad (4)$$

If we extend this estimation to all parameters  $p'$  in  $\mathcal{P}$ , we get the following estimate of the reachable set  $\mathcal{R}_t(\mathcal{P})$ :

$$\hat{\mathcal{R}}_t^p(\mathcal{P}) = \bigcup_{p' \in \mathcal{P}} \hat{\xi}_{p'}^p(t) = \{\xi_p(t) - S_p(t)p\} \oplus S_p(t)\mathcal{P}. \quad (5)$$

Thus  $\hat{\mathcal{R}}_t^p$  is an affine mapping of the initial set  $\mathcal{P}$  into  $\mathbb{R}^n$  (see Figure 1).

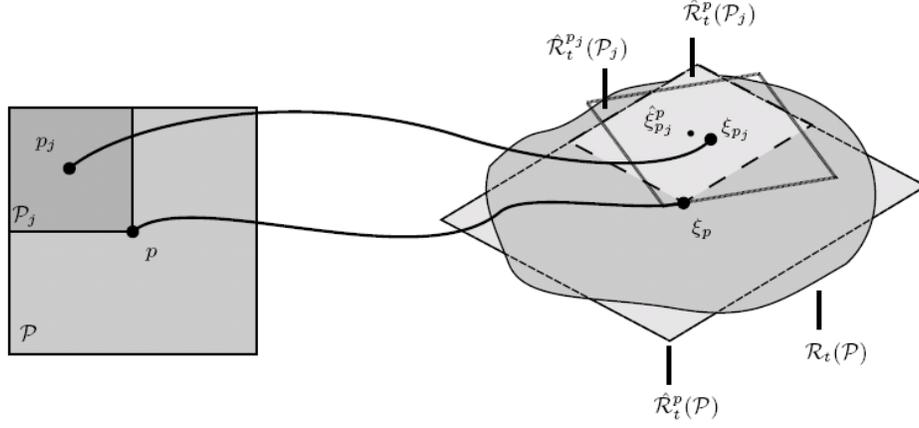
It can be shown that if the dynamics are affine, i.e., if  $f(t, x, p) = A(t, p)x + b(t, p)$ , then the estimation is exact. However, in the general case,  $\hat{\mathcal{R}}_t^p(\mathcal{P})$  is different from  $\mathcal{R}_t(\mathcal{P})$ . Since the estimation is based on a first order approximation around parameter  $p$ , it is local in the parameter space and its quality depends on how “big”  $\mathcal{P}$  is. In order to improve the estimation, we can partition  $\mathcal{P}$  into smaller subsets  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l$  and compute trajectories using new initial parameters  $p_1, p_2, \dots, p_l$  to get more precise local estimates. As a practical matter, we need to be able to estimate the benefit of such a refinement. To do so, we compare  $\hat{\mathcal{R}}_t^p(\mathcal{P}_j)$  — the estimate we get when using the “global” center,  $p$ ; to  $\hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)$  — the estimate we get when using the “local” center,  $p_j$ , and  $p'_i \in \mathcal{P}_j$ . We do this for each  $\mathcal{P}_j$ . Figure 1 illustrates the essential features of the algorithm.

**Proposition 1.** *We have*

$$d_H(\hat{\mathcal{R}}_t^p(\mathcal{P}_j), \hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)) \leq Err(\mathcal{P}, \mathcal{P}_j), \quad (6)$$

where

$$Err(\mathcal{P}, \mathcal{P}_j) = \|\xi_{p_j}(t) - \hat{\xi}_{p_j}^p(t)\| + \|S_{p_j}(t) - S_p(t)\| \|\mathcal{P}_j\|. \quad (7)$$



**Fig. 1.** Comparison between a “global” and a “local” estimate of the reachable set. The large square on the left hand side represent a region of parameter space,  $\mathcal{P}$ . The oval-shaped region on the right hand side corresponds to the true reachable set,  $\mathcal{R}_t(\mathcal{P})$ , induced by parameters  $\mathcal{P}$  at time  $t$ . The large parallelogram on the right hand side corresponds to the *estimated* reachable set,  $\hat{\mathcal{R}}_t^p(\mathcal{P})$ , using a sensitivity analysis based on trajectory labeled  $\xi_p$  which starts at point  $p \in \mathcal{P}$ . The point labeled  $\hat{\xi}_{p_j}^p$ , for example, is an estimate of where a trajectory starting at point  $p_j$  would reach at time  $t$ . If we partition  $\mathcal{P}$  and consider some particular partition,  $\mathcal{P}_j$ , we can then compare the estimated reachable sets  $\hat{\mathcal{R}}_t^p(\mathcal{P}_j)$  and  $\hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)$ , which correspond to the small light-gray and small dark gray parallelograms, respectively. We continue to refine until the distance between  $\hat{\mathcal{R}}_t^p(\mathcal{P}_j)$  and  $\hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)$  (Eq. 7) falls below some user-specified tolerance.

In other words, the difference between the global and the local estimate can be decomposed into the error introduced in the estimation  $\hat{\xi}_{p_j}^p(t)$  of the state reached at time  $t$  using  $p_j$  (first term on RHS of Eq. 7), and another term involving the difference between the local and the global sensitivity matrices and the distance from local center (second term on RHS of Eq. 7).

*Proof.* let  $y$  be in  $\hat{\mathcal{R}}_t^p(\mathcal{P}_j)$ . There exists  $p_y$  in  $\mathcal{P}_j$  such that  $y = \hat{\xi}_{p_y}^p(t)$ . We need to compare

$$\hat{\xi}_{p_y}^p(t) = \xi_p(t) + S_p(t)(p_y - p) \quad (8)$$

with

$$\hat{\xi}_{p_y}^{p_j}(t) = \xi_{p_j}(t) + S_{p_j}(t)(p_y - p_j). \quad (9)$$

By introducing

$$\hat{\xi}_{p_j}^p(t) = \xi_p(t) + S_p(t)(p_j - p) \quad (10)$$

and after some algebraic manipulations of (8), (9), and (10), we get

$$\begin{aligned} \hat{\xi}_{p_y}^p(t) - \hat{\xi}_{p_y}^{p_j}(t) &= \xi_{p_j}(t) - \hat{\xi}_{p_j}^p(t) + (S_{p_j}(t) - S_p(t))(p_y - p_j) \\ &\leq \|\xi_{p_j}(t) - \hat{\xi}_{p_j}^p(t)\| + \|S_{p_j}(t) - S_p(t)\| \|p_y - p_j\| = Err(\mathcal{P}, \mathcal{P}_j). \end{aligned} \quad (11)$$

Let  $x = \hat{\xi}_{p_y}^{p_j}(t)$  which is in  $\hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)$ , then it can be shown that  $\|y - x\| \leq Err(\mathcal{P}, \mathcal{P}_j)$  and so  $d(y, \hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)) \leq Err(\mathcal{P}, \mathcal{P}_j)$ . This is true for any  $y \in \hat{\mathcal{R}}_t^p(\mathcal{P}_j)$ , thus

$$\sup_{y \in \hat{\mathcal{R}}_t^p(\mathcal{P}_j)} d(y, \hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)) \leq Err(\mathcal{P}, \mathcal{P}_j).$$

Similarly, we can show that

$$\sup_{x \in \hat{\mathcal{R}}_t^{p_j}(\mathcal{P}_j)} d(x, \hat{\mathcal{R}}_t^p(\mathcal{P}_j)) \leq Err(\mathcal{P}, \mathcal{P}_j)$$

which proves the result.  $\square$

The quantity  $Err(\mathcal{P}, \mathcal{P}_j)$  can be easily computed from trajectories  $\xi_p$  and  $\xi_{p_j}$ , their corresponding sensitivity matrices, and  $\|\mathcal{P}_j\|$ . It has the following properties:

- If the dynamics is affine, then  $Err(\mathcal{P}, \mathcal{P}_j) = 0$ . Indeed, in this case, we have  $\hat{\xi}_{p_j}^p = \xi_{p_j}$  so the first term vanishes and  $S_p = S_{p_j}$  so the second term vanishes as well;
- If limit  $\|\mathcal{P}\|$  is 0, then limit  $Err(\mathcal{P}, \mathcal{P}_j)$  is also 0. Indeed, as  $\|\mathcal{P}\|$  decreases, so does  $\|p - p_j\|$ , and thus  $\|\xi_{p_j}(t) - \hat{\xi}_{p_j}^p(t)\|$  and  $\|\mathcal{P}_j\|$ , since  $\mathcal{P}_j$  is a subset of  $\mathcal{P}$ . We can show that the convergence is quadratic.

The computation of reachable sets at a given time  $t$  can be extended to time intervals. Assume that  $\mathcal{T}$  is a time interval of the form  $\mathcal{T} = [t_0, t_f]$ . The set reachable from  $\mathcal{P}$  during  $\mathcal{T}$  is  $\mathcal{R}_{\mathcal{T}}(\mathcal{P}) = \cup_{t \in \mathcal{T}} \mathcal{R}_t(\mathcal{P})$ . It can be approximated by simple interpolation between  $\mathcal{R}_{t_0}(\mathcal{P})$  and  $\mathcal{R}_{t_f}(\mathcal{P})$ . Of course, it may be necessary to subdivide  $\mathcal{T}$  into smaller intervals to improve the precision of the interpolation. A reasonable choice for this subdivision is to use the time steps taken by the numerical solver to compute the solution of the ODE and the sensitivity matrices.

### 3.4 Parameter Synthesis Algorithm

In this section, we state a parameter synthesis problem and propose an algorithm that provides an approximate solution. Let  $\mathcal{F}$  be a set of "bad" states. Our goal is to partition the set  $\mathcal{P}$  into safe and bad parameters. That is, we want to partition the parameters into those that induce trajectories that intersect  $\mathcal{F}$  during some time interval  $\mathcal{T}$ , and those that do not.

**Definition 2.** A solution of the parameter synthesis problem  $(Sys = (f, \mathcal{P}), \mathcal{F}, \mathcal{T})$  where  $\mathcal{F}$  is a set states and  $\mathcal{T}$  a subset of  $\mathbb{R}_{\geq 0}$ , is a partition  $\mathcal{P}_{bad} \cup \mathcal{P}_{saf}$  of  $\mathcal{P}$  such that for all  $p \in \mathcal{P}_{bad}$  (resp.  $p \in \mathcal{P}_{saf}$ ),  $\xi_p(t) \cap \mathcal{F} \neq \emptyset$  (resp.  $\xi_p(t) \cap \mathcal{F} = \emptyset$ ) for all  $t \in \mathcal{T}$ . An approximate solution is a partition  $\mathcal{P} = \mathcal{P}_{saf} \cup \mathcal{P}_{unc} \cup \mathcal{P}_{bad}$  where  $\mathcal{P}_{saf}$  and  $\mathcal{P}_{bad}$  are defined as before and  $\mathcal{P}_{unc}$  (i.e., uncertain) may contain both safe and bad parameters.

Exact solutions cannot be obtained in general, but we can try to compute an approximate solution with the uncertain subset being as small as possible. The idea is to iteratively refine  $\mathcal{P}$  and to classify the subsets into the three categories. A subset  $\mathcal{P}_j$  qualifies as safe (resp. bad) if:

1.  $\hat{\mathcal{R}}_{\mathcal{T}}^{p_j}(\mathcal{P}_j)$  is a reliable estimation of  $\mathcal{R}_{\mathcal{T}}(\mathcal{P}_j)$  based on the  $Err$  function;
2.  $\hat{\mathcal{R}}_{\mathcal{T}}^{p_j}(\mathcal{P}_j)$  does not (resp. does) intersect with  $\mathcal{F}$ .

To guarantee that the process ends, we need to ensure that each refinement introduces only subsets that are strictly smaller than the refined set.

**Definition 3 (Refining Partition).** A refining partition of a set  $\mathcal{P}$  is a finite set of sets  $\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l\}$  such that

- $\mathcal{P} = \bigcup_{j=1}^l \mathcal{P}_j$ ;
- There exists  $\gamma < 1$  such that  $\max_{j \in \{1, \dots, l\}} \|\mathcal{P}_j\| \leq \gamma \|\mathcal{P}\|$ .

Let  $\rho$  be a function that maps a set to one of its refining partitions. Our algorithm stops whenever the uncertain partition is empty, or it contains only subsets with a diameter smaller than some user-specified value,  $\delta p$ . The complete algorithm is given by Algorithm 1 below.

---

**Algorithm 1** Parameter Synthesis Algorithm
 

---

```

procedure SAFE( $\mathcal{P}, \mathcal{F}, t, \delta p, Tol$ )
   $\mathcal{P}_{\text{saf}} = \mathcal{P}_{\text{bad}} = \emptyset, \mathcal{P}_{\text{unc}} = \{\mathcal{P}\}$ 
  repeat
    Pick and remove  $\mathcal{Q}$  from  $\mathcal{P}_{\text{unc}}$  and let  $q \in \mathcal{Q}$ 
    for each  $(q_j, \mathcal{Q}_j) \in \rho(\mathcal{Q})$  do
      if  $Err(\mathcal{Q}, \mathcal{Q}_j) \leq Tol$  then
         $\triangleright$  Reach set estimation is reliable
         $\triangleright$  Reach set away from  $\mathcal{F}$ 
        if  $\hat{\mathcal{R}}_T^q(\mathcal{Q}_j) \cap \mathcal{F} = \emptyset$  then
           $\mathcal{P}_{\text{saf}} = \mathcal{P}_{\text{saf}} \cup \mathcal{Q}_j$ 
        else if  $\hat{\mathcal{R}}_T^q(\mathcal{Q}_j) \subset \mathcal{F}$  then
           $\triangleright$  Reach set inside  $\mathcal{F}$ 
           $\mathcal{P}_{\text{bad}} = \mathcal{P}_{\text{bad}} \cup \mathcal{Q}_j$ 
        else
           $\mathcal{P}_{\text{unc}} = \mathcal{P}_{\text{unc}} \cup \{(q_j, \mathcal{Q}_j)\}$ 
           $\triangleright$  Some intersection with the bad set
        end if
      end if
    else
       $\mathcal{P}_{\text{unc}} = \mathcal{P}_{\text{unc}} \cup \{(q_j, \mathcal{Q}_j)\}$ 
       $\triangleright$  Reach set estimation not enough precise
    end if
  end for
  until  $\mathcal{P}_{\text{unc}} = \emptyset$  or  $\max_{\mathcal{P}_j \in \mathcal{P}_{\text{unc}}} \|\mathcal{P}_j\| \leq \delta p$ 
  return  $\mathcal{P}_{\text{saf}}, \mathcal{P}_{\text{unc}}, \mathcal{P}_{\text{bad}}$ 
end procedure

```

---

The algorithm has been implemented within the Matlab toolbox Breach [12], which combines Matlab routines to manipulate partitions with the CVODES numerical solver, which can efficiently compute ODEs solutions with sensitivity matrices.

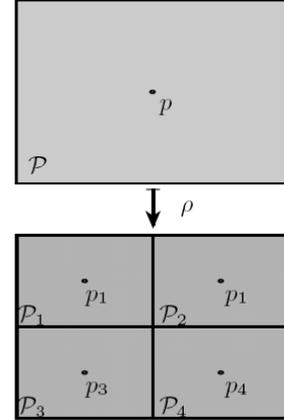
It uses rectangular partitions of the form

$$\mathcal{P}(p, \epsilon) = \{p' : p - \epsilon \leq p' \leq p + \epsilon\}$$

The refinement operator  $\rho$  is such that

$$\rho(\mathcal{P}(p, \epsilon)) = \{\mathcal{P}(p^1, \epsilon^1), \mathcal{P}(p^2, \epsilon^2), \dots, \mathcal{P}(p^l, \epsilon^l)\},$$

with  $\epsilon^k = \epsilon/2$  and  $p^k = p + (\pm \frac{\epsilon_1}{2}, \pm \frac{\epsilon_2}{2}, \dots, \pm \frac{\epsilon_n}{2})$ . This operation is illustrated in the Figure on the right.

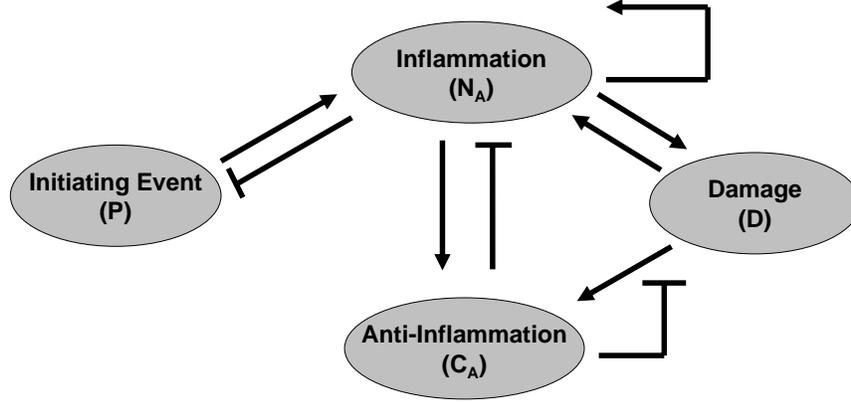


## 4 Application to Models of Acute Inflammation

We applied our method to two models of the acute inflammatory response to infection. The first is the 4-equation, 22-parameter model presented in [26], and the second is the 17-equation, 79-parameter model presented in [20]. The primary difference between these models is one of detail, and the first model can be thought of as a reduced dimensional version of the second.

The acute inflammatory response to infection has evolved to promote healing by ridding the organism of the pathogen. The actual response is a complex and carefully regulated combination of molecular and cellular cascades that exhibit both pro and anti-inflammatory behaviors. The pro-inflammatory elements are primarily responsible for eliminating the pathogen, but bacterial killing can cause collateral tissue damage.

Tissue damage, in turn, triggers an escalation in the pro-inflammatory response creating a positive feedback cycle (Figure 2). The anti-inflammatory elements counteract this cycle, thereby minimizing tissue damage and promoting healing. However, in cases of extreme infection, the delicate balance between pro and anti-inflammatory elements is destroyed, resulting in a potentially lethal amount of tissue damage.



**Fig. 2.** Cartoon representation of the 4-equation model of the acute immune response. Arrows represent up-regulation, bars represent down-regulation. Figure is adapted from Figure 1 in [26].

The 4-equation model is as follows:

$$\begin{aligned}\frac{dP}{dt} &= k_{pg}P\left(1 - \frac{P}{p_\infty}\right) - \frac{k_{pm}s_mP}{\mu_m + k_{mp}P} - k_{pm}f(N_A)P, \\ \frac{dN_A}{dt} &= \frac{s_{nr}R}{\mu_{nr} + R} - \mu_n N_A, \\ \frac{dD}{dt} &= k_{dn}f_s(f(N_A)) - \mu_d D, \\ \frac{dC_A}{dt} &= s_c + \frac{k_{cn}f(N_A + k_{cmd}D)}{1 + f(N_A + k_{cmd}D)} - \mu_c C_A,\end{aligned}$$

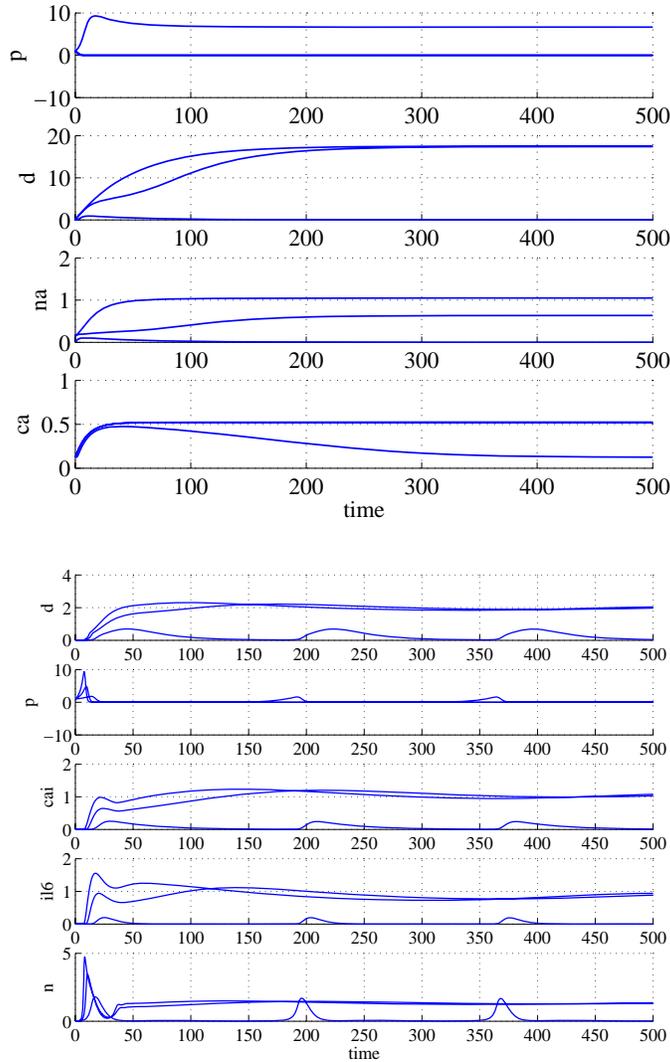
where

$$R = f(k_{nn}N_A + k_{np}P + k_{nd}D), \quad f(V) = \frac{V}{(1 + (C_A/c_\infty)^2)} \quad \text{and} \quad f_s(V) = \frac{V^6}{x_{dn}^6 + V^6}.$$

Here,  $k_*$ ,  $\mu_*$ ,  $s_*$ ,  $p_*$  are parameters, as defined in [26]. The state variables  $P$ ,  $N_A$ ,  $D$ , and  $C_A$ , correspond to the amounts of pathogen, pro-inflammatory mediators (e.g., activated neutrophils), tissue damage, and anti-inflammatory mediators (e.g., cortisol and interleukin-10), respectively. The 17-equation model, naturally, is far more detailed in terms of which mediators are modeled.

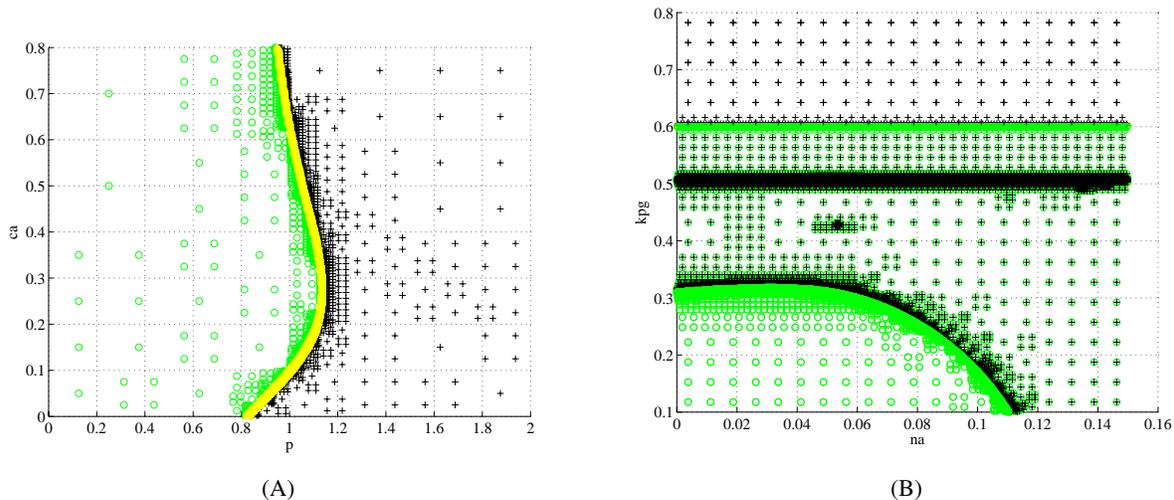
In each model, there are 3 clinically relevant outcomes: (i) a return to health, (ii) aseptic death, and (iii) septic death. Death is defined as a sustained amount of tissue damage ( $D$ ) above a specified threshold value and constitutes the undesirable or “bad” outcome we wish to avoid. Aseptic and septic death are

distinguished by whether the pathogen ( $P$ ) is cleared below a specified threshold value. Let  $\mathcal{F}_{alive}$  (resp.  $\mathcal{F}_{dead}$ ) refer to the set of states such that  $D$  is below (resp. above) some threshold  $D_{death}$ , and let  $\mathcal{F}_{septic}$  (resp.  $\mathcal{F}_{aseptic}$ ) refer to the set of states such that  $P$  is above (resp. below) some threshold  $P_{septic}$ . We can now define three sets of states corresponding to the three clinically relevant outcomes as follows: (i)  $Health = \mathcal{F}_{alive} \cap \mathcal{F}_{aseptic}$ ; (ii)  $Aseptic\ death = \mathcal{F}_{dead} \cap \mathcal{F}_{aseptic}$ ; and (iii)  $Septic\ death = \mathcal{F}_{dead} \cap \mathcal{F}_{septic}$ . In Figure 3 we present sample traces for both the 4 and 17 equation models.



**Fig. 3.** (Top) Examples trace from the 4-equation model. There are three different traces corresponding to septic death, aseptic death and health. (Bottom) Example traces from the 17-equation model; 5 of the 17 variables are shown. There are also three traces, illustrating the richer dynamics of the model. Two traces corresponds to aseptic death and the third to health with a periodic small resurgence of the pathogen. Time is measured in hours.

## 4.1 Experiments

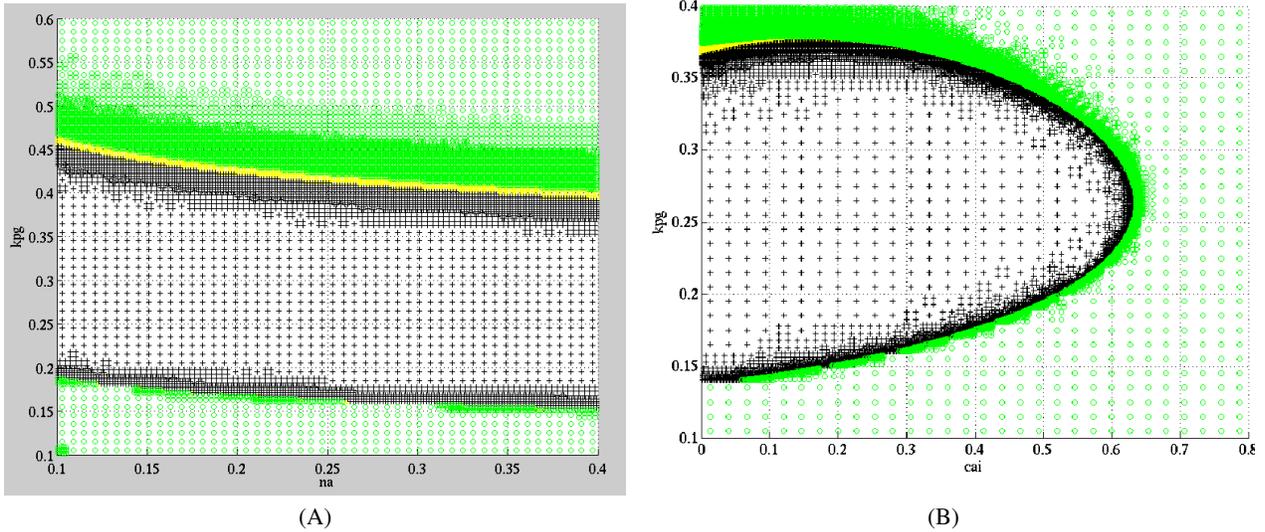


**Fig. 4.** Results obtained for the 4-equation model. Figure (A) reproduces results presented in Figure 8 of [26] with  $k_{pg} = 0.3$ , which was obtained using classical bifurcation analysis. Circles are parameter values leading to Health while crosses represent values leading to Death. Figure (B) illustrates how a pair of parameters ( $N_A$  and  $k_{pg}$ ) can be partitioned into the three possible outcomes. Circles alone lead to Health, crosses and circles lead to Aseptic Death and crosses alone lead to Septic Death. The separation between regions is induced from small uncertain regions computed by the algorithm.

We performed several experiments. In the first experiment, we validated our method by reproducing results previously obtained in [26] using bifurcation analysis. Figure 4-A contrasts the initial amount of pathogen,  $P_0$ , and the initial amount of anti-inflammatory mediators,  $C_{A0}$ . The growth rate of pathogen,  $k_{pg}$ , was set to 0.3 and other parameters to their nominal values. The region  $\mathcal{F}_{death}$  given by  $D \geq 5$  was used in our algorithm and we checked the intersection with reachable set at time 300 hours. Crosses correspond to initial values leading to death while circles lead to an healthy outcome. We can see that the resulting partition is quantitatively consistent with Figure 8 in [26]. In our second experiment, we varied growth rate of pathogen,  $k_{pg}$ , and  $N_A$ . Figure 4-(B) shows that there are three distinct regions in the  $k_{pg}$ - $N_A$  plane, corresponding to the three clinical outcomes.

We then performed several experimentations with the 17-equation model. Figures 5 (A) and (B) depict the  $k_{pg}$ - $N_A$  and  $k_{pg}$ - $C_{AI}$  planes, respectively.  $C_{AI}$  is a generic anti-inflammatory mediator. We partitioned the region using  $\mathcal{F}_{death} = D > 1.5$  and checked after time 300 hours. The 17-equation model exhibits an interesting behavior in the  $k_{pg}$ - $C_{AI}$  plane. Namely, that the separation between health and death is not monotone in the growth rate of the pathogen.

As previously mentioned, our algorithm is implemented in Matlab and uses the CVODES numerical solver. Figures 4 (A) and (B) were generated in a few seconds, and Figures 5 (A) and (B) were generated in about an hour on a standard laptop (Intel Dual Core 1.8GHz with 2 Gb of memory). We note that the algorithm could easily be parallelized.



**Fig. 5.** Results for the 17-equation model. Figure (A) illustrates the  $k_{pg}$ - $N_A$  plane, partitioned into regions leading to death (here aseptic death, represented by crosses) and regions leading to health (represented by circles). Figure (B) illustrates the  $k_{pg}$ - $C_{AI}$  plane. Interestingly, the separation is not monotone with the growth of pathogen  $k_{pg}$ .

## 5 Discussion and Conclusions

Complex models are increasingly being used to make predictions about complex phenomena in biology and medicine (e.g., [3, 25]). Such models can be potentially very useful in guiding early decisions regarding intervention, but it is often impossible to obtain accurate estimates for every parameter. Thus, it is important to have tools for explicitly examining a range of possible parameters to determine whether the behavior of the model is sensitive to those parameters that are poorly estimated. Performing this task for nonlinear models is especially challenging. We have presented an algorithm for solving the parameter synthesis problem for nonlinear dynamical models and applied it on two models of acute inflammation from the literature. The larger of the two has 17 equations and 79 parameters, demonstrating the scalability of our approach.

Our approach has several limitations. First, our refinement process implies that the number of partitions increases exponentially with the number of varying parameters. Thus, in practice, some variables must be held fixed while analyzing the behavior of the model. On the other hand, the number of state variables is not a limiting factor, as illustrated in our experiments on the 17-equation model. Second, our method relies on numerical simulations. Numerical methods are fundamentally limited in the context of verification since numerical image computation is not semi-decidable for nonlinear differential equations [23]. Moreover, there are no known methods capable of providing provable bounds on numerical errors for general nonlinear differential equations. Thus, we cannot claim to provide *formal* guarantees on the correctness of the results computed by our method. However *asymptotic* guarantees exist, meaning that results can always be improved by decreasing tolerance factors in the numerical computations. A nice feature of our approach is that it enables one to obtain qualitative results using a few simulations (e.g. a coarse partition between regions leading to qualitatively different behaviors). These qualitative results can then be made as precise as desired by focusing on smaller partitions.

There are several areas for future research. Our first order error control mechanism can be improved to make the refinements more efficient and more adaptive when nonlinear (i.e. higher order) behaviors dominate any linear dependence on parameter variations. We are also interested in developing techniques for verifying properties that are more complex than the reachability predicates we considered in this paper. For instance, temporal properties could easily be introduced in our framework. The extended system could then be used, for example, verify the possible outcomes associated with a particular medical intervention. Finally, we believe that the method could easily be used in the context of personalized medicine. In particular, given individual or longitudinal measurements from a specific patient, we could define a reachable set,  $\mathcal{F}_{obs}$ , that includes these observations (possibly convolved with a model of the measurement errors). We could then use our method to identify the set of parameters that are consistent with the observations. The refined parameters can then be used to make patient-specific predictions.

## Acknowledgments

This work is supported in part by US Department of Energy Career Award and a grant from Microsoft Research to CJL.

## References

1. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
2. R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. volume 88(7), pages 971–984. IEEE, 2000.
3. G. An. Agent based computer simulation and sirs: building a gap between basic science and clinical trials. *Shock*, 16:266273, 2001.
4. A. Annichini, E. Asarin, and A. Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In *CAV*, volume 1855 of *Lecture Notes in Computer Science*, pages 419–434. Springer, 2000.
5. E. Asarin, T. Dang, and A. Girard. Hybridization methods for verification of non-linear systems. In *ECC-CDC’05 joint conference: Conference on Decision and Control CDC and European Control Conference ECC*, 2005.
6. G. Batt, C. Belta, and R. Weiss. Model checking genetic regulatory networks with parameter uncertainty. In *HSCC*, volume 4416 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2007.
7. G. Batt, C. Belta, and R. Weiss. Model checking liveness properties of genetic regulatory networks. In *TACAS*, volume 4424 of *Lecture Notes in Computer Science*, pages 323–338. Springer, 2007.
8. G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Analysis and verification of qualitative models of genetic regulatory networks: A model-checking approach. In *IJCAI*, pages 370–375, 2005.
9. E. Cinquemani, A. Miliadis-Argeitis, and J. Lygeros. Identification of genetic regulatory networks: A stochastic hybrid approach. In *IFAC World Congress*, 2008.
10. E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
11. A. Donz, B. Krogh, and A. Rajhans. Parameter synthesis for hybrid systems with an application to simulink models. In *Proceedings of the 12th International Conference on Hybrid Systems: Computation and Control (HSCC’09)*, LNCS. Springer-Verlag, April 2009.
12. A. Donz. *Trajectory-Based Verification and Controller Synthesis for Continuous and Hybrid Systems*. PhD thesis, University Joseph Fourier, June 2007.
13. A. Donz and O. Maler. Systematic simulations using sensitivity analysis. In *HSCC’07*, LNCS, April 2007.
14. S. Drulhe, G. Ferrari-Trecate, H. de Jong, and A. Viari. Reconstruction of switching thresholds in piecewise-affine models of genetic regulatory networks. In *HSCC*, volume 3927 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2006.
15. G. E. Fainekos and G. J. Pappas. Robust sampling for mitl specifications. In *FORMATS*, Lecture Notes in Computer Science, pages 147–162. Springer, 2007.
16. E. Fargot and J.-L. Gouze. How to control a biological switch: a mathematical framework for the control of piecewise affine models of gene networks. Technical report, INRIA Sophia Antipolis, 2006.
17. G. Frehse, S. K. Jha, and B. H. Krogh. A counterexample-guided approach to parameter synthesis for linear hybrid automata. In *HSCC*, volume 4981, pages 187–200, 2008.

18. R. Ghosh and C. Tomlin. Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-notch signalling. *System Biology*, 2004.
19. T. A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond hytech: Hybrid systems analysis using interval numerical methods. In *HSCC*, volume 1790 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 2000.
20. R. Kumar. *The Dynamics of Acute Inflammation*. PhD thesis, University of Pittsburgh, 2004.
21. I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. In *HSCC*, pages 310–323, 2000.
22. I. M. Mitchell and C. J. Tomlin. Overapproximating reachable sets by hamilton-jacobi projections. *J. Symbolic Computation*, 19:1–3, 2002.
23. A. Platzer and E. M. Clarke. The image computation problem in hybrid systems model checking. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control, 10th International Conference, HSCC 2007, Pisa, Italy, Proceedings*, volume 4416 of *LNCS*, pages 473–486. Springer-Verlag, 2007.
24. A. Pnueli. The temporal logic of programs. In *Proc. 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57, 1977.
25. D. Polidori and J. Trimmer. Bringing advanced therapies to marker faster: a role for bio-simulation? *Diabetes Voice*, 48(2):28–30, 2003.
26. A. Reynolds, J. Rubin, G. Clermont, J. Day, Y. Vodovotz, and B. Ermentrout. A reduced mathematical model of the acute inflammatory response: I. derivation of model and analysis of anti-inflammation. *J Theor Biol*, 242(1):220–236, 2006.
27. R. Serban and A. C. Hindmarsh. Cvodes: the sensitivity-enabled ode solver in sundials. In *Proceedings of IDETC/CIE 2005*, Long Beach, CA., Sept. 2005.
28. O. Stursberg and B. H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In *HSCC*, pages 482–497, 2003.