



Combining set propagation with finite element methods for time integration in transient solid mechanics problems

Marcelo Forets^{a,*}, Daniel Freire Caporale^b, Jorge M. Pérez Zerpa^c

^aDepartamento de Matemática y Aplicaciones, Centro Universitario Regional del Este, Universidad de la República, Maldonado, Uruguay

^bInstituto de Física, Facultad de Ciencias, Universidad de la República, Montevideo, Uruguay

^cInstituto de Estructuras y Transporte, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

ARTICLE INFO

Article history:

Received 12 May 2021

Accepted 12 October 2021

Available online 8 November 2021

Keywords:

Reachability analysis

Finite element method

Heat transfer

Structural dynamics

Numerical verification

ABSTRACT

The Finite Element Method (FEM) is the gold standard for spatial discretization in numerical simulations for a wide spectrum of real-world engineering problems. Prototypical areas of interest include linear heat transfer and linear structural dynamics problems modeled with partial differential equations (PDEs). While different algorithms for direct integration of the equations of motion exist, exploring all feasible behaviors for varying loads, initial states and fluxes in models with large numbers of degrees of freedom remains a challenging task. In this article we propose a novel approach, based in set propagation methods and motivated by recent advances in the field of Reachability Analysis. Assuming a set of initial states and inputs, the proposed method consists in the construction of a union of sets (flowpipe) that enclose the infinite number of solutions of the spatially discretized PDE. We present the numerical results obtained in five examples to illustrate the capabilities of our approach, and compare its performance against reference numerical integration methods. We conclude that, for problems with single known initial conditions, the proposed method is accurate. For problems with uncertain initial conditions included in sets, the proposed method can compute all the solutions of the system more efficiently than numerical integration methods.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Transient problems are of great relevance in a large and diverse set of disciplines in Engineering. In structural dynamics, problems with time-varying loads are formulated in applications such as wind turbine modeling [50] or the study of nonlinear behavior of cables [40]. In heat transfer analysis, transient problems arise in applications ranging from designing massive concrete structures [49,47], to simulating the effects of COVID-19 on the respiratory tract [26].

The governing equations of the mathematical models arising in transient problems are numerically solved considering both a discretization in space and time. The Finite Element Method (FEM), emerged in the second half of the 20th century [51], has become the gold-standard approach for spatial discretization of PDEs. Regarding time integration, methods keep being actively developed [11,33,39,38,32], highlighting the challenge this represents.

In addition, all mathematical models developed and applied to any real-world problem are subject to uncertainties. The material

properties, the loads applied, the geometries, the initial conditions, etc., must be provided as an input of the models, and are always submitted to errors. Hence, the effect of uncertainties over simulation results must be quantified. In the last decades, several methods have been developed using different mathematical and numerical techniques, which can be classified into probabilistic and non-probabilistic approaches. The main probabilistic approaches are Uncertainty Propagation and Monte Carlo methods. The former is based on the calculation of probability density functions for sampling statistics and has been successfully applied in structural systems under dynamic loads [17,31,29]. Monte Carlo methods rely on repeated numerical simulations starting from different initial conditions and parameter choices [45]. Non-probabilistic approaches include interval methods [46,42], and fuzzy logic methods [41,27].

The most common practice to solve transient problems relies on exploring different behaviors by simulation and testing. However, if uncertainty is present and the application requires an exhaustive exploration of the state space, such approach becomes computationally intractable. For instance, to guarantee that a certain maximum temperature in a massive concrete structure is not exceeded [47] can be interpreted as a *verification problem*. A prominent

* Corresponding author at: Departamento de Matemática y Aplicaciones, Centro Universitario Regional del Este, Universidad de la República, Maldonado, Uruguay.
E-mail address: mforets@fing.edu.uy (M. Forets).

framework used to solve verification problems consists in approximating the set of states that are reachable by a dynamical system, from all initial states and for all admissible inputs and parameters. Such approach is called *reachability analysis* and it has been recently applied to solve verification, control, path planning and stability problems in diverse domains [6]. One of the main approaches to reachability analysis relies on *set propagation*, where the solution of an ordinary differential equation (ODE) is expressed in terms of sets rather than numbers. Several algorithms have been devised for linear [25,36,5,15] and non-linear [28,8,7,18] differential equations. These algorithms are implemented in numerical tools such as SpaceEx [21], Flowstar [18], CORA [2] or JuliaReach [20]. Despite the existence of numerous algorithms, their successful application requires expert knowledge about algorithm tuning [48].

The main motivation of our work is to integrate reachability analysis and finite element methods in structural analysis applications. Recent specific works have initiated research in such direction, as in [44], where the authors use ellipsoidal set representations. This approach does not scale for systems with many degrees of freedom. In [9] the authors develop a set propagation method based on linear programming that is applied to a heat diffusion problem with a large (1000^3) mesh obtained using finite differences. In comparison with our technique, theirs' returns reach-set approximations at time points, instead of reach-sets at time intervals as in our method. In [3] the author extended the theoretical basis of set propagation to handle large linear systems with Krylov subspace approximations of the state transition matrix acting on zonotopes. Their results are illustrated on the finite element model of a bridge subject to set-valued wind forces.

In this article we present a novel framework for the resolution of linear transient dynamics problems in solid mechanics, that builds upon set propagation techniques. Starting from the assembled FEM system of differential equations, our method is based on the construction of a set including the solutions for the first time step and propagating the set to obtain an enclosure of the solutions for posterior times. The methodology was implemented using JuliaReach¹, a state-of-the-art open source library for reachability analysis. Finite-element matrices assembly was done using the open source library ONSAS², developed for GNU-Octave/Matlab.

This article is organized as follows. In Section 2 the basic concepts of the Finite Element Method and Reachability Analysis are described, including an illustrative example complementing the presentation of the latter. In Section 3 the key components of the proposed methodology are developed. In Section 4 the results obtained considering five numerical examples (including reference examples from the literature and a realistic Engineering problem) are shown. Finally, in Section 5 the conclusions are exposed and additional proofs are available as part of the Appendix.

2. Preliminaries

2.1. Transient problems governing equations

In this section we recall the basic equations of the heat transfer and structural dynamics problems, and omit their development which can be found in the literature [49,30,10].

For the heat transfer problem, considering an isotropic solid in the region Ω with conductivity κ , density ρ and specific heat c , the governing equations are

$$\mathbf{C}_\theta \dot{\theta}(t) + \mathbf{K}_\theta \theta(t) = \mathbf{f}_\theta(t), \quad (1)$$

where the matrix $\mathbf{K}_\theta$ is the assembled convection matrix

$$\mathbf{K}_\theta = \mathbb{A}_{e=1}^{n_e} \mathbf{K}_\theta^e, \quad \mathbf{K}_\theta^e = \int_{\Omega^e} \kappa (\mathbf{B}^e)^T \mathbf{B}^e dV + \int_{\Gamma_R^e} h (\mathbf{N}^e)^T \mathbf{N}^e dS, \quad (2)$$

with n_e being the number of elements and $\mathbf{N}$ and $\mathbf{B}$ the matrices of interpolation functions and their derivatives, respectively, the matrix $\mathbf{C}_\theta$ is the assembled diffusion matrix

$$\mathbf{C}_\theta = \mathbb{A}_{e=1}^{n_e} \mathbf{C}_\theta^e, \quad \mathbf{C}_\theta^e = \int_{\Omega^e} \rho c (\mathbf{N}^e)^T \mathbf{N}^e dV, \quad (3)$$

and the vector of heat sources $\mathbf{f}_\theta(t)$ is given by

$$\begin{aligned} \mathbf{f}_\theta(t) = & \mathbb{A}_{e=1}^{n_e} \int_{\Omega^e} Q_{int}(t) \mathbf{N}^e dV + \int_{\Gamma_N^e} q_{inp}(t) \mathbf{N}^e dS \\ & + \int_{\Gamma_R^e} h \theta_\infty(t) \mathbf{N}^e dS, \end{aligned} \quad (4)$$

with Q_{int} and q_{inp} being volumetric and boundary external heat sources, h being a convection constant and θ_∞ being the external temperature.

For the structural dynamics problem, considering a solid with density ρ and linear elastic constitutive behavior given by the tensor $\mathbb{C}$, the governing equations can be written as follows

$$\mathbf{M}_u \ddot{\mathbf{u}}(t) + \mathbf{C}_u \dot{\mathbf{u}}(t) + \mathbf{K}_u \mathbf{u}(t) = \mathbf{f}_u(t), \quad (5)$$

where $\mathbf{K}_u$ is the stiffness matrix, given by:

$$\mathbf{K}_u = \mathbb{A}_{e=1}^{n_e} \mathbf{K}_u^e, \quad \mathbf{K}_u^e = \int_{\Omega^e} (\mathbf{B}_u^e)^T \mathbb{C} \mathbf{B}_u^e dV, \quad (6)$$

and $\mathbf{M}_u$ is the mass matrix, given by:

$$\mathbf{M}_u = \mathbb{A}_{e=1}^{n_e} \mathbf{M}_u^e, \quad \mathbf{M}_u^e = \int_{\Omega^e} \rho (\mathbf{N}_u^e)^T \mathbf{N}_u^e dV, \quad (7)$$

where the matrices $\mathbf{N}_u^e$ and $\mathbf{B}_u^e$ are the matrices given by the interpolation functions and gradients for the displacement fields, and $\mathbf{f}_u$ is the vector of applied loads. The viscosity (or damping) matrix $\mathbf{C}_u$ is assumed to be known or estimated as a combination of the stiffness and the mass matrices [19].

2.2. Numerical integration methods

In this section, the equations of three reference numerical time-integration methods [10] are recalled. A fixed time step Δt is considered, and the time domain is discretized in values $t_k = \Delta t \cdot k, k \in \mathbb{N}$.

The first reference numerical integration technique is the Backward Euler Method, which is used to solve the heat transfer equations. Considering an initial temperature θ_0 , the method consists in solving the following linear system for each time step:

$$(\mathbf{K}_\theta \Delta t + \mathbf{C}_\theta) \theta_{k+1} = \mathbf{f}_{\theta,k+1} \Delta t + \mathbf{C}_\theta \theta_k. \quad (8)$$

Regarding structural dynamics problems, the Newmark and the Bathe methods are presented. Assuming that initial displacements and velocities are given, after computing the initial acceleration, each method provides a different procedure to obtain subsequent displacements and velocities. The Newmark method provides the following system:

$$\begin{aligned} (b_0 \mathbf{M}_u + b_1 \mathbf{C}_u + \mathbf{K}_u) \mathbf{u}_{k+1} = & \mathbf{f}_{u,k+1} + \mathbf{M}_u (b_0 \mathbf{u}_k + b_2 \dot{\mathbf{u}}_k + \ddot{\mathbf{u}}_k) \\ & + \mathbf{C}_u (b_1 \mathbf{u}_k + \dot{\mathbf{u}}_k), \end{aligned} \quad (9)$$

where the auxiliary constants $b_0 = 4/\Delta t^2, b_1 = 2/\Delta t$ and $b_2 = 4/\Delta t$ are introduced. Update rules are provided for the velocities and accelerations.

In the Bathe method, the displacements are computed in two sub-steps, with corresponding update rules for the velocities. The linear systems corresponding to these sub-steps are the following:

¹ juliareach.com.

² onsas.org.

$$(a_0 \mathbf{M}_u + a_1 \mathbf{C}_u + \mathbf{K}_u) \mathbf{u}_{(k+1/2)} = \mathbf{f}_{u,(k+1/2)} + \mathbf{M}_u(a_0 \mathbf{u}_k + a_4 \dot{\mathbf{u}}_k + \ddot{\mathbf{u}}_k) + \mathbf{C}_u(a_1 \mathbf{u}_k + \dot{\mathbf{u}}_k), \quad (10)$$

$$(a_2 \mathbf{M}_u + a_3 \mathbf{C}_u + \mathbf{K}_u) \mathbf{u}_{k+1} = \mathbf{f}_{u,k+1} + \mathbf{M}_u(a_5 \mathbf{u}_{(k+1/2)} + a_6 \mathbf{u}_k + a_1 \dot{\mathbf{u}}_{(k+1/2)} + a_7 \dot{\mathbf{u}}_k) \dots + \mathbf{C}_u(a_1 \mathbf{u}_{(k+1/2)} + a_7 \mathbf{u}_k), \quad (11)$$

where the auxiliary constants $a_0 = 16/\Delta t^2$, $a_1 = 4/\Delta t$, $a_2 = 9/\Delta t^2$, $a_3 = 3/\Delta t$, $a_4 = 8/\Delta t$, $a_5 = 12/\Delta t^2$, $a_6 = -3/\Delta t^2$ and $a_7 = -1/\Delta t$ were introduced.

2.3. Set propagation concepts

In this section we consider set-based integration in the context of linear differential equations and introduce two essential concepts: reach-set and flowpipe. Then we define the representations used in our approach: hyperrectangles, zonotopes and support functions. The definitions are illustrated by means of a textbook example: the harmonic oscillator. More complex and larger problems are considered in Section 4.

2.3.1. Illustrative example: harmonic oscillator

The set propagation definitions introduced in this section are illustrated using a single degree of freedom second order problem given by the following equation:

$$\ddot{u}(t) + \omega^2 u(t) = 0, \quad (12)$$

where ω is a scalar and u is the unknown. This problem can be associated with a spring-mass system, where $u(t)$ is the elongation of the spring at time t and the mass and stiffness set a natural frequency ω . In this case we consider $\omega = 4\pi$.

Let us introduce the state variable $v(t) := \dot{u}(t)$ and define the vector $\mathbf{x}(t) = [u(t), v(t)]^T$. Then Eq. (12) can be written in the following first order form

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \mathbf{x}(t). \quad (13)$$

Two cases are considered for the initial conditions, first a single initial condition given by $u(0) = 1$ and $v(0) = 0$, and a second case where the initial conditions belong to a set. This problem is solved using set propagation and the results obtained for the single initial condition are shown in Fig. 1. The time step-size used is $\delta = 0.025$. The set Ω_0 from Fig. 1a includes the analytic trajectory within $[0, \delta]$, and such set is propagated to cover the solution for further time intervals as in Fig. 1b. It is worth noting that even when the initial condition is a singleton, the set propagation technique returns a sequence of sets, and such sequence guarantees an enclosure of the analytic solution at any time $t \in [0, T] \subset \mathbb{R}$. For comparison, we also plot the analytic solution of Eq. (13) (magenta solid line) and a numerical solution obtained using the Newmark's method with time step $\delta/5$ (red triangle markers). The figure shows other set representations and concepts that will be developed in the following paragraphs.

The result obtained when a set of initial conditions is considered is shown in Fig. 2, together with a few dozen trajectories with random initial conditions drawn from the initial set $\mathcal{X}_0 = \mathcal{U}_0 \times \mathcal{V}_0 = [0.9, 1.1] \times [-0.1, 0.1]$, where $\times$ denotes the Cartesian product.

In both cases, it is shown that a single set-based integration covers infinitely many trajectories in dense time, i.e. for all intermediate times between 0 and T , where $T > 0$ is the time horizon, there is a reach-set that covers all the exact solutions.

2.3.2. Trajectories, reach-sets and flowpipes

Let us consider an initial-value problem

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \quad \mathbf{x}(0) \in \mathcal{X}_0, t \in [0, T], \quad (14)$$

with state matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector at time t . The initial condition is $\mathbf{x}(0) \in \mathcal{X}_0 \subset \mathbb{R}^n$ with $\mathcal{X}_0$ a closed and convex set. In this article we will refer to any state belonging to $\mathcal{X}_0$ as an *admissible* state. The guiding principle in set-based integration is to represent reachable states using sets in the Euclidean space that are *propagated* according to the system's dynamics. The obtained sequence of sets, by construction, covers the exact trajectories for given time intervals, for any admissible initial condition or external actions. For instance, the result for the harmonic oscillator presented in Section 2.3.1 using a single initial condition is shown in Fig. 1 (green circle), and in Fig. 2 (green box) the results for a box of initial conditions are shown.

Given the system of ODEs in Eq. (14), a *trajectory* from an initial state $\mathbf{x}_0$ is the solution for a given initial condition $\mathbf{x}(0) = \mathbf{x}_0$, denoted $\varphi(\mathbf{x}_0, t) : \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}^n$, which we know is $\varphi(\mathbf{x}_0, t) = e^{\mathbf{A}t} \mathbf{x}_0$. The *reach-set* at a time point $t \in [0, T]$ is

$$\mathcal{R}^e(\mathcal{X}_0, t) := \{\varphi(\mathbf{x}_0, t) : \mathbf{x}_0 \in \mathcal{X}_0\}. \quad (15)$$

This is the set of states which are reachable at time t starting from any admissible initial condition. Generalizing Eq. (15) to time intervals is straightforward: the reach-set over $[t_0, t] \subseteq [0, T]$ is

$$\mathcal{R}^e(\mathcal{X}_0, [t_0, t]) := \{\varphi(\mathbf{x}_0, s) : \mathbf{x}_0 \in \mathcal{X}_0, \forall s \in [t_0, t]\}. \quad (16)$$

The difference between reach-sets for time points and for time intervals is that the former are evaluated at a particular time t , while the latter include *all* states reachable for any time between t_0 and t , hence $\mathcal{R}^e(\mathcal{X}_0, t) \subseteq \mathcal{R}^e(\mathcal{X}_0, [t_0, t])$. Finally, given a time step $\delta > 0$ and a collection of reach-sets $\{\mathcal{R}^e(\mathcal{X}_0, [\delta(k-1), \delta k])\}_{k=1}^N$ with $N = T/\delta$, the *flowpipe* of Eq. (14) is the set union,

$$\mathcal{F}^e(\mathcal{X}_0, [0, T]) := \bigcup_{k=1}^N \mathcal{R}^e(\mathcal{X}_0, [\delta(k-1), \delta k]). \quad (17)$$

In Eqs. (15)–(17), the superindex e is used to remark that the definitions refer to the *exact* or true sets. In practice, reachable sets can only be obtained approximately and several algorithms to numerically approximate reachable sets are known in the literature. Different methods crucially depend on how the sets are represented and the cost of operating with those representations.

2.3.3. Set representations

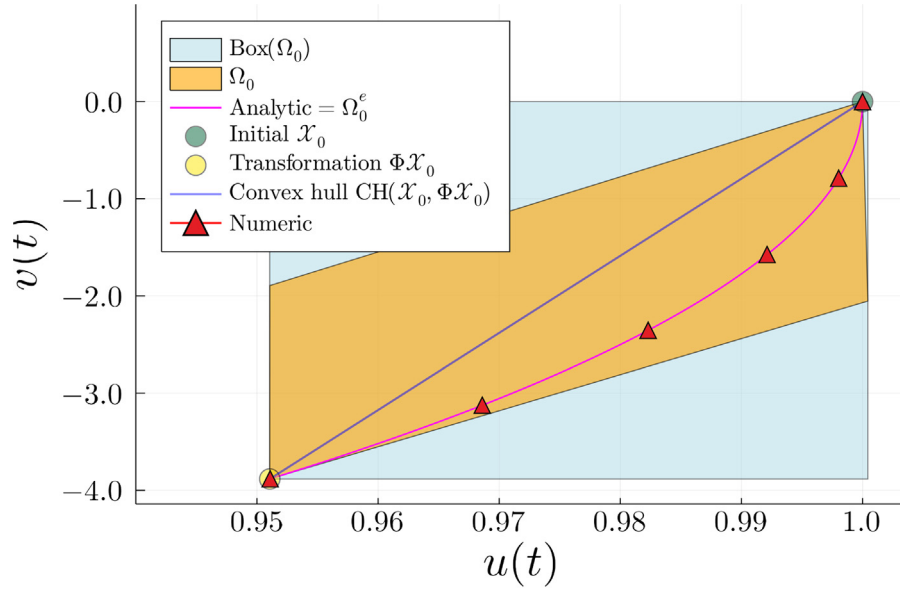
Three set representations commonly used in reachability analysis of linear systems are zonotopes, hyperrectangles and support functions. Let us briefly recall their definition.

2.3.3.1. Zonotopes. Given a vector $\mathbf{c} \in \mathbb{R}^n$ that we call *center* and a list of $p \geq n$ vectors that we call *generators*, $\mathbf{g}_1, \dots, \mathbf{g}_p, \mathbf{g}_i \in \mathbb{R}^n$, the associated zonotope is the set

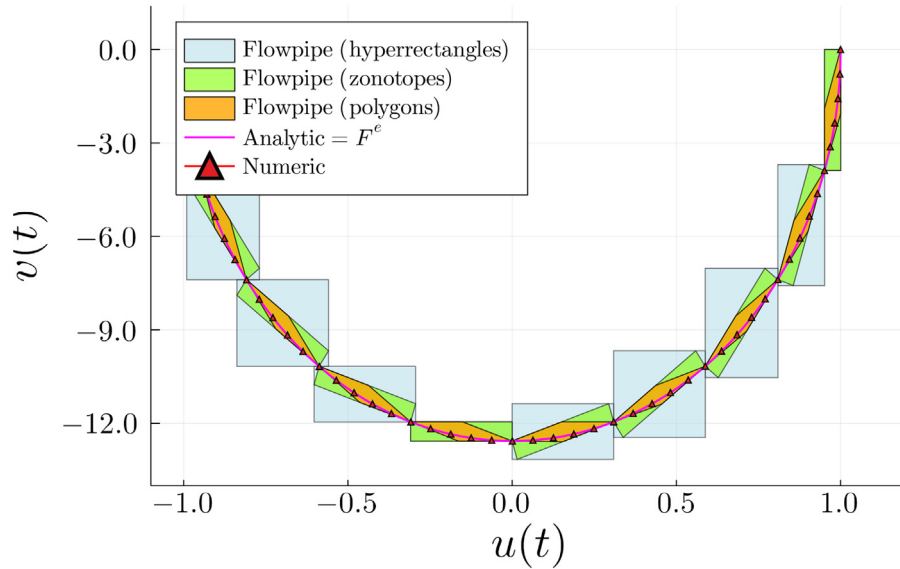
$$Z = \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{x} = \mathbf{c} + \sum_{i=1}^p \xi_i \mathbf{g}_i, \quad \xi_i \in [-1, 1] \right\}. \quad (18)$$

In compact notation we write $Z = \langle \mathbf{c}, \mathbf{G} \rangle_Z$, where each generator is stored as a column of the generators matrix $\mathbf{G} \in \mathbb{R}^{n \times p}$. Zonotopes can be interpreted in different ways, such as the Minkowski sum of line segments, or as the affine map of a unit ball in the infinity norm. The zonotope representation has been successfully used in reachability analysis because it offers a very compact representation relative to the number of vertices or faces, so it is well suited to represent high dimensional sets [24,5].

2.3.3.2. Hyperrectangles. Given vectors $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{r} \in \mathbb{R}^n$ that we call *center* and *radius* respectively, the associated hyperrectangle is the set



(a) The set Ω_0 (orange) encloses the true solution (magenta) at the endpoints and at any intermediate time between 0 and $\delta = 0.025$.



(b) Set propagation using hyperrectangles (lightblue), zonotopes (green) and to assess the zonotopic overapproximation of Ω_0 , polygons (orange).

Fig. 1. Illustration of set-based integration of the simple harmonic oscillator example with single initial condition. Shown are phase-space plots $u(t)$ vs. $v(t)$ for the first reach-set (Fig. 1a), and for the first 9 reach-sets (Fig. 1b).

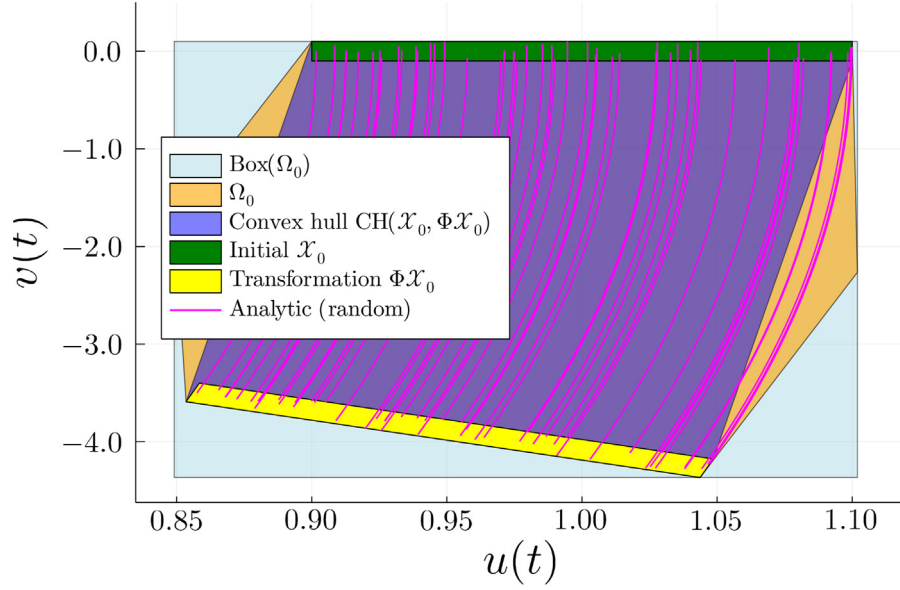
$$H = \{\mathbf{x} \in \mathbb{R}^n : |x_i - c_i| \leq r_i, \quad i = 1, \dots, n\}. \quad (19)$$

In compact notation we write $H = \langle \mathbf{c}, \mathbf{r} \rangle_H$. Every hyperrectangle is a zonotope but the converse is not true. In fact, the linear map of a hyperrectangle can be (exactly) represented as a zonotope. While computations with hyperrectangles can be performed very efficiently, using hyperrectangles for reachability can result in coarse overapproximations unless they are used in concurrence with other representations at intermediate parts of the algorithm.

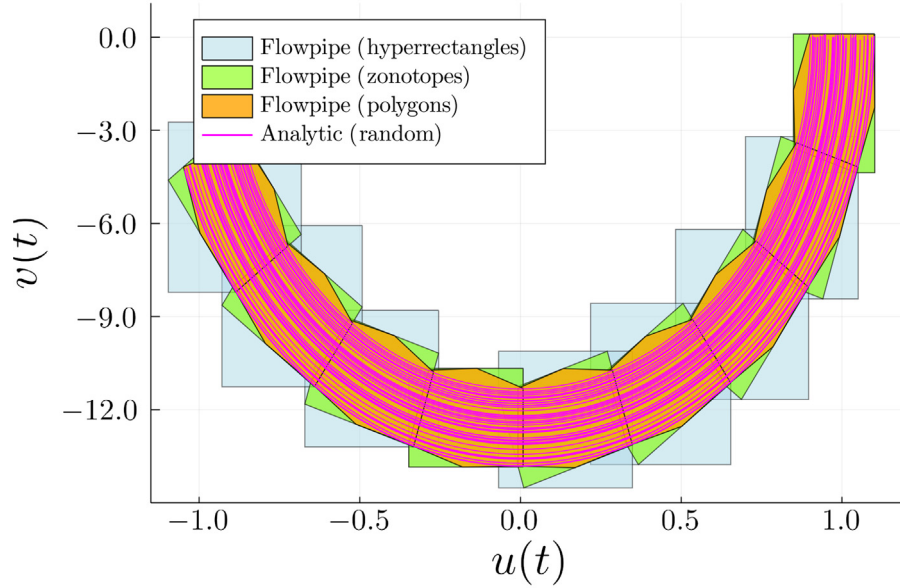
2.3.3.3. Support functions. The support function of a set $X \subset \mathbb{R}^n$ along direction $\mathbf{d} \in \mathbb{R}^n$ is the scalar

$$\rho(\mathbf{d}, X) = \max_{\mathbf{x} \in X} \mathbf{d}^T \mathbf{x}, \quad (20)$$

where the superscript T denotes transposition. The set of points in X that are the maximizers of Eq. (20) are called the *support vectors*. The intuition behind Eq. (20) is that support functions represent the farthest (signed) distance to the origin of the set X along direction $\mathbf{d}$. Given that the direction can be chosen at will, support functions can be used to obtain the solution corresponding to a linear combination of the state variables at once and at a reduced cost [34].



(a) The set Ω_0 (orange) covers the right-most trajectories for intermediate times, which naturally escape linear interpolations (violet).



(b) The flowpipe construction method for Eq. (22) is wrapping-free: the area of the sets does not increase with time.

Fig. 2. Illustration of set-based integration of the simple harmonic oscillator with distributed initial conditions. Note that a single set-based integration covers all dynamically feasible behaviors: to illustrate this fact we plot 50 analytic solutions (magenta) uniformly distributed over $\mathcal{X}_0$.

3. Methodology

In this section the proposed method is presented. Given a heat transfer or structural dynamics system of linear differential equations, the proposed algorithm can be summarized in the following three stages:

- (i) **Homogeneization and transformation into a first order system.** Transform the system of differential equations into a linear homogeneous system of first order, as in Eq. (14), using the method described in Section 3.1.

- (ii) **Conservative time discretization.** Build a set $\Omega_0 \subset \mathbb{R}^n$ that contains all exact trajectories between time 0 and time $\delta > 0$, i.e. satisfying the property:

$$\Omega_0^e := \mathcal{R}^e(\mathcal{X}_0, [0, \delta]) \subseteq \Omega_0, \quad (21)$$

using the results presented in Section 3.2.

- (iii) **Set propagation.** Set the initial set $X_0 := \Omega_0$ and the state transition matrix $\Phi := e^{A\delta}$, and compute the reachable sets for the subsequent time intervals, solving the recurrence:

$$X_{k+1} = \Phi X_k, \quad X_0 \subset \mathbb{R}^n, k = 0, 1, \dots, N-2, \quad (22)$$

where X_k is an overapproximation of the exact reachable sets for the time interval $[k\delta, (k+1)\delta]$. This is done using available set propagation strategies described in Section 3.3.

The proposed method consists in an explicit numerical approach, since Eq. (22) is used to calculate the reachable set at a later time interval from the reachable set at the current time interval. If the state dimension is large, Krylov methods are used to evaluate the action of the matrix exponential on specific directions, instead of computing the matrix exponential in full.

3.1. Homogeneization and transformation into a first order system

We now describe the first step of the method, consisting in: an algebraic transformation of the governing equations (heat transfer or structural dynamics), into a first order, homogeneous system.

3.1.1. Transformation into a first order form

The first step of the method is to reformulate the governing equations as a system of the form of Eq. (14). In the case of heat transfer problems, using that the diffusion matrix $\mathbf{C}_\theta$ is a square positive definite (thus invertible) matrix, the system given by Eq. (1) can be written as

$$\dot{\theta}(t) = -\mathbf{C}_\theta^{-1}\mathbf{K}_\theta\theta(t) + \mathbf{C}_\theta^{-1}\mathbf{f}_\theta(t). \quad (23)$$

For the structural dynamics equations, using that the matrix $\mathbf{M}_u$ is positive definite (and invertible), the system given by Eq. (5) can be written as:

$$\begin{cases} \dot{\mathbf{u}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\mathbf{M}_u^{-1}\mathbf{K}_u\mathbf{u} - \mathbf{M}_u^{-1}\mathbf{C}_u\mathbf{v} + \mathbf{M}_u^{-1}\mathbf{f}_u(t) \end{cases} \quad (24)$$

3.1.2. Homogeneization

In this stage of the method, the system of equations is converted to a homogeneous form. In order to do that, the inputs vector is integrated to the system of ODEs as a combination of auxiliary variables with a given dynamic.

Let us consider that the inputs vector $\mathbf{f}(t)$ (representing loads $\mathbf{f}_u$ or heat fluxes $\mathbf{f}_\theta$) can be decomposed as follows:

$$\mathbf{f}(t) = \sum_{i=1}^{n_f} \mathbf{f}_0^{(i)} \eta^{(i)}(t), \quad (25)$$

where n_f is the number of input components, each $\mathbf{f}_0^{(i)} \in \mathbb{R}^m$ is a known vector and the functions $\eta^{(i)}(t) : [0, T] \rightarrow \mathbb{R}$ are functions that can be expressed in terms of the solution of a first order homogeneous system of differential equations.

The property of functions $\eta^{(i)}$ can be written as follows: for each function $\eta^{(i)}$ there exists a vector of $q^{(i)}$ functions $\xi^{(i)}(t) = [\xi_1^{(i)}(t), \dots, \xi_{q^{(i)}}^{(i)}(t)]^T$, such that, for each $i = 1, \dots, n_f$:

$$\begin{cases} \eta^{(i)}(t) = \xi_1^{(i)}(t) \\ \dot{\xi}^{(i)}(t) = \mathbf{B}^{(i)}\xi^{(i)}(t) \end{cases} \quad (26)$$

where $\mathbf{B}^{(i)}$ is a $q^{(i)} \times q^{(i)}$ matrix. To exemplify definition (26), two special cases are mentioned: exponential functions (with $q = 1$) and trigonometric functions (with $q = 2$), which correspond to systems

$$\begin{cases} \eta(t) = \xi_1(t) \\ \dot{\xi}_1(t) = \alpha \cdot \xi_1(t) \end{cases} \quad \text{and} \quad \begin{cases} \eta(t) = \xi_1(t) \\ \dot{\xi}_1(t) = \xi_2(t) \\ \dot{\xi}_2(t) = -\omega^2 \xi_1(t) \end{cases}, \quad (27)$$

respectively, with α and ω being real scalars. In particular, the case $\alpha = 0$ is used to model constant functions.

Substituting Eq. (26) into Eq. (25), we can reformulate Eq. (23) as

$$\begin{cases} \dot{\theta}(t) = -\mathbf{C}_\theta^{-1}\mathbf{K}_\theta\theta(t) + \sum_{i=1}^{n_f} \mathbf{C}_\theta^{-1}\mathbf{F}_0^{(i)}\xi^{(i)}(t) \\ \dot{\xi}^{(i)}(t) = \mathbf{B}^{(i)}\xi^{(i)}(t) \quad i = 1, \dots, n_f \end{cases} \quad (28)$$

where

$$\mathbf{F}_0^{(i)} = \begin{bmatrix} \mathbf{f}_0^{(i)} & \mathbf{0}_{m \times (q^{(i)}-1)} \end{bmatrix}.$$

Similarly, Eq. (24) can be rewritten as

$$\begin{cases} \dot{\mathbf{u}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\mathbf{M}_u^{-1}\mathbf{K}_u\mathbf{u} - \mathbf{M}_u^{-1}\mathbf{C}_u\mathbf{v} + \sum_{i=1}^{n_f} \mathbf{M}_u^{-1}\mathbf{F}_0^{(i)}\xi^{(i)}(t) \\ \dot{\xi}^{(i)}(t) = \mathbf{B}^{(i)}\xi^{(i)}(t) \quad i = 1, \dots, n_f \end{cases} \quad (29)$$

obtaining then an homogeneous system of differential equations with $m + \sum_{i=1}^{n_f} q^{(i)}$ unknowns.

3.1.3. Matrix Coupled formulation and initial conditions

The systems of differential equations in Eqs. (28) and (29), can be written in a first order matrix coupled form as follows:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x}(t) \\ \xi^{(1)}(t) \\ \vdots \\ \xi^{(n_f)}(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{x}(t) \\ \xi^{(1)}(t) \\ \vdots \\ \xi^{(n_f)}(t) \end{bmatrix}, \quad (30)$$

where the vector of unknowns $\mathbf{x}(t)$ contains the state variables (θ for heat transfer or $[\mathbf{u}, \mathbf{v}]$ for structural dynamics), the unknowns in the vector $\xi^{(i)}(t)$ are called input variables and the matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{bmatrix} -\mathbf{C}_\theta^{-1}\mathbf{K}_\theta & \mathbf{C}_\theta^{-1}\mathbf{F}_0^{(1)} & \dots & \mathbf{C}_\theta^{-1}\mathbf{F}_0^{(n_f)} \\ \mathbf{0} & \mathbf{B}^{(1)} & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}^{(n_f)} \end{bmatrix}, \quad (31)$$

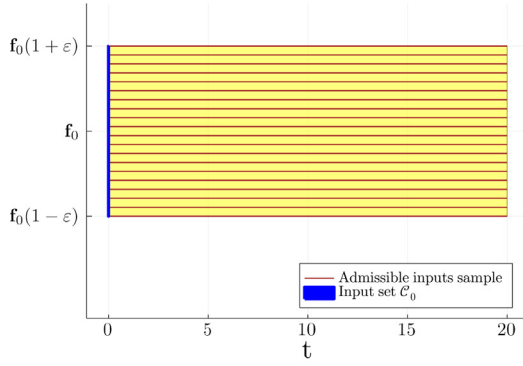
for heat transfer problems or

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ -\mathbf{M}_u^{-1}\mathbf{K}_u & -\mathbf{M}_u^{-1}\mathbf{C}_u & \mathbf{M}_u^{-1}\mathbf{F}_0^{(1)} & \dots & \mathbf{M}_u^{-1}\mathbf{F}_0^{(n_f)} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^{(1)} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}^{(n_f)} \end{bmatrix}, \quad (32)$$

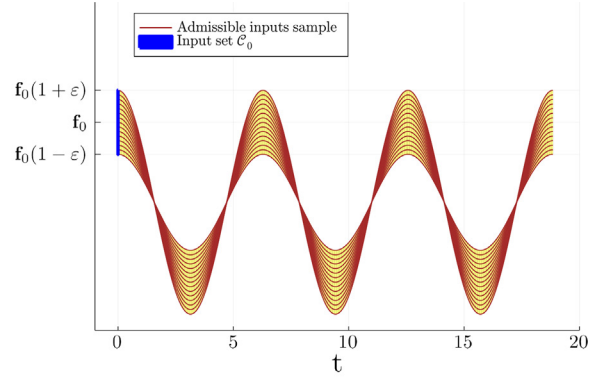
for structural dynamics problems.

The initial conditions of the system given in Eq. (30) can be defined by the Cartesian product $\mathcal{X}_0 \times \mathcal{C}_0$, where $\mathcal{X}_0$ is the set of initial conditions of the state variables and $\mathcal{C}_0$ is the set of initial conditions for the differential equations in Eq. (26).

Concerning the initial condition for the input variables in Eq. (30), that we call $\mathcal{C}_0$, they are not restricted to be a single value; in fact they can be chosen to be e.g. a hyperrectangle, in which case we can model an uncertainty of the initial conditions around nominal values for a whole family of input functions, with a single flow-pipe computation.



(a) Example of constant input model.



(b) Example of time-varying input model.

Fig. 3. Illustration of different input function assumptions. A set $\mathcal{C}_0$ is used to span all possible input functions with the specified behavior; a uniform sample is shown for each case.

To illustrate the formulation, assume that there is only one constant forcing term $\mathbf{f}_0$ ($n_f = 1$, $\mathbf{B}^{(1)} = \mathbf{0}_{1 \times 1}$), which corresponds to the case $\alpha = 0$ in Eq. (27). We take an interval $\mathcal{C}_0 = [1 - \varepsilon, 1 + \varepsilon] \subset \mathbb{R}$ for some $\varepsilon > 0$. With this initial condition, we can model all forcing functions that take any constant value within $\mathbf{f}_0 \pm \varepsilon \mathbf{f}_0$, as shown in Fig. 3a. On the other hand, in Fig. 3b we show an example that corresponds to a time-varying (sinusoidal) input function with $\omega = 1$ in Eq. (27). The three-dimensional heat transfer problem in Example 4 of Section 4.5 combines three different input terms.

3.2. Conservative time discretization

In this section we explain how to perform a time discretization that is conservative in the sense of including all admissible behaviors in dense time, and continue with the description of different methods to propagate such initial set according to the system's state transition matrix.

We recall that $\mathcal{X}_0$ is the set of initial states and, since the analytic solution of Eq. (14) at $t = \delta$ is $\varphi(\mathbf{x}_0, \delta) = \Phi \mathbf{x}_0$ for each $\mathbf{x}_0 \in \mathcal{X}_0$, then $\Phi \mathcal{X}_0$ is the set of transformed states after one time increment of size δ , i.e. $\mathcal{R}^e(\mathcal{X}_0, \delta) = \Phi \mathcal{X}_0$. The reach-set for the time interval $[0, \delta]$ is more difficult to obtain because there is no closed-form expression. Let $CH(X, Y)$ denote the *convex hull* of the set union of two sets $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^n$, i.e. the smallest convex set that contains X and Y , defined by

$$CH(X, Y) = \{z \in \mathbb{R}^n : z = \lambda x + (1 - \lambda)y, \quad \lambda \in [0, 1], x \in X, y \in Y\}. \quad (33)$$

It is clear (compare e.g. with Fig. 1a) that the analytic solution does not follow the straight line of the convex hull $CH(\mathcal{X}_0, \Phi \mathcal{X}_0)$ for intermediate times between 0 and δ , because trajectories curve due to the system's dynamics. Hence we need to enlarge, or bloat, the convex hull by a suitable amount such that all the exact reachable states Ω_0^e are covered.

To obtain a set Ω_0 satisfying Eq. (21), we propose a slight variation of the method presented in [21]. The advantage of the new method is that it doesn't require to solve an optimization problem for each direction of interest. The symbol $\mathcal{X}$ represents the symmetric interval hull of $\mathcal{X} \subset \mathbb{R}^n$, that is the smallest hyperrectangle centered at the origin that contains $\mathcal{X}$.

Proposition 1. Define the following pair of sets

$$\Omega_0^+ = CH(\mathcal{X}_0, \Phi \mathcal{X}_0 \oplus E^+(\mathbf{A}, \mathcal{X}_0, \delta)) \quad (34)$$

$$\Omega_0^- = CH(\Phi \mathcal{X}_0, \mathcal{X}_0 \oplus E^-(\mathbf{A}, \Phi \mathcal{X}_0, \delta)) \quad (35)$$

where $E^+(\mathbf{A}, \mathcal{X}_0, \delta) = \left(\mathbf{P}(\|\mathbf{A}\|, \delta) (\mathbf{A}^2 \mathcal{X}_0) \right)$ and $\mathbf{P}(\mathbf{A}, \delta) = \sum_{i=0}^{\infty} \mathbf{A}^i \delta^{i+2} / (i+2)!$. Then, the set $\Omega_0 := \Omega_0^+ \cap \Omega_0^-$ encloses Ω_0^e , i.e. $\Omega_0^e \subseteq \Omega_0$. Moreover, $\Omega_0 \rightarrow \Omega_0^e$ as $\delta \rightarrow 0$.

The proof of Propositions 1 is left to Appendix A. The intuition behind Proposition 1 is that first Eq. (34) performs one step forward in time (from $t = 0$ to $t = \delta$) with a bloating set E^+ that is a small box around the origin (E^+ tends to zero as $\delta \rightarrow 0$), and second, Eq. (35) performs one step in reverse time (from $t = \delta$ to $t = 0$) which are then intersected to remove spurious overapproximation errors.

The result in our running example is shown in Fig. 1a: note how the set Ω_0 (orange) encloses the exact (curved) solution for all intermediate times between 0 and δ . The case depicted in Fig. 2a is more interesting: it corresponds to the case in which the set of initial states $\mathcal{X}_0$ is a hyperrectangle. While the obtained set Ω_0 (orange) may not be the tightest convex set containing the exact reachable set Ω_0^e , it is nonetheless a very good approximation. Finally, let us remark that Ω_0 is in general a polygon (or a polytope in higher dimensions). In practice, either we approximate it with a hyperrectangle (see Fig. 2a in lightblue), or use lazy (i.e. on-demand) evaluations of the support function. While the former approach leads to faster runtimes for large time horizons, the latter does not introduce additional overapproximation error. The hyperrectangular approximation of Ω_0 can be easily computed using support functions.

Having obtained the set Ω_0 to fulfill the conservative time discretization requirement in Eq. (21), we proceed in the next paragraphs to explain the set-based propagation of the sequence. To solve Eq. (22), three different approaches are presented: zonotopes, hyperrectangles and support functions. Each method offers a different efficiency vs accuracy tradeoff and scales differently with the system's dimension. For Section 4 we have selected the method that is more suitable for each problem. A complete evaluation of the different methods is out of the scope of this article (see also [4] for a comparison of results obtained by different software tools on a set of benchmark problems).

```

julia> using LazySets
julia> A = [0 1; -(4π)^2 0] # state matrix
2×2 Matrix{Float64}:
 0.0  1.0
-157.914  0.0
julia> X₀ = Hyperrectangle([1.0, 0.0], [0.1, 0.1]); # initial states (center, radii)
julia> δ = 0.025; # time step

```

Listing 1. Stage 1. Initial-value problem formulation.

3.3. Set propagation

3.3.1. Set propagation using zonotopes

To solve the set-based recurrence of Eq. (22) using zonotopes, it suffices to observe that zonotopes are closed under linear maps: $\mathbf{M}Z = \langle \mathbf{M}\mathbf{c}, \mathbf{M}\mathbf{G} \rangle_Z$ is again a zonotope for any $\mathbf{M} \in \mathbb{R}^{m \times n}$, obtained by transforming the center and generators of Z by the linear map $\mathbf{M}$. Therefore if $Z_0 = \langle \mathbf{c}_0, \mathbf{G}_0 \rangle_Z$ is a zonotope, it holds that

$$Z_k = \langle \Phi^k \mathbf{c}_0, \Phi^k \mathbf{G}_0 \rangle_Z, \quad k = 0, 1, \dots, N-1. \quad (36)$$

In general, the initial set Ω_0 from Proposition 1 is not a zonotope so we overapproximate it with a hyperrectangle as a pre-processing step. Set propagation using zonotopes is illustrated in Fig. 1b for singleton resp. in Fig. 2b for distributed initial conditions (light-green).

Finally we remark that, in a simplified model of computation, the time complexity of Eq. (36) is $\mathcal{O}(Nn^2p)$, and the space complexity is $\mathcal{O}(Nnp)$. Typically, $p = n$ hence storing all intermediate zonotopes can be expensive if the state-space dimension n is sufficiently large. However, there are situations in which we are only interested in obtaining minimum and maximum bounds of the flowpipe and in that case it is more efficient and equally accurate to propagate axis-aligned hyperrectangles. This is the subject of the next subsection.

3.3.2. Set propagation using hyperrectangles

Hyperrectangular approximations are particularly useful in problems where it is relevant to find the maximum or minimum values of all the state variables; for example in the longitudinal oscillations of a bar if we are interested in computing the maximum displacements and velocities (cf. Example 4.2), or in a massive concrete structure model we may be interested in the maximum temperature achieved at any mesh point (cf. Example 4.5).

Given a zonotope Z , we say that H is a *tight* hyperrectangular approximation if H is the smallest hyperrectangle that contains Z . The following proposition formalizes the idea of approximating the zonotope flowpipe with tight hyperrectangles.

Proposition 2. Let $H_0 = \langle \mathbf{c}_0, \mathbf{r}_0 \rangle_H$ be a hyperrectangle enclosing Ω_0^c . For each $k \geq 1$, define the sequence $\mathbf{c}_k = \Phi^k \mathbf{c}_0$ and $\mathbf{r}_k = |\Phi^k| \mathbf{r}_0$, where the absolute values are taken component-wise. Then, the sequence of hyperrectangles $H_k = \langle \mathbf{c}_k, \mathbf{r}_k \rangle_H$ tightly overapproximates the solutions of the recurrence (22) for all $k = 0, \dots, N-1$.

The proof of Proposition 2 is left to Appendix B. Concerning the computational cost, the time complexity of the method in Proposition 2 is $\mathcal{O}(Nn^2)$, and the space complexity is $\mathcal{O}(Nn)$. Note that computing with hyperrectangles is comparatively less expensive than zonotopes, although the latter have more expressive power than the former.

Set propagation using hyperrectangles is illustrated in Fig. 1b for singleton initial conditions resp. in Fig. 2b for distributed initial conditions (light-blue). For the 6th reach-set the hyperrectangle and the zonotope coincide; in general though, the hyperrectangle encloses the zonotope. However, for the k -th reach-set, the maximum and minimum of Z_k are always the same to those obtained with H_k as claimed in Proposition 2.

While solving Eq. (22) using hyperrectangles scales very well with the dimension, this approach is limited to obtaining maximum (resp. minimum) values over solutions. We can interpret such computation as the maximum of the scalar product $\mathbf{d}^T \mathbf{x}(t)$ over all $\mathbf{x}(t) \in \mathcal{R}^c(\mathcal{X}_0, [\delta(k-1), \delta k])$, where $\mathbf{d}$ is any canonical direction of $\mathbb{R}^n$, i.e. of the form $\mathbf{d} = (0, \dots, 1, \dots, 0)^T$. The minimum can be obtained analogously. Using support functions, as explained in the next section, we can generalize the computation for an arbitrary direction $\mathbf{d} \in \mathbb{R}^n$.

3.3.3. Set propagation using support functions

The first systematic attempt to solve Eq. (22) using support functions was presented in [37]. The idea is to make use of the property $\rho(\mathbf{d}, \mathbf{M}\mathbf{X}) = \rho(\mathbf{M}^T \mathbf{d}, \mathbf{X})$ for any matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$. By successive application of this rule,

$$\rho(\mathbf{d}, X_k) = \rho((\Phi^T)^k \mathbf{d}, X_0), \quad k = 0, 1, \dots, N-1. \quad (37)$$

The support function of common set representations (e.g. hyperrectangles and zonotopes) is known analytically so in those cases it can be computed very efficiently. In addition, the support function of an intersection satisfies that $\rho(\mathbf{d}, X \cap Y) \leq \min\{\rho(\mathbf{d}, X), \rho(\mathbf{d}, Y)\}$, while for the case of Minkowski sums, $\rho(\mathbf{d}, X \oplus Y) = \rho(\mathbf{d}, X) + \rho(\mathbf{d}, Y)$.

Using support functions we can compute linear combinations of the state variables, i.e. we can solve the recurrence (22) along a given direction directly solving a recurrence for the support function, instead of computing the flowpipe for all variables and then extracting the relevant information. A practical application of the solution method in Eq. (37) to obtain the spatial gradient in a heat transfer problem is presented in Example 4.4.

3.4. Example code

The pseudo-code presented in Algorithm 1 illustrates the main steps of our approach. To describe with Departamento de Matemática y Aplicaciones, Centro Universitario Regional del Este, Universidad de la República, Maldonado, Uruguay more detail the implementation, a simplified version of the proposed method is presented for the harmonic oscillator from Section 2.3.1. For numeric and symbolic operations with sets, we use the LazySets.jl³ library, written in the Julia programming language [12]. The library implements all required set representations. We show the

³ The library can be downloaded from github.com/JuliaReach/LazySets.jl.

```

julia> P = sum(abs.(A)^i * δ^(i+2) / factorial(i+2) for i in 0:10) # cf. Prop. 1
2×2 Matrix{Float64}:
 0.00031508  0.00000262
 0.00041327  0.00031508

julia> E+(X0) = symmetric_interval_hull(P * symmetric_interval_hull(A^2 * X0)); E+(X0)
Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}{([0.0 0.0], [0.05477208, 0.07676220])}

julia> Φ = exp(A*δ)
2×2 Matrix{Float64}:
 0.95105652  0.02459079
-3.88322208  0.95105652

julia> Ω0 = CH(X0, Φ*X0 ⊕ E+(X0)) ∩ CH(Φ*X0, X0 ⊕ E+(Φ*X0)); # symbolic discretized states
julia> H0 = box_approximation(Ω0) # box enclosure
Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}{([0.97471, -2.13332], [0.12868, 2.23332])}

```

Listing 2. Conservative time discretization computation.

```

julia> Z6 = Zonotope(Φ^5 * H0.center, Φ^5 * genmat(H0));
Zonotope{Float64, Vector{Float64}, Matrix{Float64}}{
 [-0.16976461, -12.24853154], # center
 [-0.00000000  0.17772235; # generators matrix
 -1.61711795  0.00000000]}

```

Listing 3. Set propagation using zonotopes.

```

julia> H6 = Hyperrectangle(Φ^5 * H0.center, abs.(Φ^5) * H0.radius)
Hyperrectangle{Float64, Vector{Float64}, Vector{Float64}}{
 [-0.16976, -12.24853], # center
 [ 0.17772,  1.61712]} # radius

```

Listing 4. Set propagation using hyperrectangles.

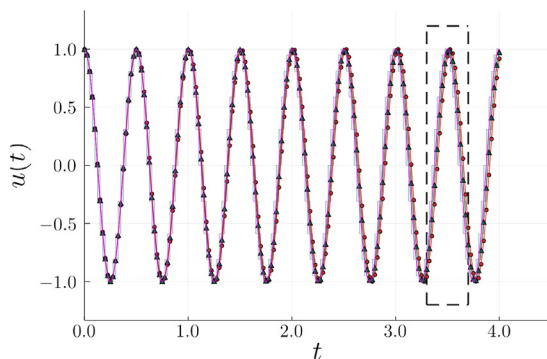
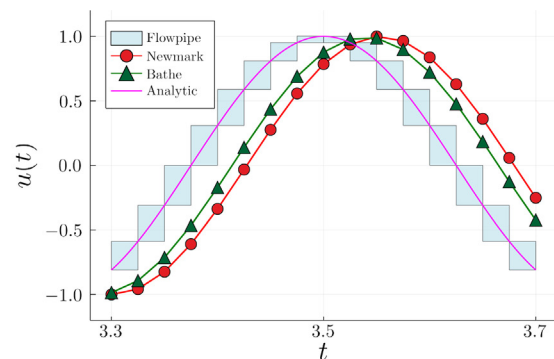
(a) Displacements for $t \in [0, 4]$.(b) Zoom for $t \in [3.3, 3.7]$.

Fig. 4. Example 1: displacements obtained using $\alpha = \delta/T = 0.05$. The analytic solution (magenta) is shown superposed with the numerical solution obtained with Newmark's method (red circles) and Bathe's method (green triangles).

Julia commands (introduced after the `julia >` prompt) followed by the resulting output.

Algorithm 1. Set propagation method pseudo-code.

```

Input:  $\mathbf{M}, \mathbf{C}, \mathbf{K}$ : FEM assembled matrices,  $\mathbf{F}$ : vector of loads,  $\mathcal{X}_0 \subseteq R^n$ : initial states set
 $\delta$ : time step increment,  $N$ : number of time steps

Output: List of reachable sets

1  $\mathbf{A}, \mathcal{C}_0 = \text{homogeneize}(\mathbf{M}, \mathbf{C}, \mathbf{K}, \mathbf{F})$  ; // use Eq.(31) or (32)
2  $\Omega_0 = \text{discretize}(\mathcal{X}_0 \times \mathcal{C}_0, \mathbf{A}, \delta)$  ; // time discretization using Prop.1
3  $\Phi = \exp(\mathbf{A}\delta)$  ; // matrix exponential
4  $X_0 = \Omega_0$  ; 347 // reachable states approximation for  $[0, \delta]$ 
5 for  $k = 0$  to  $N - 2$  do
6    $X_{k+1} = \Phi X_k$  ; // set propagation; use either Eq.(36), Prop.2 or Eq.(37)
7 end
8 return  $X_0, X_1, \dots, X_{N-1}$  ; // flowpipe approximation

```

The first step of the method is the formulation of the initial-value problem as an homogeneous, first order system which corresponds to Line 1 of the pseudo-code. In the present case, there are no external forces. The initial states are defined choosing a suitable set representation, e.g. a hyperrectangle in this case (see Listing 1).

The second step of the algorithm is the conservative time discretization, which corresponds to Line 2 of the pseudo-code. Here we computed $\mathbf{P}$ and Φ in full; for larger problems, matrix functions of exponential type are used. The operation of multiplication between matrices and sets, as well as the symmetric interval hull are already implemented in LazySets.jl. Symbolic Minkowski sums, convex hulls and intersections are also implemented in the library. Such specific features simplify writing set-based algorithms (see Listing 2).

Here we have computed an overapproximation of Ω_0 , namely a hyperrectangular enclosure.⁴ Such set is centered in $[0.97, -2.13]$ and has radius 0.13 and 2.23 on the u and v coordinates respectively, as shown in light-blue in Fig. 2a.

The third step of the method consists in set propagation (see Line 6 of the pseudo-code). To illustrate the propagation using zonotopes, let us obtain the center and generators matrix of the sixth zonotope in Fig. 2 (hence $k = 5$): (see Listing 3)

Observe that each generator is proportional to a canonical vector, which explains the axis-aligned shape of the sixth zonotope in Fig. 2. Similarly we can obtain the corresponding hyperrectangle using Proposition 2: (see Listing 4)

Finally we note that set propagation using support functions is available in the library by use of the `support_function` method (Unicode alias: ρ).

4. Numerical results

In order to obtain comparative results about the performance and accuracy of the methods discussed, five numerical examples are solved. Each example introduces a new motivation and level of complexity in the comparative process of the methods.

The reachability problems are solved using the JuliaReach library [20,14], implemented in the Julia language [12]. The computations associated to the Finite Element Method are done using the library ONSAS. The visualization is done using JuliaPlots [16] and Paraview [1], and unstructured meshes are created using GMSH [23]. For matrix functions of exponential type, the implementation in [43] is used. All the results are executed in a laptop computer running Linux OS, with an Intel(R) Core(TM) i7-8705G CPU, 3.10 GHz processor and 16 GB RAM using Julia v.1.6.1. The results presented can be reproduced using scripts publicly-available⁵. The time and memory usage reported in the profiling tables are obtained using the Julia macro `@time`, which provides the time required by the execution and the total number of bytes that were allocated.

4.1. Example 1: accuracy analysis of the harmonic oscillator

In this example, the one-dimensional harmonic oscillator presented in Section 2.3.1 is considered. We compare the set propaga-

⁴ The set Ω_0 may also be propagated symbolically, i.e. without overapproximation, using support functions.

⁵ Available at: <https://github.com/JuliaReach/SetPropagation-FEM-Examples>.

Table 1

Execution times and memory usage for the harmonic oscillator example.

	Set Propagation	Newmark	Bathe
$\delta = 0.025$	0.29 ms (140 kB)	0.44 ms (321 kB)	0.56 ms (561 kB)
$\delta = 0.05$	0.17 ms (91 kB)	0.18 ms (161 kB)	0.36 ms (282 kB)

tion results in terms of accuracy with respect to reference numerical integration methods and the analytic solution.

4.1.1. Problem definition

The governing equation is given in Eq. (12) and the parameters considered are frequency $\omega = 4\pi$ (i.e. period 0.5 s), and initial conditions $u(0) = 1, \dot{u}(0) = v(0) = 0$. Different time steps are considered: $\delta = 0.025$ s and $\delta = 0.05$ s.

4.1.2. Numerical resolution

The results obtained using time step $\delta = 0.025$ s are shown in Fig. 4. The solution obtained using the Newmark method is shown with red circles, the solution using the Bathe method is represented by green triangles and the analytic solution is represented by a solid magenta line. Regarding the solution obtained by set propagation, we represent the flowpipe using polygons and show their projection onto the $u(t)$ axis using light-blue boxes.

In Fig. 4b it can be observed that the analytic solution is contained in the flowpipe. On the other hand, the numerical methods show a considerable period elongation (after 7 periods of simulation). Execution times are reported in Table 1.

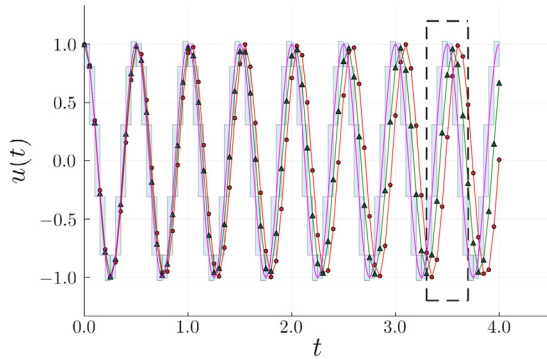
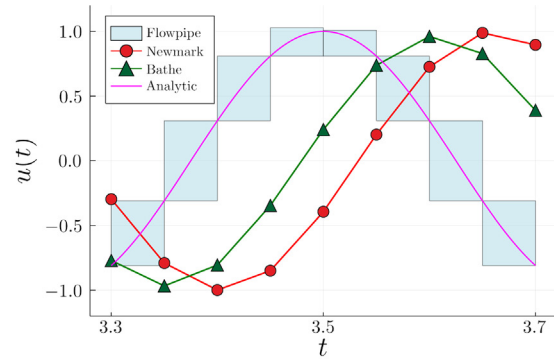
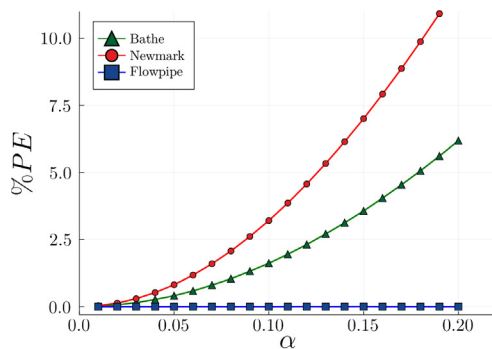
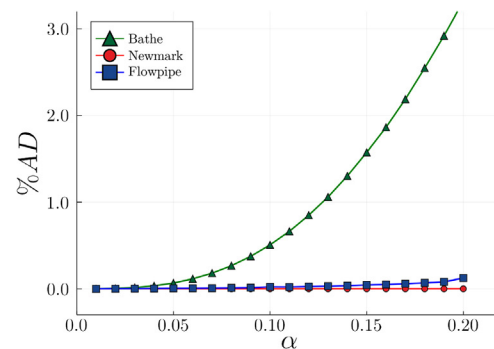
(a) Displacements for $t \in [0, 4]$.(b) Zoom for $t \in [3.3, 3.7]$.

Fig. 5. Example 1: displacements using $\alpha = \delta/T = 0.1$. The analytic solution (magenta) is shown superposed with the numerical solution obtained with Newmark's method (red circles) and Bathe's method (green triangles).



(a) Percentage period elongation (PE).



(b) Percentage amplitude decay (AD).

Fig. 6. Example 1: PE and AD as a function of the relative step size $\alpha = \delta/T$. In red (green) color, the results for Newmark (Bathe) methods. In blue, the results obtained with the set propagation approach.

The results obtained using a larger time step $\delta = 0.05$ s are shown in Fig. 5. In this case, in Fig. 5b it can be observed a greater period elongation for the numerical methods. The flowpipe still includes the analytic solution but the width of the intervals on the $u(t)$ axis is larger than in the previous case.

4.1.3. Accuracy analysis

Accuracy analysis of numerical integration methods can be done using two tools: Period Elongation (PE) and Amplitude Decay (AD) [11,10]. In this section we compute these magnitudes for the numerical results obtained and draw a comparison with respect to the set propagation method. The numerical data used to compute these magnitudes corresponds to a simulation for 50 periods, i.e. 25 s.

The PE can be computed using the following expression:

$$PE = \frac{T_{num} - T_{nat}}{T_{nat}}, \quad (38)$$

where T_{nat} is the natural period of the oscillator and T_{num} is the period of the numerical solution. For the numerical integration methods, we compute T_{num} using standard spectral analysis methods. On the other hand, to our knowledge there are no methods for the computation of PE estimates of flowpipes, thus we devised a procedure to compute T_{num} in the set-based setting. The notions used in the method are presented in Appendix C, where both PE and AD estimations are described.

In Fig. 6a the PEs computed for the Bathe, Newmark and set propagation methods are presented, where the latter is shown in

blue squares and the Bathe and Newmark methods using triangles and circles, respectively. The values of α considered are intended to reproduce the results shown in [10]. It can be seen that Newmark and Bathe methods present PE values, that match with literature results, while the set propagation results do not present PE.

The numerical amplitude A_{num} after n_A periods is $A_{num} = A(1 - AD)^{n_A}$, hence the Amplitude Decay AD is:

$$AD = 1 - \left(\frac{A_{num}}{A} \right)^{\frac{1}{n_A}}. \quad (39)$$

In Fig. 6b the computed values of AD are shown. As it is expected, the Newmark method does not present AD while the Bathe method presents a curve that matches the literature results. The set propagation method has not significant AD, with values below 0.15% for any α .

The results obtained show that the set propagation method presents advantages with respect to the numerical integration methods, in terms of PE, and competitive results in terms of AD. The set propagation method presents good results for single degree of freedom problems. In the following example, the problem is extended to increase the number of unknowns.

4.2. Example 2 - Clamped-Free Bar

We consider a uni-dimensional wave propagation problem from [39]. The purpose of this example is to evaluate the conservative time integration scheme for single initial conditions in a high-dimensional problem. Subsequent examples consider variations in the initial conditions and input parameters.

4.2.1. Problem definition

The domain consists of a bar of length $L = 200$ and cross-section area $A = 1$, formed by a linear elastic material with Young modulus $E = 30 \times 10^6$ and density $\rho = 7.3 \times 10^{-4}$. The free end of the bar is submitted to a step force $F(t) = 10,000H(t)$, where $H(t)$ is the Heaviside function. The bar is governed by the partial differential equation

$$EA \frac{\partial^2 u}{\partial x^2}(x, t) - \rho A \frac{\partial^2 u}{\partial t^2}(x, t) = 0, \quad (40)$$

where $u(x, t)$ is the displacement of the point in position x at time t , considering the axis shown in Fig. 7.

The bar is considered to be initially at rest, with $u(x, 0) = 0$ and $\dot{u}(x, 0) = 0$ for all $x \in [0, L]$. The boundary conditions are $u(0, t) = 0$, corresponding to the fixed end, and $\sigma(L, t)A = F(t)$, for the free end, where $\sigma(x, t) = E \frac{\partial u}{\partial x}(x, t)$ is the stress function at point x and time t .

The analytical solution of this problem in the continuum domain can be obtained using mode superposition [22], and it is given by:

$$u(x, t) = \frac{8FL}{\pi^2 EA} \sum_{s=1}^{\infty} \left\{ \frac{(-1)^{s-1}}{(2s-1)^2} \sin \frac{(2s-1)\pi x}{2L} \left(1 - \cos \frac{(2s-1)\pi \mu t}{2L} \right) \right\}, \quad (41)$$

where $\mu = \sqrt{E/\rho}$.

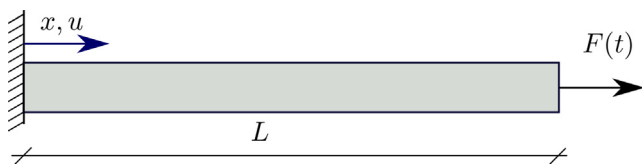


Fig. 7. Example 2: diagram of the clamped-free bar excited by end load.

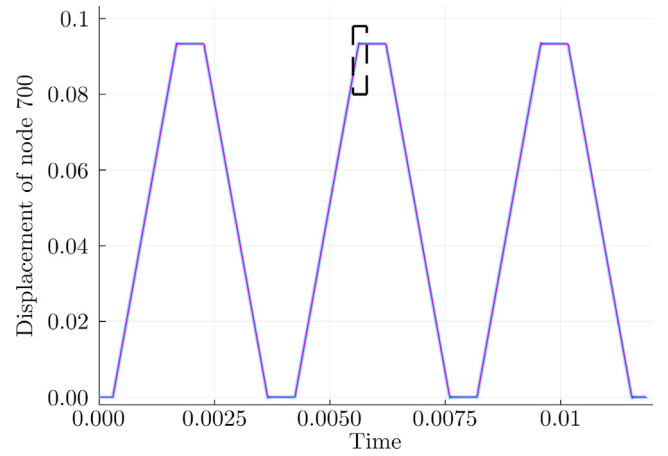


Fig. 8. Example 2: Displacement of node 700 as a function of time corresponding to the analytic solution of the PDE (magenta), the analytic solution of the ODE (blue) and the flowpipe (light-blue). The rectangle shows the zoomed region.

4.2.2. Numerical resolution

The numerical space discretization is done considering $N = 1000$ two-node finite elements with linear interpolation. The following $N \times N$ mass and stiffness matrices are obtained:

$$\mathbf{K} = \frac{EA}{\ell} \begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ 0 & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix}, \quad \mathbf{M} = \frac{\rho A \ell}{2} \begin{bmatrix} 2 & & & & 0 \\ & 2 & & & \\ & & 2 & & \\ & & & \ddots & \\ 0 & & & & 2 & 1 \end{bmatrix},$$

where $\ell = L/N$ is the length of each element. The damping matrix $\mathbf{C}$ in Eq. (5) is a zero matrix of dimension $N \times N$.

The displacements obtained for node 700 are shown in Fig. 8 and in Fig. 9 a zoom at time 5.6×10^{-3} is shown. In this case we present both the analytical solution in the continuum and the analytical solution in the discrete domain. The flowpipe computing using set propagation includes the analytic solution in the space-discretized domain, since that is the exact solution of the system of ODEs.

In Fig. 10 the velocities obtained in node 700 are shown, and in Fig. 11 a zoom is shown. The analytical solution is again within the

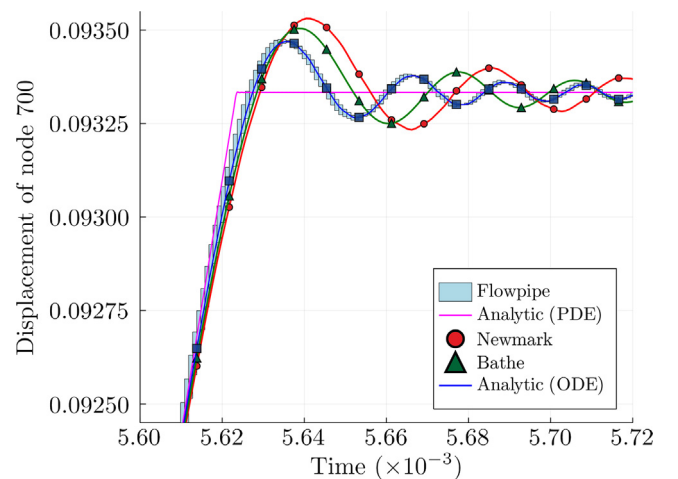


Fig. 9. Example 2: Displacement of node 700 of the clamped beam model as a function of time.

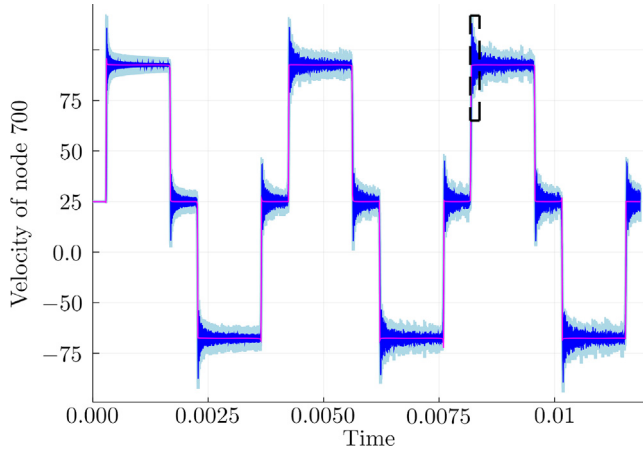


Fig. 10. Example 2: velocity of node 700 as a function of time corresponding to the analytic solution of the PDE (magenta), the analytic solution of the ODE (blue) and the flowpipe (light-blue). The rectangle shows the zoomed region.

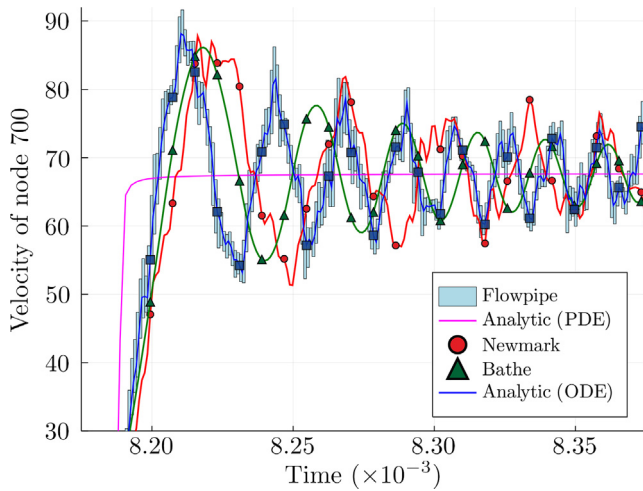


Fig. 11. Example 2: zoom for the velocity of node 700.

flowpipe, confirming the theoretically expected results. In this problem the flowpipe has been obtained using support functions along the node of interest. Execution times are reported in Table 2.

4.3. Example 3 - Two-dimensional wave propagation problem

In this example a two-dimensional wave propagation problem is considered. The geometry and load applied are inspired on the Lamb's Problem considered in [32]. The goals of this example are twofold: comparing the results regarding accuracy for a 2D wave propagation problem, and obtaining results for problems with distributed initial conditions.

4.3.1. Problem definition

The domain geometry and boundary conditions considered are shown in Fig. 12a, where the load is assumed to be given by a step function $P(t) = 2 \times 10^6 H(t)$ (in Newtons), with $H(t)$ being the

Heaviside function. Given the symmetry of the problem, half of the domain is considered. The FEM mesh, shown in Fig. 12b, was generated using GMSH and it is formed by 2235 linear triangular elements and 1180 nodes.

The solid is assumed to be submitted to a plane strain state and formed by a linear elastic isotropic material, with Young modulus $E = 1.8773 \times 10^{10}$ Pa, Poisson ratio $\nu = 0.25$ and density $\rho = 2200 \text{ kg/m}^3$.

The goal of the problem is to compute the horizontal displacement and velocity of the control node A, located at $x_A = 2560$ m and $y_A = 3200$ m, as shown in Fig. 12a. For the initial conditions, two cases are considered: a single initial condition and distributed initial conditions.

4.3.2. Single initial condition analysis

For the first analysis zero displacement and velocity initial conditions are considered, i.e. $\mathbf{u}(0) = \mathbf{0}$ and $\dot{\mathbf{u}}(0) = \mathbf{v}(0) = \mathbf{0}$. For the set propagation method such choice corresponds to $\mathcal{X}_0 = \{\mathbf{0}\}$ and $\mathcal{C}_0 = \{1 \times 10^6\}$, with $q = 1$ and $\alpha = 0$ in Eq. (27).

Given that the goal of the problem is to compute the evolution of a few state variables of the system, support functions are chosen for the set propagation. The time step size was chosen using the formula proposed in [32], then $\delta = \sqrt{\frac{2.3200^2}{2235} \cdot \frac{0.125}{3200}} \approx 0.0037$. The set propagation result was obtained using support functions along the displacements and velocity components of node A. The obtained horizontal displacements of node A for the Set Propagation, Newmark and Bathe methods are shown in Fig. 13a, where in Fig. 13b a zoom for time interval $[0.4, 0.5]$ is shown. The corresponding plots for the velocity of node A are shown in Fig. 14a. In Fig. 14b the zoom for the velocities at time interval $[0.8, 1.0]$ is shown, where it can be observed that numerical methods, and in particular Newmark, provide solutions outside of the Flowpipes (as in the previous example).

The runtimes (and memory usage) of the methods are: Newmark 3.7 s (0.6 GB), Bathe 4.7 s (1.0 GB) and Set Propagation 5.6 s (1.1 GB).

4.3.3. Distributed initial conditions analysis

Let us now consider a case in which the values of initial displacement and velocity of each node are assumed to be included in the intervals $[-\delta_u, \delta_u]$ and $[-\delta_v, \delta_v]$, respectively. This is equivalent to $\mathbf{u}(0) \in \mathcal{U}_0 = [-\delta_u, \delta_u]^{N_{dof}}$ and $\dot{\mathbf{u}}(0) = \mathbf{v}(0) \in \mathcal{V}_0 = [-\delta_v, \delta_v]^{N_{dof}}$, where $N_{dof} = 2322$ is the number of degrees of freedom of the system. In this problem, the radii of the uncertainty intervals, δ_u and δ_v , are considered as the 10 % of the maximum magnitude of displacement and velocity of node A for the previous singleton case, i.e. 3.5×10^{-7} m and 5×10^{-6} m/s.

Each initial condition $(\mathbf{u}_0, \mathbf{v}_0)$ has a corresponding solution for the horizontal velocity at node A, denoted as $v_{x,A}(\mathbf{u}_0, \mathbf{v}_0, t)$. The maximum envelope and minimum envelope functions are defined as:

$$\begin{aligned} v_{env}^+(t) &= \max_{\mathbf{u}_0 \in \mathcal{U}_0, \mathbf{v}_0 \in \mathcal{V}_0} v_{x,A}(\mathbf{u}_0, \mathbf{v}_0, t), \quad \text{and} \quad v_{env}^-(t) \\ &= \min_{\mathbf{u}_0 \in \mathcal{U}_0, \mathbf{v}_0 \in \mathcal{V}_0} v_{x,A}(\mathbf{u}_0, \mathbf{v}_0, t), \end{aligned} \quad (42)$$

respectively, and the absolute envelope is defined as:

$$v_{env}(t) = \max\{|v_{env}^+(t)|, |v_{env}^-(t)|\}. \quad (43)$$

We define the following metrics to evaluate the numerical results:

$$\|v_{env}\|_{L_\infty} = \max_{t \in [0,1]} v_{env}(t), \quad \|v_{env}\|_{L_1} = \int_0^1 |v_{env}(t)| dt. \quad (44)$$

Table 2

Execution times and memory usage for the clamped-free bar example.

Set Propagation	Newmark	Bathe
3.3 s (0.6 GB)	0.7 s (2.5 GB)	1.3 s (4.3 GB)

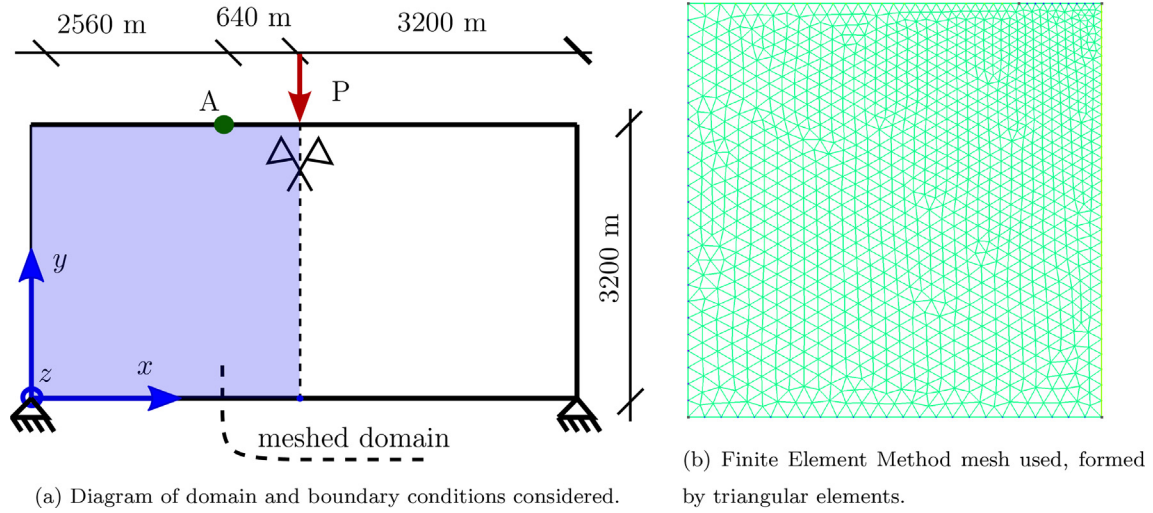


Fig. 12. Example 3: diagram and mesh of two-dimensional wave propagation problem considered.

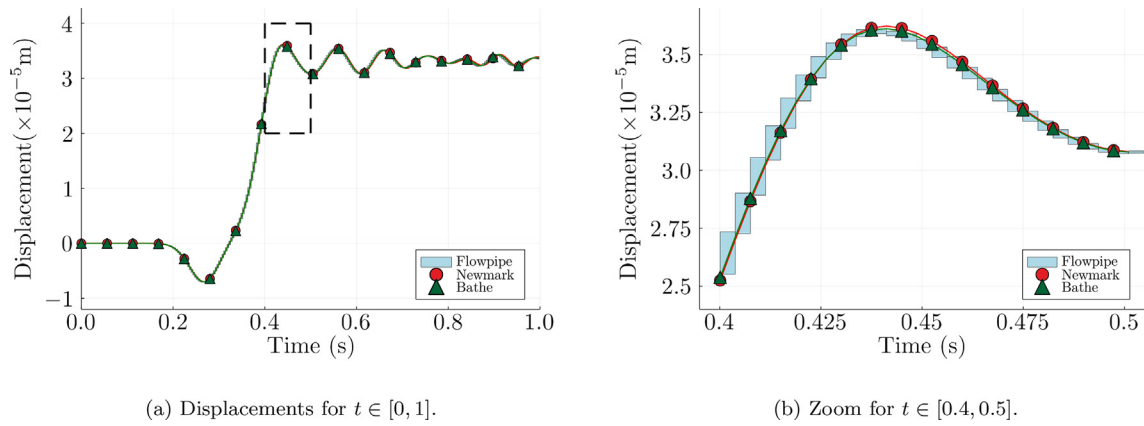


Fig. 13. Example 3: Horizontal displacement at control node A for the wave propagation problem subject to a step-load and a single initial condition.

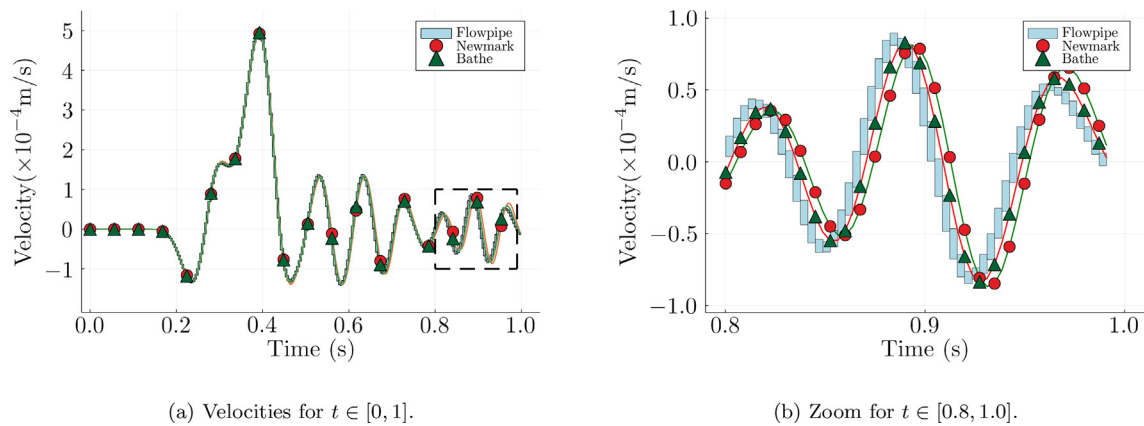


Fig. 14. Example 3: Horizontal velocity at control node A for the wave propagation problem subject to a step-load and a single initial condition.

The Set Propagation method propagates all the initial conditions, thus the computation of v_{env}^+ , v_{env}^- and v_{env} can be done executing a single analysis for $\mathcal{X}_0 = \mathcal{U}_0 \times \mathcal{V}_0$. For numerical integration methods, the most direct approach to compute this would be to perform the simulations for a given number of random

initial points in the initial conditions set. A more efficient approach is to consider only the vertices of the initial states. It can be demonstrated (and it was empirically verified) that this selection strategy provides solutions with higher v_{env} values, which seems the best scenario for numerical integration methods. The number of differ-

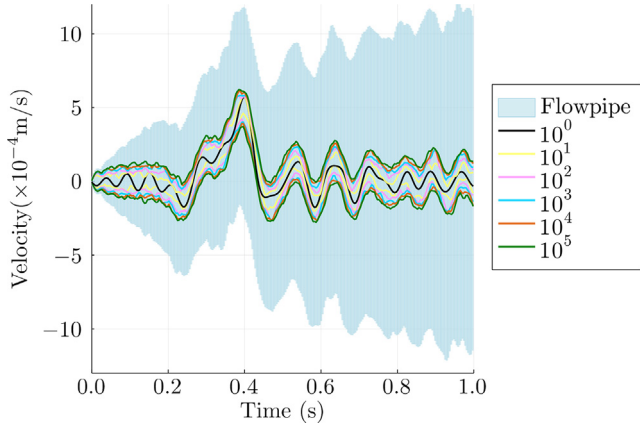


Fig. 15. Example 3: plot of the flowpipe obtained by Set Propagation and the envelope functions v_{env}^+ and v_{env}^- for 1, 10, 100, 1000, 10000, and 100000 random initial conditions using Newmark's method.

Table 3

Example 3: results of execution time and for distributed initial conditions using Newmark method and Set Propagation method.

Method	# Trajectories	Time (s)	$\ v_{env}\ _{L_1}$ (10^{-5})	$\ v_{env}\ _{L_\infty}$ (10^{-5})
Newmark	1	0.3	9.27	56.98
Newmark	10	2.0	13.52	57.53
Newmark	100	17.7	16.61	57.59
Newmark	1000	175.5	18.52	58.22
Newmark	10000	1771.4	19.98	61.18
Newmark	100000	17796.1	21.42	62.21
Set Propagation	-	8.5	81.33	122.25

ent initial conditions considering this vertex constraint is $2^{2322} \approx 10^{698}$, hence only a small fraction of such solutions can be practically evaluated.

Only the Newmark method is used as numerical integration method, considering that similar results would be obtained using Bathe's method. The time-step used is the same as in the previous case. The numerical values of v_{env}^+ and v_{env}^- are computed for the first: 1, 10, 100, 1000, 10000, and 100000 initial conditions, sampling uniformly from the vertices of $\mathcal{U}_0 \times \mathcal{V}_0$.

In Fig. 15, the functions v_{env}^+ and v_{env}^- obtained for the Newmark method and the flowpipe obtained using the Set Propagation method are shown.

The results obtained for the norms of v_{env} and the execution times are shown in Table 3. It can be observed that, as the number of trajectories increases, the norms of the solutions obtained by Newmark, also increase with no clear convergence. On the other hand, the Set Propagation method provides an upper bound that covers all the admissible trajectories.

4.4. Example 4 - One-dimensional heat transfer problem

In this example, a one-dimensional heat transfer problem is considered and interval sets are introduced for the initial values. The main goal of this problem is to evaluate the set propagation method as a tool to analyze the maximum temperature of a heat transfer problem under input uncertainty.

4.4.1. Problem definition

The domain consists of the interval $\Omega = [0, 1]$ and homogeneous Dirichlet boundary conditions are considered on both ends. The

domain is assumed to have $\kappa = 1$ as thermal conductivity, $\rho = 1$ as density and $c = 1$ as specific heat capacity, all assumed uniform and constant.

The governing equations can be written as:

$$\begin{cases} \frac{\partial T(x,t)}{\partial t} = \frac{\kappa}{\rho c} \frac{\partial^2 T(x,t)}{\partial x^2}, & x \in \Omega \\ T(x,t) = 0, & x \in \{0, 1\}, \end{cases} \quad (45)$$

where $T(x,t)$ is the temperature in position x at time t .

For the initial conditions a set of possible temperature profiles is considered. The expression of the initial condition function is given by:

$$T(x,0) = (1 + \varepsilon) \left(\sin(\pi x) + \frac{1}{2} \sin(3\pi x) \right), \quad \varepsilon \in [-0.1, 0.1], \quad (46)$$

where ε is a relative difference parameter associated with a given level of uncertainty in the initial values used for the simulations.

The Finite Element Method is used to discretize the governing equations considering a 100 two-node elements mesh with linear interpolation. It is important to remark that although the FEM discretization is used, the set of initial temperatures contains an infinite number of functions since ε belongs to an interval that is dense in $\mathbb{R}$.

4.4.2. Maximum temperature analysis

The results obtained using the set propagation and the Backward Euler methods are compared with the analytic (exact) solution of Eq. (45). The time-step used for both resolution methods is $\delta = 10^{-5}$. In the case where set propagation is used, the entire set of parameters ε from Eq. (46) is considered, while for the Backward Euler method the analysis is only done for the two extreme cases: $\varepsilon = -0.1$ and $\varepsilon = 0.1$.

The results for the temperature in the node $x = 0.5$ are shown Fig. 16a and temperature profiles at times $t = 0, 0.03$ and $t = 0.10$ are shown in Fig. 16b. We compute the flowpipe using a hyperrectangular representation. Each direction corresponds to a different node. We observe that the flowpipe borders contain and tightly match the numerical solutions from the Backward Euler method and the analytical solutions starting from the two extreme initial conditions.

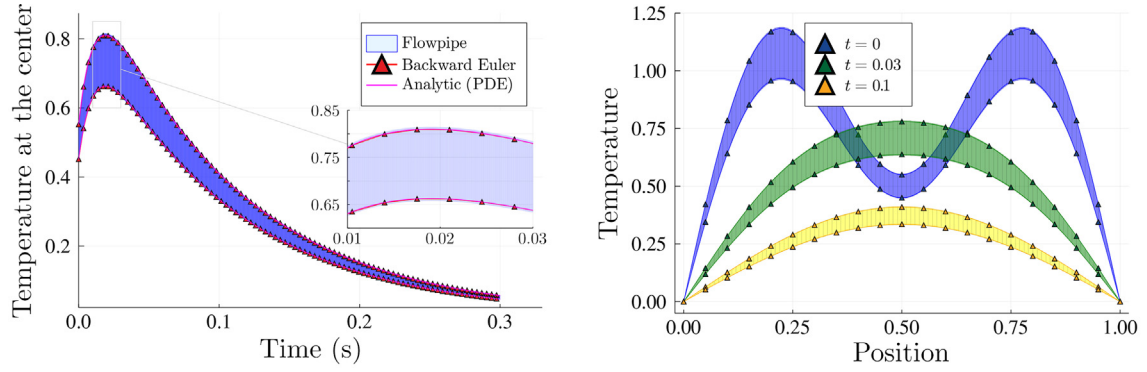
4.4.3. Maximum temperature gradient analysis

In this case, the goal of the analysis is to obtain the maximum value of the temperature gradients among all the nodes of the domain for all the admissible initial condition profiles.

Since it is not clear how to choose the initial temperature profile to obtain an extreme temperature gradient at a given time, here we repeated the calculations with Backward Euler using several different initial temperature profiles (a thousand) chosen randomly from Eq. (46). The Set Propagation method allows to compute all the solutions for the gradients, in this case, using support functions.

The results obtained for the temperature gradient profile at time $t = 0.001$ are shown in Fig. 17a and the time evolution of the temperature gradient at $x = 0.66$ are shown in Fig. 17b. We observe that the flowpipe contains all the trajectories calculated with Backward Euler. We remark that statistical strategies are needed to estimate the temperature gradient bounds if we use the Backward Euler method. On the other hand, the flowpipe provides safe bounds for this gradient with only one integration. We conclude that the Set Propagation method is an effective tool to evaluate the maximum temperature under uncertainty.

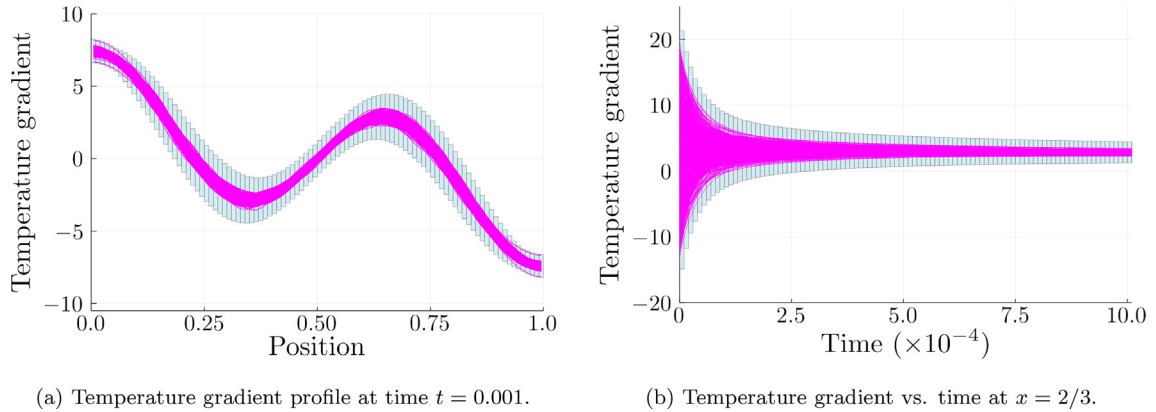
The execution times are reported in Table 4.



(a) Temperature evolution obtained at the center node of the mesh ($x = 0.5$) using the analytic solution (magenta), the Backward Euler method for extreme cases (red triangle), and the set propagation method (blue flowpipe).

(b) Temperature profiles computed at different time points: $t = 0$ (blue), $t = 0.03$ (green) and $t = 0.1$ (yellow). Results from the Backward Euler method at extreme cases are shown using triangle markers.

Fig. 16. Example 4: Results for the one-dimensional heat problem with set initial conditions obtained with the set propagation method and with the Backward Euler method.



(a) Temperature gradient profile at time $t = 0.001$.

(b) Temperature gradient vs. time at $x = 2/3$.

Fig. 17. Example 4: Flowpipe (light blue) calculation of the temperature gradient for the same case study from Fig. 16. Superposed are shown several temperature gradients computed with the Backward Euler method (magenta) with different initial states chosen randomly within the given uncertainty in the initial temperature profile.

Table 4

Execution times and memory usage for the one-dimensional heat problem.

	Set Propagation	B. Euler (1 trajectory)	B. Euler (1000 trajectories)
Profile	2.6 (113 MB)	0.42 s (107 MB)	-
Gradient	1.1 s (22 MB)	-	17 s (7.1 GB)

4.5. Example 5 - concrete casting heat of hydration

In this example a model for heat of hydration during casting of a massive concrete structure is considered. The development of reliable computational models for this kind of processes is a greatly challenging task due to the uncertainty in parameters such as the internal heat generation. In [49], for instance, different levels of heat of hydration (associated with different sizes of aggregate) are considered. The parameter values and the modeling assumptions considered are based on [13,47].

4.5.1. Problem definition

The domain considered consists in a square cuboid $\Omega = [-1, 1] \times [-1, 1] \times [0, 1]$, with 1 m height (in the z direction) and 2 m width in the other directions (x and y). In Fig. 18 the geom-

etry, boundary conditions and control points of the problem are shown.

The concrete density, the specific heat and the conductivity are considered $\rho = 2485 \text{ kg/m}^3$, $c = 0.967 \text{ kJ/(kg } ^\circ\text{C)}$ and $\kappa = 9.37 \text{ kJ/(m h } ^\circ\text{C)}$, respectively.

Regarding the boundary conditions, on the upper face ($z = 1$) a convection condition is assumed with value $h_{\text{air}} = 40 \text{ kJ/(m}^2 \text{ h } ^\circ\text{C)}$, while on the other faces it is $h_{\text{timb}} = 500 \text{ kJ/(m}^2 \text{ h } ^\circ\text{C)}$. The external temperature $T_\infty(t)$ dependence in time is assumed to be given by the following sinusoidal law:

$$T_\infty(t) = T_{\min} + T_{\text{var}} \left(\frac{1}{2} + \frac{1}{2} \sin \left(\omega t - \frac{\pi}{2} \right) \right), \quad (47)$$

where $T_{\min} = 17^\circ\text{C}$ is the minimum temperature, T_{var} is the variation or difference between the minimum and maximum temperatures and $\omega = \pi/12 \text{ h}^{-1}$ is the angular frequency. The initial temperature of all the points in the concrete is assumed $T(\mathbf{x}, 0) = T_{\min}$, which is equal to the initial external temperature $T_\infty(0)$.

The accumulated heat of hydration is assumed to be given by the exponential law:

$$Q_{\text{AH}}(t) = Q_{\text{FH}}(1 - e^{-mt}), \quad (48)$$

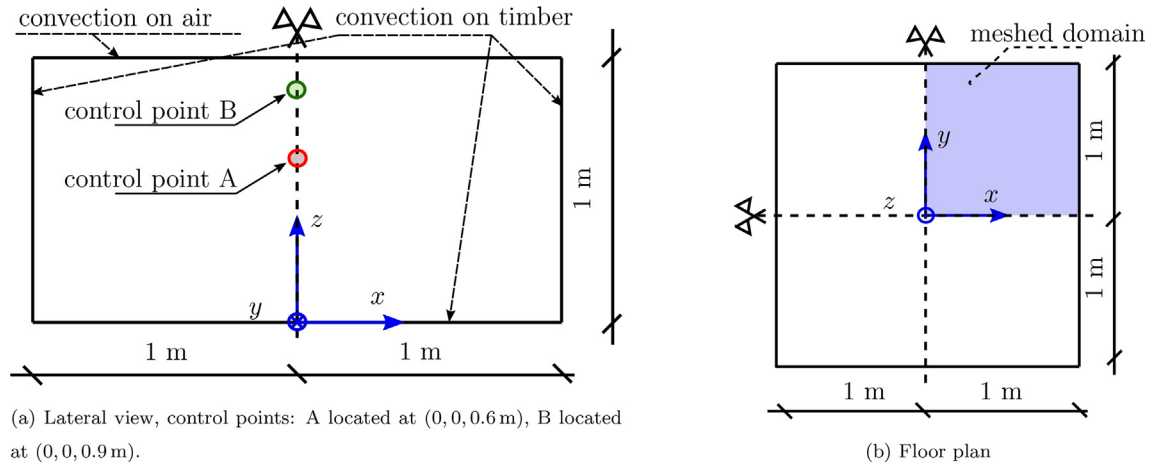


Fig. 18. Example 5: diagram of domain, boundary conditions and control points.

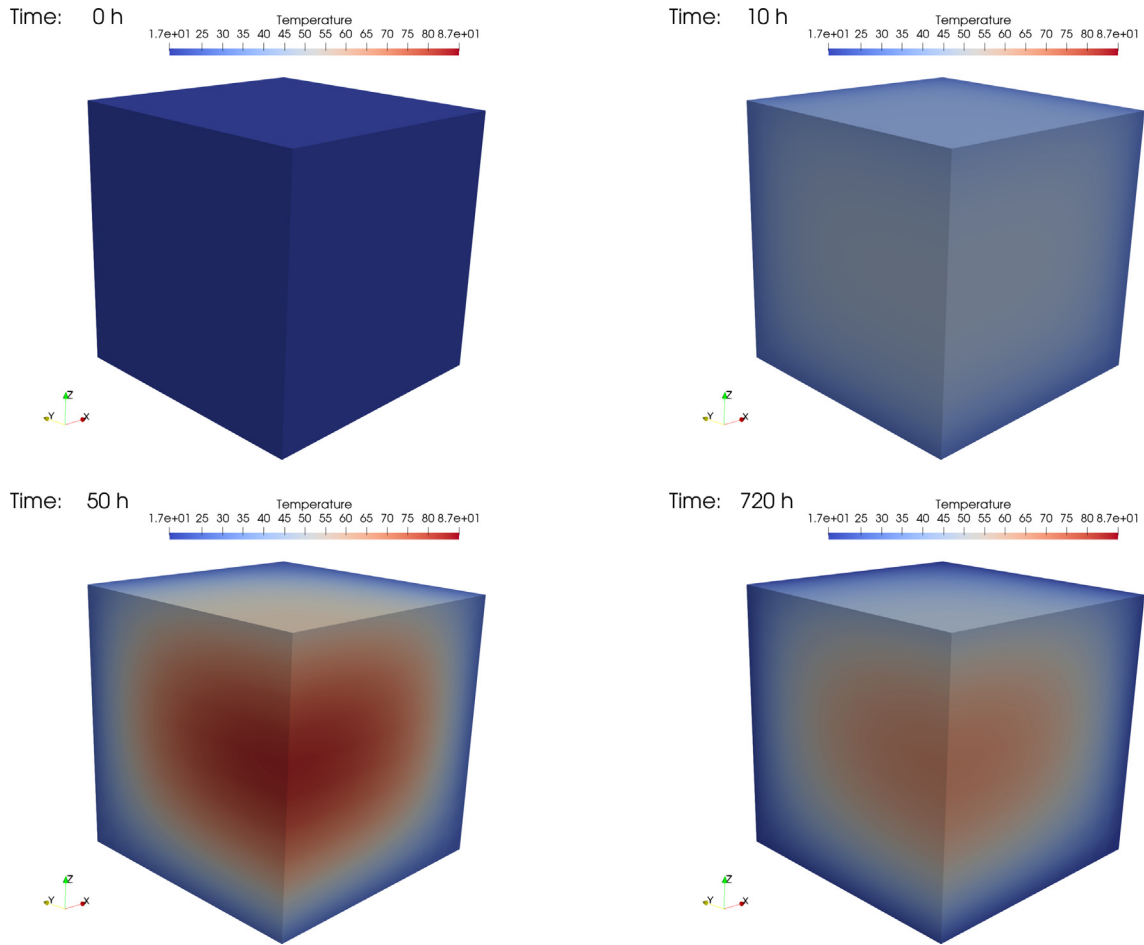


Fig. 19. Example 5: Evolution of the temperature field in the meshed domain at four different observation times: 0 h (top-left), 10 h (top-right), 50 h (bottom-left) and 720 h (bottom-right).

where Q_{FH} is the final heat of hydration in kJ/kg and $m = 7.95 \times 10^{-3} \text{ h}^{-1}$. This quantity is multiplied by the density, derived and included in the thermal-work Eq. (4) as:

$$Q_{int}(t) = Q_{FH} \cdot \rho \cdot m e^{-mt}. \quad (49)$$

4.5.2. Numerical resolution with fixed parameters

Let us consider the following parameters: $Q_{FH} = 330 \text{ kJ/kg}$ and $T_{var} = 6 \text{ }^\circ\text{C}$. We use a time step $\Delta t = 1/3 \text{ h}$ and the final time of

the simulation is 240 h (10 days). Numerical time integration is done using Backward Euler. The temperature fields obtained at times 0, 10, 50 and 720 h are shown in Fig. 19. Let us remark that by symmetry, only one quarter of the domain is meshed, and a regular structured mesh of 1331 nodes and 6000 four-node tetrahedron elements is considered.

In order to apply the set propagation method, the source term is factorized using the formulation described in Section 3.1.3. The source term for this model at time t can be written as:

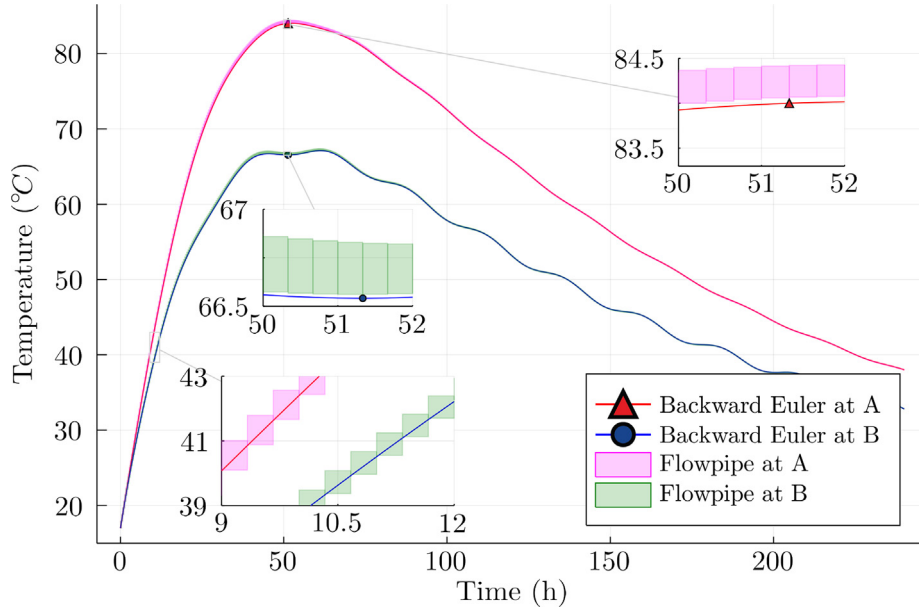


Fig. 20. Example 5: temperature evolution at control points A and B.

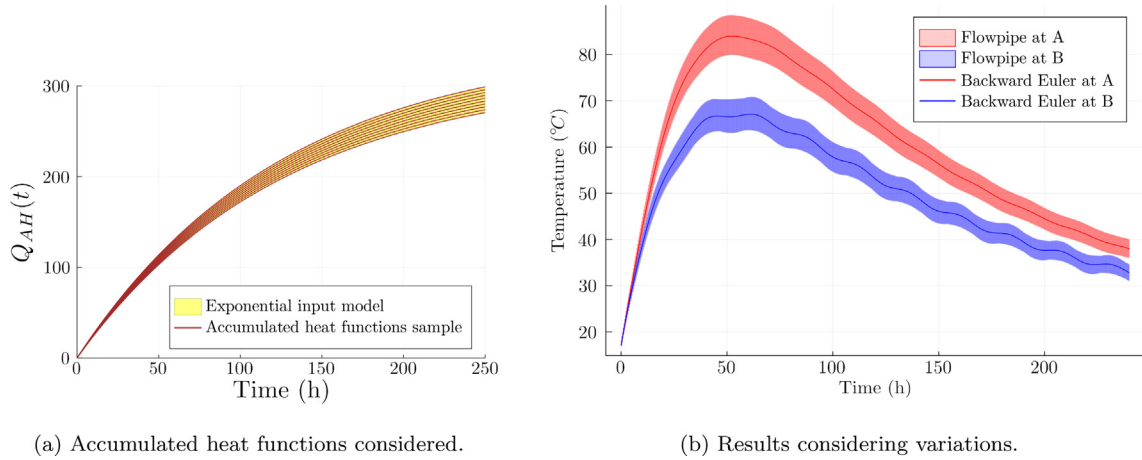


Fig. 21. Example 5: set propagation results for the heat of hydration model under variation in heat sources.

$$\mathbf{f}_\theta(t) = \mathbf{f}_Q Q_{FH} \rho m e^{-mt} + \mathbf{f}_{T_\infty} T_{var} \frac{1}{2} \sin\left(\omega t - \frac{\pi}{2}\right) + \mathbf{f}_{T_\infty} \left(T_{min} + T_{var} \frac{1}{2}\right), \quad (50)$$

where $\mathbf{f}_Q$ and $\mathbf{f}_{T_\infty}$ are vectors given by the finite element discretization integration. The terms in Eq. (50) can be factorized to the form of Eq. (25) using:

$$\begin{cases} \mathbf{f}_0^{(1)} = \mathbf{f}_{T_\infty} & \eta^{(1)}(t) = T_{min} + T_{var} \frac{1}{2} \\ \mathbf{f}_0^{(2)} = \mathbf{f}_Q \cdot \rho \cdot m & \eta^{(2)}(t) = Q_{FH} e^{-mt} \\ \mathbf{f}_0^{(3)} = \mathbf{f}_{T_\infty} & \eta^{(3)}(t) = T_{var} \frac{1}{2} \sin\left(\omega t - \frac{\pi}{2}\right). \end{cases} \quad (51)$$

The function $\eta^{(1)}(t)$ can be considered as the solution of the differential equation $\dot{y} = 0$ with initial condition $T_{min} + T_{var} \frac{1}{2}$, the function $\eta^{(2)}$ is the solution of the differential equation $\dot{y} = -my$ with initial condition Q_{FH} and the function $\eta^{(3)}$ is the solution of the differential equation $\dot{y} + \omega^2 y = 0$ with initial conditions $y(0) = -T_{var} \frac{1}{2}$ and $\dot{y}(0) = 0$.

This can be written in matrix form as:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\xi}_1^{(1)} \\ \dot{\xi}_1^{(2)} \\ \dot{\xi}_1^{(3)} \\ \dot{\xi}_2^{(3)} \end{bmatrix} = \begin{bmatrix} -\mathbf{C}^{-1}\mathbf{K} & -\mathbf{C}^{-1}\mathbf{f}_{T_\infty} & -\mathbf{C}^{-1}\mathbf{f}_Q \rho m & -\mathbf{C}^{-1}\mathbf{f}_{T_\infty} & \mathbf{0} \\ \mathbf{0} & 0 & 0 & 0 & 0 \\ \mathbf{0} & 0 & -m & 0 & 0 \\ \mathbf{0} & 0 & 0 & 0 & 1 \\ \mathbf{0} & 0 & 0 & -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \xi_1^{(1)} \\ \xi_1^{(2)} \\ \xi_1^{(3)} \\ \xi_2^{(3)} \end{bmatrix}, \quad (52)$$

Let us solve the problem considering fixed values for the parameters Q_{FH} and T_{var} . In that case, the initial values for the input variables ξ are single fixed values. The evolution of the temperatures obtained in control points A and B is shown in Fig. 20.

We observe that the Backward Euler method may underestimate the temperature values, as shown in the zoomed region in Fig. 20 for times within the interval [50, 52]h. The flowpipe is computed using support functions along the canonical directions associated to the control points A and B.

Table 5

Execution times and memory usage for the concrete casting heat of hydration example. The column for Backward Euler corresponds to solving for one initial condition.

	Set Propagation	Backward Euler
Case 1 (fixed parameters)	2.05 s (373 MB)	3.7 s (78 MB)
Case 2 (variation of heat sources)	3.1 s (555 MB)	3.8 s (78 MB)

4.5.3. Set propagation under variation in heat sources

Let us consider a variation in the parameters Q_{FH} and T_{var} . For Q_{FH} , a 5% error interval is used, then $Q_{FH} \in [313.5, 346.5] \text{ kJ/kg}$, while for T_{amb} , a 2°C variation is considered, then $T_{amb} \in [4, 8]^\circ\text{C}$. These intervals are used to construct the set $\mathcal{C}_0$ of initial values of the input variables. A plot of the set of accumulated heat functions is shown in Fig. 21a. The flowpipes corresponding to the evolution of the temperature in the control nodes are shown in Fig. 21b. The solution obtained with Backward Euler for the mean values of the parameters (as in Section 4.5.2) is represented with solid lines.

Execution times are reported in Table 5. We remark that the computational cost of the set propagation is approximately the same whether we use fixed parameters or parameters with uncertainty. Therefore, reachability analysis becomes an efficient option for safety verification in the design of massive concrete structures under uncertainty.

5. Conclusions

In this article a new approach for numerically solving the differential equations that govern linear heat transfer and structural dynamics problems is presented. The approach is based on recent advances in reachability analysis for high-dimensional linear systems. Our work is an attempt to bridge the gap between spatially discretized equations using the Finite Element Method and the incipient field of set-based numerical integration. The results of our approach are compared with reference numerical time integration methods through the resolution of five examples.

Given a system of differential equations with a set of admissible initial conditions, the set propagation method consists in computing an initial set that contains all possible trajectories within a time interval $[0, \delta]$, and propagating that set using set-based techniques. The union of the obtained sets is called *flowpipe*. Let us remark that, while standard numerical integration provides single values for each time point, our approach returns a *set* for each time interval that is guaranteed to contain all possible solutions of the differential equation for the given initial state(s) and input(s). We formalized the idea of homogenization of the governing equations subject to set-based initial conditions and inputs. We also presented a simple modification of an existing theoretical result for conservative time discretization, which ensures more accurate results at a marginal increase in the computational cost.

In Example 1 a fundamental test problem of a single degree of freedom oscillator was considered. This problem is typically used to evaluate numerical properties such as Period Elongation (PE) and Amplitude Decay (AD). The results obtained let us conclude that the set propagation method has neither PE nor AD, therefore it provides accurate results.

In Example 2 a reference uni-dimensional wave propagation problem was considered. In this example we found that the proposed method can handle a relatively high number of degrees of freedom. Execution times are in same order for the three methods evaluated. However, considerably less memory usage is required by the set propagation method, since it computes the solution only at a specific node.

In Example 3 a two-dimensional wave propagation problem was considered. For single initial conditions the three compared methods provide accurate results with similar execution times and memory usage. For distributed initial conditions, our approach is able to compute, in a single execution, an envelope of the solutions at a specified control node for all admissible conditions. On the other hand, the results obtained for the numerical integration methods, let us infer that a prohibitively large number of simulations would be required to provide a fair estimation of the envelope.

In Example 4 the Set Propagation and Backward Euler methods are compared in a unidimensional heat transfer problem. A dense set of spatially distributed initial condition functions is considered. The obtained flowpipe accurately encloses the analytic solution for all initial functions considered. Moreover, we computed the gradient of the solution using a support function approach. While numerical integration approaches such as Backward Euler require to perform several simulations for different initial functions, the proposed approach provides a robust enclosure with a single set-based integration. Let us remark that computing a high number of simulations to ensure a very high degree of coverage is computationally demanding due to the curse of dimensionality. However, the set propagation method provides robust results in a single analysis for the solution combination of the solution hence the cost can be drastically reduced.

Finally, the results obtained in Example 5 let us conclude that the set propagation method can be applied to a realistic three-dimensional massive concrete hydration problem. The problem includes a combination of different types of heat sources. The obtained flowpipe is accurate in comparison with numerical solutions, showing that the homogenization strategy can correctly handle various sources of input uncertainty. We observed that for some time intervals the solution obtained using the Backward Euler method underestimates the maximum temperature. Contrary to Example 2, in this problem the memory usage is higher for the Set Propagation method than the reference numerical method, since the flowpipe was computed at all nodes. Given the theoretical basis of reachability analysis, and the accuracy of our results, we conclude that set propagation may be a useful tool for the design of massive concrete structures.

It is important to recall that the solution of the FEM equations, or spatially discrete system of ODEs, is already in error when compared with the original exact problem in the continuum. Estimating and absorbing such error in a set-based fashion was not addressed in this article. Moreover, it is well-known that high order modes are artifacts of the discretization process, thus numerical or algorithmic dissipation is often a favorable or even required mechanism to improve the accuracy of the results when compared to the exact problem. On the other hand, solutions obtained by reachability analysis are not numerically damped, therefore do not include artificial damping behavior. This, apart from being an advantage of the method itself, may be useful to quantify how numerical solutions depart from the true trajectories of the FEM equations.

Several research directions remain open. It would be enlightening to apply the developed method in real engineering problems, in particular in the context of Example 5. For that purpose the theoretical development of other, more general input models might be of interest, as well as the evaluation of the computational cost when the size of the problems increase. The set-based approach might be extended to other scientific domains such as inverse problems and non-linear structural analysis. It would also be interesting to extend this method to integrate the differential equations given by other finite element approaches, such as Overlapping Finite Elements or the Extended Finite Element Method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the *Comisión Sectorial de Investigación Científica* of *Universidad de la República* and the *Agencia Nacional de Investigación e Innovación* for the funding of the *Timbó* portal. The authors gratefully acknowledge comments from two anonymous reviewers, to Christian Schilling for valuable discussions at various stages of this work, to Pedro Curto for useful comments in heat transfer analysis, and to Ander Gray for comments on non-probabilistic approaches.

Appendix A. Proof of Proposition 1

Using the forward-only transformations in [15] applied to the special case when the system is homogeneous,

$$\Omega_0^+ = CH(\mathcal{X}_0, \Phi \mathcal{X}_0 \oplus E^+(\mathbf{A}, \mathcal{X}_0, \delta)), \quad (\text{A.1})$$

where $E^+(\mathbf{A}, \mathcal{X}_0, \delta) = (\mathbf{P}(|\mathbf{A}|, \delta)(\mathbf{A}^2 \mathcal{X}_0))$ and $\mathbf{P}(\mathbf{A}, \delta) = \sum_{i=0}^{\infty} \mathbf{A}^i \delta^{i+2} / (i+2)!$, $\delta > 0$.

The exact reachable set at the time point $t = \delta$ is $\mathcal{R}^e(\mathcal{X}_0, \delta) = \Phi \mathcal{X}_0$. Under the transformation $t \mapsto -t$, the state transition matrix of the linear system (14) becomes $\Phi^{-1} = e^{-\mathbf{A}\delta}$. Therefore the reachable states for the time interval $[0, \delta]$ can be enclosed using Eq. (A.1) backwards in time:

$$\begin{aligned} \Omega_0^- &= CH(\Phi \mathcal{X}_0, \Phi^{-1} \Phi \mathcal{X}_0 \oplus E^+(-\mathbf{A}, \Phi \mathcal{X}_0, \delta)) \\ &= CH(\Phi \mathcal{X}_0, \mathcal{X}_0 \oplus E^+(\mathbf{A}, \Phi \mathcal{X}_0, \delta)). \end{aligned} \quad (\text{A.2})$$

Since both Ω_0^+ and Ω_0^- are sound overapproximations of $\mathcal{R}^e(\mathcal{X}_0, [0, \delta])$, we conclude that

$$\mathcal{R}^e(\mathcal{X}_0, [0, \delta]) \subseteq \Omega_0^+ \cap \Omega_0^-.$$

The convergence property follows from Lemma 3 in [21].

Appendix B. Proof of Proposition 2

Let $H_0 = \langle \mathbf{c}_0, \mathbf{r}_0 \rangle_H$ be the initial hyperrectangle. It can be represented as a zonotope with a diagonal generators matrix, $Z_0 = \langle \mathbf{c}_0, \mathbf{D}_0 \rangle_Z$ where $\mathbf{D}_0$ is the diagonal matrix whose entries are the elements of $\mathbf{r}_0$. Using Eq. (36), let $Z_k = \langle \mathbf{c}_k, \mathbf{G}_k \rangle_Z$ such that $\mathbf{c}_k = \Phi^k \mathbf{c}_0$ and $\mathbf{G}_k = \Phi^k \mathbf{D}_0$ for each $k \geq 0$. From Prop. 2.2 in [35], the support function of $Z = \langle \mathbf{c}, \mathbf{G} \rangle_Z$ along direction $\mathbf{d} \in \mathbb{R}^n$ is

$$\rho(\mathbf{d}, Z) = \mathbf{d}^T \mathbf{c} + \|\mathbf{G}^T \mathbf{d}\|_1 = \sum_{i=1}^n d_i c_i + \sum_{i=1}^n \left| \sum_{j=1}^n (\mathbf{G})_{ji} d_j \right|. \quad (\text{B.1})$$

Let $\mathbf{d} = \pm \mathbf{e}_i$ for any $i = 1, \dots, n$ be a canonical direction (and its opposite). Then,

$$\begin{aligned} \rho(\pm \mathbf{e}_i, Z_k) &= (\pm \mathbf{c}_k)_i + \sum_{i=1}^n |(\mathbf{G}_k)_{ii}| = (\pm \Phi^k \mathbf{c}_0)_i + \sum_{i=1}^n |(\Phi^k \mathbf{D}_0)_{ii}| \\ &= (\pm \Phi^k \mathbf{c}_0)_i + (\|\Phi^k\| \mathbf{r}_0)_i. \end{aligned} \quad (\text{B.2})$$

Therefore, the radius of the tight hyperrectangular overapproximation of $Z_k, H_k = \langle \mathbf{c}_k, \mathbf{r}_k \rangle_H$, along direction $\mathbf{e}_i$ is

$$(\mathbf{r}_k)_i = \frac{\rho(\mathbf{e}_i, Z_k) + \rho(-\mathbf{e}_i, Z_k)}{2} = (\|\Phi^k\| \mathbf{r}_0)_i,$$

which concludes the proof.

Appendix C. Estimation of PE and AD in Flowpipes

In this section we describe a numerical procedure for the computation of Period Elongation (PE) and Amplitude Decay (AD) for the set propagation method results presented in Example 1.

This problem can be reduced to the computation of T_{num} for Eq. (38) and A_{num} for Eq. (39). In order to compute amplitudes and periods, maximums of the flowpipe solution have to be identified.

Let us consider a flowpipe $\mathcal{F}$ formed by the union of a finite number of reach-sets $\{X_k\}_k$. The relative maximum of the displacement happens when the conditions $u(t) > 0$ and $v(t) = 0$ are met. The intersection of $\mathcal{F}$ with the set $U^+ = \{(u, v) \in \mathbb{R}^2 : (u > 0) \wedge (v = 0)\}$ results in one or several reach-sets such that their intersection with U^+ is non-empty, i.e. they include the behavior that satisfies the maximum displacement conditions (if more than one reach-set intersects U^+ , we take their set union).

To estimate a period T_{num} to the flowpipe $\mathcal{F}$ we proceed as follows. Let $\mathcal{T}_m$ be the time interval associated with the m -th intersection of $\mathcal{F}$ with U^+ . Then, we have $m \cdot T_{num} \in \mathcal{T}_m$ for any period m , therefore we can define the following time interval:

$$S_T = \bigcap_m \mathcal{T}_m \frac{1}{m}. \quad (\text{C.1})$$

Since S_T is an intersection of intervals, all containing T_{num} , then

$$T_{num} \in S_T. \quad (\text{C.2})$$

We repeated the same example for different values of α and calculated S_T in each case using 1000 periods. The value of *PE* reported in Fig. 6a corresponds to the maximum of T_{num} among all periods.

Regarding the estimation of the amplitude, using the time intersection in Eq. (C.1) we compute an interval of displacements. The lower value of displacement represents a safe estimation of the amplitude, and also the worst possible case for the Amplitude Decay. This is the value considered for the computation of *AD* in Fig. 6b.

References

- [1] Ahrens J, Geveci B, Law C. Paraview: An end-user tool for large data visualization. The visualization handbook. 2005, 717.
- [2] Althoff M. An introduction to cora 2015. In: Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems, 2015.
- [3] Althoff M. Reachability analysis of large linear systems with uncertain inputs in the krylov subspace. *IEEE Trans Autom Control* 2019;65:477–92.
- [4] Althoff M, Bak S, Bao Z, Forets M, Frehse G, Freire D, Kochdumper N, Li Y, Mitra S, Ray R, et al. ARCH-COMP20 category report: Continuous and hybrid systems with linear continuous dynamics. *EPiC Series Comput* 2020;74:16–48.
- [5] Althoff M, Frehse G. Combining zonotopes and support functions for efficient reachability analysis of linear systems. In: 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE; 2016. p. 7439–446.
- [6] Althoff M, Frehse G, Girard A. Set propagation techniques for reachability analysis. *Ann Rev Control, Robot Auton Syst* 4:2020b.
- [7] Althoff M, Stursberg O, Buss M. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In: 2008 47th IEEE Conference on Decision and Control. IEEE; 2008. p. 4042–048.
- [8] Asarin E, Dang T, Girard A. Reachability analysis of nonlinear systems using conservative approximation. In: International Workshop on Hybrid Systems: Computation and Control. Springer; 2003. p. 20–35.
- [9] Bak S, Tran H, Johnson TT. Numerical verification of affine systems with up to a billion dimensions. In: HSCC. ACM; 2019. p. 23–32.
- [10] Bathe K-J. *Finite Element Procedures*. 2nd ed. USA: Watertown; 2014.
- [11] Bathe K-J, Noh G. Insight into an implicit time integration scheme for structural dynamics. *Comput Struct* 2012;1–6. <https://doi.org/10.1016/j.compstruc.2012.01.009>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045794912000107>.

- [12] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. *SIAM Rev* 2017;59:65–98.
- [13] Bofang Z. Thermal Stresses and Temperature Control of Mass Concrete. Elsevier; 2014. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0020060383>. doi:10.1016/C2012-0-06038-3.
- [14] Bogomolov S, Forests M, Frehse G, Potomkin K, Schilling C. JuliaReach: a toolbox for set-based reachability. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control; 2019. p. 39–44.
- [15] Bogomolov S, Forests M, Frehse G, Viry F, Podelski A, Schilling C. Reach set approximation through decomposition with low-dimensional sets and high-dimensional matrices. In: Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week); 2018. p. 41–50.
- [16] Breloff T, Schwabeneder D, Borregaard MK, Christ S, Heinen J, Yuval, et al. JuliaPlots/Plots.jl: v1.13.2; 2021. URL: doi: 10.5281/zenodo.4725318. doi:10.5281/zenodo.4725318.
- [17] Capillon R, Desceliers C, Soize C. Uncertainty quantification in computational linear structural dynamics for viscoelastic composite structures. *Comput Methods Appl Mech Eng*. 305;2016:154–72. URL: <https://www.sciencedirect.com/science/article/pii/S0045782516300949>. doi: 10.1016/j.cma.2016.03.012.
- [18] Chen X, Ábrahám E, Sankaranarayanan S. Flow*: An analyzer for non-linear hybrid systems. In: International Conference on Computer Aided Verification. Springer; 2013. p. 258–63.
- [19] Clough R, Penzien J. *Dynamics of Structures*. McGraw-Hill; 1993.
- [20] Forests M, Schilling C, Caporale DF, Benet L, Sanders DP, Guadalupe S. JuliaReach, 2021. URL: <https://github.com/JuliaReach/>.
- [21] Frehse G, Guernic CL, Donzé A, Cotton S, Ray R, Lebeltel O. et al. SpaceX: Scalable verification of hybrid systems. In: International Conference on Computer Aided Verification. Springer; 2011. p. 379–95.
- [22] Gérardin M, Rixen DJ. *Mechanical vibrations: theory and application to structural dynamics*. John Wiley & Sons; 2014.
- [23] Geuzaine C, Remacle J-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int J Numer Meth Eng* 2009;79:1309–31. <https://doi.org/10.1002/nme.2579>.
- [24] Girard A, Guernic CL, Maler O. Efficient computation of reachable sets of linear time-invariant systems with inputs. In: HSCC. Springer; 2006a. p. 257–71.
- [25] Girard A, Le Guernic C, Maler O. Efficient computation of reachable sets of linear time-invariant systems with inputs. In: International workshop on hybrid systems: Computation and control. Springer; 2006b. pp. 257–71.
- [26] Grau-Bartual S, Al-Jumaily AM. Prediction of upper airway dryness and optimal continuous positive airway pressure conditions. *J Biomech* 2020;112:110037. <https://doi.org/10.1016/j.jbiomech.2020.110037>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0021929020304619>.
- [27] Haiqing L, Jinli W, Guijun W. The Mass Concrete Temperature Simulation Analysis with Fuzzy Factors. In: Fuzzy Information and Engineering 2010. p. 599–605. https://doi.org/10.1007/978-3-642-14880-4_66. URL: http://link.springer.com/10.1007/978-3-642-14880-4_66.
- [28] Henzinger TA, Ho P-H, Wong-Toi H. Algorithmic analysis of nonlinear hybrid systems. *IEEE Trans Automatic Control* 1998;43:540–54.
- [29] Hornshøj-Møller SD, Nielsen PD, Foroughi P, Abkar M. Quantifying structural uncertainties in Reynolds-averaged Navier-Stokes simulations of wind turbine wakes. *Renewable Energy* 2021;164:1550–8. <https://doi.org/10.1016/j.renene.2020.10.148>. URL: <https://www.sciencedirect.com/science/article/pii/S0960148120317237>.
- [30] Hughes TJR. *The Finite Element Method*. (1st ed.). Prentice-Hall; 1987.
- [31] Imholz M, Faes M, Vandepitte D, Moens D. Robust uncertainty quantification in structural dynamics under scarce experimental modal data: A bayesian-interval approach. *J Sound Vib* 2020;467:114983. <https://doi.org/10.1016/j.jsv.2019.114983>. URL: <https://www.sciencedirect.com/science/article/pii/S0022460X19305450>.
- [32] Kim K-T, Bathe K-J. Accurate solution of wave propagation problems in elasticity. *Comput Struct* 2021;249:106502. <https://doi.org/10.1016/j.compstruc.2021.106502>. URL: <https://www.sciencedirect.com/science/article/pii/S0045794921000249>.
- [33] Kim W, Choi SY. An improved implicit time integration algorithm: The generalized composite time integration algorithm. *Comput Struct* 2018;196:341–54. <https://doi.org/10.1016/j.compstruc.2017.10.002>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045794917303413>.
- [34] Le Guernic C. Reachability analysis of hybrid systems with linear continuous dynamics. Ph.D. thesis Université Grenoble 1 - Joseph Fourier; 2009.
- [35] Le Guernic C. Reachability analysis of hybrid systems with linear continuous dynamics. Ph.D. dissertation, Université Joseph-Fourier-Grenoble I; 2009.
- [36] Le Guernic C, Girard A. Reachability analysis of linear systems using support functions. *Nonlinear Anal: Hybrid Syst* 2010;4:250–62.
- [37] Le Guernic C, Girard A. Reachability analysis of linear systems using support functions. *Nonlinear Anal: Hybrid Syst* 2010;4:250–62. IFAC World Congress 2008.
- [38] Li E, He Z, Tang Q, Zhang G. Large time steps in the explicit formulation of transient heat transfer. *Int J Heat Mass Transf* 2017;108:2040–52. <https://doi.org/10.1016/j.ijheatmasstransfer.2017.01.065>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0017931016324607>.
- [39] Malakiyeh MM, Shojaei S, Bathe K-J. The Bathe time integration method revisited for prescribing desired numerical dissipation. *Comput Struct* 2019;212:289–98. <https://doi.org/10.1016/j.compstruc.2018.10.008>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0045794918310046>.
- [40] Martinielli L, Perotti F. Numerical analysis of the non-linear dynamic behaviour of suspended cables under turbulent wind excitation. *Int J Struct Stab Dyn* 2001;1:207–33.
- [41] Muhanna RL, Mullen RL. Formulation of fuzzy finite-element methods for solid mechanics problems. *Comput -Aided Civil Infrastruct Eng* 1999;14:107–17.
- [42] Muhanna RL, Mullen RL. Interval based finite elements for uncertainty quantification in engineering mechanics. In: IFIP Working Conference on Uncertainty Quantification. Springer; 2011. p. 265–79.
- [43] Rackauckas C, Nie Q. Differentialequations.jl - a performant and feature-rich ecosystem for solving differential equations in julia. *The J Open Res Software*, 5;2017. URL: <https://app.dimensions.ai/details/publication/pub.1085583166> and <http://openresearchsoftware.metajnl.com/articles/10.5334/jors.151/galley/245/download/>. doi:10.5334/jors.151. Exported from <https://app.dimensions.ai> on 2019/05/05.
- [44] Scacchioli A, Bayen AM, Stojadinović B. Assessment of uncertainty propagation in the dynamic response of single-degree-of-freedom structures using reachability analysis. *J Eng Mech* 2014;140:04014038.
- [45] Shinozuka M. Monte carlo solution of structural dynamics. *Comput Struct* 1972;2:855–74.
- [46] Shu-Xiang G, Zhen-zhou L. Interval arithmetic and static interval finite element method. *Appl Mathe Mech* 2001;22:1390–6.
- [47] Tahersima M, Tikalsky P. Finite element modeling of hydration heat in a concrete slab-on-grade floor with limestone blended cement. *Constr Build Mater* 2017;154:44–50. <https://doi.org/10.1016/j.conbuildmat.2017.07.176>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0950061817315180>.
- [48] Wetzlinger M, Kochdumper N, Althoff M. Adaptive parameter tuning for reachability analysis of linear systems. In: 2020 59th IEEE Conference on Decision and Control (CDC). IEEE; 2020. p. 5145–152.
- [49] Wilson E, Bathe K, Peterson F. Finite element analysis of linear and nonlinear heat transfer. *Nucl Eng Des* 1974;29:110–24. [https://doi.org/10.1016/0029-5493\(74\)90101-0](https://doi.org/10.1016/0029-5493(74)90101-0). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0029549374901010>.
- [50] Xue Z, Wang W, Fang L, Zhou J. Numerical simulation on structural dynamics of 5 MW wind turbine. *Renewable Energy* 2020;162:222–33. <https://doi.org/10.1016/j.renene.2020.08.028>. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0960148120312659>.
- [51] Zienkiewicz OC. *Introductory Lectures on the Finite Element Method*. Vienna: Springer Vienna; 1972. URL: <http://link.springer.com/10.1007/978-3-7091-2973-9>. doi:10.1007/978-3-7091-2973-9.