Reachability Analysis for One Dimensional Linear Parabolic Equations

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

* *Vanderbilt University, TN 37023, USA.*
** *Air Force Research Laboratory, USA.*

Abstract: Partial differential equations (PDEs) 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 methods for solving PDEs, from a safety verification (or falsification) perspective, methods are still needed to verify (or falsify) a system whose dynamics is specified as a PDE 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 Galerkin Finite Element Method (FEM) for a class of one-dimensional linear parabolic PDEs with *fixed but uncertain* inputs and initial conditions, which is a subclass of PDEs that is useful for modeling, for instance, heat flows. In particular, a *continuous approximate* reachable set of the parabolic PDE is computed using linear interpolation. Since a *complete conservativeness* is hardly achieved by using the approximate reachable set, to *enhance* the conservativeness, we investigate the error bound between the numerical solution and the exact analytically unsolvable solution to bloat the continuous approximate reachable set. This bloated reachable set is then used for safety verification and falsification. 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 lead the system to an unsafe state.

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

1. INTRODUCTION

Reachability analysis is the fundamental problem in safety verification of cyber-physical systems. Over the last two decades, numerous techniques and tools have been proposed for continuous and hybrid systems whose dynamics are described by linear or nonlinear ordinary differential equations (ODEs). Reachability analysis using zonotopes/support functions has been demonstrated to be the most efficient and scalable approach that can verify linear continuous and hybrid systems with up to hundreds of state variables Frehse et al. (2011); Althoff (2015). For the nonlinear case, Flow* Chen et al. (2013) utilizing Taylor model is well-known. Recently, the order-reduction abstraction Tran et al. (2017); Chou et al. (2017) and simulation-based reachability analysis Bak and Duggirala (2017b,a) have exhibited an ability to tackle the most challenging problem in reachability analysis named *state space explosion*.

Although significant works in reachablity analysis have been done for continuous and hybrid systems with ODE dynamics, little attention has been paid to the class of systems with PDE dynamics that appear in many science and engineering problems such as fluid dynamics control, heat equation and quantum mechanics. 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 a recent reachability analysis approach dealing with large scale linear systems Han and Krogh (2006). In this context, the heat equation was transformed into a continuous ODE model using a finite difference method and the safety specification of interest was given for *discrete* mesh points. It is also worth noting that the input of the heat equation benchmark was assumed to be a constant with a small *time-invariant uncertainty* while the initial states of the mesh points was represented as a bounded box.

Although the heat equation has been demonstrated to be a good benchmark for accessing the scalability of verification techniques, a deeper study should be done

⋆ The material presented in this paper is based upon work supported by the National Science Foundation (NSF) under grant numbers CNS 1464311, CNS 1713253, SHF 1527398, and SHF 1736323, the Air Force Research Laboratory (AFRL) through the AFRLs Summer of Innovation (SoI) Program under contract FA8650-12-3-7255 via subcontract number WBSC 7255 SOI VU 0001, and the Air Force Office of Scientific Research (AFOSR) through contract numbers FA9550-15-1-0258, FA9550-16-1-0246, and FA9550-18-1-0122. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of AFRL, AFOSR, or NSF.

for two reasons. Firstly, it is reasonable to have a safety specification concerned with a region in space and not only concentrated at specific mesh points. In other worlds, we are interested in *continuous-space* and not *discrete-space* reachability analysis. Second, it is crucial to have an approach that works for more general types of inputs and initial conditions, i.e., an input described by a nonlinear function in both time and space, and an initial condition defined by a nonlinear spatial function.

In this paper, we propose a *continuous reachability analysis* approach for linear parabolic equations with time-invariant uncertain nonlinear inputs and initial conditions. By "continuous", we mean that the continuity in both space and time are investigated. Our main contributions are : 1) an extension of the well-known space-time Galerkin method and linear interpolation into a continuous reachability analysis approach for one dimensional parabolic equation; 2) enhancing the conservativeness of the proposed method by investigating and utilizing the error between the numerical solution and the exact analytically unsolvable solution; 3) providing an implementation the proposed method in a prototype called *pdev*, which is available online for further experimentation and evaluation.

2. PROBLEM FORMULATION

The mathematical description of a one-dimensional linear parabolic equation with time-invariant uncertain nonlinear inputs and initial conditions is given as follows:

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

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

The equation $u(0,t) = u(L,t) = 0$ describes the *Dirichlet* boundary condition used in our problem. For simplicity, we define $\alpha = (1+\epsilon_2)$ and $\beta = (1+\epsilon_1)$ and use them as time-invariant uncertainty parameters for the rest of the paper. We also use $\dot{u}$ to denote $\frac{\partial u}{\partial t}$, u' denotes $\frac{\partial u}{\partial x}$, and u'' designates $\frac{\partial^2 u}{\partial x^2}$. The reachable set and safety verification problem for this class of system are defined as follows.

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

$$R_{[0,T]}(u) = \{\bigcup_{x,t} u(x,t) |\ u(x,t)\ \textit{satisfies}\ (1),\ 0 \le t \le T \le \infty\}.$$

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 $0 \le T_1 \le t \le T < \infty$ where $u_1, u_2, x_1, x_2, T_1, T$ are scalars, the system (1) is called safe if and only if the continuous reachable set of $u(x,t)$ of the system in the time range $[T_1, T]$ denoted by $R_{[T_1,T]}(u) \subseteq R_{[0,T]}(u)$ satisfies the safety specification.

To verify the safety of the system, the continuous reachable set $R_{[0,T]}(u)$ needs 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 numerically using FEM and linear interpolation. Thus, instead of constructing the *exact* continuous reachable set $R_{[0,T]}(u)$, we can only construct an *approximate* continuous reachable set $R_{[0,T]}(\tilde{u})$ for the system. By doing this, the ineluctable approximation error $e(x,t) = u(x,t) - \tilde{u}(x,t)$ needs to be taken into account. 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 the core steps used to obtain the approximate continuous reachable set of a parabolic equation by leveraging the well-known space-time Galerkin FEM and linear interpolation. We refer readers to Larson and Bengzon (2013) for further detail.

3.1 Approximate discrete reachable set computation

The Galerkin FEM is a powerful tool for approximating the solution of PDEs at mesh points and time steps. The general idea is that, the space of interest $[0, L]$ is discretized by a list of mesh points $x = [0 = x_0, x_1, x_2, \cdots, x_{m-1}, x_m = L]$, and 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 simplicity, we use a uniform time step $k = T/n, t_j = j \times k, 0 \le j \le n$ and a uniform space step $h = L/m, x_i = i \times h, 0 \le i \le m$. 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 both sides of the first equation in (1) by a *test function* v and integrating by parts.

$$\int_{I_n}\int_{\Omega} (\dot{u} - u'')v dx dt = \int_{I_n}\int_{\Omega} \beta f(x,t) v dx dt, \tag{2}$$

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:

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

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,

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

where

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

and

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

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

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

One can 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) - (7), the *discrete* approximate model of the system (1) can be derived as follows.

$$\tilde{u}_n = A\tilde{u}_{n-1} + \beta g_n, \quad (8)$$

where:

$$\tilde{u}_n = [\tilde{u}_{n,1}, \tilde{u}_{n,2} \cdots, \tilde{u}_{n,m-1}]^T,$$
$$A = (M + \frac{k}{2}S)^{-1}(M - \frac{k}{2}S) and g_n = (M + \frac{k}{2}S)^{-1}\bar{g}_n,$$
$$M \in \mathbb{R}^{(m-1)\times(m-1)}, M = [\int_{\Omega}\phi_i\phi_j]_{i,j},$$
$$S \in \mathbb{R}^{(m-1)\times(m-1)}, S = [\int_{\Omega}\phi_i'\phi_j']_{i,j},$$
$$\bar{g}_n = [\bar{g}_{n,1}, \bar{g}_{n,2}, \cdots, \bar{g}_{n,m-1}]^T, \bar{g}_{n,i} = \int_{I_n}\int_{\Omega} f(x,t)\phi_i dxdt.$$

M, S and $\bar{g}_n$ are called mass matrix, stiff matrix and load vector of the system (1) respectively. It should be clarified that we use the notation $\tilde{u}_n$, a scalar vector, to represent the approximate solution $\tilde{u}(x,t)$ at a specific time step $t = t_n$ and at all of mesh points x_i. Thus, $\tilde{u}_n$ is a *discrete approximate solution* of the system (1) at $t = t_n$. Bear in mind that this is different from the notation $\tilde{u}_n(x)$ which is a continuous function of the position x.

The approximate discrete reachable set of system (1) is obtained below from (8):

$$\tilde{\mathbf{u}}_\mathbf{n} = \alpha\mathbf{z}_\mathbf{n} + \beta\mathbf{l}_\mathbf{n}, \quad (9)$$

where:

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

3.2 Approximate continuous reachable set computation

Using linear interpolation, the approximate continuous reachable set of the system (1), $R_{[0,T]}(\tilde{u})$, can be constructed from the approximate discrete reachable set derived previously in two steps. The first step is the construction of the linear interpolation set in space from the approximate discrete reachable set using (5) and (6). The second step constructs the approximate continuous reachable set using (3) and the interpolation set in space obtained in the first step.

Linear interpolation set in space. From (5), (6) and (9), 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.

$$\tilde{\mathbf{u}}_\mathbf{n}(\mathbf{x}) = (\mathbf{a}_\mathbf{n}\alpha + \mathbf{b}_\mathbf{n}\beta)\mathbf{x} + \mathbf{c}_\mathbf{n}\alpha + \mathbf{d}_\mathbf{n}\beta, \quad (10)$$

where $\tilde{u}_n(x)$ is represented as a 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 depend on x as follows.

For $\mathbf{0 < x \le x_1}$:
$a_n[1] = z_n[1]/h,\ b_n[1] = l_n[1]/h,\ c_n[1] = 0,\ d_n[1] = 0.$
For $\mathbf{x_{i-1} < x \le x_i, 1 < i \le m-1}$:
$a_n[i] = (z_n[i] - z_n[i-1])/h,\ b_n[i] = (l_n[i] - l_n[i-1])/h,$
$c_n[i] = i \cdot z_n[i-1] - (i-1) \cdot z_n[i],$
$d_n[i] = i \cdot l_n[i-1] - (i-1) \cdot l_n[i].$
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. Using (4) and the spatial interpolation set computed in the previous step, the approximate continuous reachable set is obtained as a function of the time variable t, the position variable x and the uncertainty parameters (α, β) as follows.

$$\begin{aligned} R_{[0,T]}[\tilde{u}] = \{\bigcup_{x,t}\tilde{u}(x,t) \mid \tilde{u}(x,t) = (1/k)(\Delta_a\alpha + \Delta_b\beta)xt + \\ (\Delta_c\alpha + \Delta_d\beta)t/k + (\Delta_{\gamma(a)}\alpha + (\Delta_{\gamma(b)}\beta)x \\ ((\Delta_{\gamma(c)}\alpha + (\Delta_{\gamma(d)}\beta)\}, \end{aligned} \quad (11)$$

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 $\Delta_a, \Delta_b, \Delta_{\gamma(a)}, \Delta_{\gamma(b)}, \Delta_{\gamma(c)}, \Delta_{\gamma(d)} \in \mathbb{R}^{m\times n}$ defined below.

$$\begin{aligned} &\textbf{For } \mathbf{1 \le j \le n} : \\ &\Delta_a.column[j] = a_{j-1} - a_j, \\ &\Delta_b.column[j] = b_{j-1} - b_j, \\ &\Delta_c.column[j] = c_{j-1} - c_j, \\ &\Delta_d.column[j] = d_{j-1} - d_j, \\ &\Delta_{\gamma(a)}.column[j] = j \cdot a_j - (j-1) \cdot a_{j-1}, \\ &\Delta_{\gamma(b)}.column[j] = j \cdot b_j - (j-1) \cdot b_{j-1}, \\ &\Delta_{\gamma(c)}.column[j] = j \cdot c_j - (j-1) \cdot c_{j-1}, \\ &\Delta_{\gamma(d)}.column[j] = j \cdot d_j - (j-1) \cdot d_{j-1}, \\ &a_0 = b_0 = d_0 = [0,0,\cdots,0]^T,\ c_0 = z_0. \end{aligned} \quad (12)$$

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 using 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 the Galerkin FEM.

4. ERROR ANALYSIS

It is important to emphasize that the exact solution for the error $e(x,t)$ is also analytically unsolvable in general. Thus, the only way to deal with this reality is to again approximate this error.

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

$$\begin{aligned} &\dot{e} - e'' = \beta f - \dot{\tilde{u}} = r(\tilde{u}),\ 0 < x < L, \\ &e(0,t) = e(L,t) = 0, \\ &e(x,0) = 0. \end{aligned} \quad (13)$$

Using Galerkin FEM 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.

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

5. CONTINUOUS SAFETY VERIFICATION/FALSIFICATION

The previous two sections focused on the computation of the approximate continuous reachable set $R_{[0,T]}(\bar{u})$ of system (1) and the corresponding approximate continuous error reachable set $R_{[0,T]}(\tilde{e})$. Combining these two reachable sets, a more conservative approximate continuous reachable set $R_{[0,T]}(\bar{u})$ of system (1) can be obtained as follows.

$$R_{[0,T]}(\bar{u}) = \{\bigcup_{x,t} \bar{u}(x,t) |\ \bar{u}(x,t) = \tilde{u}(x,t) + \tilde{e}(x,t) = q_1(\alpha,\beta)xt + q_2(\alpha,\beta)t + q_3(\alpha,\beta)x + q_4(\alpha,\beta)\}, \quad (14)$$

where:

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

Utilizing the conservative approximate continuous reachable set $R_{[0,T]}(\bar{u})$, the continuous safety verification problem defined in Section 2 can be solved by splitting the time range $[T_1, T]$ and position range $[x_1, x_2]$ of interest into a finite number of segments with time step k and 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 problems where the time 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.

$$\begin{aligned}
&\textbf{Find } (\alpha,\beta,x,t) \textbf{ such that:}\\
&q_1[i,j]xt + q_2[i,j]t + q_3[i,j]x + q_4[i,j] < u_1,\\
&\textbf{or}:\\
&q_1[i,j]xt + q_2[i,j]t + q_3[i,j]x + q_4[i,j] > u_2, \quad (15)\\
&\textbf{subject to:}\\
&x_{i-1} < x \leq x_i,\ t_{j-1} < t \leq t_j,\\
&\alpha_1 \leq \alpha \leq \alpha_2,\ \beta_1 \leq \beta \leq \beta_2,
\end{aligned}$$

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

Algorithm 5 describes the whole process of our approach. In the next section, we discuss the conservativeness and soundness of our approach.

6. CONSERVATIVENESS AND SOUNDNESS

A continuous reachable set is said to be *completely conservative* if it contains *all* trajectories of the system while neglecting the error introduced by using floating-point computation. A verification method is sound if it uses a completely conservative continuous reachable set and handles the error in computation using floating-point.

Algorithm 5 Continuous Safety Verification/Falsification for Parabolic Equation

Input 1: L, k, h, $f(x,t)$, $u_0(x)$, $[\alpha_1, \alpha_2]$, $[\beta_1, \beta_2]$
Input 2: $u_1, u_2, T_1, T_2, x_1, x_2$ % Specification
Output: Safe/ (Unsafe, Unsafe Trace)

```
procedure INITIALIZATION
    Compute mass matrix M, stiff matrix S, A, z_0.
procedure CHECK SAFETY
    Construct R_[0,T2](ū) (14).
    Decompose [x_1, x_2] and [t_1, t_2].
    if (15) is feasible:
        Get feasible solution (α, β, x, t).
        Compute unsafe trace using (α, β, x) and (11).
        return Unsafe, unsafe trace.
    else: return Safe.
```

Our method uses floating-point computation without handling the error. Thus, our method is not sound. Additionally, the approximate continuous reachable set in our method is not completely conservative because a completely conservative error $e(x,t)$ is unobtainable. This is the most challenging problem that we are going to address in the future work. However, it is worth noting that in some specific cases such as *the stationary heat equation*, a completely conservative reachable set can be obtained.

The main goal of our approach is to achieve a high conservative guarantee and a 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 the continuous reachable set. 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

Our method is implemented in a prototype named *pdev* [1] written in *python*. We use the following parameters for evaluation. 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 initial condition and input uncertainties are $0.8 \leq \alpha \leq 1.1$ and $0.9 \leq \beta \leq 1.1$ respectively. All experiments are 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 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 position $x = 8$ with time step $k = 0.1$ and space step $h = 0.5$. Using these discrete reachable sets, we construct the bloated discrete reachable set of the

[1] https://github.com/trhoangdung/pdev

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

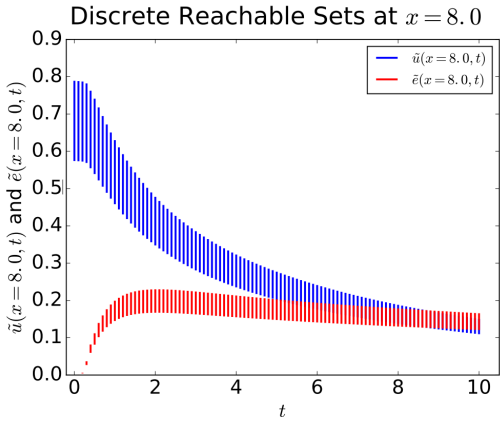


Fig. 2. **Bloated approximate discrete reachable sets of the parabolic equation at position $x = 8.0$ using time step $k = 0.1$ and space step $h = 0.5$.**

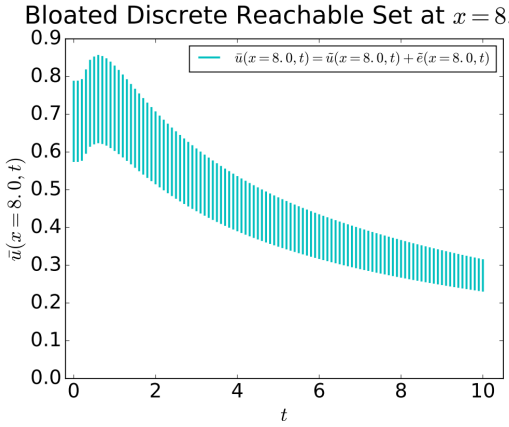
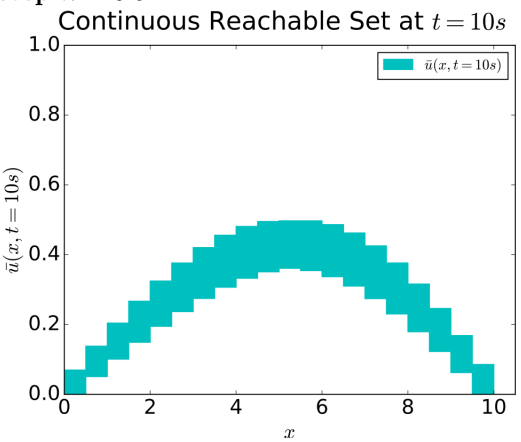


Fig. 3. **Bloated approximate continuous (in space) reachable sets of the parabolic equation at the time $t = 10s$ using time step $k = 0.1$ and space step $h = 0.5$.**



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 $\bar{u}(x, t = t_j)$ 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 $\bar{u}(x, t = t_j)$ by implementing linear interpolation in time.

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 $h = 0.5$.**

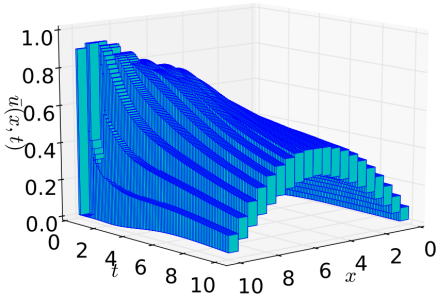
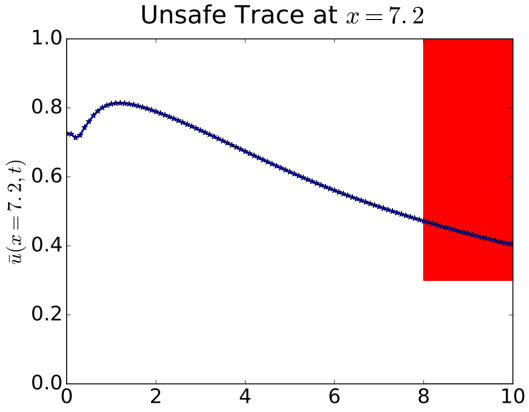


Fig. 5. **A trace showing that the parabolic equation violates its safety specification.**



Safety Verification/Falsification. The continuous safety verification/falsification problem is equivalent to finding a feasible solution (α, β, x, t) in the reachable set that satisfies the dual unsafe constraint (15). 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 time step $k = 0.1$ and space step $h = 0.5$. Therefore, the region of interest is located between three mesh points: $x = 7.5, x = 8.0$ and $x = 8.5$. Using Algorithm 5, we can find that the system violates its safety requirement. An unsafe trace depicted in Fig. 5 is produced from the Algorithm for the specific point $x = 7.2$.

Error analysis. It is important to consider the appropriate time and space steps needed to verify the safety property of the system. This is a trade off between an increase in accuracy while cognizant of computation cost. Decreasing the space step h increases the size of the discrete system, i.e., the size of the matrix A. Similarly, reducing the time step k increases the number of discrete time steps used in computing the discrete reachable set. Therefore, in order to verify the safety property with an appropriate computation cost, reasonable time step k and space step h are needed. Fig. 6 illustrates how the approximate error 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

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

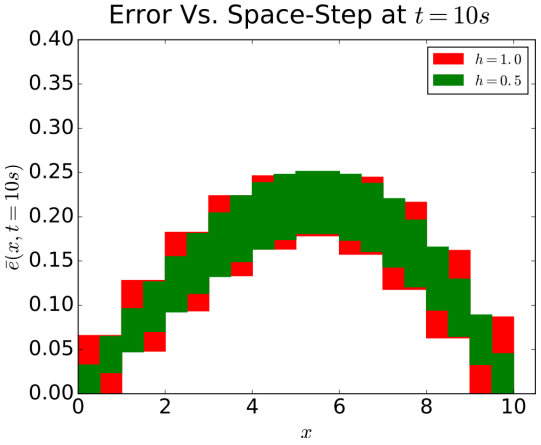


Fig. 7. **Approximate discrete reachable set of the error at time $x = 5$ with different time steps k.**

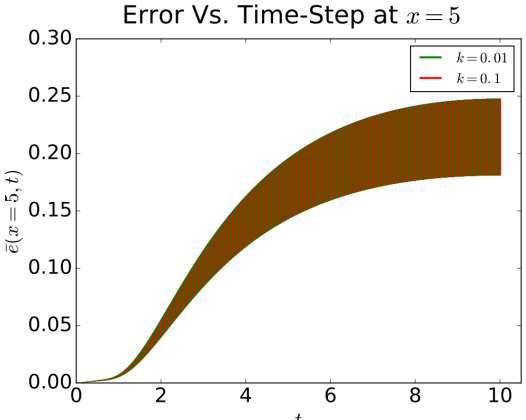


Table 1. **Computation time for one time step using different number mesh points.**

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

and consequently, a tighter/better reachable set of the system can be constructed. However, reducing the time step does not help much in reducing the approximate error $\tilde{e}$ as shown in Fig. 7. This recalls the fact that the *time-space* Galerkin FEM is numerically stable regardless of any choices of time step k.

Computation complexity. 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 time complexity $\mathcal{O}(nm)$. Therefore, the total time complexity of our reachability analysis approach is $\mathcal{O}(nm^2)$. 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 method for *one time step* with various numbers of mesh points. Table. 2 presents the computation time versus the number of time steps where the number of mesh points is fixed. The table shows that the computation time depends linearly on the number of time steps.

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

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

8. CONCLUSION

In this paper, a reachability analysis approach for linear parabolic equation is proposed based on the well-known Galerkin FEM. The conservativeness of our method is enhanced by utilizing the error caused by the Galerkin FEM to obtain a *bloated* continuous reachable set before using it to check the safety of a system. 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. Moreover, 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 to different classes/high-dimensional PDEs with different types of boundary conditions.

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.

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.