This space is reserved for the EPiC Series header, do not use it

Discrete-Space Analysis of Partial Differential Equations

Hoang-Dung Tran, TianShu Bao, Taylor T. Johnson

There are numerous examples that arise and benefit from the reachability analysis problem. In cyber-physical systems (CPS), most dynamic phenomena are described as systems of ordinary differential equations (ODEs). Previous work has been done by using zonotopes, support functions, and other geometric data structures to represent subsets of the reachable set and have been shown to be efficient. Meanwhile, a wide range of important control problems are more precisely modeled by partial differential equations (PDEs), even though not much attention has been paid to their reachability analyses. This reason motivates us to investigate the properties of these equations, especially from the reachability analysis and verification perspectives. In contrast to ODEs, PDEs have other space variables that also affect their behaviors and are more complex. In this paper, we study the discrete-space analysis of PDEs. Our ultimate goal is to propose a set of PDE reachability analysis benchmarks, and present preliminary analysis of different dimensional heat equations and wave equations. Finite difference methods (FDMs) are utilized to approximate the derivative at each mesh point with explicit order of errors. FDM will convert the PDE to a system of ODEs depending on the type of boundary conditions and discretization scheme chosen. After that, the problem can be treated as a common reachability problem and relevant conceptions and approaches can be applied and evaluated directly. We used SpaceEx to generate the plots and reachable regions for these equations given inputs and series of results are shown and analyzed.

1 Context and Origin

Reachability analysis for Cyber-Physical Systems (CPS) involving ODEs dynamics has been explicitly investigated in the last two decades which leads to the innovation of numerous efficient techniques and tools. CPS with linear ODEs dynamics can be analyzed efficiently using support function-based method implemented in SpaceEx [1] or zonotope-based method utilized in CORA [2]. Notably, the most recent tool called Hylaa leveraging simulation-based method [3,4] can analyze the linear ODEs with up to 10000 dimensions [5]. In addition to linear ODEs, the analysis of CPS with nonlinear ODEs dynamics has been solved for small-scale systems. Notable techniques for this type of dynamics are Taylor model [6] and the $\delta-$ reachability analysis [7].

Although most of CPS applications involves the ODEs dynamics, there are many practical applications relating sensing and control of PDEs such as fluid dynamics control, quantum mechanics, heat flows control, electrostatics and electrodynamics [8]. Such type of applications rises the need of new methodologies and tools to formally analyze the PDEs dynamics that depends only in time but also in space. Analyzing a system with PDEs dynamics for the whole space in which the dynamics is defined may be challenging [9]. However, a cheaper analysis can be done if we only concern about the system property at some discrete points in the space. To do that, an *approximate discrete-space model* of a PDE needs to be obtained. Fortunately, this can be done efficiently using Finite Different Method (FDM) which is basically a fundamental approach for solving PDEs in the field of numerical methods [10].

Using semi-explicit FDM, i.e., discretizing only in space and not in time, an approximate discrete-space model of a PDE can be obtained in the form of ODEs which then can be analyzed efficiently using existing verification tools. Generally, the obtained discrete-space model has finite dimensions, and it approximates the behavior of the original PDE at specific so-called *mesh points*. It is worth noting that increasing the number of mesh points in discretization grows the size of the obtained ODEs.

In this paper, we study the discrete-space analysis of some popular PDEs including parabolic and hyperbolic equations. The derivation of ODEs models from those equations is presented in detail. Beside investigating the discrete-space analysis for PDEs, our ultimate objective is to introduce PDE benchmarks to verification community. Since we can obtain arbitrary large ODEs by increasing the number of mesh points in discretization, e.g, ODEs with billion dimensions, our benchmarks are suitable for evaluating the scalability of existing or on-developing approaches. The SpaceEx and Flow* models of all benchmarks introduced in this paper can be generated automatically using a PDE-to-ODE printer provided in the *pdev* prototype [9] which is written in Python and available at: https://github.com/trhoangdung/pdev.

The paper is organized as follows. Section 2 presents in detail the derivation of discrete-space models (i.e, linear ODEs systems) of all introduced benchmarks. The corresponding algorithms to generate these models are given in Appendix. Section 3 provides analysis results of some small models of the benchmarks using SpaceEx. Section 4 discusses some open questions and concludes the paper.

2 Discrete-space Model Derivation

In this section, we present in detail how to derive the discrete-space models of PDEs using semi-explicit FDM. All benchmarks considered in our paper are linear PDEs. We particularly focus on parabolic and hyperbolic PDEs in which heat and wave equations are two well-known representatives. Several types of boundary conditions for PDEs are addressed in this paper.

2.1 Parabolic Equation

Heat-flow and diffusion-type problems are fundamental physical processes which can be modeled using parabolic PDEs. One simple example is that the heat flows on a 2-dimensional metal plate, the boundaries of the plate are *insulated* or connected to a *heat source* or free to exchange the heat with the environment. To analyze such physical process, the conditions for the boundaries need to be considered in detail. The heat-flow and diffusion-type problems can happen in every 1-dimensional, 2-dimensional or 3-dimensional physical objects.

2.1.1 One-dimensional Heat Equation

This benchmark is from [11], and is depicted in Figure 1. As shown in the figure, we have a copper rod with the length of $L = 200cm$. The top and two sides of the rod are insulted while the bottom is immersed in moving water having constant temperature of 20^oC. The rod initially has temperature of 0^oC. The mathematical model of this heat-flow problem are as follows:

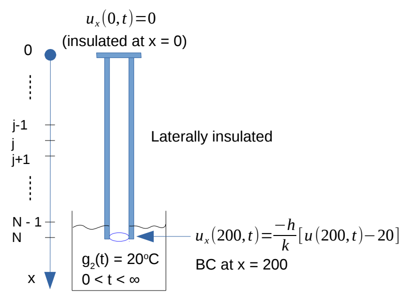

Figure 1: One-dimensional heat equation

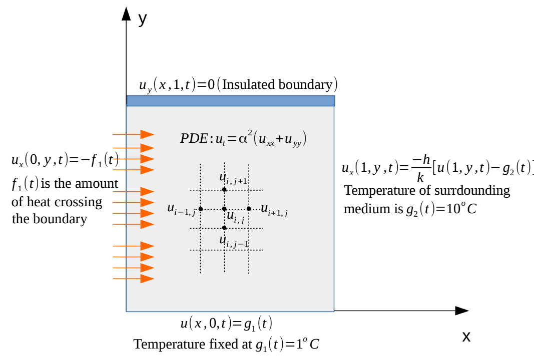

Figure 2: Two-dimensional heat equation

$$
\begin{aligned}
&PDE: u_t = \alpha^2 u_{xx}, \quad 0 < x < 200, \quad 0 < t < \infty \\
&BCs: \begin{cases} u_x(0,t) = 0 \\ u_x(200,t) = -\frac{h}{k}[u(200,t) - g_2(t)] \end{cases}, \quad 0 < t < \infty \\
&IC: u(x,0) = 0^oC, \quad 0 \leq x \leq 200
\end{aligned} \tag{1}
$$

where:
$\alpha^2 = 1.16cm^2/sec$ is the diffusivity constant for copper
$k = 0.93cal/cm \cdot sec \cdot^o C$ is the thermal conductivity of copper
$h = 1$ is the heat exchange coefficient.

By discretizing the length of the rod into N segments and then using semi-explicit finite difference method, we can obtain a discrete-space model in the form of linear ODEs with $N-1$ state variables representing the temperatures of discretized points of the rod. Let u_j be the temperature at the point j on the x-axis as shown in Figure 1. From the first boundary condition, we have $u_0 = 0$ Approximating the right hand side of the PDE using *central-difference approximation* leads to:

$$\frac{du_j}{dt} = u_{xx}^j = \frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2}, \tag{2}$$

where $\Delta x = L/N$ is the discretization step.

To obtain the discrete-space model, one more step need to be done is to approximate the boundary conditions at $x = 200$ and $x = 0$. Using *backward-difference approximation* for the boundary condition at $x = 200$, we have:

$$u_x^N = \frac{u_N - u_{N-1}}{\Delta x} = -\frac{h}{k}[u_N - g_2(t)]. \tag{3}$$

Consequently, the formula for the point N's temperature is:

$$u_N = \frac{k}{k + \Delta x \times h} u_{N-1} + \frac{\Delta x \times h}{k + \Delta x \times h} g_2(t). \tag{4}$$

Similarly, using *forward-difference approximation* for the boundary condition at $x = 0$, we have:

$$u_x^0 = \frac{u_1 - u_0}{\Delta x} = 0 \Rightarrow u_1 = u_0. \tag{5}$$

Using Equations (2), (4) and (5), the discrete-space model of the form $\dot{x} = Ax + Bv$ of the heat-flow problem is obtained with following characteristics:

$$x = [u_1, u_2 ... u_{N-1}]^T, v = g_2(t) = 20,$$
$$A = \frac{\alpha^2}{\Delta^2}\begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 1 & -2 + \frac{k}{k+\Delta x \times h} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \frac{\alpha^2 h}{\Delta x(k+\Delta x \times h)} \end{bmatrix}.$$

2.1.2 Two-dimensional Heat Equation

This benchmark is from [11] and is depicted in Figure 2. The related constants (α^2, k, h) are the same as in the one-dimensional benchmark. The square copper plate has the width and height of $W = H = 1m = 100cm$. The mathematical model of this problem are as follows.

$$\begin{aligned} &PDE: u_t = \alpha^2(u_{xx} + u_{yy}) \\ &BCs: \begin{cases} u_x(0, y, t) = -f_1(t) = -1 \\ u_x(100, y, t) = -\frac{h}{k}[u(100, y, t) - g_2(t)] \\ u(x, 0, t) = g_1(t) = 1^oC \\ u_y(x, 100, t) = 0 \end{cases}, \quad 0 < t < \infty \\ &IC: u(x, y, 0) = sin(\pi x/100), \quad 0 \leq x \leq 100 \end{aligned} \tag{6}$$

Similar to the one-dimensional heat-flow benchmark, we can obtain the discrete-space model of this two-dimensional heat-flow benchmark by discretizing the square copper plate by a number of mesh points and then using semi-explicit finite difference method. Assume that we discretize the width and the height of the plate by N and M segments along x-axis and y-axis respectively. Totally, we have $N \times M$ mesh points (MPs) and then the corresponding linear model of this benchmark has $(N-1) \times (M-1)$ state variables. The discretization steps along x-axis and y-axis are $\Delta x = W/N$ and $\Delta y = H/M$. A simple case of discretization is letting $N = M$ and consequently we have $\Delta x = \Delta y = 100/N$.

Let $u_{i,j}$ be the temperature at the mesh point (i, j) as depicted in Figure 2. Using central-difference approximation for the right hand side of the PDE leads to:

$$\begin{aligned} u_{xx}^{i,j} &= \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2}, \\ u_{yy}^{i,j} &= \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2}, \\ \frac{du_{i,j}}{dt} &= \alpha^2[\frac{u_{i-1,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} + u_{i,j+1}}{\Delta y^2} - (\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2})u_{i,j}] \end{aligned} \tag{7}$$

Using *forward-approximation* for the first boundary condition, we have:

$$u_x(0, j) = \frac{u_{1,j} - u_{0,j}}{\Delta x} = -f_1(t) \Rightarrow u_{0,j} = u_{1,j} + \Delta x f_1(t). \tag{8}$$

Using *backward-approximation* for the second boundary condition yields:

$$u_x(N,j) = \frac{u_{N,j} - u_{N-1,j}}{\Delta x} = -\frac{h}{k}[u(N,j) - g_2(t)].$$

Consequently, we have:

$$u_{N,j} = \frac{k}{k + \Delta x \times h} u_{N-1,j} + \frac{\Delta x \times h}{k + \Delta x \times h} g_2(t). \tag{9}$$

The third boundary condition for discretized mesh points is:

$$u_{i,0} = g_1(t). \tag{10}$$

Using *back-ward approximation* for the last boundary condition, we get:

$$u_y(i,M) = \frac{u_{i,M} - u_{i,M-1}}{\Delta y} = 0 \Rightarrow u_{i,M} = u_{i,M-1}. \tag{11}$$

Combining Equations (7)-(11), a linear model with $(N-1)\times(M-1)$-dimensions of the form $\dot{x} = Ax + Bv$ of two-dimensional heat-flow problem can be obtained, where:
$u = [\underbrace{u_{1,1}, ..., u_{N-1,1}}_{1^{st} \text{ column of MPs}}, \underbrace{u_{1,2}, ..., u_{N-1,2}}_{2^{nd} \text{ column of MPs}}, ..., \underbrace{u_{1,M-1}, ..., u_{N-1,M-1}}_{(M-1)^{th} \text{ column of MPs}}]^T$,
The input vector v is defined by $v = [f_1(t) \quad g_1(t) \quad g_2(t)]$.

Let $n = (N-1)\times(M-1)$, then the matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times 3}$ can be obtained by Algorithm 1.1.

2.2 Hyperbolic Equation

Wave equation is used to model a large collection of phenomena such as flows of fluids through a porous media and atmospheric flows. In this paper, we consider 1- and 2-dimensional equations with Dirichlet boundary condition and periodic boundary condition [12].

2.2.1 One-dimensional Wave Equation

$$\begin{aligned} &PDE: u_t + \alpha u_x = 0, 0 < x < 30, 0 < t < \infty \\ &BCs: u(0,t) = u(30,t), \quad \text{(periodic BCs)} \\ &ICs: \begin{cases} u(x,0) = 1, & 14 \le x \le 18 \\ u(x,0) = 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{12}$$

where α is the wave propagation speed.

By discretizing the space of interest with N uniform steps $\Delta x = 30/N$ and using forward scheme, the spacial derivative u_x at the i^{th} mesh point is approximated as:

$$u_x^i = \frac{u_{i+1} - u_i}{\Delta x},\ u_i \equiv u(i \times \Delta x, t).$$

This leads to:

$$\frac{du_i}{dt} = \frac{\alpha}{\Delta x}(u_{i+1} - u_i) \tag{13}$$

At the boundary, we have the following constrain:

$$\frac{du_{N-1}}{dt} = \frac{\alpha}{\Delta x}(u_N - u_{N-1}),\ u_N = u_0 \tag{14}$$

From (13) and (14), the discrete-space model of this wave equation is obtained in the form of $\dot{x} = Ax$, where:

$$x = [u_0, \dots, u_{N-1}]^T, A = \frac{\alpha}{\Delta x}\begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & -1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -1 & 1 \\ 1 & 0 & 0 & \dots & 0 & -1 \end{bmatrix}$$

2.2.2 Two-dimensional Wave Equation

$$\begin{aligned} &PDE : u_t + \alpha u_x + \beta u_y = 0, 0 < x < 6, 0 < y < 6, 0 < t < \infty \\ &BCs : \begin{cases} u(0,y,t) = 0, \ \text{(Dirichlet BC)} \\ u(x,0,t) = u(x,6,t), \text{(periodic BC)} \end{cases} \\ &ICs : \begin{cases} u(x,y,0) = 1, & 0 \leq x \leq 1, 0 \leq y \leq 1 \\ u(x,y,0) = 0, & \text{otherwise} \end{cases}, \end{aligned} \tag{15}$$

where $\alpha = 1cm/sec$ and $\beta = 1cm/sec$ are velocities of the wave in x and y directions respectively.

Discretizing the space of interest by $N \times M$ mesh points as in the case of two-dimensional heat equation and using backward-difference approximation, we have:

$$\begin{aligned} &u_x^{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \\ &u_y^{i,j} = \frac{u_{i,j} - u_{i,j-1}}{\Delta y}, \\ &u_{i,j} \equiv u(i \times \Delta x, j \times \Delta y), \ \Delta x = \frac{6}{N}, \ \Delta y = \frac{6}{M}. \end{aligned} \tag{16}$$

At the boundary, we have following constrain:

$$\begin{aligned} u_x^{1,j} &= \frac{u_{1,j} - u_{0,j}}{\Delta x} = \frac{u_{1,j}}{\Delta x}, \\ u_y^{i,1} &= \frac{u_{i,1} - u_{i,0}}{\Delta x} = \frac{u_{i,1} - u_{i,M}}{\Delta x}. \end{aligned} \tag{17}$$

Using (16) and (17), the discrete-space model of the two dimensional wave equation can be obtained in the form of $\dot{x} = Ax$, where $x = [\underbrace{u_{1,1}, ..., u_{N,1}}, \underbrace{u_{1,2}, ..., u_{N,2}}, ..., \underbrace{u_{1,M}, ..., u_{N,M}}]^T$. The matrix $A \in \mathbb{R}^{NM \times NM}$ can be generated from Algorithm 1.2.

3 Reachability analysis

All discrete-space models of the PDE benchmarks studied in this paper can be analyzed by existing verification tools such as SpaceEx, Flow*, CORA and Hylaa which support ODEs dynamics. It is worth noting that the traditional over-approximation tools such as SpaceEx, Flow* and CORA can analyze those models with small and medium sizes, i.e., from several to hundred state variables while the simulation-based tool Hylaa can analyze large models with thousands of state variables.

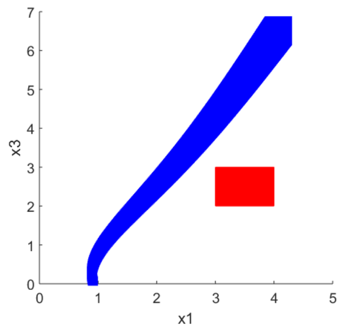

Figure 3: A reachable set of (x_1, x_3) of the discrete-model of the two dimensional heat equation using 3×3 grid of mesh points.

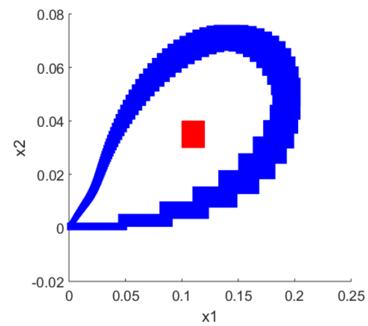

Figure 4: A reachable set of (x_1, x_2) of the discrete-model of the two dimensional wave equation using 3×3 grid of mesh points.

Figure 3 describes the reachable set of the pair (x_1, x_3) state variables of the discrete-space model of the two dimensional heat equation using SpaceEx. To obtain the discrete-space model, the space of interest is discretized by 3×3 grid of mesh points. The reachability analysis is done with the assumption that the input functions are constant in some bounded ranges. We refer the reader to the generated SpaceEx/Flow* model for the detail information. The unsafe region is the red square region defined by $3 \leq x_1 \leq 4$ and $2 \leq x_3 \leq 3$. The analysis shows that the discrete-space model satisfies its safety requirement.

Similarly, Figure 4 shows the reachable set of the pair (x_1, x_2) state variables of the discrete-space model of the two dimensional wave equation using SpaceEx. The discrete-space model is obtained by discretizing the space of interest using 3×3 grid of mesh points. The reachable set is computed with the assumption that the initial conditions are constant in some bounded ranges. The unsafe region (red square) is defined by $0.1 \leq x_1 \leq 0.12$ and $0.03 \leq x_2 \leq 0.04$. From the figure, one can conclude that the discrete-space model of the wave equation is safe.

4 Outlook

We have studied the discrete-space analysis for parabolic and hyperbolic PDEs by leveraging sem-explicit FDM and the existing verification tools that supports ODEs dynamics. By doing that, we can guarantee *to some extends* the safety of the PDEs at specific mesh points on the space. However, achieving a formal guarantee for CPS with PDEs dynamics is still a big challenge because of two reasons. Firstly, the FDM used in this paper is only an approximation method that does not give a *formal measure* about the distance between the approximate behavior of the discrete-space model at a specific mesh point and the actual behavior of the original PDEs at that point. Secondly, suppose we have a completely formal discrete-space analysis approach, we still can answer nothing about the safety of the system between two mesh points. From this observation and the fact that there are many CPS involving PDEs dynamics, there is an urgent need of novel formal methodologies to analyze the safety of PDEs not only at specific mesh points but also for a whole continuous region in the space.

The SpaceEx/Flow* models generated from the considered PDEs benchmarks are useful for testing the scalability of verification techniques and tools. A billion dimensional discrete-space model of the two-dimensional heat equation has been used to evaluate the recent simulation-based reachability analysis approach leveraging Krylov subspace method [13]. In the future, we are going to explore new types of PDEs such as elliptic and more importantly, the real safety-critical applications related to PDEs dynamics.

References

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

[2] M. Althoff, "An introduction to cora 2015." in *ARCH@ CPSWeek*, 2015, pp. 120–151.

[3] S. Bak and P. S. Duggirala, "Simulation-equivalent reachability of large linear systems with inputs."

[4] ——, "Hylaa: A tool for computing simulation-equivalent reachability for linear systems," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 2017, pp. 173–178.

[5] ——, "Direct verification of linear systems with over 10000 dimensions," in *4th International Workshop on Applied Verification of Continuous and Hybrid Systems*, ser. EPiC Series in Computing. EasyChair, 2017.

[6] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *International Conference on Computer Aided Verification*. Springer, 2013, pp. 258–263.

[7] S. Kong, S. Gao, W. Chen, and E. Clarke, "dreach: δ-reachability analysis for hybrid systems," pp. 200–205, 2015.

[8] W. A. Strauss, *Partial differential equations*. John Wiley & Sons New York, NY, USA, 1992.

[9] H.-D. Tran, W. Xiang, S. Bak, and T. T. Johnson, "Reachability analysis for one dimensional linear parabolic equation," in *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 2018)*. IFAC, 2018.

[10] G. D. Smith, *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.

[11] S. J. Farlow, *Partial differential equations for scientists and engineers*. Courier Corporation, 1993.

[12] J.W.Thomas, *Numerical partial differential equations: finite difference methods*. Springer, 1998.

[13] S. Bak, H.-D. Tran, and T. T. Johnson, "Numerical verification of affine systems with up to a billion dimensions," in *https://scirate.com/arxiv/1804.01583*, 2018.

Appendix A Algorithms for Two-dimensional Heat and Wave Equations

Algorithm 1.1 Discrete-space model of 2-dimensional heat-flow equation

Input: (N, M) % number of discretization steps along x and y-axis
Output: (A, B) % matrix of the linear ODE: $\dot{x} = Ax + Bv$

procedure INITIALIZATION
  n = $(N-1) \times (M-1)$ % number of state variables
  A $\leftarrow (n \times n)$ sparse matrix
  B $\leftarrow (n \times 3)$ sparse matrix
  $\alpha^2 \leftarrow$ diffusity constant
  $q \leftarrow \frac{h}{k}$ % heat loss constant
  $\Delta x \leftarrow \frac{W}{N} = \frac{100}{N}$ % discretization step along x-axis
  $\Delta y \leftarrow \frac{H}{M} = \frac{100}{M}$ % discretization step along y-axis
procedure FILLING A AND B
loop *for i = 0 to n:*
  $A[i,i] \leftarrow -2(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2})$
  $pos_x \leftarrow i\%(N-1)$ % x-position of i-th state variable
  $pos_y \leftarrow \textbf{int}(\frac{i - pos_x}{N-1})$ % y-position of i-th state variable
  if $pos_x - 1 \geq 0$:
    $A[i,i-1] \leftarrow \frac{1}{\Delta x^2}$
  else:
    $A[i,i] \leftarrow A[i,i] + \frac{1}{\Delta x^2}$
    $B[i,0] \leftarrow \frac{1}{\Delta x}$
  if $pos_x + 1 \leq N-2$:
    $A[i,i+1] \leftarrow \frac{1}{\Delta x^2}$
  else:
    $A[i,i] \leftarrow A[i,i] + \frac{1}{\Delta x^2(1+q\Delta x)}$
    $B[i,2] \leftarrow \frac{q}{\Delta x(1+q\Delta x)}$
  if $pos_y - 1 \geq 0$:
    $A[i,(pos_y-1)(N-1)+pos_x] \leftarrow \frac{1}{\Delta y^2}$
  else:
    $B[i,1] \leftarrow \frac{1}{\Delta y^2}$
  if $pos_y + 1 \leq M-2$:
    $A[i,(pos_y+1)(N-1)+pos_x] \leftarrow \frac{1}{\Delta y^2}$
  else:
    $A[i,i] \leftarrow A[i,i] + \frac{1}{\Delta y^2}$
end loop: $A \leftarrow \alpha^2 A$, $B \leftarrow \alpha^2 B$
return (A,B)

Algorithm 1.2 Discrete-space model of 2-dimensional wave equation

Input: (N, M) % number of discretization steps along x and y-axis
Output: A % matrix of the linear ODE: $\dot{x} = Ax$

procedure INITIALIZATION
    $n = N \times M$
    $A \leftarrow (n \times n)$ sparse matrix
    $\alpha \leftarrow$ wave propagation speed in x direction
    $\beta \leftarrow$ wave propagation speed in y direction
    $\Delta x \leftarrow$ discretization step along x-axis
    $\Delta y \leftarrow$ discretization step along y-axis
    $a \leftarrow \alpha / \Delta x$
    $b \leftarrow \beta / \Delta y$
procedure FILLING A
loop for i = 0 to n:
    $A[i, i] \leftarrow$ a + b
    $pos_x = i \bmod N$ % x-position corresponding to i-th state variable
    $pos_y = \mathrm{int}((i - pos_x)/N)$ % y-position corresponding to i-th state variable
    if $pos_y == 0$ **then**:
        $A[i, i-1] \leftarrow$ -a
        $A[i, i] \leftarrow$ b + a
    else:
        $A[i, i-1] \leftarrow$ -a
        $A[i, i] \leftarrow$ b + a
        $A[i, i-N] \leftarrow$ -b
    if pos_y == M - 1 **then**:
        $A[pos_x, i] \leftarrow$ -b
end loop
Return(A)