Reachability Analysis for One Dimensional Linear Parabolic Equation

Hoang-Dung Tran * **Weiming Xiang** ** **Stanley Bak** ***
Taylor T. Johnson ****

* *Electrical Engineering and Computer Science Department, Vanderbilt University, TN 37023, USA (e-mail: trhoangdung@gmail.com).*
** *Electrical Engineering and Computer Science Department, Vanderbilt University, TN 37023, USA (e-mail: xiangwming@gmail.com)*
*** *Aerospace Systems Directorate, Air Force Research Laboratory, USA (e-mail: stanleybak@gmail.com)*
**** *Electrical Engineering and Computer Science Department, Vanderbilt University, TN 37023, USA (e-mail: taylor.johnson@gmail.com)*

Abstract: Partial differential equations (PDE) mathematically describe a wide range of phenomena such as fluid dynamics, or quantum mechanics. Although great achievements have been accomplished in the field of numerical method for solving PDEs, from a safety verification (or falsification) perspective, methods are still needed to verify (or falsify) a system whose dynamics are specified as PDEs that may depend not only on space, but also on time. As many cyber-physical systems (CPS) involve sensing and control of physical phenomena modeled as PDEs, reachability analysis of PDEs provides novel methods for safety verification and falsification. As a first step to address this challenging problem, we propose a reachability analysis approach leveraging the well-known Finite Element Method (FEM) for a class of one-dimensional linear parabolic PDEs with *fixed but uncertain* inputs and initial conditions, which are a subclass of PDEs useful for modeling, for instance, heat flows. In particular, a *continuous approximate* reachable set of the parabolic PDE is first computed using linear interpolation. To enhance conservativeness of our approach, we investigate the error bound between the numerical solution and the exact analytically unsolvable solution to bloat the continuous approximate reachable set and then use this bloated reachable set for safety verification and falsification. In addition, in the case that the safety specification is violated, our approach produces a numerical trace to prove that there exists an initial condition and input that leads the system to an unsafe state.

Keywords: Reachability Analysis, Cyber-Physical Systems, Partial Differential Equations

1. INTRODUCTION

Reachability analysis is a fundamental problem in safety verification of cyber-physical systems. Over the last two decades, a numerous techniques and tools have been proposed for a class of continuous and hybrid systems whose dynamics are described by linear or nonlinear ordinary differential equations (ODEs). Reachability analysis using zonotopes and support functions have been proved as the most efficient and scalable approaches that can verify linear continuous and hybrid systems with up to hundred state variables Frehse et al. (2011); Althoff (2015). For nonlinear case, Flow* Chen et al. (2013) utilizing Taylor model is one of most efficient and reliable tool.

The core challenge in reachability analysis is the state explosion problem. Recently, a huge effort has been made to tackle this challenging problem. Among other techniques, simulation-based reachability analysis Bak and Duggirala (2017b,a); Duggirala et al. (2015) and order-reduction abstraction Tran et al. (2017); Chou et al. (2017); Han and Krogh (2006) are promising approaches which show a great improvement in scalability when dealing with both large scale linear and nonlinear continuous/hybrid systems.

Although rigorous works on reachablity analysis have been accomplished for continuous and hybrid systems with ODE dynamics, not much attention has been paid to a class of systems with PDE dynamics which appears in many engineering and science problems such as fluid dynamics control, heat equation, quantum mechanics and water flow. This motivates us to conduct research on this interesting but challenging problem. It should be noted that a typical parabolic equation called heat equation has been used as a benchmark for evaluating the scalability of recent reachability analysis approach dealing with large scale linear systems Han and Krogh (2006). In this context, the heat equation is transformed to a continuous ODE

⋆ Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

model using finite different method (FDM) and the safety specification of interest is given for *discrete* mesh points. It is also worth to emphasize that the input of the considered heat equation benchmark is assumed to be a constant with a small *time-invariant uncertainty* and the initial states of the mesh points are given as bounded boxes.

Even though the heat equation has been shown as a good benchmark for testing the scalability of verification techniques in dealing with large scale linear systems, a deeper study should be done for two reasons. First, it is reasonable to have a safety specification for the whole space between two mesh points and for the whole time between two time steps, not just only at specific mesh points or specific time steps. In other world, we are interested in *continous* not *discrete* reachability analysis. Second, it is required to have an approach that works for more general types of input and initial condition, i.e., an input described by a nonlinear function in both time and space and an initial condition defined by a nonlinear function in space.

In this paper, we propose an approach for *continuous reachability analysis* of a linear parabolic equation with time-invariant uncertain nonlinear input and initial condition. Our main contributions are : 1) extending the well-known finite element method and linear interpolation into *continuous* safety verification/falsification problem of one dimensional parabolic equation; 2) enhance the conservativeness of the proposed method by investigating and utilizing the error between the numerical solution and the exact analytically unsolvable solution; 3) implement the proposed method in a prototype called *pdev* that is available online for further testing and evaluation.

The rest of the paper is organized as follows: Section 2 presents the problem formulation and formally defines the safety verification problem. Section 3 gives the main steps for computing the continuous reachable set of the parabolic equation using FEM and linear interpolation. Section 4 investigates the error between the numerical solution and the exact unknown solution. Section 5 describes the continuous safety verification/falsification algorithm. Section 6 discusses deeply about the conservativeness and soundness of our approach. Section 7 illustrates shortly the implementation of the proposed approach in the *pdev* prototype and evaluates the proposed approach via a specific example. Section 8 concludes the paper and discusses some interesting directions for the future works.

2. PROBLEM FORMULATION

In this section, we present the mathematical description of the parabolic equation and give a formal safety verification definition. The problem of interest is the *continuous* safety verification of a one-dimensional linear parabolic equation with time-invariant uncertain nonlinear input and initial condition that is mathematically given in the following:

$$\begin{aligned} &\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = (1+\epsilon_1) f(x,t),\ 0 < x < L \\ &u(0,t) = u(L,t) = 0, \\ &u(x,0) = (1+\epsilon_2) u_0(x), \end{aligned} \tag{1}$$

where $f(x,t)$ is a nonlinear input function in both time and space and $u_0(x)$ is nonlinear initial condition function in space; ϵ_1, ϵ_2 are in bounded ranges that illustrates the time-invariant uncertainty in input and initial condition.

The second equation describes the boundary condition of our problem. Here we are considering the problem with *Dirichlet* boundary condition. There are two other common boundary conditions called *Neumann* and *Robin* boundary conditions Larson and Bengzon (2013). For simplification, we define $\alpha = (1+\epsilon_2)$ and $\beta = (1+\epsilon_1)$ and use them as time-invariant uncertainty parameters in the rest of the paper. We also use $\dot{u}$ to state for $\frac{\partial u}{\partial t}$, u' for $\frac{\partial u}{\partial x}$ and u'' for $\frac{\partial^2 u}{\partial x^2}$. The reachable set and safety verification problem for this class of system are defined as follows.

Definition 2.1. (*Continuous reachable set*). A continuous bounded-time reachable set of the system (1) is defined by:

$$R_{[0,T]}(u) = \{u(x,t),\ 0 \le x \le L,\ 0 \le t \le T < \infty\},$$

where $u(x,t)$ is the solution of the equation (1).

Definition 2.2. (*Continuous bounded-time safety verification*). Given a linear safety specification of the form: $u_1 \le u(x,t) \le u_2$, $0 \le x_1 \le x \le x_2 \le L$ and $0 \le T_1 \le t \le T_2 < \infty$ where $u_1, u_2, x_1, x_2, T_1, T_2$ are scalars, the system (1) is called safe if and only if the reachable set of $u(x,t)$ of the system in the time range $[T_1, T_2]$ denoted by $R_{[T_1,T_2]}(u) \subset R_{[0,T_2]}(u)$ satisfies the safety specification.

To verify the safety of the system, the continuous reachable set $R_{[0,T]}(u)$ need to be computed. The main challenge is, with a nonlinear input and initial condition, it is in general hard to solve for the exact solution $u(x,t)$ analytically. Instead, only an *approximate* solution $\tilde{u}(x,t)$ can be computed using finite element method and linear interpolation. Thus, instead of constructing the *exact* reachable set $R_{[0,T]}(u)$, we can only construct an *approximate* reachable set $R_{[0,T]}(\tilde{u})$ for the system. By doing this, the ineluctable approximation error $e(x,t) = u(x,t) - \tilde{u}(x,t)$ need to be taken into account. Then, using this error, we bloat the approximate reachable set before checking whether or not it violates the safety specification. The next section will focus on the computation of the approximate continuous reachable set in a bounded time $R_{[0,T]}(\tilde{u})$.

3. APPROXIMATE CONTINUOUS REACHABLE SET COMPUTATION

In this section, we present core steps for obtaining approximate continuous reachable set of the parabolic equation by leveraging the well-known space-time finite element method and linear interpolation. We refer readers to Larson and Bengzon (2013) for further detail about finite element method and interpolation.

The approximate continuous reachable set $R_{[0,T]}(\tilde{u})$ are obtained in two steps. First, the approximate discrete reachable set at all mesh points and time steps is computed. Then, using linear interpolation, we construct the approximate continuous reachable set using the computed approximate discrete reachable set in the first step.

3.1 Approximate discrete reachable set computation

The FEM method is a powerful tool to approximate the solution of PDEs at mesh points and time steps.

The general idea is, the space of interest $[0, L]$ is discretized by a list number of mess points $x = [0 = x_0, x_1, x_2, \cdots, x_{m-1}, x_m = L]$. Similarly, the time range $[0, T]$ of interest is also discretized into a list of time steps $t = [0 = t_0, t_1, t_2, \cdots, t_{n-1}, t_n = T]$. For simplification, we use a uniform time step $k = T/n, t_j = j \times k, 0 \leq j \leq n$ and a uniform space step $h = L/m, x_i = i \times h, 0 \leq i \leq m$. To compute the approximate discrete solution $\tilde{u}(x_i, t_j)$ of the system (1) at all time steps and mesh points, the *weak form* of the system (1) is obtained below by multiplying two sides of the first equation by a *test function* v and integrating by parts.

$$\int_{I_n}\int_{\Omega}(\dot{u} - u'')vdxdt = \int_{I_n}\int_{\Omega}\beta f(x,t)vdxdt, \quad (2)$$

where $I_n = (t_{n-1}, t_n]$ and $\Omega = (0, L)$.

If we choose a specific class of the test function v such that: $v \in V = \{v(x,t)|\ v(0,t) = v(L,t) = 0\}$, equation (2) becomes:

$$\int_{I_n}\int_{\Omega}(\dot{u}v + u'v')dxdt = \int_{I_n}\int_{\Omega}\beta f(x,t)vdxdt,\ \forall v \in V.$$

The above equation is called the weak form or the variational formulation of the system (1). We are interested in finding a *piecewise linear* approximate solution $\tilde{u}(x,t)$ of $u(x,t)$ that satisfies the variational formulation. Let,

$$\tilde{u}(x,t) = \tilde{u}_{n-1}(x)\psi_{n-1}(t) + \tilde{u}_n(x)\psi_n(t), \quad (3)$$

where

$$\psi_n(t) = \frac{t - t_{n-1}}{k}, \psi_{n-1}(t) = \frac{t_n - t}{k},\ t \in I_n, \quad (4)$$

and

$$\tilde{u}_n(x) = \tilde{u}_{n,1}\phi_1(x) + \tilde{u}_{n,2}\phi_2(x) + \cdots + \tilde{u}_{n,m-1}\phi_{m-1}(x), \quad (5)$$

with $\phi_i, 1 \leq i \leq m-1$ is a *hat function* defined by:

$$\phi_i = \begin{cases} (x - x_{i-1})/h, & x_{i-1} < x <= x_i \\ (x_{i+1} - x)/h, & x_i < x <= x_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

It is easy to see that $\phi_i \in V$. Thus, the piecewise linear approximate solution $\tilde{u}(x,t)$ of $u(x,t)$ satisfies the following discrete variational formulation:

$$\int_{I_n}\int_{\Omega}(\dot{\tilde{u}}\phi_i + \tilde{u}'\phi_i')dxdt = \int_{I_n}\int_{\Omega}\beta f(x,t)\phi_i dxdt \quad (7)$$

From (3), (4) and (5), we have:

$$\dot{\tilde{u}}(x,t) = \tilde{u}_{n-1}(x)\dot{\psi_{n-1}}(t) + \tilde{u}_n(x)\dot{\psi_n}(t) = \frac{\tilde{u}_n(x) - \tilde{u}_{n-1}(x)}{k}, \quad (8)$$

and

$$\tilde{u}'(x,t) = \tilde{u}'_{n-1}(x)\psi_{n-1}(t) + \tilde{u}'_n(x)\psi_n(t), \quad (9)$$

where

$$\tilde{u}'_n(x) = \tilde{u}_{n,1}\phi'_1(x) + \tilde{u}_{n,2}\phi'_2(x) + \cdots + \tilde{u}_{n,m-1}\phi'_{m-1}(x). \quad (10)$$

Inserting (8) and (9) into (7) and using the fact that $\int_{I_n} dt = k$ and $\int_{I_n}\psi_n(t) = \int_{I_n}\psi_{n-1}(t) = k/2$, we have:

$$\int_{\Omega}\tilde{u}_n(x)\phi_i dx - \int_{\Omega}\tilde{u}_{n-1}(x)\phi_i dx + \frac{k}{2}\int_{\Omega}\tilde{u}'_{n-1}\phi'_i dx + \frac{k}{2}\int_{\Omega}\tilde{u}'_n\phi'_i dx = \int_{I_n}\int_{\Omega}\beta f(x,t)\phi_i dxdt, \forall i, 1 \leq i \leq m-1 \quad (11)$$

Using (5) and (10) yields:

$$\begin{aligned} &\int_{\Omega}\tilde{u}_n(x)\phi_i dx = (\int_{\Omega}\phi_i\phi_1 dx)\tilde{u}_{n,1} + \\ &(\int_{\Omega}\phi_i\phi_2 dx)\tilde{u}_{n,2} + \cdots + (\int_{\Omega}\phi_i\phi_{m-1}dx)\tilde{u}_{n,m-1}, \\ &\int_{\Omega}\tilde{u}'_n(x)\phi'_i dx = (\int_{\Omega}\phi'_i\phi'_1 dx)\tilde{u}_{n,1} + \\ &(\int_{\Omega}\phi'_i\phi'_2 dx)\tilde{u}_{n,2} + \cdots + (\int_{\Omega}\phi'_i\phi'_{m-1}dx)\tilde{u}_{n,m-1}. \end{aligned} \quad (12)$$

Using (12), (11) can be written as:

$$(M + \frac{k}{2}S)\tilde{u}_n = (M - \frac{k}{2}S)\tilde{u}_{n-1} + \beta\bar{g}_n, \quad (13)$$

where

$\tilde{u}_n = [\tilde{u}_{n,1}, \tilde{u}_{n,2} \cdots, \tilde{u}_{n,m-1}]^T$,

$M \in \mathbb{R}^{(m-1)\times(m-1)}, M = [\int_{\Omega}\phi_i\phi_j]_{i,j}$,

$S \in \mathbb{R}^{(m-1)\times(m-1)}, S = [\int_{\Omega}\phi'_i\phi'_j]_{i,j}$,

$\bar{g}_n = [\bar{g}_{n,1}, \bar{g}_{n,2}, \cdots, \bar{g}_{n,m-1}]^T, \bar{g}_{n,i} = \int_{I_n}\int_{\Omega}f(x,t)\phi_i dxdt.$

M, S and $\bar{g}_n$ are called mass matrix, stiff matrix and load vector of the system (1) respectively. Further computing these matrices and vector shows that M and S are symmetric matrices. Thus, the inversion of $(M + \frac{k}{2}S)$ exists and is easy to compute. Consequently, the *discrete* approximate model of the system (1) can be derived as follows.

$$\tilde{u}_n = A\tilde{u}_{n-1} + \beta g_n, \quad (14)$$

where $A = (M + \frac{k}{2}S)^{-1}(M - \frac{k}{2}S)$ and $g_n = (M + \frac{k}{2}S)^{-1}\bar{g}_n$.

It should be clarified that we use the notation $\tilde{u}_n$ as a scalar vector stating for the approximate solution $\tilde{u}(x,t)$ at specific time step $t = t_n$ and at all of mess points x_i. Thus, $\tilde{u}_n$ is a *discrete approximate solution* of the system (1) at $t = t_n$. Keep in mind that it is different from the notation $\tilde{u}_n(x)$ which is a function of the position x.

The approximate discrete reachable set of system (1) is obtained below from (14):

$$\tilde{\mathbf{u}}_\mathbf{n} = \alpha\mathbf{z}_\mathbf{n} + \beta\mathbf{l}_\mathbf{n}, \quad (15)$$

where

$$\begin{aligned} z_n &= Az_{n-1} = A^2 z_{n-2} = \cdots = A^n z_0, \\ z_0 &= [u_0(x_1), u_0(x_2), \cdots, u_0(x_{m-1})]^T, \\ l_n &= g_n + Al_{n-1} = g_n + Ag_{n-1} + A^2 l_{n-2}, \\ &= \cdots = \Sigma_{j=0}^{n} A^j g_{n-j},\ l_0 = [0, 0, .., 0]^T. \end{aligned}$$

We have obtained the *approximate discrete reachable set* of system (1) using space-time FEM method. In the next subsection, we construct the *approximate continuous reachable set* of the system using linear interpolation.

3.2 Approximate continuous reachable set computation

Using linear interpolation, the approximate continuous reachable set of the system (1) can be constructed from the approximate discrete reachable set derived previously in two steps. The first step is constructing the linear interpolation set in space from the approximate discrete reachable

set using (5) and (6). The second step is constructing the approximate continuous reachable set using (3) and the interpolation set in space of the first step.

Linear interpolation set in space. From (5), (6) and (15), for every x, $0 < x < L$, the linear interpolation set in space at the time step $t = t_n$ can be constructed in the following.

$$\tilde{\mathbf{u}}_\mathbf{n}(\mathbf{x}) = (\mathbf{a_n}\alpha + \mathbf{b_n}\beta)\mathbf{x} + \mathbf{c_n}\alpha + \mathbf{d_n}\beta, \tag{16}$$

where $\tilde{u}_n(x)$ is represented as one-dimensional array, i.e., $\tilde{u}_n(x) \in \mathbb{R}^m$ and a_n, b_n, c_n and $d_n \in \mathbb{R}^m$ are vectors whose values depends on x as follows.

For $\mathbf{0 < x \leq x_1}$:
$a_n[1] = z_n[1]/h,\ b_n[1] = l_n[1]/h,\ c_n[1] = 0,\ d_n[1] = 0.$
For $\mathbf{x_{i-1} < x \leq x_i, 1 < i \leq m - 1}$:
$a_n[i] = (z_n[i] - z_n[i-1])/h,\ b_n[i] = (z_n[i] - z_n[i-1])/h,$
$c_n[i] = i \cdot z_n[i-1] - (i-1) \cdot z_n[i],$
$d_n[i] = i \cdot l_n[i-1] - (i-1) \cdot l_n[i].$
For $\mathbf{x_{m-1} < x <= x_m = L}$:
$a_n[m] = -z_n[m-1]/h,\ b_n[m] = -l_n[m-1]/h,$
$c_n[m] = mZ_n[m-1],\ d_n[m] = ml_n[m-1].$

Approximate continuous reachable set. In this step, we are interested in the approximate continuous reachable set $\tilde{u}(x,t)$ of the system (1) at anytime $t \in [0, T]$ and at any $x \in [0, L]$. Using (4) and the interpolation set in space computed in previous step, the approximate continuous reachable set is obtained as a function of time variable t, the position variable x and the uncertainty parameters (α, β) as follows.

$$\begin{aligned} R_{[0,T]}[\tilde{u}] = \{\tilde{u}(x,t) \mid \tilde{u}(x,t) = (1/k)(\Delta_a\alpha + \Delta_b\beta)xt + \\ (\Delta_c\alpha + \Delta_d\beta)t/k + (\Delta_{\gamma(a)}\alpha + (\Delta_{\gamma(b)}\beta)x \\ ((\Delta_{\gamma(c)}\alpha + (\Delta_{\gamma(d)}\beta)\}, \end{aligned} \tag{17}$$

where $R_{[0,T]}(\tilde{u})$ is represented as a two-dimensional array, i.e., $R_{[0,T]}(\tilde{u}) \in \mathbb{R}^{m \times n}$ with the associate coefficient matrices $\Delta_a, \Delta_b, \Delta_{\gamma(a)}, \Delta_{\gamma(b)}, \Delta_{\gamma(c)}, \Delta_{\gamma(d)} \in \mathbb{R}^{m \times n}$ are defined below.

$$\begin{aligned} &\textbf{For } \mathbf{1 \leq j \leq n} : \\ &\Delta_a.column[j] = a_{j-1} - a_j, \\ &\Delta_b.column[j] = b_{j-1} - b_j, \\ &\Delta_c.column[j] = c_{j-1} - c_j, \\ &\Delta_d.column[j] = d_{j-1} - d_j, \\ &\Delta_{\gamma(a)}.column[j] = j \cdot a_j - (j-1) \cdot a_{j-1}, \\ &\Delta_{\gamma(b)}.column[j] = j \cdot b_j - (j-1) \cdot b_{j-1}, \\ &\Delta_{\gamma(c)}.column[j] = j \cdot c_j - (j-1) \cdot c_{j-1}, \\ &\Delta_{\gamma(d)}.column[j] = j \cdot d_j - (j-1) \cdot d_{j-1}, \\ &a_0 = b_0 = d_0 = [0, 0, \cdots, 0]^T,\ c_0 = z_0. \end{aligned} \tag{18}$$

with $(a_j,\ b_j,\ c_j,\ d_j)$ from the interpolation set in space at time step $t = t_j$.

The safety verification problem can be solved with the constructed approximate continuous reachable set if we neglect the error between the approximate solution $\tilde{u}(x,t)$ and the exact unknown solution $u(x,t)$. However, we can enhance the conservetiveness of using the approximate continuous reachable set by further analyzing the ineluctable error $e(x,t) = u(x,t) - \tilde{u}(x,t)$ caused by numerical approach. In the next section, both theoretical and computable bounds of the error are discussed in detail.

4. ERROR ANALYSIS

In this section, we first discuss about the theoretical bound of the error $e(x,t)$. Then, we show that this error $e(x,t)$ can also be approximated using FEM method.

4.1 Theoretical error bound

Theorem 4.1. For any $x \in (x_{i-1}, x_i]$ and $t \in (t_{j-1}, t_j]$, the error $e(x,t)$ between the exact solution and the numerical solution of the system (1) using the above space-time FEM method satisfies:

$$\|e(x,t)\|_\infty \leq \|e(x_{i-1}, t_{j-1})\|_\infty + k\|\dot{e}\|_\infty + h\|e'\|_\infty, \tag{19}$$

where $\|\cdot\|_\infty$ denotes the infinity norm of a function or a vector.

Proof 4.1. The proof is given in the Appendix.

The Theorem 4.1 theoretically shows that the error between the numerical solution $\tilde{u}(x,t)$ and the exact unknown solution $u(x,t)$ can be reduced if we decrease the time step k and the space step h. However, this is also certainly increase the computation cost in safety verification. It is interesting to emphasize that the exact solution for the error $e(x,t)$ is also analytically unsolvable in general. The only way to deal with this problem is again using FEM method and linear interpolation to approximate this error which is described in detail in the next subsection.

4.2 Approximate continuous error computation

Recall that $\tilde{u}(x,t)$ is linear function in x. Thus, we have $\tilde{u}''(x,t) = 0$. Using this fact, it is easy to see that the error $e(x,t)$ is the solution of the following equation:

$$\begin{aligned} &\dot{e} - e'' = \beta f - \dot{\tilde{u}} = r(\tilde{u}),\ 0 < x < L, \\ &e(0,t) = e(L,t) = 0, \\ &e(x,0) = 0. \end{aligned} \tag{20}$$

Again, using FEM method and linear interpolation with the same time step k and space step h, the approximate continuous reachable set $R_{[0,T]}(\tilde{e})$ of $e(x,t)$ can be constructed as follows.

$$\begin{aligned} R_{[0,T]}(\tilde{e}) = \{\tilde{e}(x,t) \mid \tilde{e}(x,t) = (1/k)(\Delta_{a_e}\alpha + \Delta_{b_e}\beta)xt + \\ (\Delta_{c_e}\alpha + \Delta_{d_e}\beta)t/k + (\Delta_{\gamma(a_e)}\alpha + (\Delta_{\gamma(b_e)}\beta)x + \\ ((\Delta_{\gamma(c_e)}\alpha + (\Delta_{\gamma(d_e)}\alpha)\}. \end{aligned}$$

We have analyzed the error between the exact unknown solution and the numerical one. In the next section, a more conservative approximate continuous reachable set of the system (1) is constructed by utilizing the approximate continuous reachable set of the error.

5. CONTINUOUS SAFETY VERIFICATION/FALSIFICATION

The previous two sections focus on the computation of the approximate continuous reachable set $R_{[0,T]}(\tilde{u})$ of the system (1) and its corresponding approximate continuous reachable set of the error $R_{[0,T]}(\tilde{e})$. Combining these two

reachable sets, a more conservative approximate continuous reachable set $R_{[0,T]}(\bar{u})$ of the system (1) can be obtained as follows.

$$\begin{aligned} R_{[0,T]}(\bar{u}) = \{\bar{u}(x,t) =& \tilde{u}(x,t) + \tilde{e}(x,t) \\ =& q_1(\alpha,\beta)xt + q_2(\alpha,\beta)t + \\ & q_3(\alpha,\beta)x + q_4(\alpha,\beta)\}, \end{aligned} \tag{21}$$

where:

$$\begin{aligned} q_1(\alpha,\beta) &= (1/k)[(\Delta_a + \Delta_{a_e})\alpha + (\Delta_b + \Delta_{b_e})\beta], \\ q_2(\alpha,\beta) &= (1/k)[(\Delta_c + \Delta_{c_e})\alpha + (\Delta_d + \Delta_{d_e})\beta], \\ q_3(\alpha,\beta) &= (\Delta_{\gamma(a)} + \Delta_{\gamma(a_e)})\alpha + (\Delta_{\gamma(b)} + \Delta_{\gamma(b_e)})\beta, \\ q_4(\alpha,\beta) &= (\Delta_{\gamma(c)} + \Delta_{\gamma(c_e)})\alpha + (\Delta_{\gamma(c)} + \Delta_{\gamma(c_e)})\beta. \end{aligned}$$

Utilizing the conservative approximate continuous reachable set $R_{[0,T]}(\bar{u})$, the continuous safety verification problem defined in Section 2 can be easily solved by splitting the interested time range $[t_1, t_2]$ and the interested position range $[x_1, x_2]$ into a finite number of segments along with the time step k and the space step h. In other words, a large continuous safety verification/falsification problem can be decomposed into a finite number of small continuous safety verification problem where the interested time range and position ranges are $t_{j-1} < t \leq t_j$ and $x_{i-1} < x \leq x_i$ respectively. Then, verifying whether or not the system violates the safety specification is solving the following problem.

$$\begin{aligned} &\textbf{Find } (\alpha,\beta,x,t) \textbf{ such that:} \\ &q_1[i,j]xt + q_2[i,j]t + q_3[i,j]x + q_4[i,j] < u_1, \\ &\textbf{or}: \\ &q_1[i,j]xt + q_2[i,j]t + q_3[i,j]x + q_4[i,j] > u_2, \\ &\textbf{subject to:} \\ &x_{i-1} < x \leq x_i,\ t_{j-1} < t \leq t_j, \\ &\alpha_1 \leq \alpha \leq \alpha_2,\ \beta_1 \leq \beta \leq \beta_2, \end{aligned} \tag{22}$$

where $[\alpha_1, \alpha_2]$ and $[\beta_1, \beta_2]$ are the given bounded ranges of the uncertainty parameters (α, β).

The Algorithm 5 describes the whole process of our approach for continuous safety verification/falsification of the parabolic equation. In the next section, we discuss deeply about the conservativeness and soundness of our approach.

6. CONSERVATIVENESS AND SOUNDNESS

In this section, we focus on the conservativeness and the soundness of our method. A continuous reachable set is said *completely conservative* if it contains *all* trajectories of the system when the error in computation using floating-point is neglected. A verification method is sound if it uses a completely conservative continuous reachable set and handles the error in computation using floating-point.

Our method uses floating-point computation and neglect its error. Thus, our method is not sound. The approximate continuous reachable set in our method for the parabolic equation (1) is also not completely conservative because we can not obtain a completely conservative error $e(x,t)$. This is an interesting and challenging problem that we are going to address in future work. However, it is worth to point out that for some specific cases such as *a stationary heat equation* described in the following, a completely conservative reachable set can be obtained.

Algorithm 5 Continuous Safety Verification/Falsification for Parabolic Equation

Input 1: Parabolic equation parameters: L, k, h, $f(x,t)$, $u_0(x)$, $[\alpha_1, \alpha_2]$, $[\beta_1, \beta_2]$ % Length of the rod, time step, space step, input function, initial condition and ranges of perturbation.
Input 2: Safety Specification: $u_1, u_2, T_1, T_2, x_1, x_2$
Output: Safe/ (Unsafe, Unsafe Trace)

```
1: procedure INITIALIZATION
2:     Compute k and h
3:     Compute mass matrix M
4:     Compute stiff matrix S
5:     Compute matrix A (14)
6:     Compute initial condition z_0
7: procedure CHECK SAFETY
8:     Construct R_[0,T2](ū) (21)
9:     Decompose [x1, x2] and [t1, t2]
10:    Solve (22)
11:    if (22) is feasible:
12:        Get feasible solution (α, β, x, t)
13:        Compute unsafe trace using (α, β, x) and (17)
14:        return Unsafe, unsafe trace
15:    else: return Safe
```

$$\begin{aligned} -u'' &= \beta f(x,t),\ 0 < x < L, \\ u(0,t) &= u(L,t) = 0, \\ u(x,0) &= \alpha u_0(x). \end{aligned} \tag{23}$$

The completely conservative reachable set can be obtained for the stationary heat equation because the fact that a completely conservative approximate error $\bar{e}(x,t)$ can be computed in this case using the following Proposition.

Proposition 6.1. The approximate continuous error $\tilde{e}(x,t)$ in computing approximate continuous solution $\tilde{u}(x,t)$ of the stationary heat equation (23) satisfies the following inequality:

$$\|e(x,t) - \tilde{e}(x,t)\|_\infty \leq h^2 \|r(\tilde{u})\|_\infty, \tag{24}$$

where $r(\tilde{u}) = \beta f(x,t) - \dot{\tilde{u}}(x,t)$.

Proof 6.1. The proof is given in the Appendix.

The above theorem shows that a completely conservative approximate error $\bar{e}(x,t)$ can be achieved by bloating the approximate error $\tilde{e}(x,t)$ by $h^2\|r(\tilde{u})\|$.

Although soundness is the ultimate goal of formal method, it is hardly achieved in practice because handling floating-point error in computation slows down significantly the performance of verification techniques and therefore, a real sound verification method is difficult to apply for a practical problem due to its small scalibility. Most of well-known tools such SpaceEx, CORA, C2E2 and Hylaa use floating-point computation without handling the floating-point error and their usefulness have been proved via many practical problems.

The main goal of our approach is to achieve a high conservative guarantee with high scalability for continuous safety verification/falsification of PDEs. Thus, we neglect the floating-point error and enhance the conservativeness by further investigating the error in computing continuous reachable set using FEM method. In the next section, we will illustrate shortly the implementation of our method and evaluate it in detail via a specific example.

7. IMPLEMENTATION AND EVALUATION

7.1 Implementation

A prototype of our method called *pdev* is available online at https://github.com/trhoangdung/pdev for convenient evaluation. We use python and its standard packages including *scipy*, *numpy*, *sympy* and *matlibplot* for computation, checking feasibility and visualization. Utilizing the symbolic computation package *sympy* allows user to easily test the method with any nonlinear input and initial condition. For efficient representation and computation, the mass matrix M, stiff matrix S are represented in sparse form. The interpolation set in space $\tilde{u}_n(x)$ and all related approximate continuous reachable sets are represented as one/two-dimensional arrays.

7.2 Evaluation

We evaluate our method using the following parameters for the parabolic equation. The rod length is $L = 10$. The input function is $f(x,t) = e^{-x-t},\ 0.2 \leq x \leq 0.4$ and the initial condition is $u_0(x) = sin(x/L)$. The ranges for the uncertainty in initial condition and input are $0.8 \leq \alpha \leq 1.1$ and $0.9 \leq \beta \leq 1.1$ respectively. The experiment is done on a computer with the following configuration: Intel Core i7-6700 CPU @ 3.4GHz × 8 Processor, 62.8 GiB Memory, 64-bit Ubuntu 16.04.3 LTS OS.

Reachability analysis. To construct the approximate continuous reachable set of the parabolic equation, the discrete reachable set of the approximate solution $\tilde{u}(x_i, t_j)$ at each mesh point and each time step is first computed. At the same time, we also compute the discrete reachable set of the corresponding approximate error $\tilde{e}(x_i, t_j)$. Fig. 1 presents the discrete reachable sets of the approximate solution $\tilde{u}(x = 8, t)$ and the approximate error $\tilde{e}(x = 8, t)$ at the position $x = 8$ with the time step $k = 0.1$ and the space step $h = 0.5$. Using these discrete reachable sets, we construct the bloated discrete reachable set of the approximate solution $\bar{u}(x_i, t_j) = \tilde{u}(x_i, t_j) + \tilde{e}(x_i, t_j)$ for all mesh points and time steps as shown in Fig. 2. From the bloated discrete reachable set, we then construct the interpolation set in space $\tilde{u}(x, t = t_j)$ as depicted in Fig. 3. Finally, the approximate continuous reachable set shown in Fig. 4 of the parabolic equation $\bar{u}(x,t)$ is constructed from $\tilde{u}(x, t = t_j)$ by implementing linear interpolation in time.

Safety Verification/Falsification. After constructing the approximate continuous reachable set, we can verify the safety of the parabolic equation. The continuous safety verification/falsification problem is equivalent to finding a feasible solution (α, β, x, t) satisfies the dual unsafe constraint (22). Assume that we want to verify whether or not the system satisfies the following safety specification:

$$0.0 \leq u(x,t) \leq 0.3,\ \forall (x,t):\ 7.2 \leq x \leq 8.3,\ 8.0 \leq t \leq 10.0$$

It should be noted that we are using the time step $k = 0.1$ and the space step $h = 0.5$. Therefore, the region that we are interested in $7.2 \leq x \leq 8.3$ is located between three mesh points $x = 7.5, x = 8.0$ and $x = 8.5$. Using Algorithm 5, we can check that the system violates its safety requirement. An unsafe trace is produced from the Algorithm for a specific point $x = 7.2$ as depicted in Fig. 5.

Fig. 1. **Approximate discrete reachable sets of the parabolic equation at the position $x = 8.0$ using time step $k = 0.1$ and space step $h = 0.5$.**

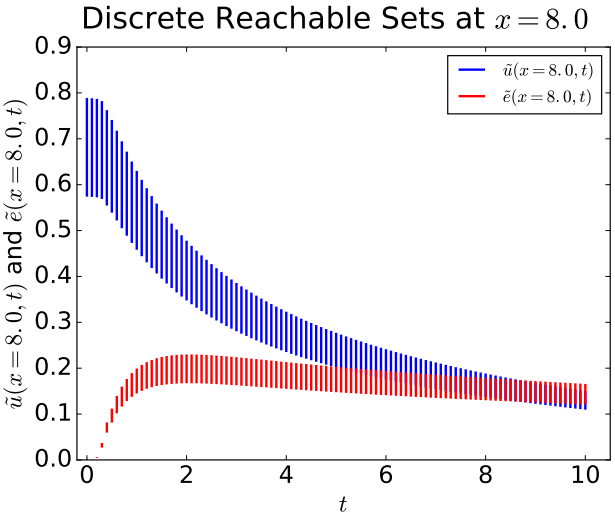
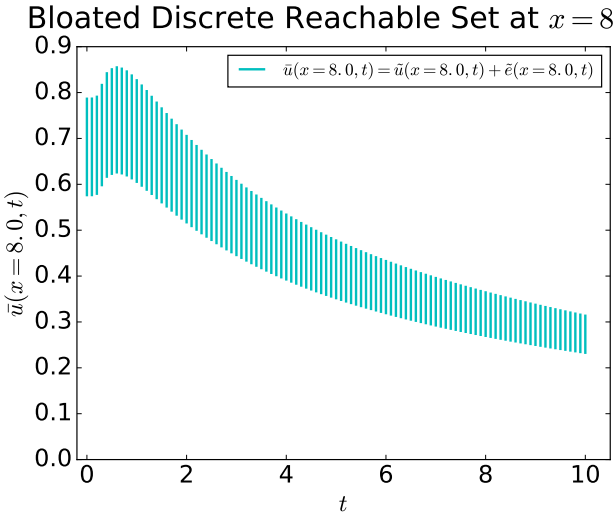


Fig. 2. **Bloated approximate discrete reachable sets of the parabolic equation at the position $x = 8.0$ using time step $k = 0.1$ and space step $h = 0.5$.**



Error analysis. Since our reachability analysis approach relies on FEM method, it is interesting and worth to know that what are appropriate time step and space step that we can use to verify the safety property of the system. This is a trade off between the accuracy and computation cost. Decreasing the space step h increases the size of the discrete system, i.e., the size of the discrete system matrix A. Similarly, reducing the time step k increases the number of discrete time steps we need to compute the discrete reachable set. Therefore, to verify the safety property with an appropriate computation cost, we need to have reasonable time step k and space step h. Fig. 6 illustrates how the approximate error caused by using FEM method varies with different space steps. It can be seen that, reducing space step can help to produce a tighter/better reachable set of the approximate error and consequently, a tighter/better reachable set of the system can be constructed.

Fig. 3. **Bloated approximate continuous (in space) reachable sets of the parabolic equation at the time $t = 10s$ using time step $k = 0.1$ and space step $h = 0.5$.**

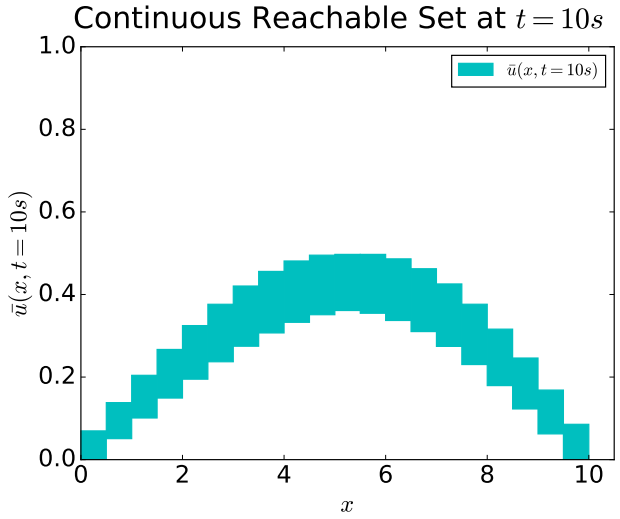
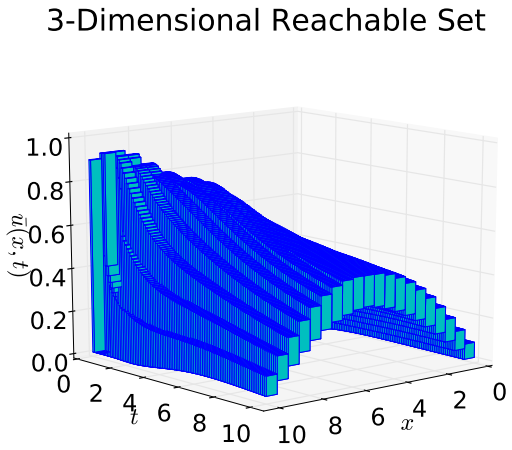


Fig. 4. **Bloated approximate continuous reachable sets of the parabolic equation for all t in $[0, 10s]$ using time step $k = 0.1$ and space step $h = 0.5$.**



It can be seen in Fig. 7 that the approximate discrete reachable set of the error at the specific mesh point $x = 5$ using the time step $k = 0.1$ is contained inside the one with $k = 0.01$. The figure shows that reducing the time step does not help much in reducing the approximate error $\tilde{e}$. This may be caused from the fact that we are using *space-time* FEM method which is numerically stable regardless any selected time step. This is an interesting point that should be investigated further to choose a good time step for seeking an appropriate computation cost.

Computation complexity. A theoretical complexity of our reachability analysis approach can be easily derived by analyzing the time and memory complexity for each step of the whole process. One can see that the time complexity for computing the discrete reachable set is $\mathcal{O}(nm^2)$, where n is the number of time steps and m is the number of mesh points. Constructing the interpolation set in space and the continuous reachable set has the time complexity of $\mathcal{O}(nm)$. Therefore, the total time complexity of our reachability analysis approach is $\mathcal{O}(nm^2)$. It is also easy to see that the memory complexity of our approach is $\mathcal{O}(m^2 + nm)$, where $\mathcal{O}(m^2)$ is for storing the matrix A and $\mathcal{O}(nm)$ is for storing the interpolation set in space and the continuous reachable set. Table. 1 shows the computation time of our reachability analysis approach for *one time step* with different number of mesh points. Table. 2 presents the computation time versus the number of time steps where the number of mesh points is fixed. It is easy to see from the table that the computation time depends linearly on the number of time steps.

Fig. 5. **A trace to show that the parabolic equation violates its safety specification.**

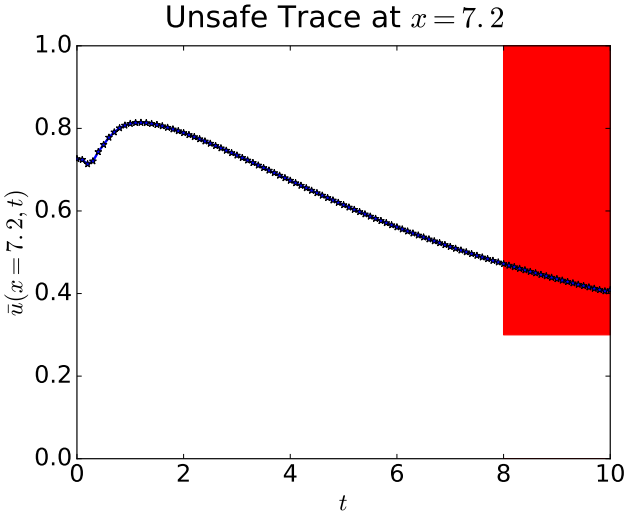


Fig. 6. **Approximate continuous reachable set of the error at time $t = 10s$ with different space steps.**

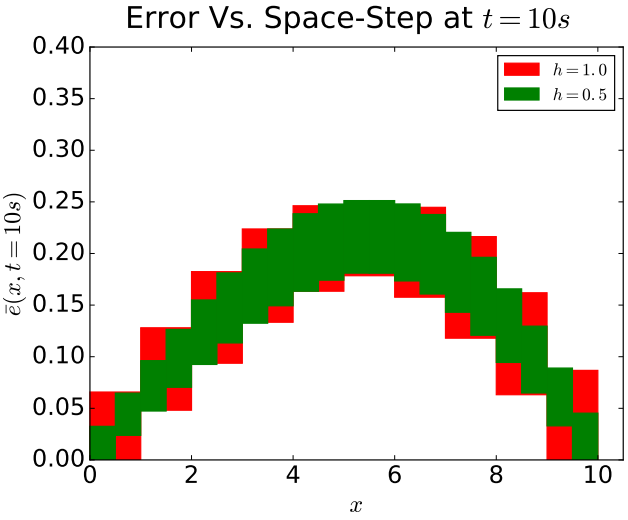


Table 1. **Computation time for one time step using different number mesh points.**

$m = 10$	$m = 20$	$m = 40$	$m = 80$	$m = 100$
$2.75s$	$5.79s$	$11.95s$	$25.1s$	$33.1s$

8. CONCLUSION

A reachability analysis approach for linear parabolic equation is proposed based on the well-known FEM method.

Fig. 7. **Approximate discrete reachable set of the error at time $x = 5$ with different time steps k.**

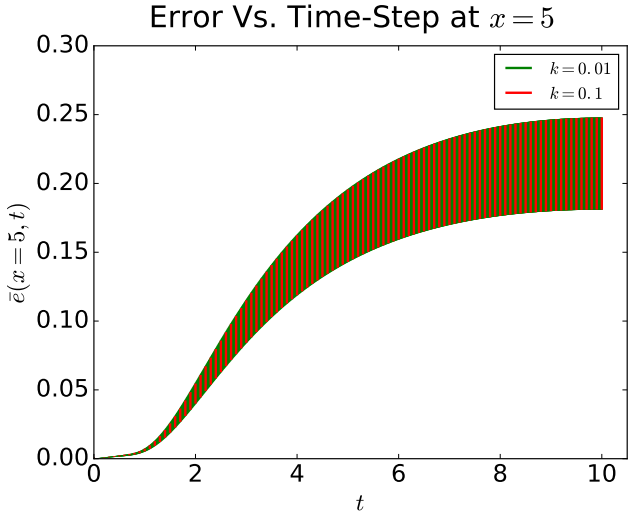


Table 2. **Computation time versus the number of time steps using fixed number mesh points ($m = 20$).**

$n = 50$	$n = 100$	$n = 200$	$n = 1000$	$n = 2000$
$278.5s$	$548.2s$	$1110.04s$	$6136.4s$	$14718.6s$

The conservativeness of our method is enhanced by utilizing the error caused by FEM method to obtain a *bloated* continuous reachable set before using it to check the system safety. The evaluation section has shown that our method is practically applicable where the safety verification/falsification problem can be solved efficiently with an appropriate computation cost since we do not need to choose a too small time step or space step to achieve our goal. Another good point is that the time complexity of our method is smaller than the traditional reachability analysis methods because our approach is simulation-equivalent.

Achieving a complete conservativeness of the proposed approach is the main goal of our future work beside extending it for different classes/high-dimensional PDEs with different types of boundary conditions.

9. APPENDIX

9.1 Proof for Theorem 4.1

For any $x \in (x_{i-1}, x_i]$ and $t \in (t_{j-1}, t_j]$, we have:

$$e(x,t) = e(x,t_{j-1}) + \int_{t_{j-1}}^{t} \dot{e}(x,t)dt.$$

Thus,

$$\begin{aligned} |e(x,t)| &\leq |e(x,t_{j-1})| + \int_{t_{j-1}}^{t} |\dot{e}(x,t)|dt, \\ &\leq |e(x,t_{j-1})| + k \cdot \sup |\dot{e}(x,t)|_{t\in(t_{j-1},t_j]}. \end{aligned} \tag{25}$$

Similarly, we also have:

$$\begin{aligned} |e(x,t)| &\leq |e(x_{i-1},t| + \int_{x_{i-1}}^{x} |e'(x,t)|dx, \\ &\leq |e(x_{i-1},t)| + h \cdot \sup |e'(x,t)|_{x\in(x_{i-1},x_i]}. \end{aligned} \tag{26}$$

Using (25) for $e(x_{i-1}, t)$ yields:

$$|e(x_{i-1},t)| \leq |e(x_{i-1},t_{j-1})| + k \cdot \sup |\dot{e}(x,t)|_{t\in(t_{j-1},t_j]}. \tag{27}$$

Combining (26) and (27) leads to:

$$\begin{aligned} |e(x,t)| \leq & |e(x_{i-1},t_{j-1})| + k \cdot \sup |\dot{e}(x,t)|_{t\in(t_{j-1},t_j]} + \\ & h \cdot \sup |e'(x,t)|_{x\in(x_{i-1},x_i]}. \end{aligned}$$

The proof is completed.

9.2 Proof for Proposition 6.1

This Proposition can be derived easily based on the fact of using FEM method Larson and Bengzon (2013) that:

$$\|e - \tilde{e}\|_\infty \leq h^2 \|e''\|_\infty = h^2 \left\|f - \dot{\tilde{u}}\right\|_\infty = h^2 \|r(\tilde{u})\|_\infty.$$

REFERENCES

Althoff, M. (2015). An introduction to cora 2015. In *Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems*.

Bak, S. and Duggirala, P.S. (2017a). Hylaa: A tool for computing simulation-equivalent reachability for linear systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, 173–178. ACM.

Bak, S. and Duggirala, P.S. (2017b). Simulation-equivalent reachability of large linear systems with inputs. In *International Conference on Computer Aided Verification*, 401–420. Springer.

Chen, X., Ábrahám, E., and Sankaranarayanan, S. (2013). Flow*: An analyzer for non-linear hybrid systems. In *International Conference on Computer Aided Verification*, 258–263. Springer.

Chou, Y., Chen, X., and Sankaranarayanan, S. (2017). A study of model-order reduction techniques for verification. In *International Workshop on Numerical Software Verification*, 98–113. Springer.

Duggirala, P.S., Mitra, S., Viswanathan, M., and Potok, M. (2015). C2E2: a verification tool for stateflow models. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 68–82. Springer.

Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., and Maler, O. (2011). Spaceex: Scalable verification of hybrid systems. In *Computer Aided Verification*, 379–395. Springer.

Han, Z. and Krogh, B.H. (2006). Reachability analysis of large-scale affine systems using low-dimensional polytopes. In *Hybrid Systems: Computation and Control*, 287–301. Springer.

Larson, M.G. and Bengzon, F. (2013). *The finite element method: Theory, implementation, and applications*, volume 10. Springer Science & Business Media.

Tran, H.D., Nguyen, L.V., Xiang, W., and Johnson, T.T. (2017). Order-reduction abstractions for safety verification of high-dimensional linear systems. *Discrete Event Dynamic Systems*, 27(2), 443–461.