

# Rigorous Model-Based Safety Analysis for Nonlinear Continuous-Time Systems

Youdong Lin<sup>1</sup> and Mark A. Stadtherr<sup>2</sup>

Department of Chemical and Biomolecular Engineering  
University of Notre Dame, Notre Dame, IN 46556, USA

July 22, 2008  
(revised, October 21, 2008)

<sup>1</sup>Current address: LINDO Systems, Inc., 1415 North Dayton St., Chicago, IL USA 60642

<sup>2</sup>Author to whom all correspondence should be addressed. Phone: (574) 631-9318; Fax: (574) 631-8366;  
E-mail: markst@nd.edu

## **Abstract**

A method is presented for the quantitative, model-based safety analysis of nonlinear continuous-time hybrid systems. This method uses the region-transition-model (RTM) framework of Huang et al. (2002), together with a recently developed technique (Lin & Stadtherr, 2007c) for the rigorous global analysis of nonlinear, continuous-time systems with uncertain initial conditions and/or parameters. Given an operating region described by bounds on possible initial conditions, inputs and model parameters, and a finite time horizon, the method can determine which operating subregions lead to safe operation. Numerical examples are presented that demonstrate the effectiveness of the method. This approach can supplement and complement the more qualitative techniques that are widely used for hazard identification and safety analysis.

**Keywords:** Safety; Hazard identification; Hybrid systems; Dynamic systems; Nonlinear systems; Interval analysis;

# 1 Introduction

In the design and operation of chemical processes, safety is clearly a critical concern. In the analysis of process safety, a key early step is hazard identification. In industrial practice, this is typically done using the qualitative, experience-based techniques of Process Hazards Analysis (PHA) (also called Preliminary Hazard Analysis). There is a wide range of methods for PHA (Venkatasubramanian & Preston, 1995; Crowl & Louvar, 2002), from simple “what if?” and checklist approaches to detailed HAZard and OPerability (HAZOP) studies. Of these, the use of HAZOP analysis is generally preferred in the chemical process industries. In this type of analysis, a multidisciplinary team of experts systematically examines the process Piping and Instrumentation Diagram (P&ID) to determine the potential hazards resulting from deviations from normal operation. HAZOP studies tend to be labor-, knowledge- and time-intensive. However, automated tools are available (e.g., Zhao et al., 2005a,b) to facilitate HAZOP analysis.

A recognized drawback of the qualitative techniques for hazard identification is that they do not well account for the potential complexity of the chemical process being analyzed, including the interactions of different effects. Thus, it may be difficult to determine whether specific hazards are physically realizable, and if so how they might occur. Because of such difficulties, methods for qualitative analysis are generally designed to produce very conservative results, perhaps leading to significant “overdesign.” For a more realistic understanding of possible hazards, a quantitative, model-based approach to safety analysis can supplement and complement the usual qualitative techniques.

Chemical processes can be modeled fundamentally as hybrid dynamic systems, characterized by a strong coupling between continuous state dynamics and discrete events (Barton & Pantelides, 1994). In the context of hazard identification, such a hybrid system involves both nonterminal safe states and terminal states which may be safe or unsafe. All unsafe states are considered terminal,

since once such a state is reached a hazard has been identified. In general, each nonterminal state is associated with a set of differential-algebraic equations and variables. Transitions between states are triggered when certain logical conditions are satisfied. Using such a state transition representation, two approaches have been used to examine the system of interest and identify potential hazards: reachability analysis and worst-case analysis. In reachability analysis, the problem is, given a set of possible initial conditions, to identify the states the system can reach over an infinite time horizon (Clarke et al., 1986; Moon et al., 1992; Park & Barton, 1997). Worst-case analysis is optimization based, and the problem is, given a finite time horizon and set of possible initial conditions and controls, to minimize the time required to reach an unsafe state (Dimitriadis et al., 1996, 1997; Srinivasan et al., 1997, 1998). For either of these strategies, a global analysis of the model is needed in order to guarantee the validity of the results, which in turn has restricted these approaches to linear systems and a limited number of chemical processes. To obtain a more realistic approach to safety analysis, Huang et al. (2002) have proposed a region-transition-model (RTM) framework for uncertain, nonlinear systems. While this can be formulated in terms of continuous-time systems, Huang et al. (2002) actually implemented the RTM approach using a discrete-time approximation; that is, they used difference equations to describe the evolution of the continuous states. However, as emphasized by Barton et al. (2006), for nonlinear systems the use of differential equations, as opposed to difference equations, often provides a more realistic physical model, which may be necessitated by the demands of many real world applications.

We present here an approach for quantitative, model-based safety analysis based on *continuous-time* hybrid systems. This method uses the RTM framework of Huang et al. (2002), together with a recently developed technique (Lin & Stadtherr, 2007c) for the rigorous global analysis of nonlinear, continuous-time systems with uncertain initial conditions and/or parameters. The goal is to solve the problem stated by Huang et al. (2002), “given a set of possible initial conditions and inputs

and a finite time horizon, identify the set of initial conditions and inputs which lead to unsafe behavior,” but to do so for continuous-time, not discrete-time, systems. The remainder of this article is organized as follows. In the next section, we present the mathematical formulation of the problem to be solved. This is followed by a section that provides background on tools used, namely interval analysis and Taylor models, and a section summarizing the method used to bound the state variables. Then a section is presented in which we outline the approach for rigorous, model-based safety analysis. Finally, we present the results of numerical examples: linear and nonlinear tank flow problems and a nonlinear batch reactor problem.

## 2 Problem Definition

It is assumed that the system of interest is described by a model with the following characteristics.

1. There is a finite number of normal (safe) states. There is one and only one active state at any time, except at times corresponding to events (state transitions). Associated with each normal state  $s_i$  is a set of ordinary differential equations (ODEs) describing the system,

$$\dot{\mathbf{x}} = \mathbf{f}^{(i)}(\mathbf{x}, \boldsymbol{\theta}), \quad (1)$$

where  $\mathbf{x}$  is the state vector (length  $m$ ) and  $\boldsymbol{\theta}$  is a time-invariant “parameter” vector (length  $p$ ) that includes uncertain model parameters, inputs, and disturbances. Models that are non-autonomous, or that involve parameters with known time dependence, can be easily converted into the form of Eq. (1) by the introduction of additional state variables. It is assumed that the model has a unique solution for any given set of initial conditions and parameter values. Initially the system is in state  $s_1$ .

2. There is a set of possible state transitions (discrete events). A transition from a source state

$s_i$  to a destination state  $s_r$  occurs when the logical condition

$$l_{ir}(\mathbf{x}(t), \boldsymbol{\theta}) \leq 0 \quad (2)$$

is satisfied. It is assumed that the transition is instantaneous.

3. For simplicity, it is assumed that all normal states are described by the same state vector  $\mathbf{x}$ , and that the value of  $\mathbf{x}$  does not change during a state transition. It should be emphasized that the problem can be formulated (Huang et al., 2002) and solved without use of this assumption. Initially, at  $t = 0$ ,  $\mathbf{x} = \mathbf{x}_0$ .
4. The initial state values are uncertain or are operating decisions, and are within a specified interval  $\mathbf{X}_0$ . That is,  $\mathbf{x}_0 \in \mathbf{X}_0$ . Similarly, the components of the parameter vector  $\boldsymbol{\theta}$  are uncertain or are operating or design decisions, and are within a specified interval  $\boldsymbol{\Theta}$ . That is,  $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ . The interval  $\mathbf{Z} = (\mathbf{X}_0, \boldsymbol{\Theta})^T$  then defines an “operating space” that encompasses uncertainties and the range of possible operating/design decisions for the process.
5. In order to use the technique of Lin & Stadtherr (2007c) for rigorous global analysis of nonlinear, continuous-time systems with uncertain initial conditions and/or parameters, it is assumed that  $\mathbf{f}^{(i)}$  is  $(k - 1)$ -times continuously differentiable with respect to  $\mathbf{x}$ , and  $(q + 1)$ -times continuously differentiable with respect to  $\mathbf{x}_0$  and  $\boldsymbol{\theta}$ , where  $k$  is the order of the truncation error in the interval Taylor series (ITS) method to be used in the integration procedure, and  $q$  is the order of Taylor model to be used to represent dependence on  $\mathbf{x}_0$  and  $\boldsymbol{\theta}$  (to be discussed in Sections 3.2 and 4).

The problem then is to identify the regions of the operating space  $\mathbf{Z} = (\mathbf{X}_0, \boldsymbol{\Theta})^T$  that lead to safe operation, and to do so with mathematical and computational certainty. If there are some safe states that correspond to successful operation (for example, in terms of a product specification), then the regions of the operating space leading to successful operation should also be identified.

The end result is to distinguish regions of  $\mathcal{Z}$  leading to safe and successful operation from those leading to unsafe or unsuccessful operation.

### 3 Background

The method described here uses interval analysis and Taylor models. As background, a brief introduction to these topics is provided. Additional background is available elsewhere.

#### 3.1 Interval analysis

A real interval  $X = [\underline{X}, \overline{X}]$  is defined by  $X = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}$ . We use an underline to indicate the lower bound of an interval and an overline to indicate the upper bound. A real interval vector  $\mathbf{X} = (X_1, X_2, \dots, X_n)^T \subset \mathbb{R}^n$  has  $n$  real interval components and can be interpreted geometrically as an  $n$ -dimensional rectangle or box. Unless noted otherwise, we use uppercase to indicate intervals and lowercase (or uppercase with underline or overline) to indicate real numbers. Arithmetic operations with intervals are defined by  $X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}$ , where  $\text{op} \in \{+, -, \times, \div\}$ . Interval versions of the elementary functions can be similarly defined. For dealing with exceptions, such as division by an interval containing zero, extended models for interval arithmetic are available, often based on the extended real system  $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$ . The concept of containment sets (csets) provides a valuable framework for constructing models for interval arithmetic with consistent handling of exceptions (Hansen & Walster, 2004; Pryce & Corliss, 2006). Implementations of interval arithmetic and elementary functions are readily available, and recent compilers from Sun Microsystems directly support interval arithmetic and an interval data type. Several good introductions to interval analysis, including interval arithmetic and other aspects of computing with intervals, are available (Jaulin et al., 2001; Hansen & Walster, 2004; Neumaier, 1990; Kearfott, 1996).

For an arbitrary function  $f(\mathbf{x})$ , the interval extension  $F(\mathbf{X})$  encloses the range of  $f(\mathbf{x})$  over  $\mathbf{X}$ . It is often computed by substituting  $\mathbf{X}$  into  $f(\mathbf{x})$  and then evaluating the function using interval arithmetic. This “natural” interval extension may be wider than the actual range of function values, though it always includes the actual range. This overestimation of the function range is due to the “dependency” problem, which may arise when a variable occurs more than once in a function expression. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not recognized when the natural interval extension is computed. Another source of overestimation that may arise in the use of interval methods is the “wrapping” effect. This occurs when an interval is used to enclose (wrap) a set of results that is not an interval. One approach that can be used to address both the dependency and wrapping problems is the use of Taylor models, as described in the next subsection.

### 3.2 Taylor models

Makino & Berz (1996) have described a remainder differential algebra (RDA) approach for bounding function ranges and controlling the dependency problem of interval arithmetic (Makino & Berz, 1999). This method is based on representing a function with a model consisting of a Taylor polynomial and an interval remainder bound. Based on a Taylor expansion about the point  $\mathbf{x}_0 \in \mathbf{X} \subset \mathbb{R}^n$ , the  $q$ -th order Taylor model  $T_f$  of  $f(\mathbf{x})$  consists of a  $q$ -th order polynomial function in  $(\mathbf{x} - \mathbf{x}_0)$ ,  $p_f$ , and an interval remainder bound  $R_f$ , such that  $f \in T_f = p_f + R_f$  for all  $\mathbf{x} \in \mathbf{X}$ . The function  $f$  is then bounded by seeking bounds on the Taylor model  $T_f$ , which is also denoted  $T_f = (p_f, R_f)$ .

Arithmetic operations with Taylor models can be done using the remainder differential algebra described by Makino & Berz (1996, 1999, 2003), which includes addition and multiplication, as well

as reciprocal and intrinsic functions. Using these, it is possible to start with simple functions such as the constant function  $f(x) = k$ , for which  $T_f = (k, [0, 0])$ , and the identity function  $f(x_i) = x_i$ , for which  $T_f = (x_{i0} + (x_i - x_{i0}), [0, 0])$ , and then to compute Taylor models for very complicated functions. Therefore, by using simple operator overloading with RDA operations, it is possible to compute a Taylor model for any function representable in a computer environment. It has been shown that, compared to other rigorous bounding methods, the Taylor model often yields sharper bounds for modest to complicated functional dependencies (Makino & Berz, 1996, 1999; Neumaier, 2003).

An interval bound on a Taylor model  $T = (p, R)$  over  $\mathbf{X}$  is denoted by  $B(T)$  and is given by  $B(T) = B(p) + R$ , where  $B(p)$  is an interval bound on the polynomial part  $p$ . The range bounding of the polynomial  $B(p) = P(\mathbf{X} - \mathbf{x}_0)$  is an important issue, which directly affects the performance of Taylor model methods. However, exact range bounding of an interval polynomial is NP hard, and direct evaluation using interval arithmetic is very inefficient, often yielding only loose bounds. Various bounding schemes (Makino & Berz, 2004, 2005; Neumaier, 2003) have been used, mostly focused on exact bounding of the first- and second-order polynomial terms. However, exact bounding of a general interval quadratic is also computationally expensive (in the worst case, exponential in the number of variables). Thus, in our implementation of Taylor models (Lin & Stadtherr, 2006, 2007a,c), we have used a compromise approach, in which only the first-order and the diagonal second-order terms are considered for exact bounding, and other terms are evaluated directly.

This bounding scheme for Taylor models can be exploited in performing *constraint propagation*. Information provided by a constraint can be used to eliminate incompatible values from the domain of its variables. This domain reduction can then be propagated to all constraints on that variable, where it may be used to further reduce the domains of other variables. This is the process known

as constraint propagation (Jaulin et al., 2001; Hansen & Walster, 2004). It is widely used in various forms (e.g., hull consistency) in connection with interval methods and Taylor models. We have shown previously how efficient constraint propagation schemes using Taylor models can be developed for inequality constraints (Lin & Stadtherr, 2006), bound constraints (Lin & Stadtherr, 2007b), and equality constraints (Lin et al., 2008). In the method for safety analysis described below, we use the procedure given by Lin & Stadtherr (2006) for constraint propagation with Taylor models on an inequality constraint  $c(\mathbf{x}) \leq 0$ , with  $\mathbf{x} \in \mathbf{X}$ . Using this procedure, regions of  $\mathbf{X}$  that are *guaranteed* not to satisfy the constraint can be identified and removed, thus yielding a region  $\mathbf{X}'$  in which it is possible to satisfy the constraint, and a region  $\mathbf{X} \setminus \mathbf{X}' = \{\mathbf{x} \in \mathbf{X} \mid \mathbf{x} \notin \mathbf{X}'\}$  in which it is impossible to satisfy the constraint.

## 4 Bounding the State Variables

In the context of safety analysis and hazard identification, in order to guarantee the reliability of the results obtained, we need a solver for nonlinear ODEs that can compute rigorous, verified bounds on the state variables for the case in which the initial values and parameters are given by intervals. Interval methods (also called validated methods or verified methods) for ODEs (Moore, 1966), provide a natural approach for computing the desired enclosure of the state variables.

Traditional interval methods (Moore, 1966) usually consist of two processes applied at each integration step. In the first process, existence and uniqueness of the solution are proved using the Picard-Lindelöf operator and the Banach fixed point theorem, and a rough enclosure of the solution is computed. In the second process, a tighter enclosure of the solution is computed. In general, both processes are realized by applying interval Taylor series (ITS) expansions with respect to time, and using automatic differentiation to obtain the Taylor coefficients. An excellent review of the traditional interval methods has been given by Nedialkov et al. (1999), and more recent

work has been reviewed by Neher et al. (2007). For addressing this problem, there are various packages available, including AWA (Lohner, 1992), VNODE (Nedialkov, 1999; Nedialkov et al., 2001) COSY VI (Berz & Makino, 1998) and ValEncIA-IVP (Rauh et al., 2006). In this study, we will use a new validating solver (Lin & Stadtherr, 2007c) for parametric ODEs, which is used to produce guaranteed bounds on the solutions of dynamic systems with interval-valued initial states and parameters, and which offers significant performance improvements over the popular VNODE package. The method makes use, in a novel way, of the Taylor model approach (Makino & Berz, 1996, 1999, 2003) to deal with the dependency and wrapping problems on the uncertain quantities (parameters and initial values). We will summarize here the basic ideas of the method used.

The ODE problem of interest is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}_0 \in \mathbf{X}_0, \quad \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad (3)$$

where  $t \in [t_0, t_N]$  for some  $t_N > t_0$ , and  $\mathbf{X}_0$  and  $\boldsymbol{\Theta}$  represent enclosures of initial values and parameters, respectively. It is desired to determine a verified enclosure of all possible solutions to this IVP. We denote by  $\mathbf{x}(t; t_j, \mathbf{X}_j, \boldsymbol{\Theta})$  the set of solutions  $\{\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \mid \mathbf{x}_j \in \mathbf{X}_j, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ , where  $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta})$  denotes a solution of  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$  for the initial condition  $\mathbf{x} = \mathbf{x}_j$  at  $t_j$ . We will summarize a method for determining enclosures  $\mathbf{X}_j$  of the state variables at each time step  $j = 1, \dots, N$ , such that  $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \boldsymbol{\Theta}) \subseteq \mathbf{X}_j$ .

Assume that at  $t_j$  we have an enclosure  $\mathbf{X}_j$  of  $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \boldsymbol{\Theta})$ , and that we want to carry out an integration step to compute the next enclosure  $\mathbf{X}_{j+1}$ . Then, in the first phase of the method, the goal is to find a step size  $h_j = t_{j+1} - t_j > 0$  and a rough enclosure  $\widetilde{\mathbf{X}}_j$  of the solution such that a unique solution  $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \in \widetilde{\mathbf{X}}_j$  is guaranteed to exist for all  $t \in [t_j, t_{j+1}]$ , all  $\mathbf{x}_j \in \mathbf{X}_j$ , and all  $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ . We apply a traditional interval method, with high order enclosure, to the parametric ODEs by using an interval Taylor series (ITS) with respect to time. That is, we determine  $h_j$  and

$\widetilde{\mathbf{X}}_j$  such that for  $\mathbf{X}_j \subseteq \widetilde{\mathbf{X}}_j^0$ ,

$$\widetilde{\mathbf{X}}_j = \sum_{i=0}^{k-1} [0, h_j]^i \mathbf{F}^{[i]}(\mathbf{X}_j, \Theta) + [0, h_j]^k \mathbf{F}^{[k]}(\widetilde{\mathbf{X}}_j^0, \Theta) \subseteq \widetilde{\mathbf{X}}_j^0. \quad (4)$$

Here  $k$  denotes the order of the Taylor series,  $\widetilde{\mathbf{X}}_j^0$  is an initial estimate of  $\widetilde{\mathbf{X}}_j$ , and the coefficients  $\mathbf{F}^{[i]}$  are interval extensions of the Taylor coefficients  $\mathbf{f}^{[i]}$  of  $\mathbf{x}(t)$  with respect to time, which can be obtained recursively in terms of  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \theta)$ . When Eq. (4) is satisfied, it demonstrates (Corliss & Rihm, 1996) that there exists a unique solution  $\mathbf{x}(t; t_j, \mathbf{x}_j, \theta) \in \widetilde{\mathbf{X}}_j$  for all  $t \in [t_j, t_{j+1}]$ , all  $\mathbf{x}_j \in \mathbf{X}_j$ , and all  $\theta \in \Theta$ .

In the second phase of the method, we compute a tighter enclosure  $\mathbf{X}_{j+1} \subseteq \widetilde{\mathbf{X}}_j$ , such that  $\mathbf{x}(t_{j+1}; t_0, \mathbf{X}_0, \Theta) \subseteq \mathbf{X}_{j+1}$ . This is done by using an ITS approach to compute  $\mathbf{T}_{\mathbf{x}_{j+1}}(\mathbf{x}_0, \theta)$ , a Taylor model of  $\mathbf{x}_{j+1}$  in terms of the initial values  $\mathbf{x}_0$  and parameters  $\theta$ , and then obtaining the enclosure  $\mathbf{X}_{j+1} = B(\mathbf{T}_{\mathbf{x}_{j+1}})$ . For the Taylor model computations, we begin by representing the interval initial states and parameters by the Taylor models (identity functions)  $\mathbf{T}_{\mathbf{x}_0}$  and  $\mathbf{T}_\theta$ , respectively. Then, we can determine Taylor models  $\mathbf{T}_{\mathbf{f}^{[i]}}$  of the interval Taylor series coefficients  $\mathbf{f}^{[i]}(\mathbf{x}_j, \theta)$  by using RDA operations to compute  $\mathbf{T}_{\mathbf{f}^{[i]}} = \mathbf{f}^{[i]}(\mathbf{T}_{\mathbf{x}_j}, \mathbf{T}_\theta)$ . Using an interval Taylor series for  $\mathbf{x}_{j+1}$  with coefficients given by  $\mathbf{T}_{\mathbf{f}^{[i]}}$ , and using the mean value theorem, one can obtain  $\mathbf{T}_{\mathbf{x}_{j+1}}(\mathbf{x}_0, \theta)$ , the desired Taylor model of  $\mathbf{x}_{j+1}$  in terms of the parameters  $\theta$  and initial states  $\mathbf{x}_0$ . To control the wrapping effect, the state enclosures are propagated using a new type of Taylor model consisting of a polynomial and a *parallelepiped* (as opposed to an interval) remainder bound. Complete details of the computation of  $\mathbf{T}_{\mathbf{x}_{j+1}}$  are given by Lin & Stadtherr (2007c). An implementation of this approach, called VSPODE (Verifying Solver for Parametric ODEs), has been developed and tested by Lin & Stadtherr (2007c), who compared its performance with results obtained using the popular VNODE package (Nedialkov, 1999; Nedialkov et al., 2001). For the test problems used, VSPODE provided tighter enclosures on the state variables than VNODE, and required significantly less computation time. Information about the availability of VSPODE can

be obtained by contacting the authors.

## 5 Safety Analysis Method

Consider again the safety analysis problem described in Section 2. The core problem is to determine regions of the operating space in which state transitions occur. A transition from a state  $s_i$  to another state  $s_r$  occurs when  $l_{ir}(\mathbf{x}(t), \boldsymbol{\theta}) \leq 0$ . Thus, if bounds on  $l_{ir}(\mathbf{x}(t), \boldsymbol{\theta})$  can be determined, it is possible to check for the possibility of transitions. To do this we need bounds on the state variables  $\mathbf{x}(t)$ , which can be obtained using VSPODE. The bounds  $\mathbf{X}_j$ ,  $j = 1, \dots, N$ , computed by VSPODE are valid at the corresponding times  $t_j$ ; that is, at the endpoints of the integration steps. However, transitions may also occur during an integration step. Thus, bounds that are valid over entire integration steps are needed. These can be obtained from the coarse enclosures  $\widetilde{\mathbf{X}}_j$ ,  $j = 1, \dots, N$ , which bound the state variables for all  $t \in [t_j, t_{j+1}]$ , all  $\mathbf{x}_j \in \mathbf{X}_j$ , and all  $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ , and are thus critical in determining the occurrence of system transition during a period of time.

Since overestimation of  $\widetilde{\mathbf{X}}_j$  can lead to poor performance in the method described here, we first describe a simple technique to tighten these bounds. This is based on the idea that over a small time step, it is usually possible to show that most or all of the component functions of  $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \dot{\mathbf{x}}$  are bounded either above zero or below zero, making most or all of the component trajectories of  $\mathbf{x}(t)$  monotonic. If this is true, then the tighter endpoint enclosures  $\mathbf{X}_j$  and  $\mathbf{X}_{j+1}$  can be used to determine bounds over the entire time step. This is shown schematically in Fig. 1, for a trajectory  $x(t)$  that has been shown to be monotonically increasing based on the initial coarse enclosure  $\widetilde{X}_j$ . An improved  $\widetilde{X}_j$  can be obtained by taking the interval hull of  $X_j$  and  $X_{j+1}$ . Note that the curves connecting  $X_j$  and  $X_{j+1}$  cannot go outside the bounds given by the improved  $\widetilde{X}_j$  without violating the known monotonicity of  $x(t)$ .

In general, the procedure used for improvement of the course enclosure  $\widetilde{\mathbf{X}}_j$  is:

1. For time step  $j + 1$ , with  $\mathbf{X}_j$  known, use VSPODE to compute  $\widetilde{\mathbf{X}}_j$  and then  $\mathbf{X}_{j+1}$ .
2. For each variable  $i = 1, \dots, m$ ,
  - (a) Compute  $F_i = F_i(\widetilde{\mathbf{X}}_j, \Theta)$  using interval arithmetic.
  - (b) If  $\underline{F}_i \geq 0$  ( $x_i$  monotonically increases with respect to time) or  $\overline{F}_i \leq 0$  ( $x_i$  monotonically decreases with respect to time), then  $\widetilde{X}_{i,j} = \square\{X_{i,j}, X_{i,j+1}\}$ , where  $\square$  indicates the interval hull.

To test a region  $\mathbf{Z}$  for transition from state  $s_i$  to  $s_r$ , integration with VSPODE is combined with evaluations of  $l_{ir}$ . If there are multiple possible destination states  $s_r$ , then different parts of  $\mathbf{Z}$  may transition to different destination states. However, during an integration step, it is assumed that only one transition may occur in any particular part of  $\mathbf{Z}$ . At integration step  $j + 1$ , and for all possible destination states  $s_r$ ,

1. Compute the interval  $L_{ir} = l_{ir}(\widetilde{\mathbf{X}}_j, \Theta)$  for  $t \in [t_j, t_{j+1}]$  using interval arithmetic, where  $\widetilde{\mathbf{X}}_j$  is the improved coarse enclosure.
  - (a) If  $\underline{L}_{ir} > 0$ , then a transition to  $s_r$  does not occur for any point in  $\mathbf{Z}$ , for all  $t \in [t_j, t_{j+1}]$ . Mark the region as FALSE and proceed to consider the next possible destination state.
  - (b) If  $\overline{L}_{ir} \leq 0$ , then the transition to  $s_r$  does occur at all points in  $\mathbf{Z}$ , for all  $t \in [t_j, t_{j+1}]$ . Mark the region as TRUE and move it to the working list  $\mathcal{L}_r$  for state  $s_r$ . Stop integration and set  $\mathbf{Z} = \emptyset$ .
  - (c) Otherwise, the transition to  $s_r$  may or may not occur for all points in  $\mathbf{Z}$ , and the region is UNDECIDED.

For a region to be marked TRUE, we need to show that the transition to  $s_r$  occurs at all points in  $\mathbf{Z}$ , but not necessarily for all times  $t \in [t_j, t_{j+1}]$ . In fact, it is only necessary to show that a transition to  $s_r$  may occur at one point in time. Therefore, in the next step, to continue testing an UNDECIDED region, we only consider the time  $t = t_{j+1}$ , since a Taylor model state enclosure  $\mathbf{T}_{\mathbf{x}_{j+1}}$  is available there.

2. To bound  $l_{ir}$ , compute  $T_{l_{ir}} = l_{ir}(\mathbf{T}_{\mathbf{x}_{j+1}}, \mathbf{T}_{\theta})$  for  $t = t_{j+1}$  using Taylor model operations.
  - (a) If  $\overline{B(T_{l_{ir}})} \leq 0$ , then the transition to  $s_r$  will occur at all points in  $\mathbf{Z}$  at  $t = t_{j+1}$ . Mark the region as TRUE and move it to the working list  $\mathcal{L}_r$  for state  $s_r$ . Stop integration and set  $\mathbf{Z} = \emptyset$ .
  - (b) If  $\underline{B(T_{l_{ir}})} > 0$ , the transition to  $s_r$  does not occur for any point in  $\mathbf{Z}$ , but this has been shown only for  $t = t_{j+1}$ . Thus, keep the mark as UNDECIDED.
  - (c) Otherwise, perform constraint propagation (see Section 3.2) using  $T_{l_{ir}}$  with the constraint  $l_{ir} \geq 0$  to reduce the region  $\mathbf{Z}$  to  $\mathbf{Z}'$ .
    - i. For all points in the region  $\mathbf{Z} \setminus \mathbf{Z}'$ , it is guaranteed that  $l_{ir} < 0$  at  $t = t_{j+1}$ . Thus, the transition to  $s_r$  will occur at all points in  $\mathbf{Z} \setminus \mathbf{Z}'$  at  $t = t_{j+1}$ . Mark this region as TRUE and move it to the working list  $\mathcal{L}_r$  for state  $s_r$ .
    - ii. Reset  $\mathbf{Z} = \mathbf{Z}'$ . Mark the remaining region  $\mathbf{Z}$  as UNDECIDED and proceed to consider the next possible destination state.

This procedure for testing a region  $\mathbf{Z}$  for transition from state  $s_i$  to  $s_r$  at time step  $j + 1$  is summarized in Fig. 2. After all possible destination states have been considered, the possible outcomes for the region  $\mathbf{Z}$  that was tested are: 1) The entire region is marked FALSE; integration proceeds with the next time step. 2) The entire region is marked TRUE; integration in state  $s_i$  stops. 3) The entire region is marked UNDECIDED; integration proceeds with the next time step.

4) The region is subdivided into subregions that are TRUE for one of the possible destination states, and a subregion that remains UNDECIDED; integration proceeds with the next time step in the UNDECIDED region. Integration proceeds until a specified time horizon is reached, or until  $\mathbf{Z} = \emptyset$  (meaning the entire region has been marked TRUE).

To deal with regions that remain UNDECIDED, and that are larger than some desired tolerance, a subdivision strategy is needed. This is incorporated into the overall algorithm that follows.

1. Initialize. Consider the initial state  $s_1$  and initial operating region  $\mathbf{Z}^{(0)}$ , with a specified time horizon. Set the region size tolerance vector  $\epsilon$ . The components of  $\epsilon$  provide tolerances for the corresponding components of the operating region. Set the working list for state  $s_1$  as  $\mathcal{L}_1 = \{\mathbf{Z}^{(0)}\}$ . Set the working lists for all other states to be empty, that is,  $\mathcal{L}_{i \neq 1} = \emptyset$ . Set the results lists for all states to be empty, that is,  $\mathcal{R}_i = \emptyset$ .
2. Iterate, beginning with  $\mathcal{L}_1$ , over all working lists  $\mathcal{L}_i$ . Iterate over all subregions  $\mathbf{Z}^{(k)}$  stored in  $\mathcal{L}_i$ .
  - (a) Integrate using VSPODE, testing for transitions, as described above. Note that during integration any subregions of  $\mathbf{Z}^{(k)}$  that are marked TRUE are excluded from  $\mathbf{Z}^{(k)}$  and stored in the working list for the corresponding destination state. Integration proceeds until the specified time horizon is reached, or until  $\mathbf{Z}^{(k)} = \emptyset$  (meaning the entire region has been marked TRUE).
  - (b) If  $\mathbf{Z}^{(k)} = \emptyset$ , go to (f).
  - (c) If the time horizon was reached during integration, check for final-time transitions. If any is TRUE, move  $\mathbf{Z}^{(k)}$  to the results list for the corresponding state. Go to (f).
  - (d) If  $\mathbf{Z}^{(k)}$  is marked FALSE, put it in the results list  $\mathcal{R}_i$  for state  $s_i$ . Go to (f).
  - (e) If  $\mathbf{Z}^{(k)}$  is marked as UNDECIDED

- i. If the width all components of  $\mathbf{Z}^{(k)}$  is smaller than the corresponding components in the tolerance vector  $\boldsymbol{\epsilon}$ , store it in the UNDECIDED results list  $\mathcal{R}_U$ , and go to (f).
- Note that the width of an interval vector is the maximum of the component widths.
- ii. Else if  $\mathbf{Z}^{(k)}$  has been sufficiently reduced (30% reduction in volume), return to (a).
  - iii. Else, bisect  $\mathbf{Z}^{(k)}$  and put the resulting subregions at the front of  $\mathcal{L}_i$  and go to (f).
- (f) Go to the next subregion in  $\mathcal{L}_i$ . If  $\mathcal{L}_i = \emptyset$ , process another nonempty working list.
3. Terminate. At termination all working lists  $\mathcal{L}_i$  will be empty. The initial operating space  $\mathbf{Z}^{(0)}$  will now be divided into non-overlapping subregions which have been stored in results lists corresponding to the eventual states achieved, or stored in the UNDECIDED list if no determination could be made.

## 6 Examples

In this section, we present the results of numerical experiments on three example problems to illustrate the theoretical and computational aspects of the proposed approach for safety analysis. The first and third example problems were taken from Huang et al. (2002), who used discrete-time models to do the analysis. Since the models used here are continuous-time, and thus not the same as those used by Huang et al. (2002), no direct comparisons are made to their results. The first problem involves a linear model, and the last two involve nonlinear models. All example problems were solved on an Intel Pentium 4 3.2GHz workstation running Red Hat Linux. The VSPODE package (Lin & Stadtherr, 2007c), with a  $k = 17$  order interval Taylor series,  $q = 5$  order Taylor model, QR approach for wrapping, and automatic variable step size (unless otherwise noted) was used to integrate the continuous dynamic systems.

## 6.1 Linear tank flow

This simple problem involves flow into and out of a tank, and considers the possibility of underflow and overflow, shown in Fig. 3 (Huang et al., 2002). The tank system has three states: state  $s_1$ , normal operation (safe); state  $s_2$ , underflow (unsafe); and state  $s_3$ , overflow (unsafe). The normal operation state is governed by the ODE

$$\frac{dV}{dt} = F_{\text{in}} - \alpha V, \quad t \in [0, t_{\text{H}}], \quad V(0) = V_0, \quad (5)$$

where  $V$  is the (real-valued) fluid volume in the tank,  $F_{\text{in}}$  is the (real-valued) time-invariant inlet flow rate,  $\alpha$  is a time-invariant constant, and  $t_{\text{H}}$  is the time horizon. The unsafe states (underflow and overflow) are considered terminal states, therefore there is no need to provide models for these two states. The possible transitions at any given time are: 1) Transition from normal state  $s_1$  to underflow state  $s_2$  when  $V \leq V_{\text{min}}$ . Thus, this transition occurs when  $l_{12} = V - V_{\text{min}} \leq 0$ . 2) Transition from normal state  $s_1$  to overflow state  $s_3$  when  $V \geq V_{\text{max}}$ . Thus, this transition occurs when  $l_{13} = V_{\text{max}} - V \leq 0$ .

For the safety analysis of this system, we consider the operating region defined by the initial tank level  $V_0 \in [2, 3] \text{ m}^3$  and the inlet flow rate  $F_{\text{in}} \in [0, 1] \text{ m}^3/\text{s}$ , with a constant  $\alpha = 0.15 \text{ s}^{-1}$ . A region size tolerance of  $\epsilon = 0.01$  is used for both  $V_0$  and  $F_{\text{in}}$ . Applying the algorithm described above, for a time horizon of  $t_{\text{H}} = 10 \text{ s}$ , gives the results shown in Fig. 4. Here it can be seen that the  $(V_0, F_{\text{in}})$  operating space has been split into regions corresponding to safe and unsafe operation (underflow or overflow), with small undecidable regions separating the safe and unsafe regions. The calculations required 0.62 s of CPU time and involved 1125 subregion tests. The undecidable regions account for about 1.37% of the total operating space tested. Repeating the analysis for a time horizon of  $t_{\text{H}} = 100 \text{ s}$  with constant step size  $h = 2 \text{ s}$  yields the results given in Fig. 5. This was obtained in 21.7 s of CPU time and 862 subregion tests. The undecidable regions account

for about 1.29% of the overall operating space tested. These results correspond to the theoretical results for an infinite time horizon, which can be derived analytically for this simple system (Huang et al., 2002).

A similar analysis can also be done for the case of uncertainty in the model parameter  $\alpha$ , which we now assume is known only within  $\pm 5\%$  of its nominal value, that is  $\alpha \in A = [0.142, 0.158]$ . A region size tolerance of  $\epsilon = 0.001$  is set for  $\alpha$ . The results of the safety analysis for  $t_H = 10$  s are shown in Fig. 6, which was obtained in 24.7 s of CPU time and 28987 subregion tests. Because of the uncertainty in  $\alpha$ , the undecidable regions now grow to about 9.85% of the  $(V_0, F_{in})$  space tested. Note that the complete operating space is now three dimensional, with  $\alpha$  as the third dimension, and that Fig. 6 is a projection into the  $(V_0, F_{in})$  space. Regions in Fig. 6 are marked as safe, overflow or underflow only when this is true for *all* values  $\alpha \in A$  in the  $\alpha$  dimension. Fig. 7 shows the results obtained for the longer time horizon  $t_H = 100$  s, with constant step size  $h = 2$  s. For this case, the computation time was 617 s of CPU time with 31139 subregion tests. The undecidable regions account for about 10.80% of the  $(V_0, F_{in})$  space tested.

## 6.2 Nonlinear tank flow

The algorithm described here is also applicable to nonlinear systems. In this example, we consider again the tank flow problem described above, but now with a nonlinear relationship for flow from the tank. For this problem, the normal operating (safe) state is described by

$$\frac{dV}{dt} = F_{in} - \alpha\sqrt{V}, \quad t \in [0, t_H], \quad V(0) = V_0, \quad (6)$$

For the safety analysis of this system, we again use the operating region defined by  $V_0 \in [2, 3]$  m<sup>3</sup> and  $F_{in} \in [0, 1]$  m<sup>3</sup>/s, with a constant  $\alpha = 0.15$  m<sup>3/2</sup>/s and region size tolerances of  $\epsilon = 0.01$  for both  $V_0$  and  $F_{in}$ . Results of the safety analysis for this case, with a time horizon of  $t_H = 10$  s, are given in Fig. 8, which was obtained with 10.1 s of CPU time and 777 subregion tests. The

undecidable regions account for about 1.35% of the total operating space tested. For  $t_H = 100$  s, the results are shown in Fig. 9. This required 28.0 s of CPU time and 850 subregion tests. The undecidable regions account for about 1.52% of the total operating space.

The case of an uncertain  $\alpha$  is also considered again, with  $\alpha \in A = [0.142, 0.158]$ , and region size tolerance  $\epsilon = 0.001$ . Fig. 10 shows the results for  $t_H = 10$  s, which required 396 s of CPU time and 18745 subregion tests, and Fig. 11 shows the results for  $t_H = 100$  s, which required 1259 seconds of CPU time and 27297 subregion tests. As described above, this is a projection in the  $(V_0, F_{in})$  space. The undecidable regions account for about 5.98% of the  $(V_0, F_{in})$  space tested in the  $t_H = 10$  s case, and 6.19% for  $t_H = 100$  s. This example demonstrates that it is very easy to apply the algorithm described above to nonlinear systems. In the next example, a model with more complicated nonlinearities is considered.

### 6.3 Exothermic batch reactor

In this example, the process is a first-order exothermic reaction  $A \rightarrow B$  in a batch reactor fitted with a segmented, variable-area cooling jacket. The process model is given by the following nonlinear ODE system (mass and energy balance)

$$\begin{aligned} \frac{dX}{dt} &= k_0 \exp\left(-\frac{E_a}{RT}\right) (1 - X) \\ \frac{dT}{dt} &= \frac{UA}{C_{A0}VC_p}(T_a - T) - \frac{\Delta H_R k_0}{C_p} \exp\left(-\frac{E_a}{RT}\right) (1 - X), \end{aligned} \quad (7)$$

where  $X$  is the (real-valued) conversion and  $T$  is the (real-valued) reactor temperature. Other symbols and their nominal values are shown in Table 1. The coolant temperature  $T_a$  and heat transfer constant  $UA$  are adjustable.

As shown in Fig. 12 (Huang et al., 2002), the system has four possible states, with states  $s_2$ ,  $s_3$  and  $s_4$  being terminal states:

State  $s_1$ : Normal (safe) operation.

State  $s_2$ : Overheated (unsafe) operation; this occurs when the reactor temperature exceeds 540 K. Thus, the transition from  $s_1$  to  $s_2$  occurs when  $l_{12} = 540 - T < 0$ .

State  $s_3$ : Successful (and safe) run completion; this occurs at the final time of  $t_H = 1500$  s when the conversion  $X$  is greater than or equal to 97.5%. This is a final-time transition from  $s_1$  to  $s_3$  that occurs when  $l_{13} = 0.975 - X \leq 0$ .

State  $s_4$ : Off-spec (but safe) run; this occurs at the final time of  $t_H = 1500$  s when the conversion  $X$  is less than 97.5%. This is a final-time transition from  $s_1$  to  $s_4$  that occurs when  $l_{14} = X - 0.975 < 0$ .

We first consider the case of an operating region given by initial temperature  $T_0 \in [310, 540]$  K and coolant temperature  $T_a \in [290, 310]$  K, with the heat transfer constant at its nominal value of  $UA = 3$  W/K. A region size tolerance of  $\epsilon = 1$  is used for both  $T_0$  and  $T_a$ . The results of the safety analysis obtained using the procedure described above are shown in Fig. 13. The computation required 1886.9 s of CPU time and 1299 subregion tests. The undecidable regions account for about 1.66% of the overall  $(T_0, T_a)$  operating region.

As a second case, we consider the operating region defined by initial temperature  $T_0 \in [310, 540]$  K and heat transfer constant  $UA \in [0, 6]$  W/K, with the coolant temperature at its nominal value of  $T_a = 298$  K. The region size tolerance for  $T_0$  is set to 1, and for  $UA$  to 0.05. The safety analysis results are shown in Fig. 14, which was determined using 4487.58 seconds of CPU time and 2969 subregion tests. Here the undecidable regions account for about 0.96% of the total  $(T_0, UA)$  operating region.

As a final case, we consider the three-dimensional operating region given by initial temperature  $T_0 \in [310, 540]$  K, coolant temperature  $T_a \in [290, 310]$  K, and heat transfer constant  $UA \in [2.5, 3.5]$  W/K. The region size tolerances for  $T_0$  and  $T_a$  are set to 1, and to 0.05 for  $UA$ . The full three-

dimensional results are shown in Fig. 15. This computation required 98649 seconds of CPU time and 48440 subregion tests. The undecidable regions account for about 4.80% of the overall  $(T_0, T_a, UA)$  operating region. In Fig. 15, the subregions that appear in each of the results lists  $\mathcal{R}_2$  (overheat),  $\mathcal{R}_3$  (successful run),  $\mathcal{R}_4$  (insufficient conversion), and  $\mathcal{R}_U$  (undecided) are also depicted (delineated by white lines).

## 7 Concluding Remarks

We have demonstrated here a strategy for quantitative, model-based safety analysis for nonlinear, continuous-time hybrid systems. This method uses the region-transition-model (RTM) framework of Huang et al. (2002), together with a recently developed technique (Lin & Stadherr, 2007c) for the rigorous global analysis of nonlinear, continuous-time systems with uncertain initial conditions and/or parameters. Given an operating region described by bounds on possible initial conditions, inputs and model parameters, and a finite time horizon, the method can determine which operating subregions lead to safe operation. This approach can supplement and complement the more qualitative techniques that are widely used for hazard identification and safety analysis.

## Acknowledgments

This work was supported in part by the U. S. Department of Energy under Grant DE-FG02-05CH11294.

## References

- Barton, P. & Pantelides, C. (1994). Modeling of combined discrete-continuous processes. *AIChE Journal*, 40, 966–979.
- Barton, P. I., Lee, C. K., & Yunt, M. (2006). Optimization of hybrid systems. *Computers and Chemical Engineering*, 30, 1576–1589.
- Berz, M. & Makino, K. (1998). Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4, 361–369.
- Clarke, E., Emerson, E., & Sistla, A. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8, 244–263.
- Corliss, G. F. & Rihm, R. (1996). Validating an a priori enclosure using high-order Taylor series. In G. Alefeld & A. Frommer, eds., *Scientific Computing: Computer Arithmetic, and Validated Numerics*, pp. 228–238, Akademie Verlag, Berlin.
- Crowl, D. A. & Louvar, J. F. (2002). *Chemical Process Safety: Fundamentals with Applications*. Upper Saddle River, NJ: Prentice Hall.
- Dimitriadis, V., Hackenberg, J., Shah, N., & Pantelides, C. (1996). A case study in hybrid process safety verification. *Computers and Chemical Engineering*, 20, S503–S508.
- Dimitriadis, V. D., Shah, N., & Pantelides, C. C. (1997). Modeling and safety verification of discrete/continuous processing systems. *AIChE Journal*, 43, 1041–1059.
- Hansen, E. & Walster, G. W. (2004). *Global Optimization Using Interval Analysis*. New York: Marcel Dekker.

- Huang, H., Adjiman, C. S., & Shah, N. (2002). Quantitative framework for reliable safety analysis. *AIChE Journal*, 48, 78–96.
- Jaulin, L., Kieffer, M., Didrit, O., & É Walter (2001). *Applied Interval Analysis*. London: Springer-Verlag.
- Kearfott, R. B. (1996). *Rigorous Global Search: Continuous Problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Lin, Y., Enszer, J. A., & Stadtherr, M. A. (2008). Enclosing all solutions of two-point boundary value problems for ODEs. *Computers and Chemical Engineering*, 32, 1714–1725.
- Lin, Y. & Stadtherr, M. A. (2006). Deterministic global optimization for parameter estimation of dynamic systems. *Industrial and Engineering Chemistry Research*, 45, 8438–9448.
- Lin, Y. & Stadtherr, M. A. (2007a). Deterministic global optimization of nonlinear dynamic systems. *AIChE Journal*, 53, 866–875.
- Lin, Y. & Stadtherr, M. A. (2007b). Guaranteed state and parameter estimation for nonlinear continuous-time systems with bounded-error measurements. *Industrial and Engineering Chemistry Research*, 46, 7198–7207.
- Lin, Y. & Stadtherr, M. A. (2007c). Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, 57, 1145–1162.
- Lohner, R. J. (1992). Computations of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In J. Cash & I. Gladwell, eds., *Computational Ordinary Differential Equations*, pp. 425–435, Oxford, UK: Clarendon Press.
- Makino, K. & Berz, M. (1996). Remainder differential algebras and their applications. In M. Berz,

- C. Bischof, G. Corliss, & A. Griewank, eds., *Computational Differentiation: Techniques, Applications, and Tools*, pp. 63–74, Philadelphia: SIAM.
- Makino, K. & Berz, M. (1999). Efficient control of the dependency problem based on Taylor model methods. *Reliable Computing*, 5, 3–12.
- Makino, K. & Berz, M. (2003). Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4, 379–456.
- Makino, K. & Berz, M. (2004). Taylor model range bounding schemes. In *Third International Workshop on Taylor Methods*, Miami Beach, FL.
- Makino, K. & Berz, M. (2005). Verified global optimization with Taylor model-based range bounders. *Transactions on Computers*, 11, 1611–1618.
- Moon, I., Powers, G., Burch, J., & Clarke, E. (1992). Automatic verification of sequential control systems using temporal logic. *AIChE Journal*, 38, 67–75.
- Moore, R. E. (1966). *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall.
- Nedialkov, N. S. (1999). *Computing Rigorous Bounds on the Solution of An Initial Value Problems for An Ordinary Differential Equation*. Ph.D. thesis, University of Toronto, Toronto, Canada.
- Nedialkov, N. S., Jackson, K. R., & Corliss, G. F. (1999). Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105, 21–68.
- Nedialkov, N. S., Jackson, K. R., & Pryce, J. D. (2001). An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliable Computing*, 7, 449–465.
- Neher, M., Jackson, K. R., & Nedialkov, N. S. (2007). On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45, 236–262.

- Neumaier, A. (1990). *Interval Methods for Systems of Equations*. Cambridge, UK: Cambridge University Press.
- Neumaier, A. (2003). Taylor forms - Use and limits. *Reliable Computing*, 9, 43–79.
- Park, T. & Barton, P. (1997). Implicit model checking of logic-based control systems. *AIChE Journal*, 43, 2246–2260.
- Pryce, J. D. & Corliss, G. F. (2006). Interval arithmetic with containment sets. *Computing*, 78, 251–276.
- Rauh, A., Auer, E., & Hofer, E. P. (2006). ValEncIA-IVP: A case study of validated solvers for initial value problems. In *12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2006)*, Duisburg, Germany.
- Srinivasan, R., Dimitriadis, V., Shah, N., & Venkatasubramanian, V. (1998). Safety verification using a hybrid knowledge-based mathematical programming framework. *AIChE Journal*, 44, 361–371.
- Srinivasan, R., Dimitriadis, V. D., Shah, N., & Venkatasubramanian, V. (1997). Integrating knowledge-based and mathematical programming approaches for process safety verification. *Computers and Chemical Engineering*, 21, S905–S910.
- Venkatasubramanian, V. & Preston, M. (1995). A perspective on intelligent systems for process hazards analysis. In J. F. Davis, G. Stephanopoulos, & V. Venkatasubramanian, eds., *Proceedings of the First Intelligent Conference on Intelligent Systems in Process Engineering*, pp. 160–171, CACHE.
- Zhao, C., Bhushan, M., & Venkatasubramanian, V. (2005a). PHASuite: An automated HAZOP

analysis tool for chemical processes, Part I: Knowledge engineering framework. *Transactions IChemE, Part B*, 83, 509–532.

Zhao, C., Bhushan, M., & Venkatasubramanian, V. (2005b). PHASuite: An automated HAZOP analysis tool for chemical processes, Part II: Implementation and case study. *Transactions IChemE, Part B*, 83, 533–548.

Table 1: Batch reactor parameters.

Parameter	Description	Value
$k_0$	Kinetic rate constant	$0.022 \text{ s}^{-1}$
$C_{A0}$	Initial concentration of A	$10 \text{ mol/m}^3$
$V$	Volume of the reactor	$0.1 \text{ m}^3$
$C_p$	Total heat capacity	$60 \text{ J/mol K}$
$E_a$	Activation energy	$6000 \text{ J/mol}$
$R$	Gas constant	$8.314 \text{ J/mol/K}$
$\Delta H_R$	Heat of reaction	$-140,000 \text{ J/mol}$
$UA$	Heat transfer constant	$3 \text{ W/K}$
$T_a$	Coolant temperature	$290 \text{ K}$

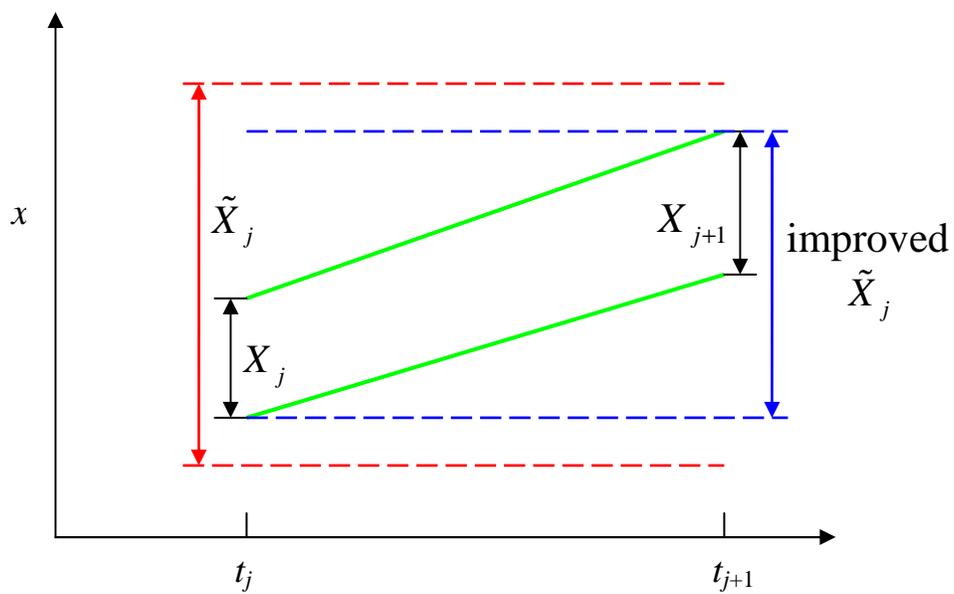


Figure 1: Improvement of the coarse enclosure  $\tilde{X}_j$  based on monotonicity of  $x(t)$  for  $t \in [t_j, t_{j+1}]$ .

See text for complete discussion.

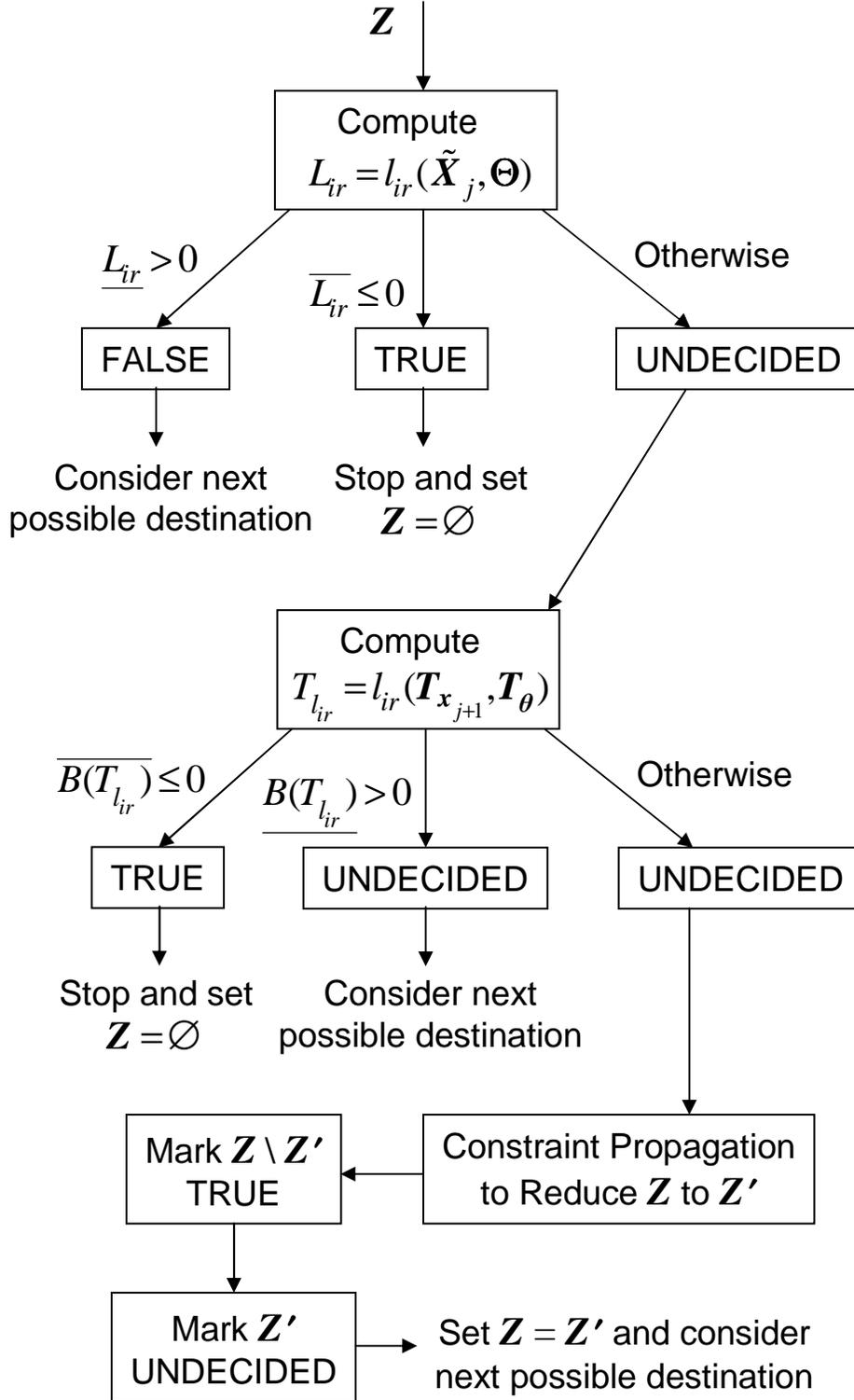


Figure 2: Summary of procedure for testing a region  $\mathbf{Z} = (\mathbf{X}_0, \Theta)^T$  for transition from state  $s_i$  to  $s_r$  at time step  $j + 1$ . See text for complete details.

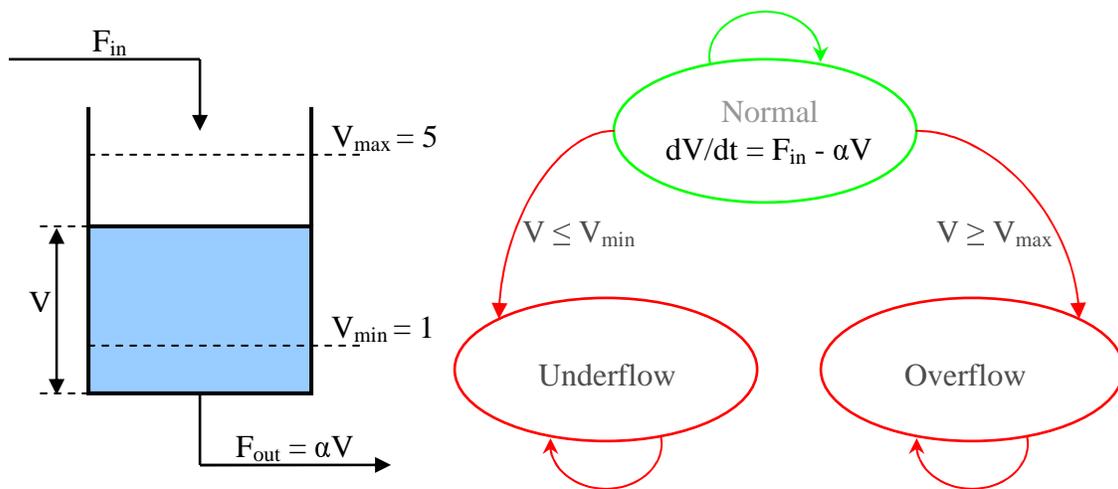


Figure 3: State-based representation of tank flow problem (Huang et al., 2002).

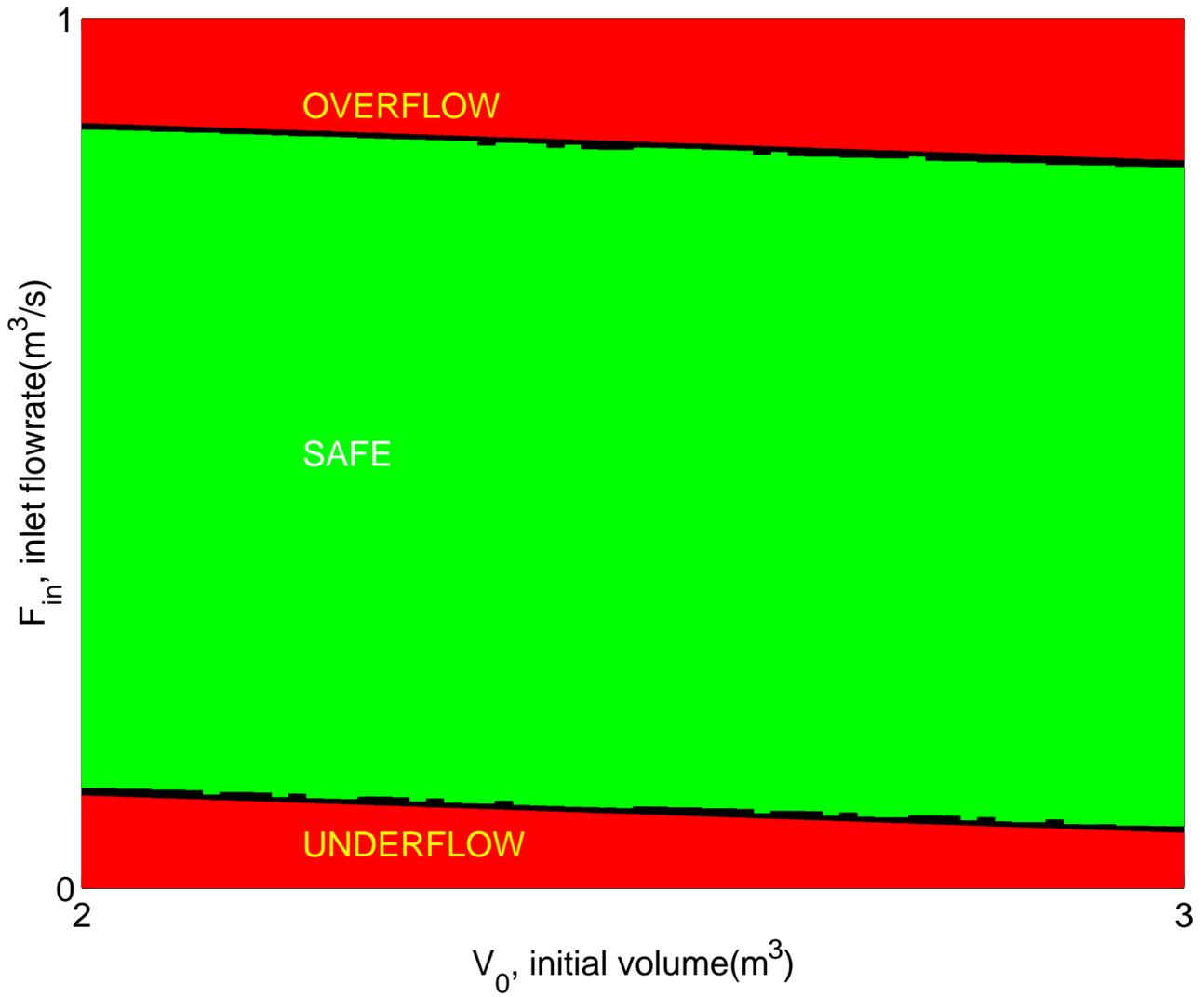


Figure 4: Safety analysis for linear tank flow problem with time horizon  $t_H = 10$  s.

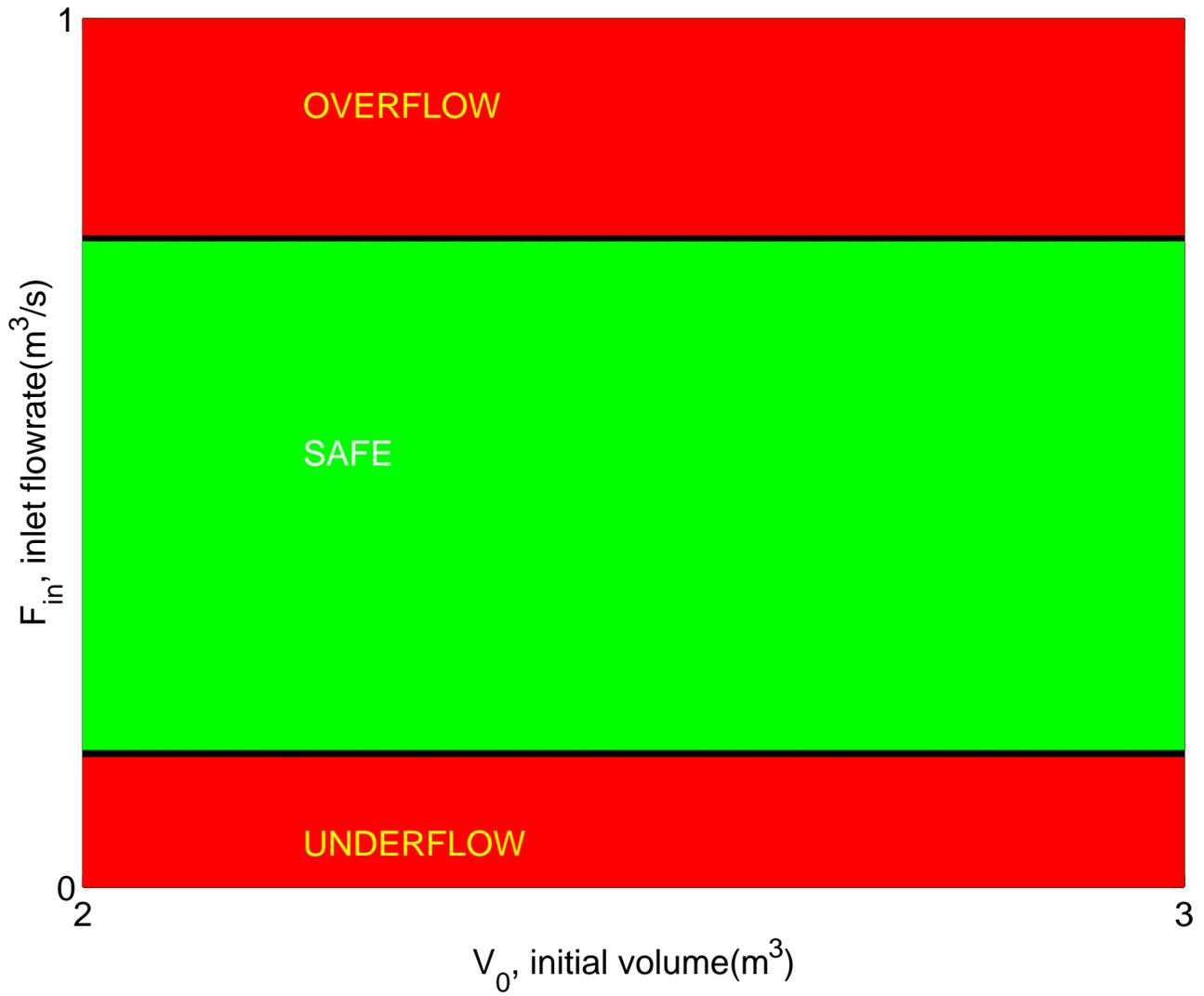


Figure 5: Safety analysis for linear tank flow problem with time horizon  $t_H = 100$  s.

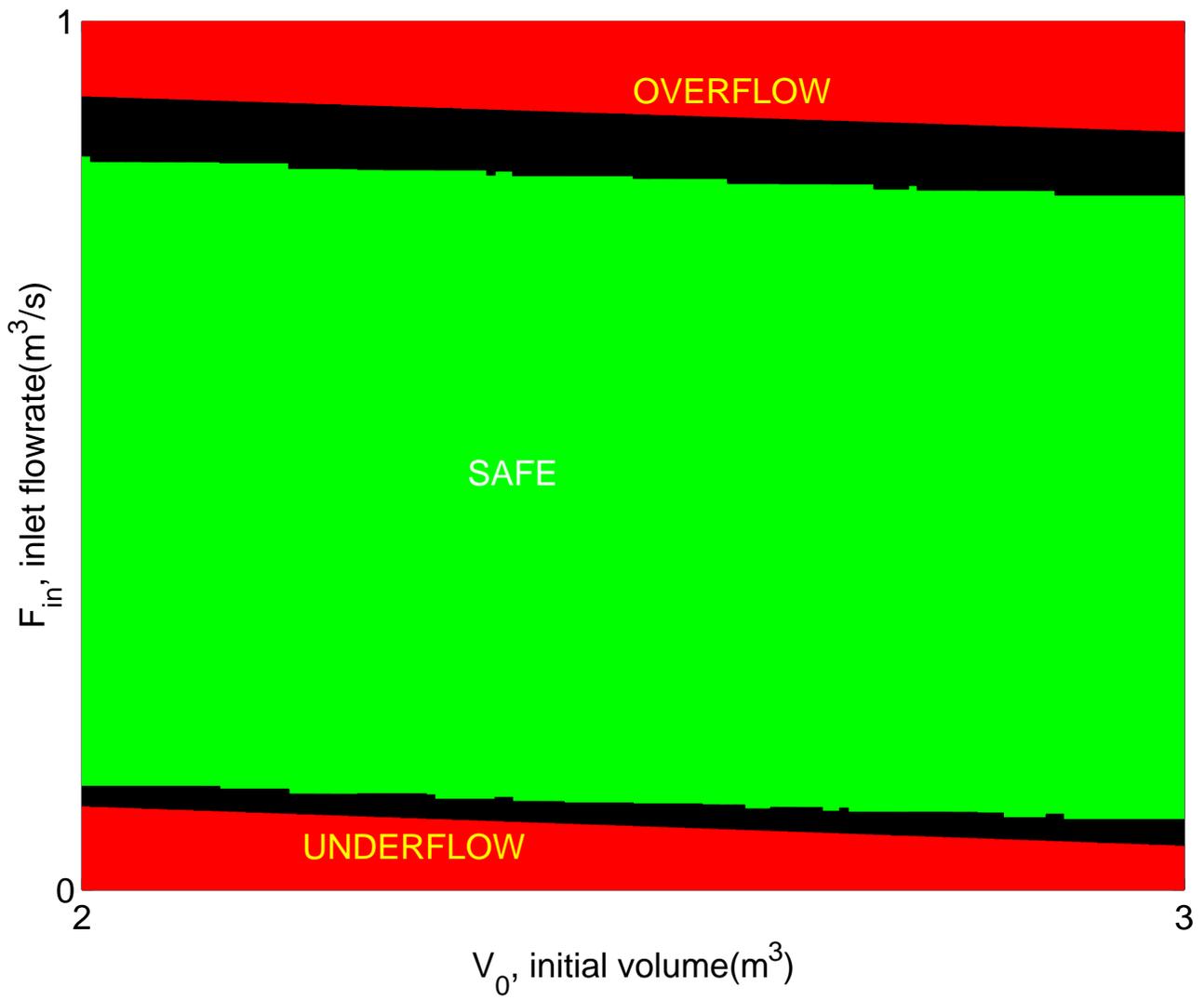


Figure 6: Safety analysis for linear tank flow problem with uncertain  $\alpha$  and time horizon  $t_H = 10$  s.

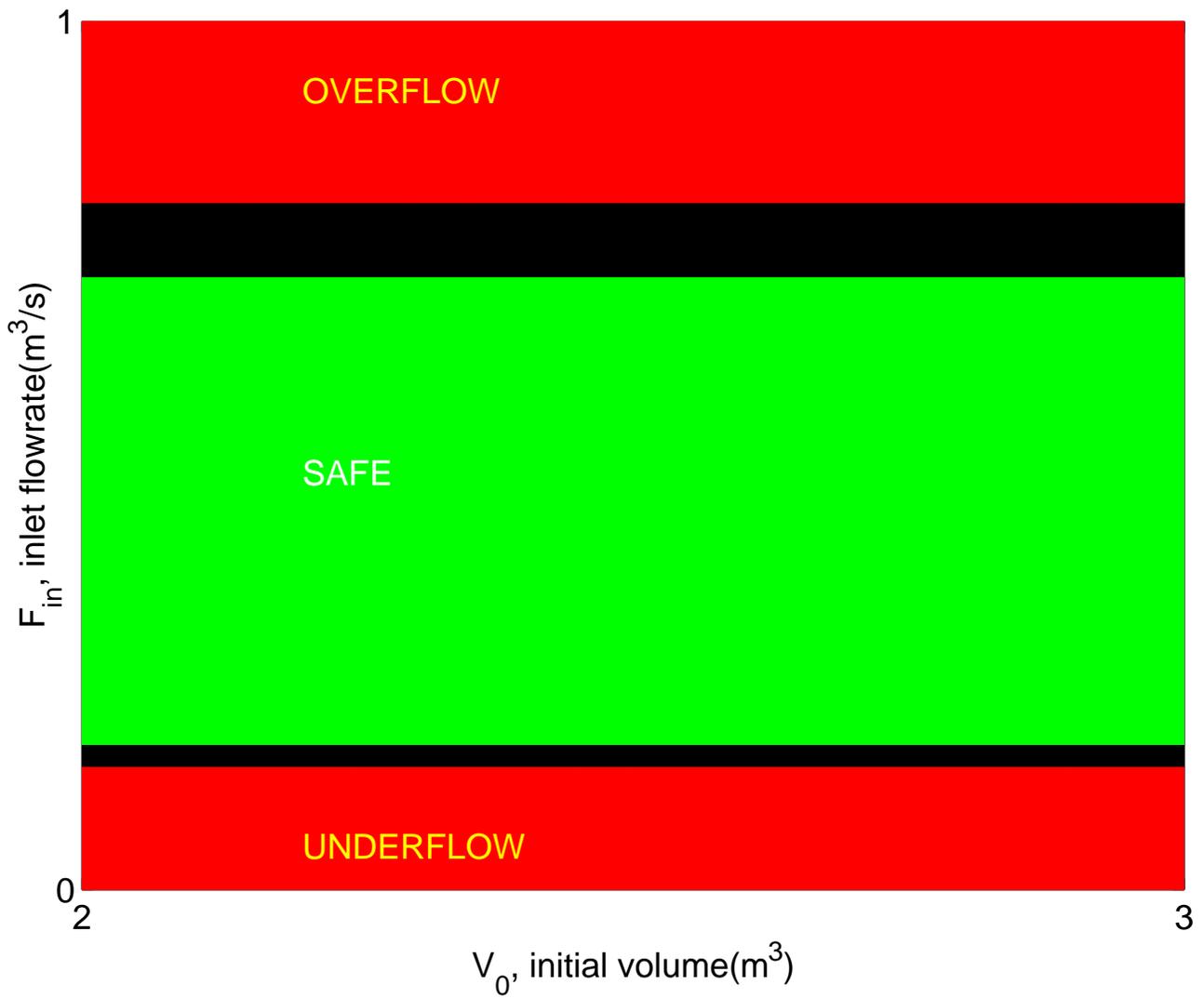


Figure 7: Safety analysis for linear tank flow problem with uncertain  $\alpha$  and time horizon  $t_H = 100$  s.



Figure 8: Safety analysis for nonlinear tank flow problem with time horizon  $t_H = 10$  s.

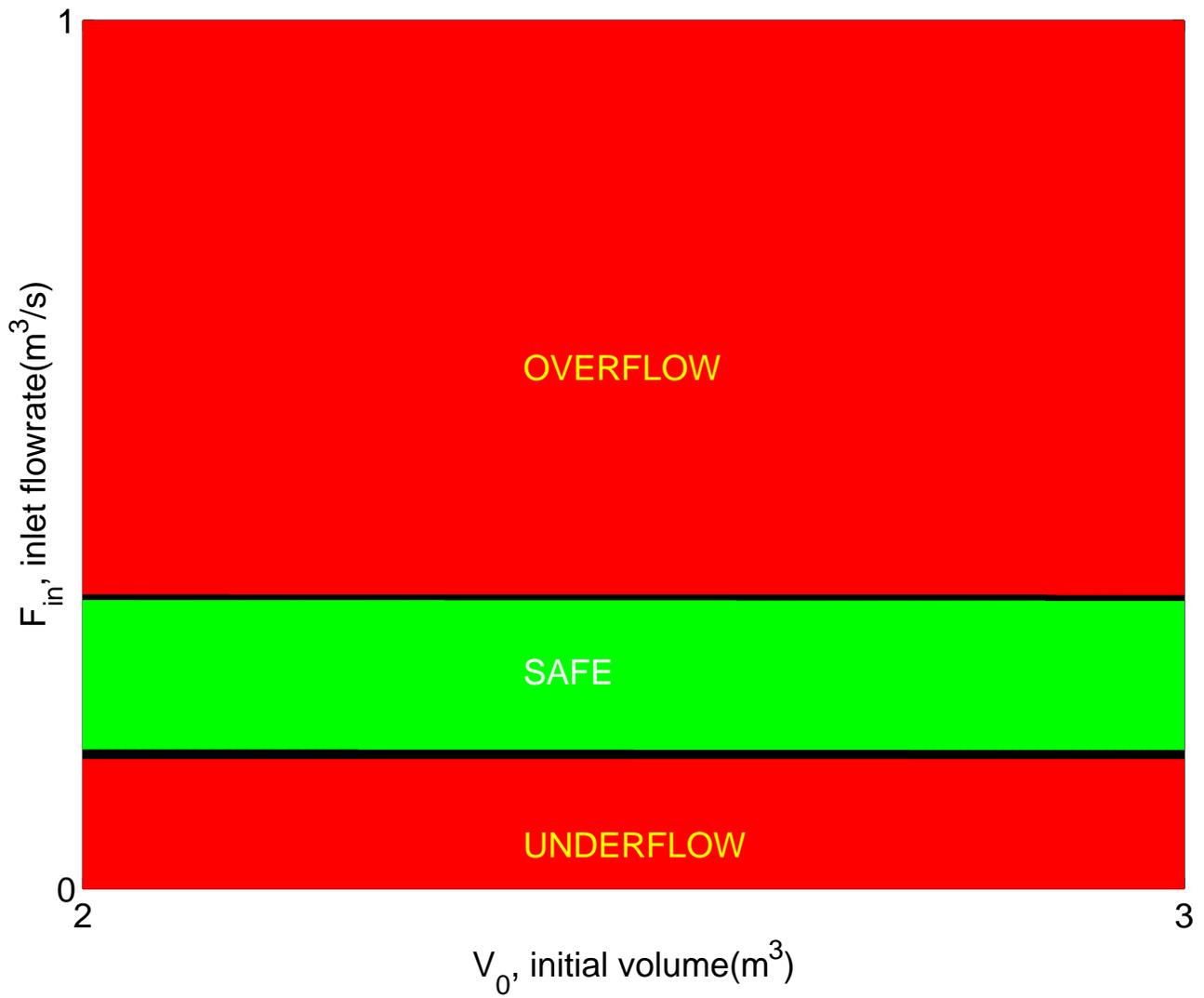


Figure 9: Safety analysis for nonlinear tank flow problem with time horizon  $t_H = 100$  s.



Figure 10: Safety analysis for nonlinear tank flow problem with uncertain  $\alpha$  and time horizon  $t_H = 10$  s.



Figure 11: Safety analysis for nonlinear tank flow problem with uncertain  $\alpha$  and time horizon  $t_H = 100$  s.

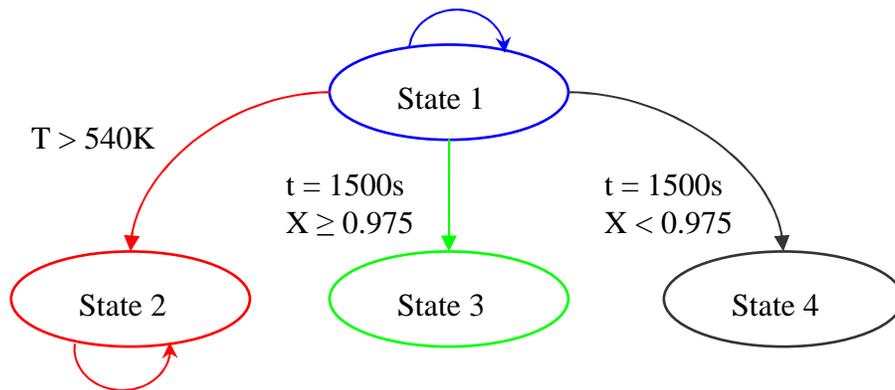


Figure 12: State-based representation of the batch reactor problem (Huang et al., 2002).

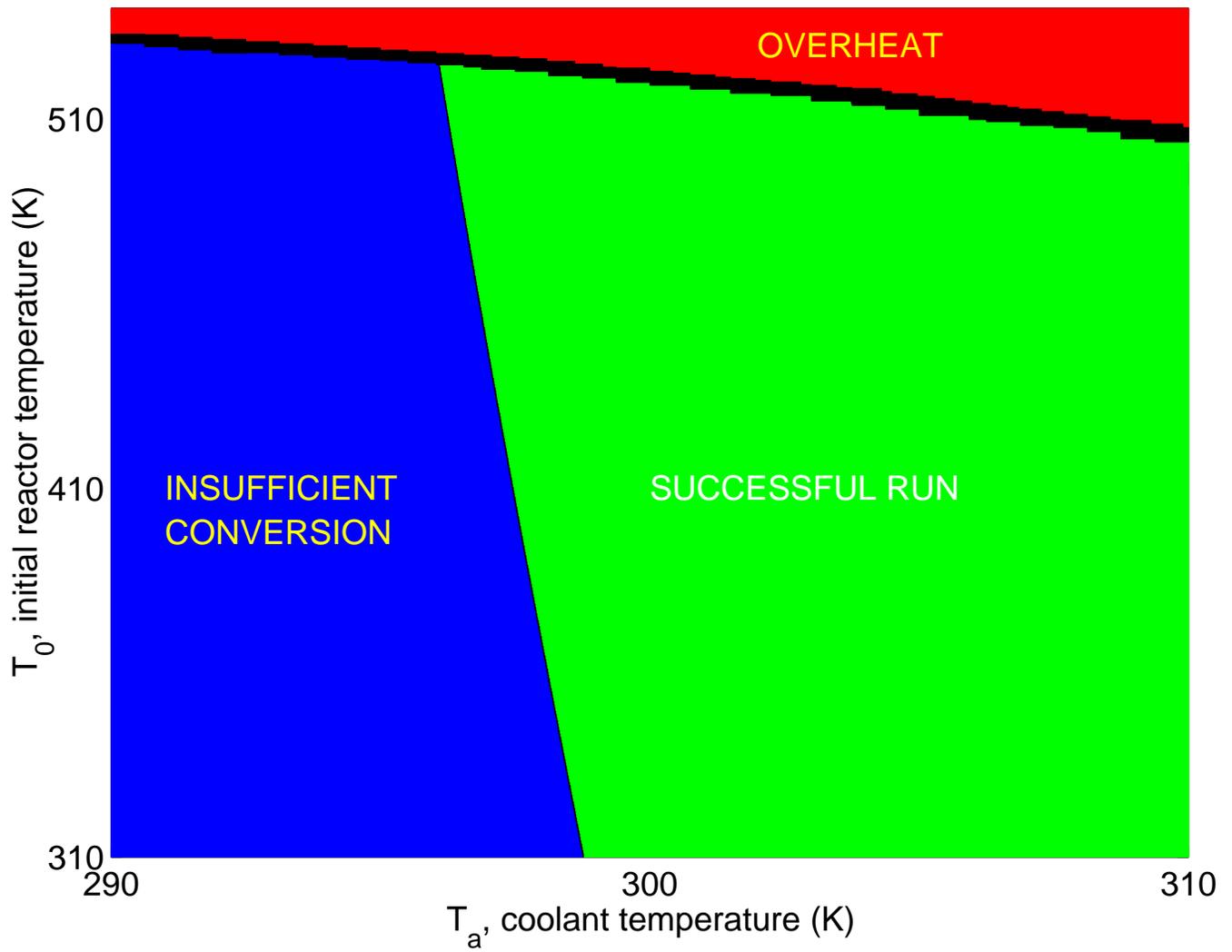


Figure 13: Safety analysis for the batch reactor problem in  $(T_0, T_a)$  operating space.

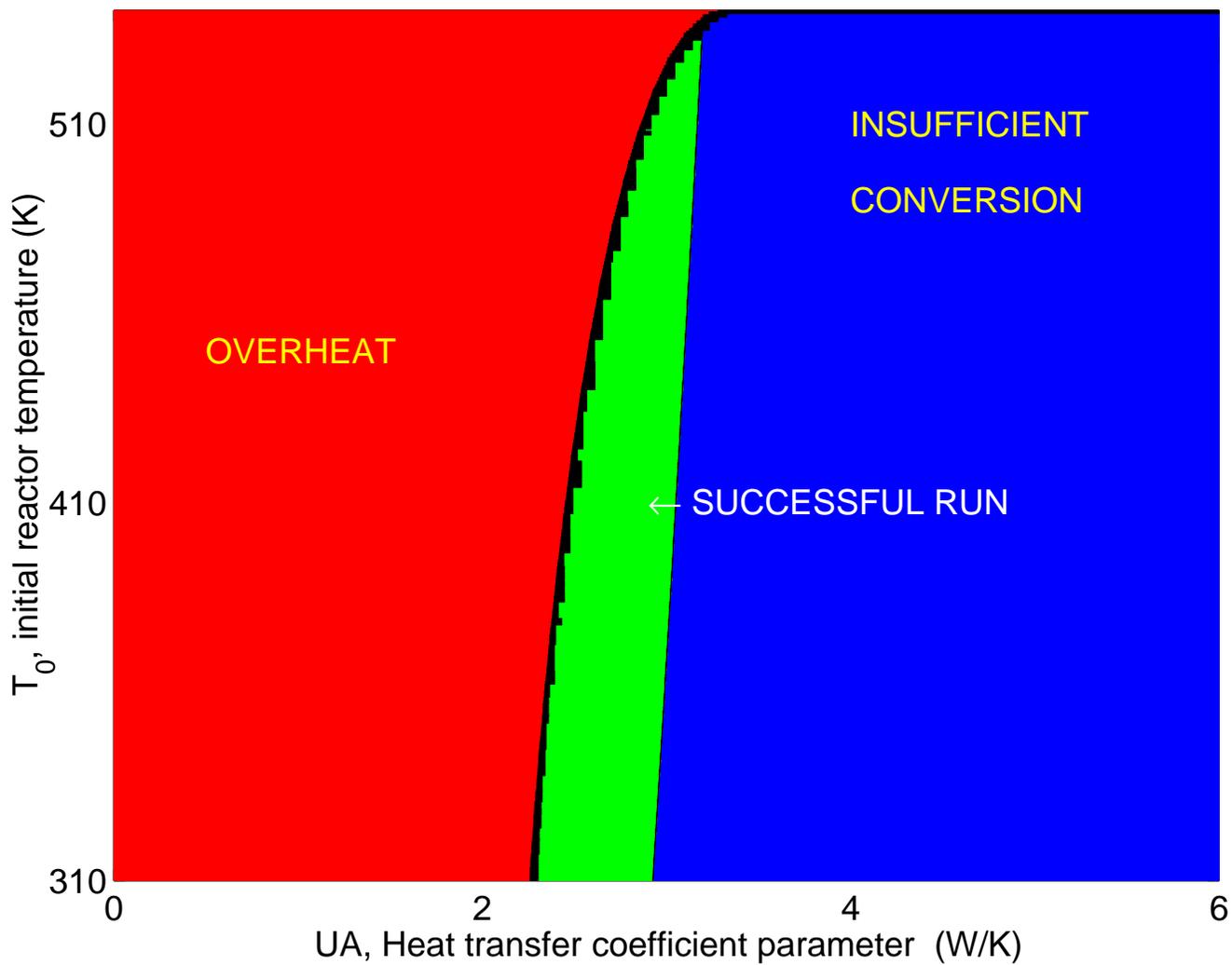


Figure 14: Safety analysis for the batch reactor problem in  $(T_0, UA)$  operating space.

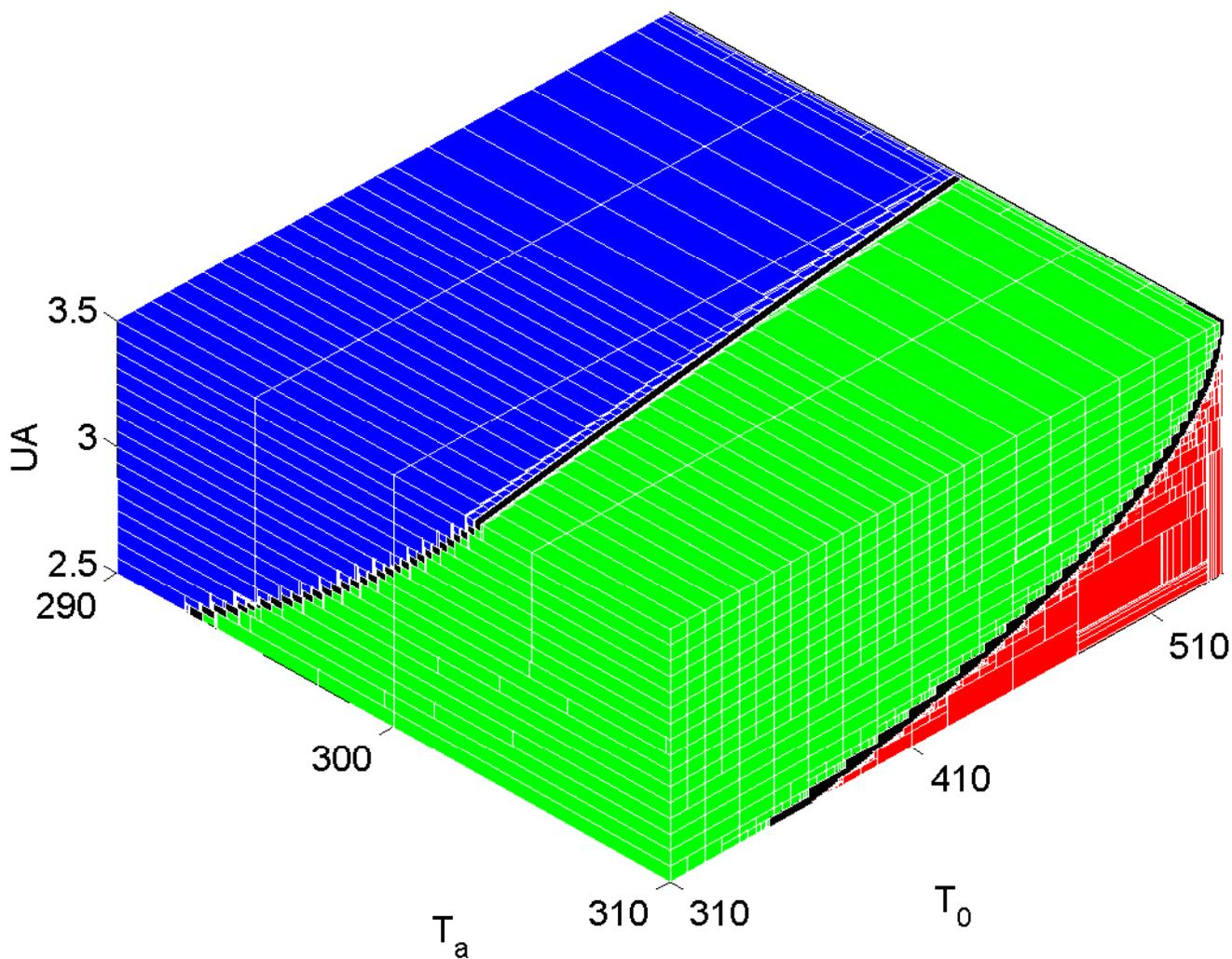


Figure 15: Safety analysis for the batch reactor problem in  $(T_0, T_a, UA)$  operating space. Green region denotes safe, successful run; Red region denotes overheating (unsafe operation); Blue region denotes safe, but unsuccessful run (insufficient conversion); Black region denotes undecided.