

DISCRETE TRANSPARENT BOUNDARY CONDITIONS FOR THE TWO-DIMENSIONAL LEAP-FROG SCHEME: APPROXIMATION AND FAST IMPLEMENTATION

CHRISTOPHE BESSE^{1,*}, JEAN-FRANÇOIS COULOMBEL¹ AND PASCAL NOBLE²

Abstract. We develop a general strategy in order to implement approximate discrete transparent boundary conditions for finite difference approximations of the two-dimensional transport equation. The computational domain is a rectangle equipped with a Cartesian grid. For the two-dimensional leap-frog scheme, we explain why our strategy provides with explicit numerical boundary conditions on the four sides of the rectangle and why it does not require prescribing any condition at the four corners of the computational domain. The stability of the numerical boundary condition on each side of the rectangle is analyzed by means of the so-called normal mode analysis. Numerical investigations for the full problem on the rectangle show that strong instabilities may occur when coupling stable strategies on each side of the rectangle. Other coupling strategies yield promising results.

Mathematics Subject Classification. 65M06, 65M12.

Received September 6, 2019. Accepted July 25, 2020.

1. INTRODUCTION

In this article, we are concerned with the construction and numerical implementation of discrete transparent boundary conditions for linear transport equations. Our goal is to derive numerical boundary conditions that minimize the parasitic wave reflections at the boundary of the computational domain. This research area has been increasingly active since the pioneering work by Engquist and Majda [19] and our goal here is to explain why following the same “small frequency approximation” strategy as in [19] can be successful at the fully discrete level when one deals with a rectangular computational domain. Namely, we shall construct fully discrete, approximate transparent boundary conditions for the two-dimensional leap-frog scheme on a rectangle. Our numerical boundary conditions will not be exactly transparent since the derivation of such conditions on a rectangle seems to be out of reach. Our strategy is to start from the *exact* transparent boundary conditions for a *half-space* and, as in [19], to *localize* them with respect to the tangential variable by means of a suitable “small frequency approximation”. This general strategy may be applied to any finite difference scheme. In terms of wave reflection magnitude, our numerical simulations show that our approximate transparent boundary conditions

Keywords and phrases. Transport equation, leap-frog schemes, transparent boundary conditions, stability.

¹ Institut de Mathématiques de Toulouse, UMR5219, Université de Toulouse, CNRS, Université Paul Sabatier, F-31062 Toulouse Cedex 9, France.

² Institut de Mathématiques de Toulouse, UMR5219, Université de Toulouse, CNRS, INSA, F-31077 Toulouse, France.

*Corresponding author: jean-francois.coulombel@math.univ-toulouse.fr

are more efficient than more standard, purely local extrapolation strategies, hence, in our opinion, the relevance of implementing discrete time convolution operators as described below.

We focus here on the discretization of the two-dimensional transport equation by means of the leap-frog scheme. The nice feature of this approximation is that it does not dissipate high frequency signals – and therefore allows for a precise analysis of wave reflections on each side of the rectangle – and, moreover, its stencil exhibits some “dimensional splitting”. The latter feature will be helpful in our construction since we shall not have to develop a specific numerical treatment for the four corners of the computational domain. Hence we shall focus on the numerical boundary conditions on each of the four sides of the rectangle and on their coupling through the eight mesh points that are closest to the four corners. More simple, purely local, outflow strategies [1, 21, 24, 35] may be implemented for dissipative schemes (*e.g.*, the Lax–Wendroff scheme). Even though these extrapolation boundary conditions are much cheaper from a computational point of view and are quite efficient in terms of absence of wave reflection for *dissipative* schemes, they only give poor intuition for more complex wave propagation phenomena modeled for instance by the Schrödinger or Airy equations. Moreover, our simulations clearly show that local extrapolation boundary conditions for the leap-frog scheme give rise to larger reflected waves than those generated by the boundary conditions we shall construct below. For inflow boundaries, high order local numerical treatments based on the inverse Lax–Wendroff method [16, 20, 31, 33, 36] have also been constructed and analyzed in the past. As for extrapolation strategies at outflow boundaries, these numerical treatments may yield good convergence properties. However, they generate wave reflections whose magnitude is larger than what we observe with our nonlocal (in time) procedure. Despite a more involved implementation and a higher computational cost, we thus believe that nonlocal strategies as those examined below may be relevant.

An alternative strategy to the one developed below would consist in starting from the exact transparent boundary condition for the underlying partial differential equation (the continuous problem) and then trying to discretize it. This strategy has been examined, at least, in [2, 8, 23], and the overall conclusion is that there is no general answer for the construction of such absorbing boundary conditions if one wishes to maintain stability. We therefore rather follow the approach developed in [3, 18] and later used in a wide variety of contexts, and thus construct (approximate) transparent boundary conditions that are adapted to the considered finite difference approximation.

Exact Discrete Transparent Boundary Conditions (DTBC in what follows) are analytically available in two space dimensions only when the computational domain is a half-space ($\mathbb{Z}^{d-1} \times \mathbb{N}$) or the product of a half-line by a torus ($(\mathbb{Z}/J\mathbb{Z})^{d-1} \times \mathbb{N}$). By “analytically available”, we mean that DTBC can be expressed by a well-defined operator, even though its practical implementation may be difficult. We refer for instance to [2–4, 15] and references therein for various aspects of the theory, the main features of which are recalled below. In either case, the discrete spatial domain is a cylinder of the form $\mathcal{D} \times \mathbb{N}$ with no tangential boundary (the variables in \mathcal{D}) so DTBC can be derived by performing either a Fourier transform or a Fourier series decomposition with respect to the spatial tangential variables. In the end, this approach yields a one-dimensional problem that is parametrized by the tangential frequency¹. In that framework, the two-dimensional DTBC thus take the form of a *tangential pseudo-differential operator* that is intrinsically *nonlocal*. In order to be implemented on a rectangular computational domain, one therefore has to make the DTBC *local*, which we achieve here as in [19] by using a suitable “small frequency approximation”, which is meaningful (at least formally) for smooth solutions. The frequency cut-off amounts to retaining only finitely many terms in the Taylor expansion of the symbol of the DTBC operator and rewriting the Taylor expansion as a trigonometric polynomial (up to higher order terms). This cut-off strategy is implemented here in such a way that the stencil used for the numerical boundary conditions is not wider than that of the interior numerical scheme. Higher order strategies could be relevant but would require a specific treatment when getting closer to the corner. We do not investigate this issue here.

¹The same strategy applies when the computational domain is a disk by using polar coordinates.

As in all numerical approximation problems, a crucial feature for the efficiency of the numerical scheme is stability. Following the general result of [15], the DTBC for the leap-frog scheme on a half-space do not meet the strongest possible stability properties because the leap-frog scheme exhibits *glancing* wave packets (even in one space dimension). On a half-space, the exact DTBC meet some kind of *neutral* stability which calls for special care since a slight modification in the numerical boundary conditions may yield *violent instabilities*. In what follows, we show that the zero, first and second order tangential frequency cut-off – as described above – maintain neutral stability when implemented on a half-space (the so-called Godunov–Ryabenkii condition is satisfied [22]). In other words, our approximate transparent boundary conditions are as spectrally stable as the exact ones on a half-space and we therefore feel rather safe to implement them on a rectangular geometry. If instabilities occur in the rectangular geometry, they will necessarily stem from the coupling between the various numerical boundary conditions on each side of the rectangle. A thorough stability analysis on the rectangle seems to be out of reach at the present time. It is not even fully understood at the continuous level! The numerical investigations that we report below show that stability in the rectangle should not be taken for granted. Some coupling strategies exhibit good stability – and convergence – properties while at least one displays violent instabilities. In a future work, we shall extend our strategy to fully discretized hyperbolic systems and to implicit difference schemes for dispersive equations.

Let us now give the plan of our work. In Section 2, we introduce the two-dimensional transport equation and the leap-frog scheme on a rectangular computational domain. The goal is then to construct and analyze approximate transparent boundary conditions. For later use and also for the ease of reading, we first go back to the corresponding one-dimensional problem in Section 3. We adapt the general strategy of [3, 18], which was recently generalized in [15], to the particular case of the leap-frog scheme. Some of the tools used in [15] are complemented here by some explicit expansions, based on Legendre polynomials, which were already used in [3, 12, 18]. Such explicit representation formulae lower the implementation cost of our method and give a sharp asymptotic behavior for the convolution coefficients, which would not be possible by only using the tools in [15]. We then analyze the stability of the one-dimensional DTBC by means of the normal mode decomposition and show that they satisfy the so-called Godunov–Ryabenkii condition [22]. Since the numerical implementation and computational cost of DTBC is sometimes considered as heavy, we then discuss the fast implementation of approximate DTBC by the so-called sum of exponential approach. We relax some of the constraints used in [3, 18] for the free parameters in this approximation, and show that the resulting approximation is efficient from a numerical cost point of view and also from the wave reflection magnitude point of view. Section 4 extends this approach to the two-dimensional problem by first considering the exact DTBC on a half-space and then using low frequency truncations in order to derive local (in the tangential space variable) approximate DTBC. We analyze, for the half-space problem, the stability of our approximate DTBC and then report on some numerical simulations on the rectangle.

2. THE TRANSPORT EQUATION AND THE LEAP-FROG APPROXIMATION

In this article, we consider the linear advection equation in two space dimensions:

$$\begin{cases} \partial_t u + \mathbf{c} \cdot \nabla u = 0, & t \geq 0, \quad (x, y) \in \mathbb{R}^2, \\ u|_{t=0} = u_0, \end{cases} \quad (2.1)$$

where the space coordinates are denoted (x, y) , the velocity reads $\mathbf{c} = (c_x, c_y)^T$ and $\nabla = (\partial_x, \partial_y)^T$. It is always assumed from now on that the velocity \mathbf{c} is nonzero. To fix ideas, we shall always take c_x and c_y to be nonnegative, but this has no impact on the analysis below. The solution to (2.1) reads:

$$u(t, x, y) = u_0(x - c_x t, y - c_y t),$$

so if the initial condition u_0 is supported, say in the square $[-L, L] \times [-L, L]$, the solution vanishes in the larger square $[-2L, 2L] \times [-2L, 2L]$ after some time $T \sim L/|\mathbf{c}|$. Given a finite difference approximation to (2.1)

associated with a Cartesian grid, our goal in this article is to propose a systematic construction of *transparent*, or *approximate transparent* numerical boundary conditions² in order to simulate the advection of the initial condition through the computational domain and eventually its exit through the numerical boundary. Our goal is to minimize as much as possible parasitic wave reflections on the numerical boundary.

In all what follows, the computational domain is a fixed rectangle $[x_\ell, x_r] \times [y_b, y_t]$ (ℓ, r, b, t stand for left, right, bottom and top). We consider some space steps $\delta x, \delta y > 0$ such that the ratios:

$$\frac{x_r - x_\ell}{\delta x} =: J + 1, \quad \frac{y_t - y_b}{\delta y} =: K + 1,$$

define some integers J and K (that are meant to be large). Eventually, the time step $\delta t > 0$ will always be chosen so that the parameters

$$\mu_x := c_x \frac{\delta t}{\delta x}, \quad \mu_y := c_y \frac{\delta t}{\delta y},$$

are fixed (or at worse vary between two fixed positive constants). These parameters should satisfy some stability requirement, the so-called Courant–Friedrichs–Lewy condition, as explained below. The grid points are defined as (x_j, y_k) , with:

$$\forall j = 0, \dots, J + 1, \quad \forall k = 0, \dots, K + 1, \quad x_j := x_\ell + j \delta x, \quad y_k := y_b + k \delta y.$$

The interior points correspond to $1 \leq j \leq J$ and $1 \leq k \leq K$. The four sides of the rectangle (the numerical boundary) correspond to $j \in \{0, J + 1\}$ and $k \in \{0, K + 1\}$.

Letting now $u_{j,k}^n$ denote the approximation of the exact solution to (2.1) at (x_j, y_k) and time $n \delta t$, the two-dimensional leap-frog scheme reads:

$$u_{j,k}^{n+2} - u_{j,k}^n + \mu_x (u_{j+1,k}^{n+1} - u_{j-1,k}^{n+1}) + \mu_y (u_{j,k+1}^{n+1} - u_{j,k-1}^{n+1}) = 0, \quad (2.2)$$

which holds for $1 \leq j \leq J$ and $1 \leq k \leq K$, that is at all interior points. The definition of the numerical scheme requires to prescribe numerical boundary conditions for $u_{0,k}^{n+2}$, $u_{J+1,k}^{n+2}$, $k = 1, \dots, K$, and $u_{j,0}^{n+2}$, $u_{j,K+1}^{n+2}$, $j = 1, \dots, J$. A crucial observation for what follows is that the update of the numerical solution in the interior does not require to prescribe any numerical boundary condition at the corners (this is because $u_{j,k}^{n+2}$ does not depend on $u_{j\pm 1,k\pm 1}^\sigma$ at any earlier time index σ). In all what follows, the numerical solution $(u_{j,k}^n)$ will therefore be defined for $(j, k) \in \{0, \dots, J + 1\} \times \{0, \dots, K + 1\} \setminus \{(0, 0), (0, K + 1), (J + 1, 0), (J + 1, K + 1)\}$. We shall be careful in our definition of the numerical boundary conditions to make sure that the four corner values are not involved either.

Let us note that on the whole space \mathbb{Z}^2 , the ℓ^2 -stability condition for (2.2) reads:

$$\mu_x + \mu_y \leq \text{cfl} < 1, \quad (2.3)$$

where cfl is a free positive parameter to be chosen. We consider the case $c_x, c_y \geq 0$, hence $\mu_x, \mu_y \geq 0$, so we do not write absolute values in (2.3). The inequality (2.3) gives an upper bound for δt in terms of $\delta x, \delta y$. The bound (2.3), or its one-dimensional analogue that we shall discuss below, is always assumed to hold from now on.

We wish to construct approximate transparent boundary conditions for (2.2) on each side of the rectangle. For ease of reading and also for future use on the two-dimensional problem (2.2), we first go back to the one-dimensional problem and briefly recall the construction of DTBC for the one-dimensional leap-frog scheme. We discuss the various possible implementations of the DTBC and approximations by means of sums of exponentials. This technique was used for instance in [3] in the context of the Schrödinger equation and later extended to other contexts. Once the tools have been set in one space dimension, we shall feel free to use them in two space dimensions whenever it makes sense.

²Sometimes these are also called non-reflecting or absorbing numerical boundary conditions.

3. DISCRETE TRANSPARENT BOUNDARY CONDITIONS FOR THE ONE-DIMENSIONAL LEAP-FROG SCHEME

3.1. The numerical scheme

We now consider the one-dimensional transport equation:

$$\begin{cases} \partial_t u + c_x \partial_x u = 0, & t \geq 0, \quad x \in \mathbb{R}, \\ u|_{t=0} = u_0, \end{cases} \quad (3.1)$$

with an initial condition u_0 that is supported in an interval $[x_\ell; x_r]$. For concreteness, we shall assume from now on $c_x > 0$ so that the initial condition is transported towards the right and eventually exits the initial support $[x_\ell; x_r]$. We are interested here in simulating the solution to (3.1) on the fixed interval $[x_\ell; x_r]$ by imposing suitable non-reflecting boundary conditions at the boundary points x_ℓ, x_r .

As in the previous section, we choose a mesh size $\delta x > 0$ such that

$$\frac{x_r - x_\ell}{\delta x} =: J + 1,$$

is an integer. The time step $\delta t > 0$ is chosen in such a way that the parameter μ_x defined by:

$$\mu_x := c_x \frac{\delta t}{\delta x},$$

is a fixed positive constant. Keeping the notation $x_j := x_\ell + j \delta x$, $j = 0, \dots, J + 1$, and letting now u_j^n denote the approximation of the solution to (3.1) at x_j and time $n \delta t$, the one-dimensional leap-frog scheme reads

$$u_j^{n+2} - u_j^n + \mu_x (u_{j+1}^{n+1} - u_{j-1}^{n+1}) = 0, \quad j = 1, \dots, J. \quad (3.2)$$

The parameter μ_x is fixed by imposing:

$$\mu_x \leq \text{cfl} < 1, \quad (3.3)$$

which is a necessary and sufficient ℓ^2 stability requirement for (3.2) on the whole real line $j \in \mathbb{Z}$, see [22, 29]. Since μ_x is positive in our framework, we thus fix $\mu_x \in (0, 1)$ and let $\delta x, \delta t$ vary accordingly.

The numerical scheme (3.2) on the interval $1 \leq j \leq J$ requires a definition for the boundary values u_0^{n+1} , u_{J+1}^{n+1} , and a definition for the initial data (u_j^0) , (u_j^1) . The initial condition (u_j^0) is defined by setting

$$\forall j = 0, \dots, J + 1, \quad u_j^0 := u_0(x_j),$$

where u_0 is the initial condition for (3.1). In the numerical simulations reported below, we have chosen $[x_\ell, x_r] = [-3, 3]$ and $u_0(x) = \exp(-10x^2)$. The first time step value (u_j^1) is defined by imposing the second order Lax-Wendroff scheme in the interior domain, namely:

$$\forall j = 1, \dots, J, \quad u_j^1 := u_j^0 - \frac{\mu_x}{2} (u_{j+1}^0 - u_{j-1}^0) + \frac{\mu_x^2}{2} (u_{j+1}^0 - 2u_j^0 + u_{j-1}^0).$$

The boundary values u_0^1 and u_{J+1}^1 are set equal to zero for simplicity (which is almost exact for the Gaussian initial condition that we consider). We now recall how to derive the DTBC for the scheme (3.2).

3.2. Derivation of DTBC

The goal of this paragraph is to derive the DTBC for the leap-frog scheme (3.2). We follow the methodology of [18], which was originally designed for the Schrödinger equations, and therefore consider the numerical scheme:

$$u_j^{n+2} - u_j^n + \mu_x (u_{j+1}^{n+1} - u_{j-1}^{n+1}) = 0, \quad j \in \mathbb{Z}, \quad (3.4)$$

on the whole real line \mathbb{Z} , assuming that the initial conditions $(u_j^0)_{j \in \mathbb{Z}}, (u_j^1)_{j \in \mathbb{Z}}$ belong to ℓ^2 and vanish outside of the interval $\{1, \dots, J\}$. The stability condition (3.3) is assumed to hold. The DTBC are first computed in terms of the so-called \mathcal{Z} -transform of the sequences $(u_j^n)_{n \in \mathbb{N}}$. Let us briefly recall the definition of the \mathcal{Z} -transform (we do not pay too much attention here to the convergence of the Laurent series involved in the computations below and rather remain at the level of formal series).

Definition 3.1. Let $(u_n)_{n \in \mathbb{N}} \in \mathbb{C}^{\mathbb{N}}$ be a sequence. The \mathcal{Z} -transform of $(u_n)_{n \in \mathbb{N}}$, which is denoted \hat{u} , is defined by:

$$\hat{u}(z) := \sum_{n=0}^{\infty} \frac{u_n}{z^n},$$

the series being well-defined and holomorphic in $\{z \in \mathbb{C}, |z| > R\}$ where $1/R$ is the convergence radius of $\sum_{n=0}^{\infty} u_n w^n$ (here w is a placeholder for $1/z$ and the radius R may be zero or infinite).

We now let $\hat{u}_j(z)$, $j \in \mathbb{Z}$, denote the \mathcal{Z} -transform of the sequence $(u_j^n)_{n \in \mathbb{N}}$:

$$\hat{u}_j(z) := \sum_{n=0}^{\infty} u_j^n z^{-n}.$$

It is shown in [15] that the above series converges for $|z| > 1$ thanks to the ℓ^2 -stability of the leap-frog scheme³. Moreover, the sequence $(\hat{u}_j(z))_{j \in \mathbb{Z}}$ is square integrable. We apply the \mathcal{Z} -transform to the numerical scheme (3.4) and obtain:

$$(z - z^{-1}) \hat{u}_j + \mu_x (\hat{u}_{j+1} - \hat{u}_{j-1}) = 0, \quad (3.5)$$

for $j \leq 0$ and $j \geq J+1$. (On the interval $\{1, \dots, J\}$, the initial conditions give rise to a nonzero source term on the right hand side of (3.5).) The equation (3.5) is a recurrence relation and we therefore look for the solutions to the characteristic equation:

$$\kappa^2 + \frac{z - z^{-1}}{\mu_x} \kappa - 1 = 0. \quad (3.6)$$

The two roots to (3.6) are given by

$$\kappa_{\pm} := \frac{z^{-1} - z \pm \sqrt{z^2 + 2(2\mu_x^2 - 1) + z^{-2}}}{2\mu_x}, \quad (3.7)$$

with the standard definition of the square root (the branch cut is \mathbb{R}^-). When $|z| \rightarrow +\infty$, $|\kappa_-| \rightarrow +\infty$ and $|\kappa_+| \rightarrow 0$. Moreover, κ_{\pm} do not belong to \mathbb{S}^1 for $|z| > 1$ so we have $|\kappa_+| < 1$ and $|\kappa_-| > 1$ for $|z| > 1$. We thus relabel these two roots as a *stable* and *unstable* one and write from now on:

$$\kappa_s^0(z) := \frac{z^{-1} - z + \sqrt{z^2 + 2(2\mu_x^2 - 1) + z^{-2}}}{2\mu_x}, \quad \kappa_u^0(z) := \frac{z^{-1} - z - \sqrt{z^2 + 2(2\mu_x^2 - 1) + z^{-2}}}{2\mu_x}. \quad (3.8)$$

The reason for the superscript 0 will be clear in the next section when we deal with the two-dimensional problem. The recurrence relation (3.5) and the decay at infinity of $(\hat{u}_j(z))_{j \in \mathbb{Z}}$ implies:

$$\hat{u}_{J+1}(z) = \kappa_s^0(z) \hat{u}_J(z), \quad \hat{u}_0(z) = \frac{1}{\kappa_u^0(z)} \hat{u}_1(z). \quad (3.9)$$

It now remains to compute the Laurent series expansion of both κ_s^0 and $1/\kappa_u^0$ and to perform the inverse \mathcal{Z} -transform in order to write the numerical boundary conditions (3.9) in the original discrete time variable n . Since we have $\kappa_s^0(z) \kappa_u^0(z) = -1$, see (3.6), we can equivalently rewrite (3.9) as:

$$\hat{u}_{J+1}(z) = \kappa_s^0(z) \hat{u}_J(z), \quad \hat{u}_0(z) = -\kappa_s^0(z) \hat{u}_1(z). \quad (3.10)$$

³This is where the condition $\mu_x < 1$ becomes necessary.

There are at least two ways to compute the Laurent series expansion of κ_s^0 for the leap-frog scheme, which we now detail in order to compare the possible benefits and drawbacks of each method. The first method is rather in the spirit of [18] while the second one follows [11, 15].

An explicit expansion in terms of Legendre polynomials. The first method to obtain this expansion relies on the explicit formula (3.8) and on the expansion

$$\frac{1}{\sqrt{1-2xt+t^2}} = \sum_{n=0}^{\infty} P_n(x) t^n, \quad (3.11)$$

where P_n denotes the n th-Legendre polynomial [32]. This family of polynomials can be equivalently defined by the first two terms $P_0(x) := 1$, $P_1(x) := x$ and by the recurrence formula:

$$\forall n \geq 1, \quad (n+1)P_{n+1}(x) := (2n+1)xP_n(x) - nP_{n-1}(x). \quad (3.12)$$

Starting from (3.8), we have

$$\begin{aligned} \sqrt{z^2 + 2(2\mu_x^2 - 1)z + z^{-2}} &= z \frac{1 - 2(1 - 2\mu_x^2)z^{-2} + z^{-4}}{\sqrt{1 - 2(1 - 2\mu_x^2)z^{-2} + z^{-4}}} \\ &= z(1 - 2\alpha_x z^{-2} + z^{-4}) \sum_{n=0}^{\infty} P_n(\alpha_x) z^{-2n}, \end{aligned}$$

with $\alpha_x := 1 - 2\mu_x^2 \in (-1, 1)$. We therefore derive the expansion

$$2\mu_x z \kappa_s^0(z) = 1 - z^2 + (z^2 - 2\alpha_x + z^{-2}) \sum_{n=0}^{\infty} P_n(\alpha_x) z^{-2n}. \quad (3.13)$$

We focus below on the right boundary condition at point $x_{J+1} = x_r$. Going back to (3.10) and using the above expansion (3.13) for κ_s^0 , the DTBC (3.10) reads

$$2\mu_x z \hat{u}_{J+1}(z) = \left(1 + \sum_{n=0}^{\infty} B_n z^{-2n}\right) \hat{u}_J(z),$$

with

$$\forall n \in \mathbb{N}, \quad B_n := P_{n+1}(\alpha_x) - 2\alpha_x P_n(\alpha_x) + P_{n-1}(\alpha_x),$$

and it is understood that P_{-1} vanishes (in the expression of B_0). We compute $B_0 = -\alpha_x$, and thanks to the recurrence relation (3.12), we get the shorter expression

$$\forall n \geq 1, \quad B_n = \frac{P_{n-1}(\alpha_x) - P_{n+1}(\alpha_x)}{2n+1},$$

and finally

$$2\mu_x z \hat{u}_{J+1}(z) = \left(2\mu_x^2 + \sum_{n=1}^{\infty} \frac{P_{n-1}(\alpha_x) - P_{n+1}(\alpha_x)}{2n+1} z^{-2n}\right) \hat{u}_J(z). \quad (3.14)$$

Recall that in (3.14), we use the notation $\alpha_x = 1 - 2\mu_x^2$.

We now look for an efficient way to implement the coefficients of the above series without using the expression of the Legendre polynomials (which would be computationally rather cheap anyway). Let us define the sequence $(s_n^0)_{n \geq 0}$ such that $s_0^0 := \mu_x$, $s_1^0 := \mu_x(1 - \mu_x^2)$, and $s_n^0 := (P_{n-1}(\alpha_x) - P_{n+1}(\alpha_x))/((4n+2)\mu_x)$ for $n \geq 2$. After various simplifications using (3.12), we obtain

$$\forall n \geq 2, \quad s_n^0 = \frac{2n-1}{n+1} (1 - 2\mu_x^2) s_{n-1}^0 - \frac{n-2}{n+1} s_{n-2}^0. \quad (3.15)$$

The DTBC (3.14) therefore reads

$$\hat{u}_{J+1}(z) = \sum_{n=0}^{\infty} s_n^0 z^{-2n-1} \hat{u}_J(z). \quad (3.16)$$

Performing the inverse \mathcal{Z} -transform, we get⁴:

$$u_{J+1}^{n+2} = \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_J^{n+1-2m}. \quad (3.17)$$

The recurrence relation (3.15) may be efficiently implemented in a computer code, which gives access to the coefficients in the numerical boundary condition (3.17) up to any prescribed final time index N_f . Going back to (3.10), we also derive the “left” numerical boundary condition:

$$u_0^{n+2} = - \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_1^{n+1-2m}. \quad (3.18)$$

In view of understanding the stability properties of the numerical boundary condition (3.18), it might be important to determine the asymptotics of the sequence $(s_n^0)_{n \in \mathbb{N}}$. Let us recall that s_n^0 equals $(P_{n-1}(\alpha_x) - P_{n+1}(\alpha_x))/((4n+2)\mu_x)$ for $n \geq 2$ (here $\mu_x \in (0, 1)$ and $\alpha_x = 1 - 2\mu_x^2$). We use the so-called Laplace formula for the Legendre polynomials, see Theorem 8.21.2 of [32]:

$$\forall \theta \in (0, \pi), \quad P_n(\cos \theta) = \sqrt{\frac{2}{\pi n \sin \theta}} \cos \left(\left(n + \frac{1}{2} \right) \theta - \frac{\pi}{4} \right) + O(n^{-3/2}), \quad (3.19)$$

where the remainder term is even uniform with respect to θ on every compact set of the form $[\varepsilon, \pi - \varepsilon]$, $\varepsilon > 0$. We fix the angle $\theta_x \in (0, \pi)$ such that $\cos \theta_x = \alpha_x = 1 - 2\mu_x^2$, and therefore $\sin \theta_x = 2\mu_x \sqrt{1 - \mu_x^2}$. We then apply the Laplace formula and obtain after a few simplifications:

$$s_n^0 = \frac{(1 - \mu_x^2)^{1/4}}{\sqrt{\pi \mu_x n^3}} \sin \left(\left(n + \frac{1}{2} \right) \theta_x - \frac{\pi}{4} \right) + O(n^{-5/2}). \quad (3.20)$$

This asymptotic behavior can be verified on numerical experiments.

Determining inductively the expansion. As observed in [11, 15], the Laurent series expansion of κ_s^0 can also be computed inductively by using the equation (3.6). The main benefit is that the method does not rely on an explicit knowledge of a power series expansion such as the one involving the Legendre polynomials (which is useful only for three point schemes). The methodology below is therefore more flexible in view of being generalized to numerical schemes with larger stencils (as the one considered in [11]).

We know that κ_s^0 tends to zero at infinity so we may plug its Laurent series expansion⁵ $\sum_{n \geq 1} \sigma_n^0 z^{-n}$ into (3.6) and obtain

$$(z - z^{-1}) \sum_{n=1}^{\infty} \sigma_n^0 z^{-n} + \mu_x \left(\left(\sum_{n=1}^{\infty} \sigma_n^0 z^{-n} \right)^2 - 1 \right) = 0,$$

which is found to be equivalent to $\sigma_1^0 = \mu_x$ and

$$\forall n \in \mathbb{N}, \quad \sigma_{n+2}^0 = \sigma_n^0 - \mu_x \sum_{m=1}^n \sigma_m^0 \sigma_{n+1-m}^0, \quad (3.21)$$

⁴It is understood in (3.17) that the summation holds over all integers m between 0 and $(n+1)/2$. In case n is even, then the summation holds over all integers m between 0 and $n/2$.

⁵It is proved in [15] that the Laurent series is convergent for $|z| > 1$ which follows from the splitting of the two roots of (3.6) for $|z| > 1$.

where we use the convention $\sigma_0^0 = 0$. It is not difficult to see that if n is even, then $\sigma_n^0 = 0$. Let now $n = 2p + 1$ be odd, $p \geq 0$. An easy computation gives

$$\sigma_{2p+3}^0 = \sigma_{2p+1}^0 - \mu_x \sum_{m=0}^p \sigma_{2m+1}^0 \sigma_{2(p-m)+1}^0.$$

This means that the sequence $(s_n^0)_{n \in \mathbb{N}}$ appearing in (3.17) and that satisfies the recurrence relation (3.15) also satisfies

$$s_{n+1}^0 = s_n^0 - \mu_x \sum_{p=0}^n s_p^0 s_{n-p}^0, \quad (3.22)$$

with $s_0^0 = \mu_x$ (s_n^0 coincides with σ_{2n+1}^0). Observe that the value $s_1^0 = \mu_x(1 - \mu_x^2)$ that we have found in the previous paragraph is consistent with (3.22). It seems less clear to derive the asymptotic behavior (3.20) by starting from (3.22) rather than from the more explicit representation (3.14). Both approaches thus seem to have their own interest.

Given the initial conditions $(u_j^0)_{0 \leq j \leq J+1}$ and $(u_j^1)_{0 \leq j \leq J+1}$ as described above (u^0 is determined by the initial condition for the transport equation and u^1 is determined by applying the Lax–Wendroff method), both of which satisfy $u_0^0 = u_0^1 = u_{J+1}^0 = u_{J+1}^1 = 0$, the leap-frog scheme on the interval $\{1, \dots, J\}$ with DTBC reads:

$$u_j^{n+2} = u_j^n - \mu_x (u_{j+1}^{n+1} - u_{j-1}^{n+1}), \quad 1 \leq j \leq J, \quad n \in \mathbb{N}, \quad (3.23a)$$

$$u_0^{n+2} = - \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_1^{n+1-2m}, \quad (3.23b)$$

$$u_{J+1}^{n+2} = \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_J^{n+1-2m}, \quad (3.23c)$$

where the parameter μ_x in (3.23) is fixed such that $0 \leq \mu_x < 1$ and the sequence $(s_n^0)_{n \in \mathbb{N}}$ is defined by $s_0^0 = \mu_x$, $s_1^0 = \mu_x(1 - \mu_x^2)$ and the recurrence relation (3.15) for $n \geq 2$ (or equivalently (3.22), but this is slightly less efficient from a numerical point of view). We recall that we consider the case $c_x \geq 0$ in (3.1), hence the sign for μ_x , but the other case $c_x \leq 0$ can be dealt with the same arguments (assuming $-1 < \mu_x \leq 0$).

3.3. Stability analysis on a half-line

In this short paragraph, we explain why the normal mode analysis predicts “neutral stability” for any of the two half-line problems where the leap-frog scheme (3.23a) is considered on the half-line $\{j \leq J\}$ or $\{j \geq 1\}$, in combination with either (3.23b) or (3.23c). Let us focus on the case $\{j \geq 1\}$, since the other case is entirely similar.

The normal mode analysis consists in determining the solutions of the leap-frog scheme (3.23a) of the form $u_j^n = z^n v_j$, with $|z| > 1$, and $(v_j) \in \ell^2$. Among such sequences, the final question is to determine whether there exist nonzero ones that satisfy the numerical boundary condition (3.23b). Following [22], we shall say that the Godunov–Ryabenkii condition is satisfied if there is no nonzero sequence $(v_j) \in \ell^2$ such that $u_j^n = z^n v_j$ is a solution to the leap-frog scheme that satisfies (3.23b). Otherwise, we shall say that z is an unstable eigenvalue. Of course, unstable eigenvalues preclude (in the most violent way) stability estimates hence a desirable feature of any numerical boundary condition is to satisfy the Godunov–Ryabenkii condition.

Plugging the ansatz $u_j^n = z^n v_j$ in the leap-frog scheme, we find that the sequence (v_j) should belong to ℓ^2 and satisfy

$$\forall j \geq 1, \quad (z^2 - 1)v_j + \mu_x z(v_{j+1} - v_{j-1}) = 0.$$

The computation of the ℓ^2 -solutions to this recurrence relation follows by solving the characteristic equation (3.6) and by determining among the roots to (3.6) which have modulus less than 1. This classification has

already been performed in the previous paragraph, so for $|z| > 1$, we can conclude that the sequence (v_j) is given by

$$\forall j \geq 1, \quad v_j = v_0 \kappa_s^0(z)^j, \quad v_0 \in \mathbb{C}.$$

Inserting into (3.23b), we have to determine whether among such sequences (that are parametrized by their initial state v_0), we can have

$$v_0 = - \sum_{0 \leq m \leq (n+1)/2} s_m^0 z^{-2m-1} v_1.$$

Taking the limit $n \rightarrow \infty$, we thus need to determine whether there can hold

$$v_0 = -\kappa_s^0(z) v_1 = \frac{1}{\kappa_u^0(z)} v_1 = \frac{\kappa_s^0(z)}{\kappa_u^0(z)} v_0,$$

where the last equality follows from the expression of v_1 . In other words, we need to determine whether there can hold $\kappa_s^0(z) = \kappa_u^0(z)$ for some complex number z with $|z| > 1$. This is clearly impossible since $\kappa_s^0(z)$ has modulus < 1 and $\kappa_u^0(z)$ has modulus > 1 . This means that the so-called Godunov–Ryabenkii condition holds (see [22]): there does not exist any unstable eigenvalue for the half-space problem $\{j \geq 1\}$ with the numerical boundary condition (3.23b). Let us observe however that the roots κ_s^0 and κ_u^0 coincide when z equals one of the four values

$$\pm i \mu_x \pm \sqrt{1 - \mu_x^2},$$

for which the characteristic equation (3.6) has a double root. These values of z correspond to the *glancing* spatial frequencies κ (for which the associated group velocity vanishes). This is a neutral stability case, whose continuous counterpart has been analyzed in details in Chapter 7 of [10], see also [35] for other examples of this situation.

3.4. Fast implementation of approximate DTBC with sums of exponentials

The computation of the DTBC (3.23b) and (3.23c) at nodes x_ℓ and x_r and time $t^{n+2} = (n+2)\delta t$ requires $n/2$ sums and $n/2$ multiplications. Therefore, the total cost of discrete convolution computations for a full simulation up to time $T_f = N_f \delta t$ is $\mathcal{O}(N_f^2)$. Since δt is related to δx by the CFL condition (3.3), a fine space grid will make the time step small and as a consequence the number of time steps N_f large. Since the leap-frog scheme (3.2) is explicit, each time iteration requires $\mathcal{O}(J)$ operations for interior nodes x_j , $j = 1, \dots, J$, which gives $\mathcal{O}(JN_f)$ operations for a full simulation. Thus, the total number of operations is $\mathcal{O}(N_f^2 + N_f J)$. If $N_f > J$, which corresponds to *large time* simulations, the main part of the computational cost is related to the computation of discrete convolutions. Moreover, it is necessary to store in memory the evolution of the solution at the two boundary nodes. It may therefore be useful to reduce this part of the computation by applying a more “local” formula.

In [3, 17], a fast convolution procedure to compute approximation of discrete convolutions $C_n(v) := \sum_{k=0}^n v_k \nu_{n-k}$ was introduced. We assume that the convolution coefficients $(\nu_k)_{k \in \mathbb{N}}$ are given and satisfy a decay property similar to (3.20) that allows to derive their approximation by “sums of exponentials”. In order to present the idea and the efficiency of the method, let us assume for a moment that we are able to compute an approximation $(\tilde{\nu}_k)_{k \in \mathbb{N}}$ of the convolution terms $(\nu_k)_{k \in \mathbb{N}}$ given by

$$\nu_k \approx \tilde{\nu}_k = \sum_{m=1}^M b_m q_m^{-k}, \quad k \geq 0, \quad (3.24)$$

where the integer M , the coefficients b_m and the complex numbers q_m , all of them satisfying $|q_m| > 1$, have to be defined. Denoting $\tilde{C}_n(v) := \sum_{k=0}^n v_k \tilde{\nu}_{n-k}$, we compute

$$\tilde{C}_n(v) = \sum_{k=0}^n v_k \sum_{m=1}^M b_m q_m^{-(n-k)} = \sum_{m=1}^M b_m \sum_{k=0}^n v_k q_m^{-(n-k)} =: \sum_{m=1}^M C_m^{(n)}(v).$$

The efficiency of the method relies on the derivation of a recurrence relation to compute the terms $C_m^{(n)}(v)$:

$$C_m^{(n)}(v) = b_m \sum_{k=0}^{n-1} v_k q_m^{-(n-k)} + b_m v_n = b_m q_m^{-1} \sum_{k=0}^{n-1} v_k q_m^{-(n-1-k)} + b_m v_n,$$

so we get the recurrence relation

$$C_m^{(n)}(v) = q_m^{-1} C_m^{(n-1)}(v) + b_m v_n. \quad (3.25)$$

Assuming $v_0 = 0$, the initial value for each $C_m^{(n)}(v)$ is given by $C_m^{(0)}(v) = 0$. It should be understood that in a practical implementation, each value $C_m^{(n)}(v)$ is stored in memory so the recurrence (3.25) represents only three operations at each time iteration.

Accordingly, the original discrete convolution $C_n(v)$ is approximated by

$$C_n(v) = \sum_{k=0}^n v_k \nu_{n-k} \approx \tilde{C}_n(v) = \sum_{m=1}^M C_m^{(n)}(v), \quad (3.26)$$

and we have replace the computation of a nonlocal discrete convolution that involves n operations by the sum of M terms computed by the explicit local operations (3.25). In this case, the total number of operations is $\mathcal{O}(N_f(J + M))$.

It remains to derive the approximation (3.24) of the sequence $(\nu_k)_{k \in \mathbb{N}}$ by $(\tilde{\nu}_k)_{k \in \mathbb{N}}$. The sum-of-exponential formula (3.24) is frequent and various techniques are available to compute the coefficients $(b_m)_m$ and $(q_m)_m$ (see *e.g.*, [13, 14]). We present here a modified version of the algorithm derived in [3]. Let us consider the power series

$$f(x) := \sum_{k \in \mathbb{N}} \nu_k x^k, \quad \text{for } |x| < 1,$$

associated with the original sequence $(\nu_k)_{k \in \mathbb{N}}$ (which we assume to satisfy a polynomial bound of the form $|\nu_k| \lesssim k^\alpha$, $\alpha \in \mathbb{R}$, so the power series converges on the unit disk). We then consider its $[N, M]$ Padé approximant, see [5, 7], which we denote $\tilde{f}(x) = P_N(x)/Q_M(x)$, where P_N and Q_M are respectively polynomials of degree N and M . We assume Q_M to be unitary in order to normalize the polynomials. Here we shall always consider the case $N < M$. The power series expansion of \tilde{f} at 0 is given by

$$\tilde{f}(x) = \sum_{k \in \mathbb{N}} \tilde{\nu}_k x^k,$$

with $\tilde{\nu}_k = \nu_k$ for $0 \leq k \leq N + M$ (by definition of the Padé approximants).

Let us now assume that Q_M has M simple roots q_m with $|q_m| > 1$ for $1 \leq m \leq M$. So, we have

$$Q_M(x) = \prod_{m=1}^M (x - q_m), \quad \text{and} \quad Q'_M(x) = \sum_{k=1}^M \prod_{m \neq k} (x - q_m),$$

and thus

$$Q'_M(q_m) = \prod_{k \neq m} (q_m - q_k).$$

Let us now define the coefficients

$$b_m := -\frac{P_N(q_m)}{q_m Q'_M(q_m)}, \quad 1 \leq m \leq M. \quad (3.27)$$

So, we obtain

$$\begin{aligned} \sum_{m=1}^M \frac{b_m q_m}{q_m - x} &= \sum_{m=1}^M \frac{P_N(q_m)}{Q'_M(q_m)} \frac{1}{x - q_m} = \sum_{m=1}^M \frac{P_N(q_m)}{Q'_M(q_m)} \frac{\prod_{k \neq m} (x - q_k)}{\prod_{k=1}^M (x - q_k)} \\ &= \sum_{m=1}^M \frac{P_N(q_m)}{Q'_M(x)} \frac{\prod_{k \neq m} (x - q_k)}{Q'_M(q_m)}. \end{aligned}$$

Thereby,

$$\sum_{m=1}^M \frac{b_m q_m}{q_m - x} = \frac{1}{Q_M(x)} \sum_{m=1}^M P_N(q_m) \prod_{k \neq m} \frac{x - q_k}{q_m - q_k} =: \frac{R_{M-1}(x)}{Q_M(x)}.$$

We are going to show that the polynomial R_{M-1} in the latter equality equals P_N . Indeed, we have $R_{M-1}(q_m) = P_N(q_m)$ for any $1 \leq m \leq M$ and $d^\circ(R_{M-1}) \leq M - 1$. Since P_N is of degree $N \leq M - 1$, by uniqueness of the interpolating polynomial, we have $R_{M-1} = P_N$. As a partial conclusion, we have

$$\tilde{f}(x) = \frac{P_N(x)}{Q_M(x)} = \sum_{m=1}^M \frac{b_m q_m}{q_m - x},$$

where the coefficients b_m are defined in (3.27). Writing $q_m - x = q_m(1 - x/q_m)$, and using $|q_m| > 1$, then for $|x| \leq 1 < |q_m|$, we obtain

$$\frac{1}{q_m - x} = q_m^{-1} \sum_{k=0}^{\infty} \left(\frac{x}{q_m} \right)^k.$$

So,

$$\tilde{f}(x) = \sum_{m=1}^M b_m \sum_{k=0}^{\infty} \left(\frac{x}{q_m} \right)^k = \sum_{k=0}^{\infty} \sum_{m=1}^M b_m q_m^{-k} x^k.$$

Let us recall that the power series expansion of \tilde{f} at the origin reads $\tilde{f}(x) = \sum_{k \geq 0} \tilde{\nu}_k x^k$ so proceeding by identification, we have

$$\forall k \geq 0, \quad \tilde{\nu}_k = \sum_{m=1}^M b_m q_m^{-k},$$

and the coefficients $\tilde{\nu}_k$ satisfy the property

$$\tilde{\nu}_k = \nu_k \quad \text{for } 0 \leq k \leq M + N.$$

The computation of $(\tilde{\nu}_k)_{k \in \mathbb{N}}$ requires to determine the $[N, M]$ Padé approximant of f , the roots q_m of Q_M (praying for them to be simple) and the coefficients b_m given by (3.27).

Remark 3.2. In [3], the authors suggested to take $N = M - 1$. Unfortunately, when we apply the latter approximation strategy to the function $\kappa_s^0(1/x)$, we are not always able to verify the assumption $\inf_m |q_m| > 1$ on the roots of Q_M , even when these roots are computed with (very) high accuracy. Our choice to take N “free” and not necessarily equal to $M - 1$ allows us to find Padé approximants for which we can verify $\inf_m |q_m| > 1$. We present below some experiments to emphasize this remark.

Remark 3.3. In [3], the authors justified the convergence of the approximate coefficients $\tilde{\nu}_k$ to ν_k as $M \rightarrow \infty$. Their result was based on the so-called Baker–Gammel–Wills conjecture, which is unfortunately now known to be false, see [6] and references therein. That does not mean however that the above procedure is meaningless, as the numerical results in [3] and those we present below confirm. The main difficulty in applying the above approach is that there does not seem to be any general result available to make sure that the condition $\inf_m |q_m| > 1$ holds in a systematic way (not mentioning the condition that the roots q_m should be simple...).

In [3], a Maple code is proposed to calculate the Padé approximant of f . The use of Maple insures to not be limited to double precision computations. This is very important to compute accurately the coefficients of the monomials of P_N and Q_M , and accordingly to determine the roots of Q_M with high accuracy. This process is however quite limited since it relies on formal calculus softwares and is also restricted to small values of M (typically $M \leq 20$). We therefore present below an alternative self-contained procedure which allows us to consider much larger values of M (say, $M \leq 200$) provided that N is not close to M (say $N \leq 50$). When N is close to M , the coefficients of P_N and Q_M become extremely large and the numerical computations become rather unstable. The $[N, M]$ Padé approximant \tilde{f} of f reads

$$\tilde{f}(x) = \frac{P_N(x)}{Q_M(x)} = \frac{\sum_{j=0}^N \mathfrak{p}_j x^j}{\sum_{j=0}^M \mathfrak{q}_j x^j},$$

where we fix for instance $\mathfrak{q}_0 = 1$, and we restrict to the case $N < M$. (The normalization convention for Q_M is not the same as above but this does not affect the previous arguments.) We therefore have to compute $N + M + 1$ unknown coefficients, which are solutions of the $N + M + 1$ equations given by

$$D^{(n)} [Q_M(x) f(x)]|_{x=0} = D^{(n)} [P_N(x)]|_{x=0}, \quad 0 \leq n \leq N + M, \quad (3.28)$$

where $D^{(n)}$ denotes the n th derivative with respect to x . Indeed, thanks to the property $Q_M(0) \neq 0$, (3.28) can be seen to be equivalent to the property

$$D^{(n)} [f(x)]|_{x=0} = D^{(n)} \left[\frac{P_N(x)}{Q_M(x)} \right]|_{x=0}, \quad 0 \leq n \leq N + M,$$

which defines the $[N, M]$ Padé approximant of f . The equations (3.28) reduce to

$$\sum_{k=1}^n \mathfrak{q}_k \nu_{n-k} - \mathfrak{p}_n = -\nu_n, \quad \text{for } 0 \leq n \leq N, \quad (3.29)$$

and

$$\sum_{k=1}^{\min(n, M)} \mathfrak{q}_k \nu_{n-k} = -\nu_n, \quad \text{for } N < n \leq N + M. \quad (3.30)$$

The first equation in (3.29) leads to $\mathfrak{p}_0 = \nu_0 = f(0)$, and the remaining equations in (3.29) and (3.30) can be recast as a linear system:

$$S X = Y, \quad (3.31)$$

where $S \in \mathbb{M}_{N+M, N+M}(\mathbb{R})$ and X, Y are vectors in \mathbb{R}^{N+M} . The matrix S in (3.31) is given by (the first row and column below present the indices for the sake of clarity):

	1	2	\dots	N	$N+1$	M	$M+1$	\dots	$N+M$
1	ν_0						-1		
2	ν_1	\ddots						\ddots	
\vdots	\vdots	\ddots	\ddots					\ddots	
N	ν_{N-1}	\dots	ν_1	ν_0					-1
$N+1$	ν_N	\dots	\dots	ν_1	ν_0		0		
\vdots	\vdots			\vdots	ν_1	\ddots		\ddots	
\vdots	\vdots			\vdots	\vdots	\ddots		\ddots	
M	ν_{M-1}	\dots	\dots	ν_{M-N}	ν_{M-N+1}	\dots	ν_1	ν_0	0
$M+1$	ν_M	\dots	\dots	ν_{M-N+1}	ν_{M-N}	\dots	ν_1	0	
\vdots	\vdots			\vdots	\vdots		\vdots		
\vdots	\vdots			\vdots	\vdots		\vdots		
$N+M$	ν_{N+M-1}	\dots	\dots	ν_M	ν_{M-1}	\dots	\dots	ν_N	0

The unknown in (3.31) is $X = (\mathbf{q}_1, \dots, \mathbf{q}_M, \mathbf{p}_1, \dots, \mathbf{p}_N)^T$, and the right hand side is $Y = (-\nu_1, -\nu_2, \dots, -\nu_{N+M})^T$. The linear system (3.31) thus displays block matrices

$$\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix},$$

with $A \in \mathbb{M}_{M,M}(\mathbb{R})$, $B \in \mathbb{M}_{M,N}(\mathbb{R})$ et $C \in \mathbb{M}_{N,M}(\mathbb{R})$. The matrix A is a subtriangular Toeplitz matrix. Its inverse (assuming that ν_0 is nonzero) is also a subtriangular Toeplitz matrix whose first column is easy to compute by solving the linear system $Ax = e_1$ where e_1 is the first vector of the canonical basis of \mathbb{R}^M . The other columns of A^{-1} are deduced from the Toeplitz structure.

Using the Schur complement method, we can compute the unknowns by solving successively

$$CA^{-1}B\mathbf{p} = CA^{-1}Y_1 - Y_2,$$

and then

$$\mathbf{q} = A^{-1}(Y_1 - B\mathbf{p}).$$

In order to get high accuracy both in the computations of \mathbf{p} and \mathbf{q} , but also of the roots q_m of Q_M , we use the floating-point arithmetic with arbitrary accuracy Python library `mpmath`. The advantage of using the library `mpmath` is its ability to control the accuracy of floating-point arithmetic. For all the numerical experiments in this paper, we set the decimal accuracy to 80, meaning the library uses 266 bits to hold an approximation of the numbers that is accurate to 80 decimal places.

Going back to our numerical scheme (3.23), we wish to approximate the sequence (s_n^0) given by (3.15) by a sum of exponentials. We apply the above algorithm, which is legitimate since $s_0^0 \neq 0$ (s_n^0 plays the above role of ν_n for all n). We present below a comparison between the thousand first elements of the sequence (s_n^0)

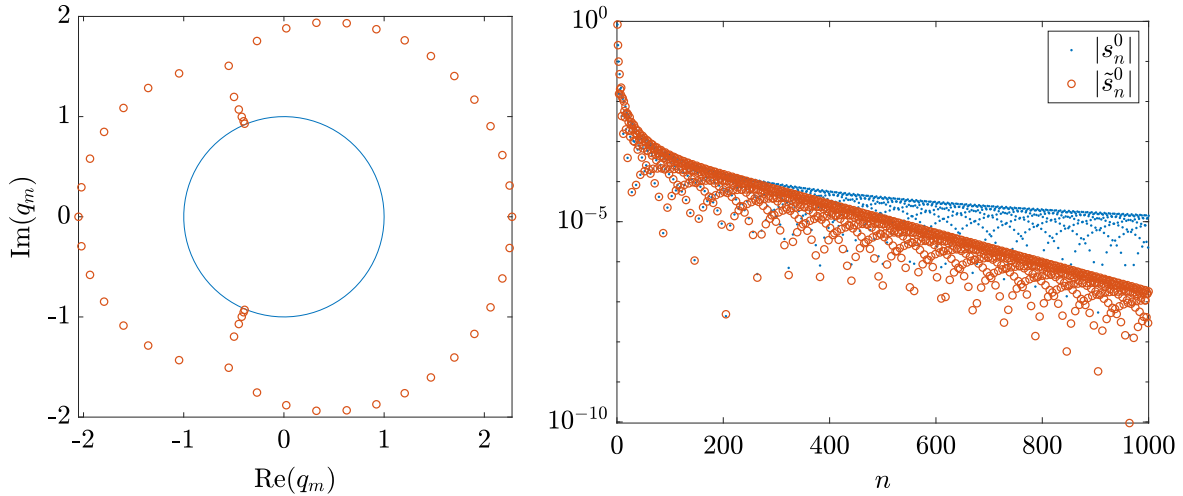


FIGURE 1. Position of the roots of Q_M (left), and $|s_n^0|$ vs. $|\tilde{s}_n^0|$ (right) for $(M, N) = (50, 10)$.

given by (3.15) and their sum of exponentials approximations (\tilde{s}_n^0) for various pairs (N, M) with $N < M$. The numerical parameters are chosen to satisfy the CFL condition (3.3) with $\text{cfl} = 5/6$. The transport velocity is $c = 1$. The space and time steps are $\delta x = 6/1000$ and $\delta t = 5/1000$ (which corresponds to the space interval $[-3, 3]$ and choosing a thousand grid points). On Figures 1–4, we plot on the left the position of the roots (q_m) of Q_M in the complex plane, compared with the unit circle, and the evolution of \tilde{s}_n^0 compared with that of s_n^0 for $0 \leq n \leq 1000$. We first choose $M = 50$ with $N = 10$ (see Fig. 1), then $M = 50$ with $N = 49$ (see Fig. 2), then $M = 100$ with $N = 30$ (see Fig. 3), and eventually $M = 100$ with $N = 99$ (see Fig. 4). On Figures 2 and 4, one root of the polynomial Q_M is relatively big compared with the other roots and we therefore do not represent it. They respectively take the values $-4.6 \cdot 10^{17}$ and $-1.2 \cdot 10^{29}$, which means that some of the coefficients of Q_M are so large that any computation involving the roots of Q_M should be taken with much care. We see that the more N is close to M , the more \tilde{s}_n^0 are close to s_n^0 for a large interval of values of n . The quality of the approximations for $(M, N) = (50, 49)$ and $(M, N) = (100, 30)$ is equivalent; the case $(M, N) = (100, 30)$ is however much more stable since all roots of the polynomial Q_M remain within a “small” ball of the complex plane while Q_M has a very large root in the case $(M, N) = (50, 49)$. When $(M, N) = (100, 99)$, one of the roots that we computed for Q_M has a modulus less than one, which generates blow up for sequence (\tilde{s}_n^0) (see Fig. 4). This does not prove however that Q_M has a root of modulus less than 1, because the computation is so unstable (due to the very large coefficients in Q_M) that the final result should be taken with care even with the decimal accuracy set to 80.

3.5. Numerical experiments

We present here some numerical tests with the following parameters:

$$u_0(x) = \exp(-10x^2), \quad x_\ell = -3, \quad x_r = 3, \quad T = 10, \quad c = 1.$$

The number of grid points is $J + 1 = 1000$, and the leap-frog scheme (3.2) is implemented with a CFL parameter $\mu_x = 5/6$ (the time step δt is computed accordingly). Since the initial condition is roughly concentrated in the interval $[-1, 1]$, the exact solution to the transport equation more or less vanishes on the interval $[x_\ell, x_r]$ after the time 4. We measure below the amplitude of the reflected wave at the right boundary x_r depending on the choice of numerical boundary conditions.

We present below the evolution of the logarithm $\log_{10} |u_j^n|$ rather than the evolution of the solution u_j^n itself. This gives a much more visible representation of wave reflections and their magnitude. Let us observe that in

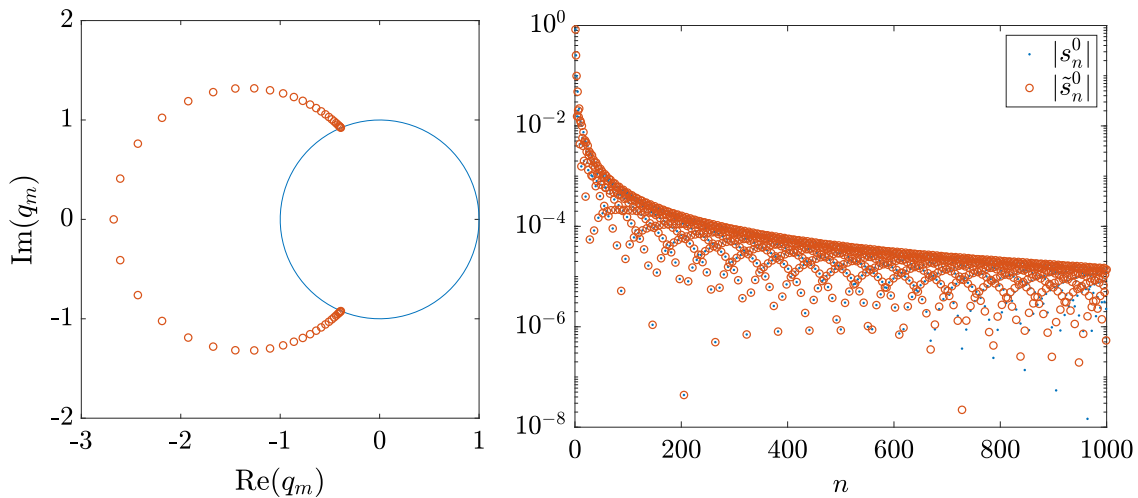


FIGURE 2. Position of the roots of Q_M (left), and $|s_n^0|$ vs. $|\tilde{s}_n^0|$ (right) for $(M, N) = (50, 49)$.

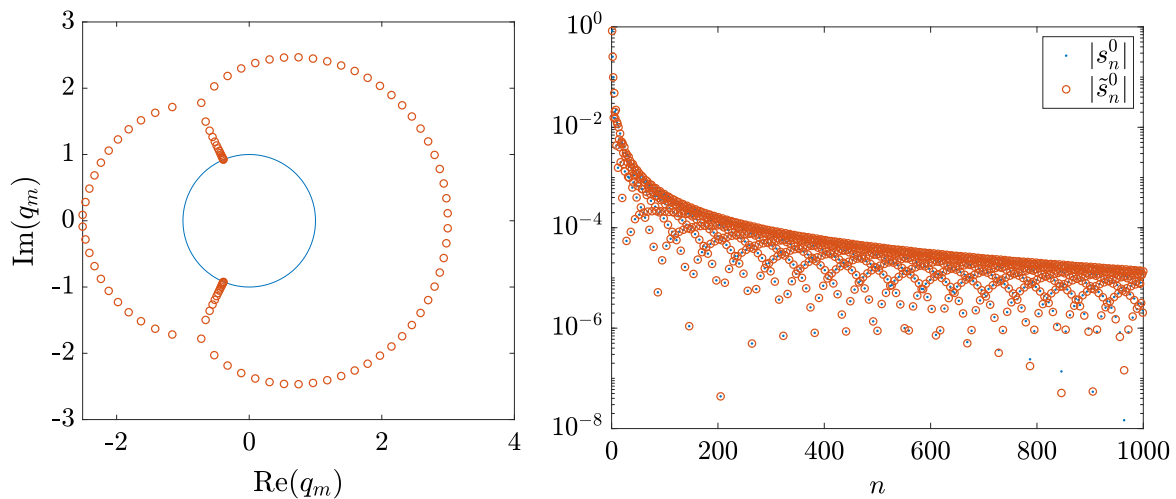


FIGURE 3. Position of the roots of Q_M (left), and $|s_n^0|$ vs. $|\tilde{s}_n^0|$ (right) for $(M, N) = (100, 30)$.

Figures 5 and 6 below, we always observe a left going wave emanating from the initial condition, this wave being highly oscillatory (in both space and time) and having magnitude 10^{-8} . This is due to the fact that the leap-frog scheme (3.2) supports the wave $(-1)^{j+n}$ which has group velocity $-c$ and thus travels backwards [34]. In Figure 5, we first compare the leap-frog scheme with exact DTBC (3.23) (left) with the leap-frog scheme implemented with the more classical Neumann type boundary condition $u_{J+1}^{n+2} = u_J^{n+1}$, see [35] (accordingly the numerical boundary condition at x_0 is $u_0^{n+2} = u_1^{n+1}$). After the theoretical exit time for the continuous solution, the DTBC strategy leaves a solution of amplitude 10^{-16} in the whole domain while the easier (local) Neumann strategy exhibits multiple wave reflections.

We then show on Figure 6 the evolution of the logarithm $\log_{10} |u_j^n|$ when we do not implement the exact DTBC (3.23b) and (3.23c) but rather their approximation by sums of exponentials as explained in the preceding paragraph. The numerical simulation with the parameters $(M, N) = (50, 6)$ (left of Fig. 6) and $(M, N) = (50, 49)$

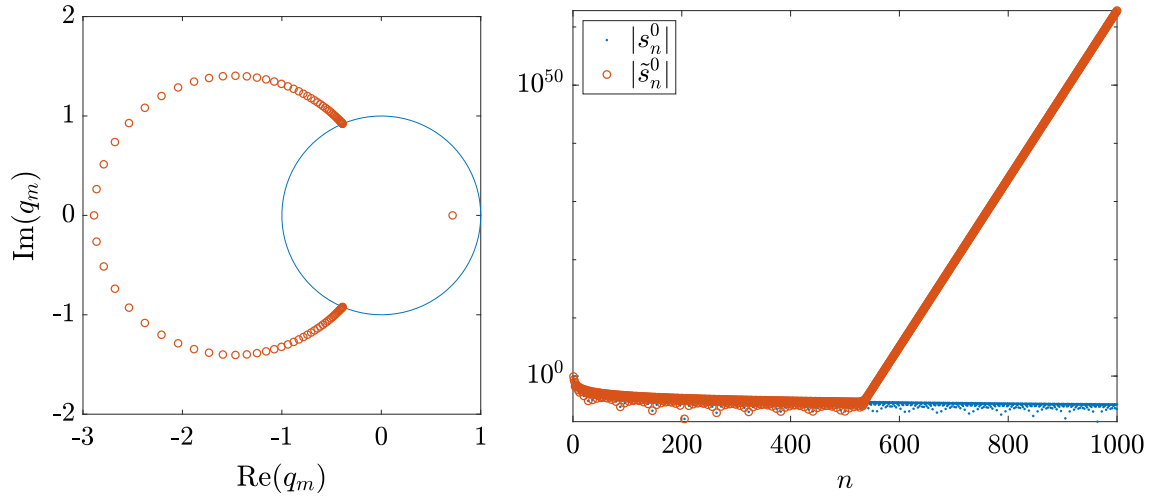


FIGURE 4. Position of the roots of Q_M (left), and $|s_n^0|$ vs. $|\tilde{s}_n^0|$ (right) for $(M, N) = (100, 99)$.

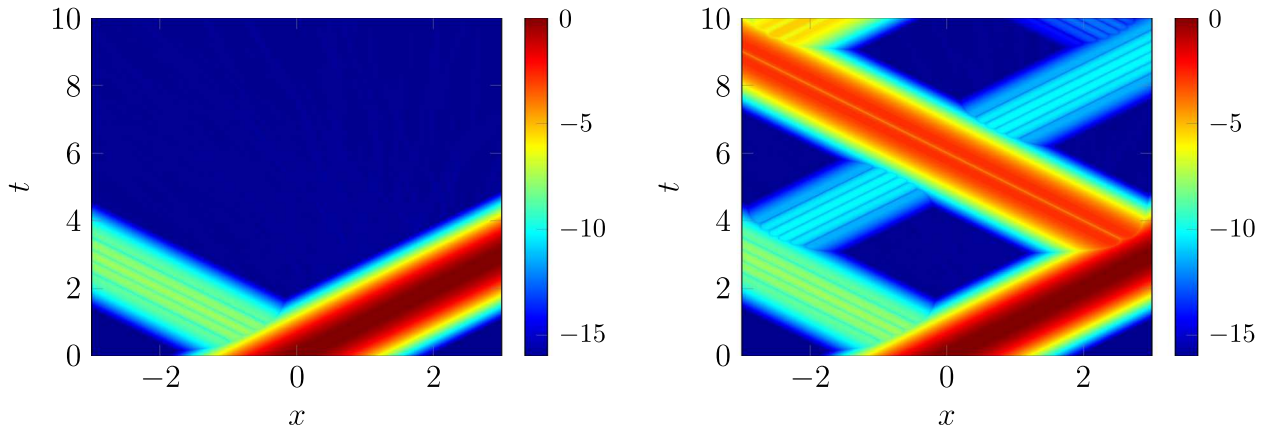


FIGURE 5. Evolution of $\log_{10}|u_j^n|$ for DTBC (3.23) (left) and homogeneous Neumann BC $u_0^{n+2} = u_1^{n+1}$, $u_{J+1}^{n+2} = u_J^{n+1}$ (right).

(right of Fig. 6) for the degrees of the Padé approximant. In each of these two cases, the condition that all roots of Q_M are simple and lie outside the unit circle has been checked numerically. We observe that in the numerical boundary conditions (3.23b) and (3.23c), the determination of u_0^{n+2} and u_{J+1}^{n+2} relies on the values u_1^σ and u_J^σ for either only odd or even values of σ . When we implement the approximate numerical boundary condition (3.26), we thus need to consider separately the cases where n is either odd or even, and we therefore introduce two sequences $(C_m^{(2p)})$ and $(C_m^{(2p+1)})$ for each root q_m of the polynomial Q_M . The numerical results presented in Figure 6 exhibit very low reflection amplitudes. In the case $(M, N) = (50, 49)$, the overall accuracy is comparable with the exact DTBC. Note however that the condition $\inf_m |q_m| > 1$ for this approximation to make sense is not granted and the implementation of the sum of exponential approximation requires first solving for P_N and Q_M and then identifying the roots of Q_M (which is not necessarily easier than implementing (3.15) and (3.23)).

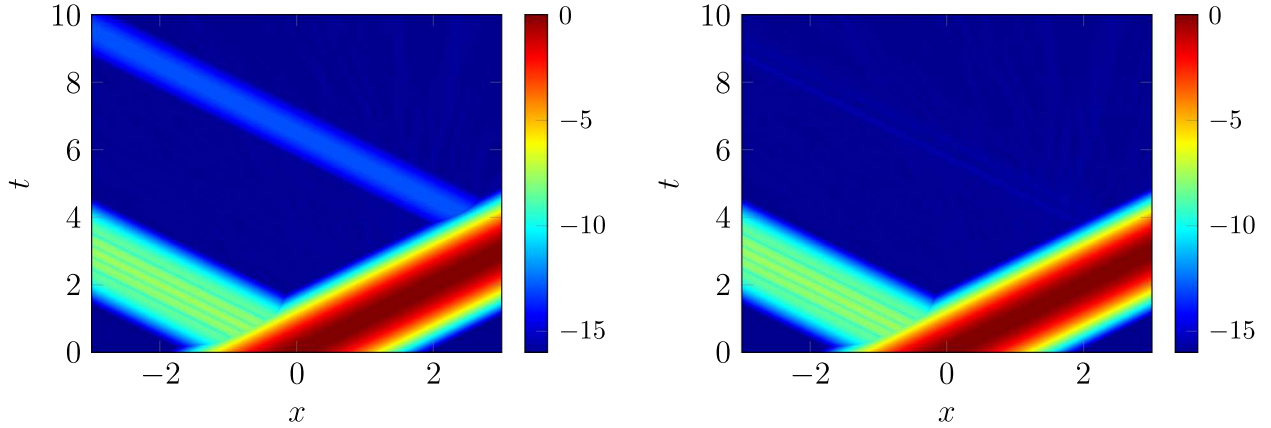


FIGURE 6. Evolution of $\log_{10} |u_j^n|$ for the sum of exponentials BC (3.26): $(M, N) = (50, 6)$ (left) and $(M, N) = (50, 49)$ (right).

4. APPROXIMATE DTBC FOR THE TWO-DIMENSIONAL LEAP-FROG SCHEME ON A RECTANGLE

4.1. Exact DTBC on a half-space and local approximations

In this paragraph, we go back to the leap-frog scheme (2.2) for the linear advection equation (2.1) in two space dimensions. Our first goal is to derive the exact DTBC for (2.2) when that scheme is considered on a half-space, be it $\{j \geq 1\}$, $\{j \leq J\}$, $\{k \geq 1\}$ or $\{k \leq K\}$. This means that we shall consider the leap-frog scheme (2.2) in the whole plane $(j, k) \in \mathbb{Z}^2$ with initial data supported, say, in the half-space $\{j \geq 1\}$, and we shall derive the numerical boundary conditions satisfied by the solution to (2.2) at $j = 0$. This is exactly the problem considered in [15] in the general framework of multistep finite difference schemes. For ease of reading and in order to stick to the notation of Section 3, we consider from now on the case where the initial data for (2.2) are supported in $\{j \geq 1\}$, and it is understood that in (2.2) the tangential variable k lies in \mathbb{Z} . We consider the step function $u_j^n(\cdot)$ defined by

$$u_j^n(y) := u_{j,k}^n, \quad \forall y \in [y_k, y_{k+1}), \quad k \in \mathbb{Z},$$

with $y_k = y_b + k \delta y$ for all $k \in \mathbb{Z}$ and not only for $k = 0, \dots, K + 1$. Then (2.2) reads

$$u_j^{n+2}(y) - u_j^n(y) + \mu_x (u_{j+1}^{n+1}(y) - u_{j-1}^{n+1}(y)) + \mu_y (u_j^{n+1}(y + \delta y) - u_j^{n+1}(y - \delta y)) = 0,$$

so applying first a partial Fourier transform with respect to y and then the \mathcal{Z} -transform in time, we are led (with obvious notation) to the recurrence relation

$$(z^2 - 1) \hat{u}_j(z, \eta) + \mu_x z (\hat{u}_{j+1}(z, \eta) - \hat{u}_{j-1}(z, \eta)) + 2i \mu_y \sin(\eta \delta y) z \hat{u}_j(z, \eta) = 0, \quad (4.1)$$

which holds for all $j \leq 0$ (in the half-space $\{j \geq 1\}$ where the initial data are supported, the analogous recurrence relation displays a nonzero source term on the right hand side).

The solution to the recurrence relation (4.1) is computed by considering the characteristic equation:

$$(z^2 - 1) \kappa + \mu_x z (\kappa^2 - 1) + 2i \mu_y \sin \theta z \kappa = 0, \quad (4.2)$$

where θ is a placeholder for the rescaled frequency $\eta \delta y$. For any real number θ and $|z| > 1$, the roots to (4.1) do not belong to \mathbb{S}^1 . Therefore one has modulus < 1 and the other one has modulus > 1 (see the general splitting argument in [15]). We label the two roots in such a way that:

$$|\kappa_s(z, \theta)| < 1, \quad |\kappa_u(z, \theta)| > 1,$$

whenever $|z| > 1$ and $\theta \in \mathbb{R}$. The discrete boundary condition (on the Fourier- \mathcal{Z} -transform side) thus reads

$$\hat{u}_0(z, \eta) = \frac{1}{\kappa_u(z, \eta \delta y)} \hat{u}_1(z, \eta), \quad (4.3)$$

which is reminiscent of (3.9). The analogous numerical boundary condition at $j = J + 1$ (for the other half space problem) is

$$\hat{u}_{J+1}(z, \eta) = \kappa_s(z, \eta \delta y) \hat{u}_J(z, \eta).$$

If we now apply the inverse Fourier- \mathcal{Z} -transform to (4.3), this would lead to the exact, nonlocal DTBC for (2.2) on a half space. Nonlocality refers here both to time and to the tangential variable k . In other words, determining $u_{0,k}^{n+2}$ requires the knowledge of $u_{1,k'}^\sigma$ for all $k' \in \mathbb{Z}$ and at all time steps σ up to $n+1$. The nonlocality in time is not so harmful from a computational point of view (as we have seen in one space dimension) but the nonlocality with respect to k is much more problematic since the set to which k belongs is infinite. We therefore propose to follow the idea developed by Engquist and Majda [19] for continuous problems and to localize the exact DTBC (4.3) in the k -direction. This is performed by means of an asymptotic expansion with respect to $\theta = \eta \delta y$ in (4.3). At a formal level, this amounts more or less to assuming that the discrete solution displays only bounded tangential frequencies η and to performing an asymptotic expansion with respect to δy (which is meant to be small in practice).

Let us therefore go back to (4.2) and observe that again the product of the two roots κ_s and κ_u equals -1 . Hence (4.3) reads

$$\hat{u}_0(z, \eta) = -\kappa_s(z, \eta \delta y) \hat{u}_1(z, \eta),$$

and we now expand the stable root $\kappa_s(z, \theta)$ to (4.2) with respect to θ . Up to the third order, this expansion reads

$$\kappa_s(z, \theta) = \kappa_s^0(z) + 2i \sin \theta \kappa_s^1(z) - 4 \sin^2 \frac{\theta}{2} \kappa_s^2(z) + O(\theta^3), \quad (4.4)$$

where $\kappa_s^0(z)$ is the stable root for the one-dimensional problem (which corresponds to $\theta = 0$ in (4.2)) and the choice of writing $\sin \theta$ rather than θ , as well as $\sin^2 \theta/2$ rather than θ^2 , has been made in order to exhibit amplification factors that are obviously linked with tangential finite difference operators (the centered first order derivative and the discrete Laplacian). At this stage, localizing the exact DTBC (4.3) with respect to k amounts to using in (4.3) finitely many terms of the Taylor expansion (4.4) (*e.g.*, either the first term, or the two/three first terms). Namely, on the Fourier- \mathcal{Z} -transform, our localization procedure for (4.3) yields one of the three following choices (in increasing order of approximation):

$$\hat{u}_0(z, \eta) = -\kappa_s^0(z) \hat{u}_1(z, \eta), \quad (4.5a)$$

$$\hat{u}_0(z, \eta) = -\kappa_s^0(z) \hat{u}_1(z, \eta) - 2i \sin(\eta \delta y) \kappa_s^1(z) \hat{u}_1(z, \eta), \quad (4.5b)$$

$$\hat{u}_0(z, \eta) = -\kappa_s^0(z) \hat{u}_1(z, \eta) - 2i \sin(\eta \delta y) \kappa_s^1(z) \hat{u}_1(z, \eta) + 4 \sin^2 \frac{\eta \delta y}{2} \kappa_s^2(z) \hat{u}_1(z, \eta). \quad (4.5c)$$

After performing an inverse Fourier transform with respect to η , (4.5) reads:

$$\hat{u}_{0,k}(z) = -\kappa_s^0(z) \hat{u}_{1,k}(z), \quad (4.6a)$$

$$\hat{u}_{0,k}(z) = -\kappa_s^0(z) \hat{u}_{1,k}(z) - \kappa_s^1(z) (\hat{u}_{1,k+1}(z) - \hat{u}_{1,k-1}(z)), \quad (4.6b)$$

$$\hat{u}_{0,k}(z) = -\kappa_s^0(z) \hat{u}_{1,k}(z) - \kappa_s^1(z) (\hat{u}_{1,k+1}(z) - \hat{u}_{1,k-1}(z)) - \kappa_s^2(z) (\hat{u}_{1,k+1}(z) - 2\hat{u}_{1,k}(z) + \hat{u}_{1,k-1}(z)), \quad (4.6c)$$

where the hat notation refers here to the \mathcal{Z} -transform only. In order to perform the inverse \mathcal{Z} -transform in (4.6) and write down the approximate non-reflecting boundary conditions in the physical variables, we need to compute the Laurent series expansion of the functions κ_s^1, κ_s^2 that appear on the right hand side of (4.6b) and (4.6c). (The Laurent series expansion of κ_s^0 has already been derived in the analysis of the one-dimensional problem.) This is achieved below with either of the two methods that we have already used in the one-dimensional case.

Expansions based on Legendre (and Tchebychev) polynomials. Let us recall that we have already determined the Laurent series expansion of the function κ_s^0 in (4.6). We have written it under the form

$$\kappa_s^0(z) = \sum_{n=1}^{\infty} \frac{\sigma_n^0}{z^n} = \sum_{n=0}^{\infty} \frac{s_n^0}{z^{2n+1}},$$

where the sequence $(s_n^0)_{n \geq 0}$ is determined by either (3.15) or (3.22). The two first values are $s_0^0 = \mu_x$ and $s_1^0 = \mu_x(1 - \mu_x^2)$. Let us eventually recall that the expression of $\kappa_s^0(z)$ is given by (3.8).

We are now going to determine the Laurent series expansion of $\kappa_s^1(z)$. The root $\kappa_s(z, \theta)$ to (4.2) is simple for $\theta \in \mathbb{R}$ and $|z| > 1$. It thus depends holomorphically on z and is \mathcal{C}^∞ with respect to θ . Differentiating (4.2) with respect to θ at $\theta = 0$, and identifying $\kappa_s^1 = \partial_\theta \kappa_s(z, 0)/(2i)$, we obtain

$$(z^2 - 1 + 2\mu_x z \kappa_s^0(z)) \kappa_s^1(z) = -\mu_y z \kappa_s^0(z), \quad (4.7)$$

which yields

$$\begin{aligned} \kappa_s^1(z) &= -\frac{\mu_y \kappa_s^0}{\sqrt{(z - z^{-1})^2 + 4\mu_x^2}} = -\frac{\mu_y}{2\mu_x} \left\{ 1 + \frac{z^{-2} - 1}{\sqrt{(z^{-2} - 1)^2 + 4\mu_x^2 z^{-2}}} \right\} \\ &= -\frac{\mu_y}{2\mu_x} \sum_{n=1}^{\infty} \frac{P_{n-1}(\alpha_x) - P_n(\alpha_x)}{z^{2n}}, \end{aligned} \quad (4.8)$$

by using (3.8) and (3.11). We can therefore write the Laurent series expansion of κ_s^1 under the form:

$$\kappa_s^1(z) = \sum_{n=0}^{\infty} \frac{s_n^1}{z^{2n}}, \quad (4.9)$$

with $s_0^1 = 0$ and

$$\forall n \geq 1, \quad s_n^1 := \frac{\mu_y}{2\mu_x} (P_n(\alpha_x) - P_{n-1}(\alpha_x)). \quad (4.10)$$

We can, for instance, compute (recall $s_0^1 = 0$):

$$s_1^1 = -\mu_x \mu_y, \quad s_2^1 = -\mu_x \mu_y (2 - 3\mu_x^2).$$

The expression (4.10) is only seemingly singular with respect to μ_x for $\mu_x \ll 1$. Indeed, we recall $\alpha_x = 1 - 2\mu_x^2$ and since all Legendre polynomials satisfy $P_n(1) = 1$, the expression $P_n(\alpha_x) - P_{n-1}(\alpha_x)$ can be written as $\mu_x^2 Q_n(\mu_x^2)$ for some polynomial Q_n . We therefore see from (4.10) that all coefficients s_n^1 read $\mu_y \mu_x Q_n(\mu_x^2)$ for some real polynomial Q_n . Using again the Laplace formula (3.19), we also get the asymptotic behavior:

$$s_n^1 = -\frac{\mu_y}{\mu_x} \sqrt{\frac{\tan(\theta_x/2)}{\pi n}} \sin\left(n\theta_x - \frac{\pi}{4}\right) + O(n^{-3/2}), \quad (4.11)$$

where the angle $\theta_x \in (0, \pi)$ is still defined by $\cos \theta_x = 1 - 2\mu_x^2$. The asymptotic behavior (4.11) can be verified by numerical experiments. Comparing with (3.20), we observe that the sequence (s_n^1) has a slower decay than (s_n^0) , but it decays to zero nevertheless. This is due to the fact that κ_s^1 has (four) singularities of the form $(z/z_0 - 1)^{-1/2}$, $z_0 \in \mathbb{S}^1$, as z approaches the unit circle \mathbb{S}^1 , while κ_s^0 is continuous up to \mathbb{S}^1 .

The Laurent series expansion of κ_s^2 can be derived by following more or less the same lines. By differentiating twice (4.2) with respect to θ and plugging the expression (4.7) of κ_s^1 , we first get:

$$\begin{aligned}\kappa_s^2(z) &= -4z\kappa_s^1(z)\frac{\mu_y + \mu_x\kappa_s^1(z)}{z^2 - 1 + 2\mu_x z\kappa_s^0(z)} \\ &= \frac{4\mu_y^2 z^2}{(z^2 - 1 + 2\mu_x z\kappa_s^0(z))^2} \frac{\kappa_s^0(z)(z^2 - 1 + \mu_x z\kappa_s^0(z))}{z^2 - 1 + 2\mu_x z\kappa_s^0(z)} \\ &= \frac{4\mu_y^2 z^2}{(z^2 - 1 + 2\mu_x z\kappa_s^0(z))^2} \frac{\mu_x z}{z^2 - 1 + 2\mu_x z\kappa_s^0(z)}.\end{aligned}\quad (4.12)$$

In other words, we have derived the relation

$$\kappa_s^2(z) = \frac{4\mu_x\mu_y^2}{z^3} \frac{1}{(1 - z^{-2})^2 + 4\mu_x^2} \frac{1}{\sqrt{(1 - z^{-2})^2 + 4\mu_x^2}},$$

and it only remains to use (3.11) as well as the generating function of the Tchebychev polynomials of the second kind (see [32]):

$$\frac{1}{1 - 2xt + t^2} = \sum_{n=0}^{\infty} U_n(x)t^n, \quad (4.13)$$

to derive

$$\kappa_s^2(z) = \frac{4\mu_x\mu_y^2}{z^3} \sum_{n=0}^{\infty} \left(\sum_{m=0}^n U_m(\alpha_x) P_{n-m}(\alpha_x) \right) z^{-2n}.$$

Writing the Laurent series expansion of κ_s^2 under the form:

$$\kappa_s^2(z) = \sum_{n=0}^{\infty} \frac{s_n^2}{z^{2n+1}}, \quad (4.14)$$

we have derived the relations $s_0^2 = 0$ and

$$\forall n \geq 1, \quad s_n^2 = 4\mu_x\mu_y^2 \sum_{m=0}^{n-1} U_m(\alpha_x) P_{n-1-m}(\alpha_x), \quad (4.15)$$

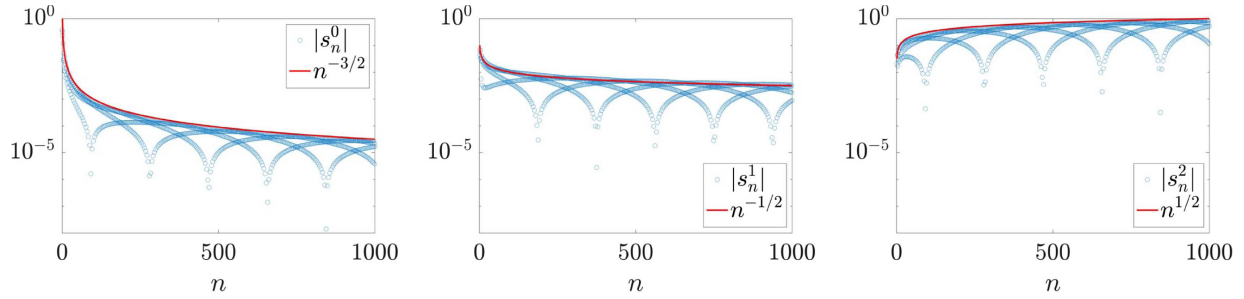
with $\alpha_x = 1 - 2\mu_x^2$. We can for instance compute

$$s_1^2 = 4\mu_x\mu_y^2, \quad s_2^2 = 12\mu_x\mu_y^2(1 - 2\mu_x^2).$$

It does not seem so easy to infer from (3.19) and from the relation

$$U_m(\cos \theta) = \frac{\sin((m+1)\theta)}{\sin \theta},$$

the asymptotic behavior of $(s_n^2)_{n \in \mathbb{N}}$. However, we verify below on some numerical experiments that $(s_n^2)_{n \in \mathbb{N}}$ does not seem to be bounded. It actually seems to grow like \sqrt{n} , see Figure 7, up to an oscillating behavior similar to the one we have exhibited in (3.20) and (4.11). This growth in \sqrt{n} is consistent with the singularities of the form $(z/z_0 - 1)^{-3/2}$ displayed by κ_s^2 . The rigorous justification of this asymptotic behavior is left to a future work.

FIGURE 7. Asymptotic behaviour of s_n^0 , s_n^1 and s_n^2 .

Inductive determination of the expansions. As we have already seen in the analysis of the one-dimensional case, the Laurent series of κ_s^0 can be also determined inductively by plugging the expansion

$$\kappa_s^0(z) = \sum_{n=1}^{\infty} \frac{\sigma_n^0}{z^n},$$

by then first identifying σ_1^0 and then deriving a recurrence formula which gives the expression of σ_n^0 in terms of $\sigma_1^0, \dots, \sigma_{n-1}^0$. This led to the formula (3.21). This strategy can be extended to the “correctors” κ_s^1 and κ_s^2 . Namely, the function κ_s^1 is given by the relation (4.7). Since $z\kappa_s^0(z)$ has a finite limit at infinity, we already see on (4.7) that κ_s^1 tends to zero at infinity. It actually decays like z^{-2} . Hence its Laurent series expansion reads

$$\kappa_s^1(z) = \sum_{n=1}^{\infty} \frac{\sigma_n^1}{z^n},$$

and the coefficients $(\sigma_n^1)_{n \geq 1}$ are computed by plugging the latter expression in (4.7) (which itself has been obtained by differentiating (4.2) with respect to θ). After a few simplifications, we obtain $\sigma_1^1 = 0$ and the recurrence relation

$$\forall n \geq 1, \quad \sigma_{n+1}^1 - \sigma_{n-1}^1 + 2\mu_x \sum_{m=0}^n \sigma_m^1 \sigma_{n-m}^0 = -\mu_y \sigma_n^0,$$

where we use the convention $\sigma_0^1 = 0$. We easily deduce that all odd coefficients σ_{2p+1}^1 vanish (recall that the even coefficients σ_{2m}^0 vanish), and the even coefficients σ_{2p}^1 , which we also write s_p^1 to be consistent with (4.9), satisfy $s_0^1 = 0$ and

$$\forall n \geq 0, \quad s_{n+1}^1 = s_n^1 - 2\mu_x \sum_{m=0}^n s_m^1 s_{n-m}^0 - \mu_y s_n^0. \quad (4.16)$$

Recalling the initial values $s_0^0 = \mu_x$ and $s_1^0 = \mu_x(1 - \mu_x^2)$, we recover for instance from (4.16) the two first values $s_1^1 = -\mu_x \mu_y$ and $s_2^1 = -\mu_x \mu_y(2 - 3\mu_x^2)$ as in the previous method.

We can follow the same lines to derive the Laurent series expansion of κ_s^2 , by starting from the relation (4.12) and plugging (observing on (4.12) that κ_s^2 tends to zero at infinity):

$$\kappa_s^2(z) = \sum_{n=1}^{\infty} \frac{\sigma_n^2}{z^n},$$

which yields $\sigma_1^2 = 0$ and the recurrence relation

$$\forall n \geq 1, \quad \sigma_{n+1}^2 - \sigma_{n-1}^2 + 2\mu_x \sum_{m=0}^n \sigma_m^2 \sigma_{n-m}^0 = -4\mu_y \sigma_n^1 - 4\mu_x \sum_{m=0}^n \sigma_m^1 \sigma_{n-m}^1.$$

Consistently with (4.14), we find that all even coefficients σ_{2p}^2 vanish, and the odd coefficients $\sigma_{2p+1}^2 = s_p^2$ satisfy $s_0^2 = 0$ as well as the recurrence relation

$$\forall n \geq 0, \quad s_{n+1}^2 = s_n^2 - 2\mu_x \sum_{m=1}^n s_m^2 s_{n-m}^0 - 4\mu_y s_{n+1}^1 - 4\mu_x \sum_{m=1}^n s_m^1 s_{n+1-m}^1.$$

For instance, we can recover the first value $s_1^2 = s_0^2 - 4\mu_y s_1^2 = 4\mu_x \mu_y^2$ as given by the previous method.

Independently of the method we choose to compute the coefficients in the Laurent series expansion of κ_s^1 and κ_s^2 , we can implement them numerically up to any prescribed final time index N_f . We now go back to (4.6). Once we have performed the inverse Fourier \mathcal{Z} -transform, the approximate DTBC (4.6) respectively read:

$$u_{0,k}^{n+2} = - \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_{1,k}^{n+1-2m}, \quad (4.17a)$$

$$u_{0,k}^{n+2} = - \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_{1,k}^{n+1-2m} - \sum_{1 \leq m \leq (n+2)/2} s_m^1 (u_{1,k+1}^{n+2-2m} - u_{1,k-1}^{n+2-2m}), \quad (4.17b)$$

$$\begin{aligned} u_{0,k}^{n+2} = & - \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_{1,k}^{n+1-2m} - \sum_{1 \leq m \leq (n+2)/2} s_m^1 (u_{1,k+1}^{n+2-2m} - u_{1,k-1}^{n+2-2m}) \\ & - \sum_{1 \leq m \leq (n+1)/2} s_m^2 (u_{1,k+1}^{n+1-2m} - 2u_{1,k}^{n+1-2m} + u_{1,k-1}^{n+1-2m}), \end{aligned} \quad (4.17c)$$

depending on the tangential accuracy we aim at achieving on the boundary.

Of course, the asymptotic expansion that we have derived for $\kappa_s(z, \theta)$ also provides the approximate DTBC at the right boundary $j = J + 1$:

$$u_{J+1,k}^{n+2} = \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_{J,k}^{n+1-2m}, \quad (4.18a)$$

$$u_{J+1,k}^{n+2} = \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_{J,k}^{n+1-2m} + \sum_{1 \leq m \leq (n+2)/2} s_m^1 (u_{J,k+1}^{n+2-2m} - u_{J,k-1}^{n+2-2m}), \quad (4.18b)$$

$$\begin{aligned} u_{J+1,k}^{n+2} = & \sum_{0 \leq m \leq (n+1)/2} s_m^0 u_{J,k}^{n+1-2m} + \sum_{1 \leq m \leq (n+2)/2} s_m^1 (u_{J,k+1}^{n+2-2m} - u_{J,k-1}^{n+2-2m}) \\ & + \sum_{1 \leq m \leq (n+1)/2} s_m^2 (u_{J,k+1}^{n+1-2m} - 2u_{J,k}^{n+1-2m} + u_{J,k-1}^{n+1-2m}). \end{aligned} \quad (4.18c)$$

Let us observe that the boundary conditions (4.17) and (4.18) are nonlocal in time but they are local in space (unlike the original exact DTBC). The boundary conditions (4.17) and (4.18) only involve a three point stencil, as depicted on Figure 8 (left picture) on the example of the point located closest to the upper right corner. We could have pushed the Taylor expansion with respect to θ further by including for instance the following term θ^3 . The problem is that there is no linear combination of 1 , $\cos \theta$ and $\sin \theta$ that behaves like θ^3 at the origin. In other words, if one writes θ^3 as a trigonometric polynomial $p(\theta)$ up to an $O(\theta^4)$ term, then the trigonometric polynomial p has degree at least 2, which means that the associated finite difference operator in k will involve at least 4 points (and more likely five if one wishes to keep some symmetry). Hence this extension causes some trouble when getting close to the four corners of the rectangle because the very last boundary points close to the corner would require a specific treatment. We therefore do not pursue this higher order extension for the time being and leave it to future investigations.

At this stage, we have made the numerical boundary conditions on the “left” and “right” boundaries, resp. $\{j = 0\}$ and $\{j = J + 1\}$, explicit. The analysis can be carried out almost word for word for the “bottom” and

“top” boundaries, namely $\{k = 0\}$ and $\{k = K + 1\}$. The only thing is to observe that the indices j, k in (2.2) can be exchanged by simply switching the roles of μ_x and μ_y . From now on, we let $(t_n^0)_{n \geq 0}$, $(t_n^1)_{n \geq 0}$, $(t_n^2)_{n \geq 0}$ denote the sequences obtained by the same procedure as $(s_n^0)_{n \geq 0}$, $(s_n^1)_{n \geq 0}$, $(s_n^2)_{n \geq 0}$, but switching the roles of μ_x and μ_y . For instance, the sequence $(t_n^0)_{n \geq 0}$ is defined by $t_0^0 = \mu_y$, $t_1^0 := \mu_y(1 - \mu_y^2)$, and

$$\forall n \geq 2, \quad t_n^0 = \frac{2n-1}{n+1}(1 - 2\mu_y^2)t_{n-1}^0 - \frac{n-2}{n+1}t_{n-2}^0.$$

Then the approximate DTBC that we consider on the bottom and top boundaries read

$$u_{j,0}^{n+2} = - \sum_{0 \leq m \leq (n+1)/2} t_m^0 u_{j,1}^{n+1-2m}, \quad (4.19a)$$

$$u_{j,0}^{n+2} = - \sum_{0 \leq m \leq (n+1)/2} t_m^0 u_{j,1}^{n+1-2m} - \sum_{1 \leq m \leq (n+2)/2} t_m^1 (u_{j+1,1}^{n+2-2m} - u_{j-1,1}^{n+2-2m}), \quad (4.19b)$$

$$\begin{aligned} u_{j,0}^{n+2} = & - \sum_{0 \leq m \leq (n+1)/2} t_m^0 u_{j,1}^{n+1-2m} - \sum_{1 \leq m \leq (n+2)/2} t_m^1 (u_{j+1,1}^{n+2-2m} - u_{j-1,1}^{n+2-2m}) \\ & - \sum_{1 \leq m \leq (n+1)/2} t_m^2 (u_{j+1,1}^{n+1-2m} - 2u_{j,1}^{n+1-2m} + u_{j-1,1}^{n+1-2m}), \end{aligned} \quad (4.19c)$$

and

$$u_{j,K+1}^{n+2} = \sum_{0 \leq m \leq (n+1)/2} t_m^0 u_{j,K}^{n+1-2m}, \quad (4.20a)$$

$$u_{j,K+1}^{n+2} = \sum_{0 \leq m \leq (n+1)/2} t_m^0 u_{j,K}^{n+1-2m} + \sum_{1 \leq m \leq (n+2)/2} t_m^1 (u_{j+1,K}^{n+2-2m} - u_{j-1,K}^{n+2-2m}), \quad (4.20b)$$

$$\begin{aligned} u_{j,K+1}^{n+2} = & \sum_{0 \leq m \leq (n+1)/2} t_m^0 u_{j,K}^{n+1-2m} + \sum_{1 \leq m \leq (n+2)/2} t_m^1 (u_{j+1,K}^{n+2-2m} - u_{j-1,K}^{n+2-2m}) \\ & + \sum_{1 \leq m \leq (n+1)/2} t_m^2 (u_{j+1,K}^{n+1-2m} - 2u_{j,K}^{n+1-2m} + u_{j-1,K}^{n+1-2m}). \end{aligned} \quad (4.20c)$$

Before reporting on several numerical simulations using the boundary conditions (4.17)–(4.20), we first study whether each subcase in these boundary conditions, for instance (4.17a)–(4.17c) satisfy some basic stability requirements. This stability analysis for *half-space problems* is carried out in the following paragraph.

4.2. Stability analysis

The stability analysis of initial boundary value problems for hyperbolic systems in regions with corners is a delicate matter, see for instance [9, 27, 28, 30] and references therein for a partial account of the theory (of which much, if not all, remains to be done). The discrete counterpart is not easier and has been left (almost) undone so far. At a theoretical level, we do not claim to study the interaction of either one of the numerical boundary conditions in (4.20) with either of the conditions in (4.18) at the upper right corner of the rectangle. Such an analytical study seems to be out of reach at the present time. However we shall report below on some interesting numerical observations which call for a lot of care in the coupling of numerical strategies on each side of the rectangle. Namely, we shall show on one example that coupling two specific numerical boundary conditions on the top and right boundaries, namely (4.20c) and (4.18c) yields strong instabilities (even though the two separate half-space problems seem to be stable, as our partial proof below indicates). This phenomenon is well-known in the PDE context, see an elementary example in [26] or more elaborate examples in [9, 30].

We now examine a more simple problem which consists in studying the stability of the half-space problem where the leap-frog scheme (2.2) holds in the half space $\{(j, k) \in \mathbb{Z}^2, j \geq 1\}$ and either of the numerical

boundary conditions (4.17a)–(4.17c) is imposed on the boundary $\{j = 0\}$. Following what we have done in the one-dimensional case, the stability analysis is carried out here by means of the so-called normal mode analysis. Extending slightly what we have done in the one-dimensional case, the normal mode analysis amounts here to determining the solutions to the leap-frog scheme (2.2) which are of the form:

$$u_{j,k}^n = z^n \exp(i k \theta) v_j,$$

with $|z| > 1$, $\theta \in \mathbb{R}$ and $(v_j)_{j \geq 0} \in \ell^2$, that also satisfy the homogeneous numerical boundary conditions, be they (4.17a), (4.17b) or (4.17c). The arguments are quite similar in each of the three cases (except that we shall not be able to carry them out completely in the last case). We deal below with (4.17a)–(4.17c) in increasing order of complexity.

The “zero order” numerical boundary condition (4.17a). First of all, we plug the expression $u_{j,k}^n = z^n \exp(i k \theta) v_j$ in (2.2) and find that $(v_j)_{j \geq 0}$ must satisfy the recurrence relation

$$\forall j \geq 1, \quad (z^2 - 1) v_j + \mu_x z (v_{j+1} - v_{j-1}) + 2i \sin \theta z v_j = 0.$$

(Compare with (4.1).) Since $(v_j)_{j \geq 0}$ should decay to zero at infinity, we get

$$\forall j \geq 0, \quad v_j = v_0 \kappa_s(z, \theta)^j, \quad v_0 \in \mathbb{C},$$

where we recall that $\kappa_s(z, \theta)$ is the only root of modulus < 1 to (4.2) for $|z| > 1$ and $\theta \in \mathbb{R}$ (and $1/\kappa_s(z, \theta) = -\kappa_u(z, \theta)$ where $\kappa_u(z, \theta)$ is the only root of modulus > 1 to the same equation).

Going back to (4.17a), or rather to its Fourier- \mathcal{Z} -transform counterpart (4.5a), we now need to determine whether there holds

$$1 = -\kappa_s^0(z) \kappa_s(z, \theta),$$

or equivalently

$$\kappa_u(z, \theta) = \kappa_s^0(z) \quad (= \kappa_s(z, 0))$$

for some pair (z, θ) with $|z| > 1$ and $\theta \in \mathbb{R}$. The latter relation can never hold because $|\kappa_u(z, \theta)| > 1$ and $|\kappa_s^0(z)| < 1$ so the Godunov–Ryabenkii condition holds as for the one-dimensional problem which we have considered in the previous section. Here again, there does not exist any unstable eigenvalue for the half-space problem $\{j \geq 1\}$ with the numerical boundary condition (4.17a). The same argument would apply for any of the other half space problems in $\{j \leq J\}$, $\{k \geq 1\}$ or $\{k \leq K\}$ with their corresponding “zero order” numerical boundary condition.

The “first order” numerical boundary condition (4.18b). We can use part of the previous analysis, namely the determination of the “stable” solutions to (2.2) of the prescribed normal mode form. However we now consider the numerical boundary condition (4.18b) or rather its Fourier- \mathcal{Z} -transform counterpart (4.5b), so we need to determine whether there holds

$$1 = -\kappa_s^0(z) \kappa_s(z, \theta) - 2i \sin \theta \kappa_s^1(z) \kappa_s(z, \theta),$$

or equivalently

$$\kappa_u(z, \theta) = \kappa_s^0(z) + 2i \sin \theta \kappa_s^1(z), \tag{4.21}$$

for some pair (z, θ) with $|z| > 1$ and $\theta \in \mathbb{R}$. Let us recall that κ_s^0 is a root to (3.6) and that κ_s^1 is given by the relation (4.7).

Let us first consider the case $\theta = 0 (\pi)$, that is, $\sin \theta = 0$. Then $\kappa_u(z, \theta) = \kappa_u^0(z)$ and (4.21) can not hold since the left hand side has modulus > 1 while the right hand side has modulus < 1 . We therefore assume from now on $\sin \theta \neq 0$. Let us assume for a moment that the equality (4.21) holds for some pair (z, θ) with $|z| > 1$ and

$\sin \theta \neq 0$. Then the right hand side of (4.21) must be a solution to the second degree polynomial equation (4.2). Plugging the right hand side of (4.21) in (4.2) and expanding, we eventually get⁶:

$$(2i \sin \theta)^2 z \kappa_s^1(z) (\mu_y + \mu_x \kappa_s^1(z)) = 0.$$

Simplifying by $\sin \theta$, we get either $\kappa_s^1(z) = 0$ or $\mu_y + \mu_x \kappa_s^1(z) = 0$. Let us exclude the first of these two options. For $|z| > 1$, the stable root $\kappa_s^0(z)$ to (3.6) does not vanish. Hence we see on the relation (4.7) that $\kappa_s^1(z)$ does not vanish either. This means that if (4.21) holds for some pair (z, θ) , then there necessarily holds $\mu_y + \mu_x \kappa_s^1(z) = 0$. We use again (4.7) to compute:

$$\begin{aligned} \mu_y + \mu_x \kappa_s^1(z) &= \mu_y \frac{z^2 - 1 + \mu_x z \kappa_s^0(z)}{z^2 - 1 + 2\mu_x z \kappa_s^0(z)} \\ &= -\frac{\mu_x \mu_y \kappa_u^0(z)}{z^2 - 1 + 2\mu_x z \kappa_s^0(z)}, \end{aligned}$$

and the latter quantity can vanish only if $\mu_y = 0$ (recall that we have assumed $\mu_x \neq 0$). At this point of the analysis, we have shown that the only possibility for (4.21) to hold is to have $\mu_y = 0$, that is $c_y = 0$. However, in that case, we have $\kappa_u(z, \theta) = \kappa_u^0(z)$ for all θ , see (4.2), and $\kappa_s^1(z) = 0$ (see (4.7)). Hence (4.21) can never hold because for $\mu_y = 0$, the right hand side has modulus < 1 while the left hand side has modulus > 1 . In other words, the Godunov-Ryabenkii condition holds and there does not exist any unstable eigenvalue for the half-space problem $\{j \geq 1\}$ with the numerical boundary condition (4.17b). Of course, the same arguments would apply for any of the other half space problems.

The “second order” numerical boundary condition (4.18c). We can use the same approach as above. Verifying the Godunov–Ryabenkii condition amounts to determining whether there exists a pair (z, θ) with $|z| > 1$ and $\theta \in \mathbb{R}$ such that:

$$\kappa_u(z, \theta) = \kappa_s^0(z) + 2i \sin \theta \kappa_s^1(z) - 4 \sin^2 \frac{\theta}{2} \kappa_s^2(z). \quad (4.22)$$

We verify again that (4.22) can not hold if $\theta = 0(\pi)$ or if $\mu_y = 0$. Hence we assume from now on $\sin \theta \neq 0$ and $\mu_y \neq 0$. Let us assume that the relation (4.22) holds, meaning that not only the right hand side of (4.22) is a root to (4.2) but also that it is the only root of modulus > 1 . We first plug the right hand side of (4.22) in (4.2) and collect the terms in increasing powers of $\sin \theta/2$ (we use again the equation (3.6) satisfied by κ_s^0 , the relation (4.7) satisfied by κ_s^1 and we also use the relation (4.12) satisfied by κ_s^2). After simplifying (quite a bit...), we eventually get the polynomial equation

$$\begin{aligned} i \sin \frac{\theta}{2} ((z^2 - 1)^4 + 8\mu_x^2 z^2 (z^2 - 1)^2 + 16\mu_x^2 (\mu_x^2 - \mu_y^2) z^4) \\ - 4 \cos \frac{\theta}{2} \mu_y z (z^2 - 1) ((z^2 - 1)^2 + 4\mu_x^2 z^2) = 0. \end{aligned} \quad (4.23)$$

Since $\sin \theta$ is nonzero, then $\sin \theta/2$ is also nonzero, and (4.23) is consequently an eight degree polynomial equation in z . Changing z into iz , (4.23) becomes a polynomial equation with real coefficients (after simplifying by i). Moreover, the roots of (4.23) are nonzero and are invariant by the transformation $z \rightarrow -1/z$. Let us also recall that we are interested in determining the roots of (4.23) that satisfy $|z| > 1$.

We have not been able to obtain a complete proof of the facts we claim below, hence we do not have a complete proof for the verification of the Godunov–Ryabenkii condition, but repeated numerical experiments indicate the following facts:

⁶Here we use the equation (3.6) satisfied by κ_s^0 as well as the relation (4.7) satisfied by κ_s^1 .

- For $\theta \in (0, \theta_{\max})$, where the angle θ_{\max} is defined by

$$\tan \frac{\theta_{\max}}{2} = \frac{2\mu_y(1-\mu_x^2)}{(1-\mu_x^2)^2 - \mu_x^2\mu_y^2},$$

(4.23) has a unique root z_θ verifying $|z_\theta| > 1$. This root is a purely imaginary number (one other root of (4.23) is $-1/z_\theta$ since the roots of (4.23) are invariant under the transformation $z \rightarrow -1/z$, and the six other roots have modulus one).

- For $\theta \in [\theta_{\max}, \pi)$, all roots to (4.23) have modulus one (the value θ_{\max} is obtained by considering the case where $-i$ is a (double) root to (4.23)).

For (4.23) to have a root z that satisfies $|z| > 1$, we must necessarily have $\theta \in (0, \theta_{\max})$, and in that case the remaining question is to determine whether the quantity

$$\kappa_s^0(z_\theta) + 2i \sin \theta \kappa_s^1(z_\theta) - 4 \sin^2 \frac{\theta}{2} \kappa_s^2(z_\theta)$$

is either the stable or the unstable root to (4.2). On the numerical experiments we have conducted, we have verified the property:

$$\left| \kappa_s^0(z_\theta) + 2i \sin \theta \kappa_s^1(z_\theta) - 4 \sin^2 \frac{\theta}{2} \kappa_s^2(z_\theta) \right| < 1, \quad (4.24)$$

for $\theta \in (0, \theta_{\max})$, meaning that the right hand side of (4.22) may coincide with a root (4.2) for some specific values of (z, θ) but in that case it coincides with the stable root $\kappa_s(z, \theta)$ and not with the unstable root $\kappa_u(z, \theta)$ (which would complete the verification of the Godunov–Ryabenkii condition). The inequality (4.24) can be shown rigorously in the regime where θ is small, for in that case we have $z_\theta \rightarrow \infty$, see (4.22), and all functions $\kappa_s^0, \kappa_s^1, \kappa_s^2$ tend to zero at infinity, so the modulus of

$$\kappa_s^0(z_\theta) + 2i \sin \theta \kappa_s^1(z_\theta) - 4 \sin^2 \frac{\theta}{2} \kappa_s^2(z_\theta)$$

tends to zero as θ tends to zero. We can also verify (4.24) in the limit regime $\theta \rightarrow \theta_{\max}$ for in that case we get $z_\theta \rightarrow -i$ and we compute

$$\kappa_s^0(z_\theta) \rightarrow \frac{i(1-\sqrt{1-\mu_x^2})}{\mu_x}, \quad \kappa_s^1(z_\theta) \rightarrow \frac{\mu_y(1-\sqrt{1-\mu_x^2})}{2\mu_x\sqrt{1-\mu_x^2}}, \quad \kappa_s^2(z_\theta) \rightarrow -\frac{i\mu_y^2\mu_x}{2(1-\mu_x^2)^{3/2}}.$$

We can then verify numerically that the inequality

$$\left| \frac{i(1-\sqrt{1-\mu_x^2})}{\mu_x} + 2i \sin \theta_{\max} \frac{\mu_y(1-\sqrt{1-\mu_x^2})}{2\mu_x\sqrt{1-\mu_x^2}} + 4 \sin^2 \frac{\theta_{\max}}{2} \frac{i\mu_y^2\mu_x}{2(1-\mu_x^2)^{3/2}} \right| < 1$$

holds for all positive values of μ_x, μ_y such that $\mu_x + \mu_y < 1$ (here we use the classical formula $\sin \theta_{\max} = 2t/(1+t^2)$ and $\sin^2 \theta_{\max}/2 = t^2/(1+t^2)$ with $t := \tan \theta_{\max}/2$). Hence it does seem that (4.22) can not hold for some pair (z, θ) , which means that the Godunov–Ryabenkii condition is verified (though a complete analytical proof of this fact still remains open). We postpone this rigorous justification to a future work.

4.3. Numerical experiments on a rectangle

We go back to our original motivation which is the simulation of the leap-frog scheme (2.2) on the rectangle $\{1 \leq j \leq J, 1 \leq k \leq K\}$ with some approximate DTBC on each side. In the numerical simulations reported below, the computations were run on the rectangle $(x, y) \in (-3, 3) \times (-2, 2)$ and on the time interval $t \in [0, 8]$.

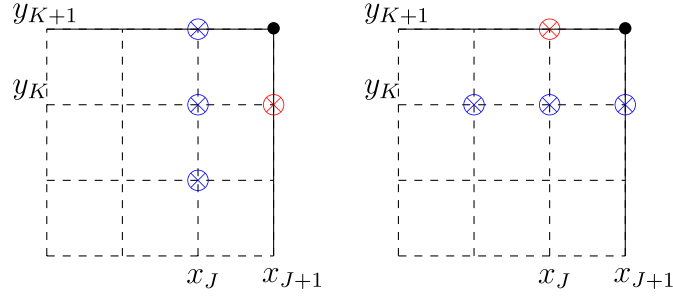


FIGURE 8. The stencil of the numerical boundary conditions (4.18b) and (4.20b) near the upper right corner. *Left*: the points (in blue crosses) involved in the computation of $u_{J+1,K}^{n+2}$ (in red). *Right*: the points (in blue crosses) involved in the computation of $u_{J,K+1}^{n+2}$ (in red).

The discretization parameters are $J = 300$ and $K = 200$ (from which one can deduce the values of the space steps $\delta x, \delta y$). The time step δt is chosen in order to satisfy the CFL condition

$$\mu_x + \mu_y = \frac{1}{2}.$$

(Recall the ℓ^2 -stability condition (2.3) for (2.2) on the whole space.) The initial datum that we consider for the transport equation (2.1) is a Gaussian, namely $u_0(x, y) = \exp(-5(x^2 + y^2))$. The discrete initial condition $(u_{j,k}^0)$ is defined by setting

$$\forall j = 0, \dots, J+1, \quad \forall k = 0, \dots, K+1, \quad u_{j,k}^0 := u_0(x_j, y_k).$$

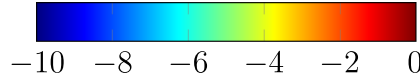
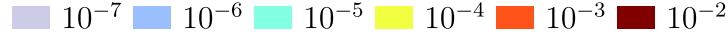
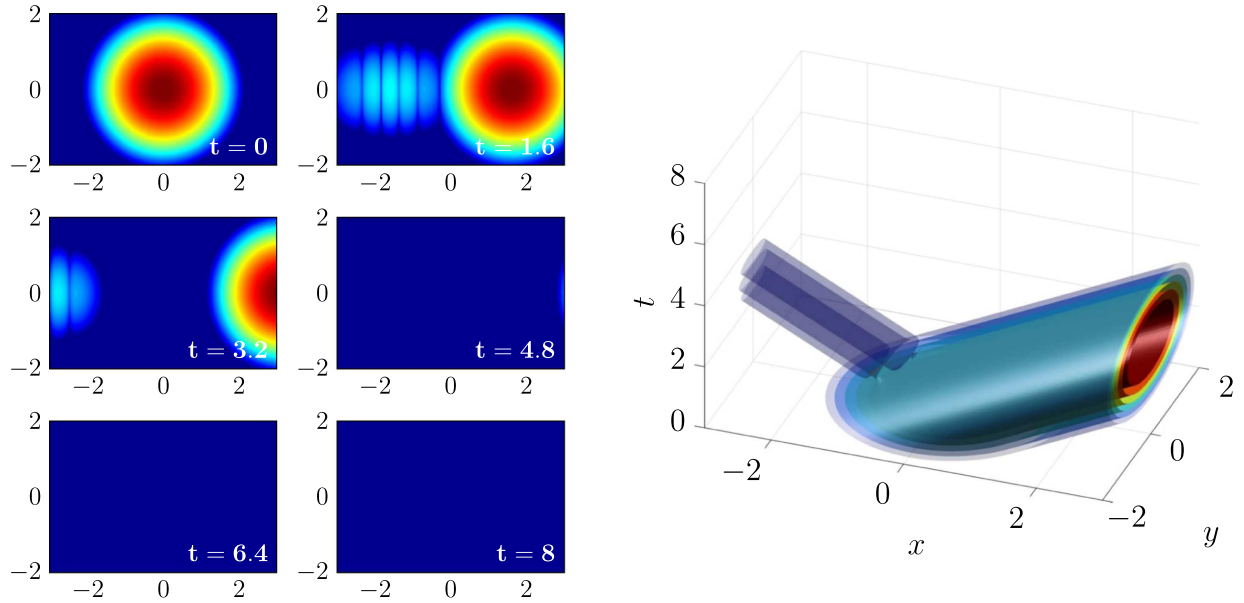
The first time step value $(u_{j,k}^1)$ is defined by imposing the second order two-dimensional Lax–Wendroff scheme [25], namely:

$$\begin{aligned} \forall j = 1, \dots, J, \quad \forall k = 1, \dots, K, \\ u_{j,k}^1 := & u_{j,k}^0 - \frac{\mu_x}{2} (u_{j+1,k}^0 - u_{j-1,k}^0) - \frac{\mu_y}{2} (u_{j,k+1}^0 - u_{j,k-1}^0) \\ & + \frac{\mu_x^2}{2} (u_{j+1,k}^0 - 2u_{j,k}^0 + u_{j-1,k}^0) + \frac{\mu_y^2}{2} (u_{j,k+1}^0 - 2u_{j,k}^0 + u_{j,k-1}^0) \\ & + \frac{\mu_x \mu_y}{4} (u_{j+1,k+1}^0 - u_{j+1,k-1}^0 - u_{j-1,k+1}^0 + u_{j-1,k-1}^0). \end{aligned}$$

The boundary values for u^1 (all other relevant values of j, k) are set equal to zero for simplicity.

In the numerical simulations below, we try various coupling strategies of numerical boundary conditions on the “right” and “top” boundaries $\{j = J+1\}$ and $\{k = K+1\}$. We first make it clear on Figure 8 that the two sides of the upper right corner are coupled through the boundary conditions if one chooses for instance to impose (4.18b) on $j = J+1$ and (4.20b) on $k = K+1$. The coupling is weak though since the boundary conditions (4.18b) and (4.20b) are explicit so the trace values $u_{J+1,K}^{n+2}$ and $u_{J,K+1}^{n+2}$ are coupled only through the preceding time steps.

In order to highlight the wave reflections on each side of the rectangle, we show below on some two and three dimensional figures the logarithm of the (absolute value of the) solution. The logarithm of the solution is represented on two-dimensional figures at various time steps. The full time evolution is represented on a three dimensional figure. Logarithmic representations can be far more enlightening if one wishes to reveal the magnitude of the reflected waves. The color legend for all 2D logarithmic representations reported below is given in Figure 9 and the color legend for all 3D logarithmic representations reported below is given in Figure 10.

FIGURE 9. Legend of the 2D representation of the evolution of $\log_{10} |u_{j,k}^n|$.FIGURE 10. Legend of the 3D representation of the evolution of $\log_{10} |u_{j,k}^n|$.FIGURE 11. Evolution of $\log_{10} |u_{j,k}^n|$ (left: six successive times, right: full time evolution) for $\mathbf{c} = (1, 0)$ and DTBC of order 0.

We first report on some cases where the initial Gaussian condition is transported towards one edge of the rectangle and does not give rise (in the considered time interval) to multiple wave reflections. The first case is depicted in Figure 11 where the chosen velocity is $\mathbf{c} = (1, 0)$ (and therefore all boundary conditions (4.18a)–(4.18c) coincide since $\mu_y = 0$). We call them DTBC of order 0 since (4.18a) amounts to not retaining any tangential dependence. The results are of course as accurate as in one space dimension (because the scheme (2.2) is one-dimensional in that case and the tangential index k only enters as a parameter). We now turn on the tangential dependence by considering the case $\mathbf{c} = (1, 0.1)$. Both μ_x and μ_y are positive then. On the right and left faces of the rectangle, we either implement:

- The approximate DTBC (4.18a) and (4.17a). These are called DTBC of order 0; accordingly our choice of approximate DTBC for the top and bottom faces of the rectangle are (4.20a) and (4.19a). The results are shown in Figure 12. The reflected wave at the (right) outflow boundary has magnitude 10^{-3} .
- The approximate DTBC (4.18b) and (4.17b). These are called DTBC of order 1; accordingly our choice of approximate DTBC for the top and bottom faces of the rectangle are (4.20b) and (4.19b). The results are shown in Figure 13. The reflected wave at the (right) outflow boundary now has magnitude 10^{-5} , which shows significant improvement.

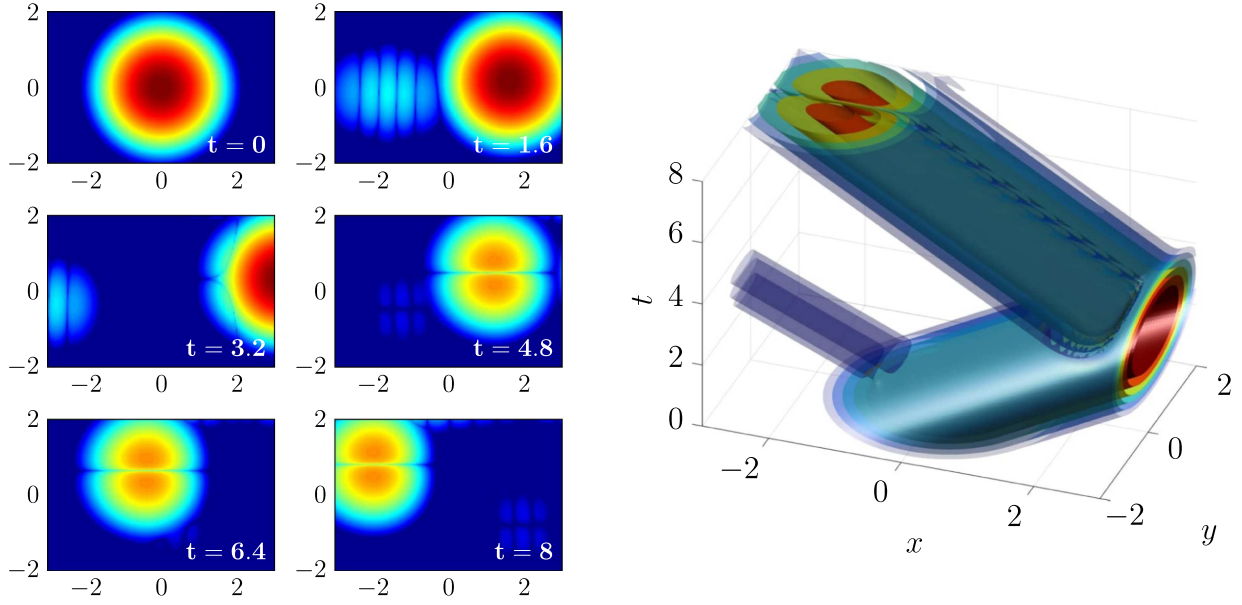


FIGURE 12. Evolution of $\log_{10} |u^n_{j,k}|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 0.1)$ and DTBC of order 0.

- The approximate DTBC (4.18c) and (4.17c). These are called DTBC of order 2; there is a subtlety here because our choice of approximate DTBC for the top and bottom faces of the rectangle are still (4.20b) and (4.19b), and not (4.20c) and (4.19c) as one might expect. This is further explained below. The results are shown in Figure 14. The reflected wave at the (right) outflow boundary now has magnitude 10^{-8} and is therefore not visible. This shows again significant improvement.

The overall conclusion that can be drawn from these numerical results is that, up to stability issues (which can be challenging), adding more tangential terms in the Taylor expansion (4.5) seems to lead to more accurate results, which was to be expected of course since we consider smooth solutions. The numerical simulations quantify the gain of adding each new term in (4.5b) and (4.5c) with respect to (4.5a).

We now report on more intricate cases that exhibit multiple wave reflections. Namely, we now choose the velocity $\mathbf{c} = (1, 0.3)$. With this velocity, the initial Gaussian function should leave the rectangle through the right edge of the boundary. Because the numerical boundary conditions (4.18) are not exactly transparent, wave reflection occurs. For the leap-frog scheme, the reflection takes place according to the billiard law. A (small amplitude highly oscillating) wave then moves towards the top face of the rectangle and a second wave reflection occurs. For this second reflection, the small tangential frequency assumption that motivated the expansion (4.5) is not any longer valid. Hence it is likely that there will be little absorption through the top boundary. The numerical simulations confirm that expectation. We follow the same plan as for the case $\mathbf{c} = (1, 0.1)$ and report on some simulations with increasing tangential approximation. We thus implement:

- The zero order DTBC (4.18a), (4.17a), (4.20a), and (4.19a) on the four sides of the rectangle. This case is shown in Figure 15; after the first reflection on the right face, the wave has magnitude 10^{-3} . The magnitude remains of order 10^{-3} after the second reflection on the top boundary.
- The first order DTBC (4.18b), (4.17b), (4.20b), and (4.19b) on the four sides of the rectangle. This case is shown in Figure 16; after the first reflection on the right face, the wave has magnitude 10^{-5} . The magnitude remains of order 10^{-5} after the second reflection on the top boundary. The overall accuracy is improved.

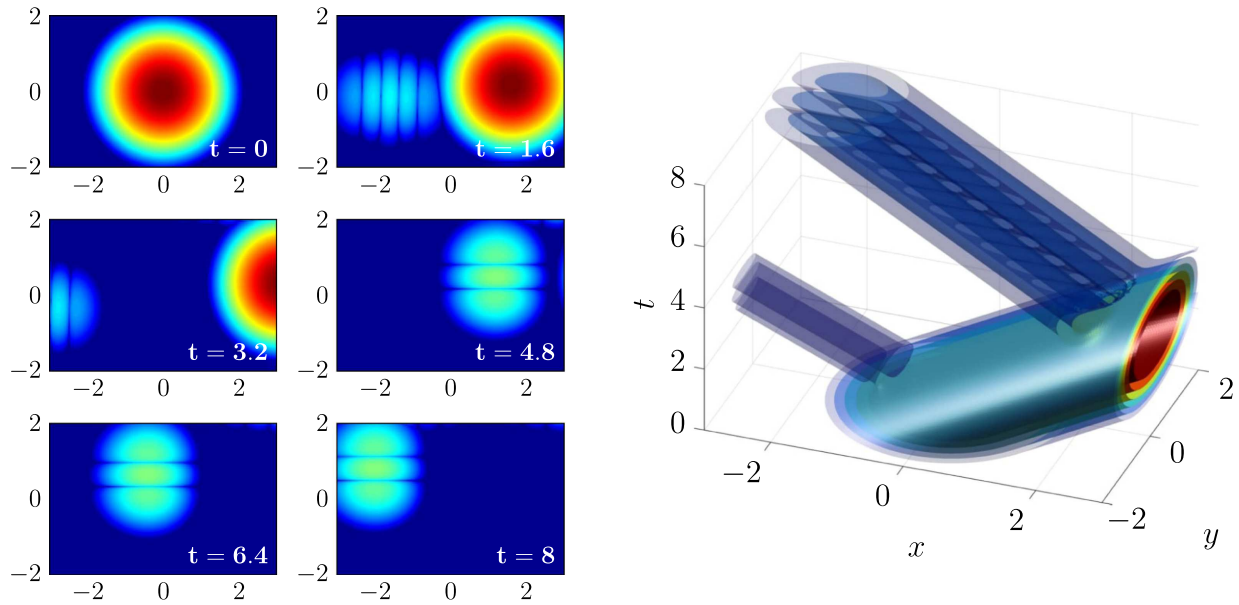


FIGURE 13. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 0.1)$ and DTBC of order 1.

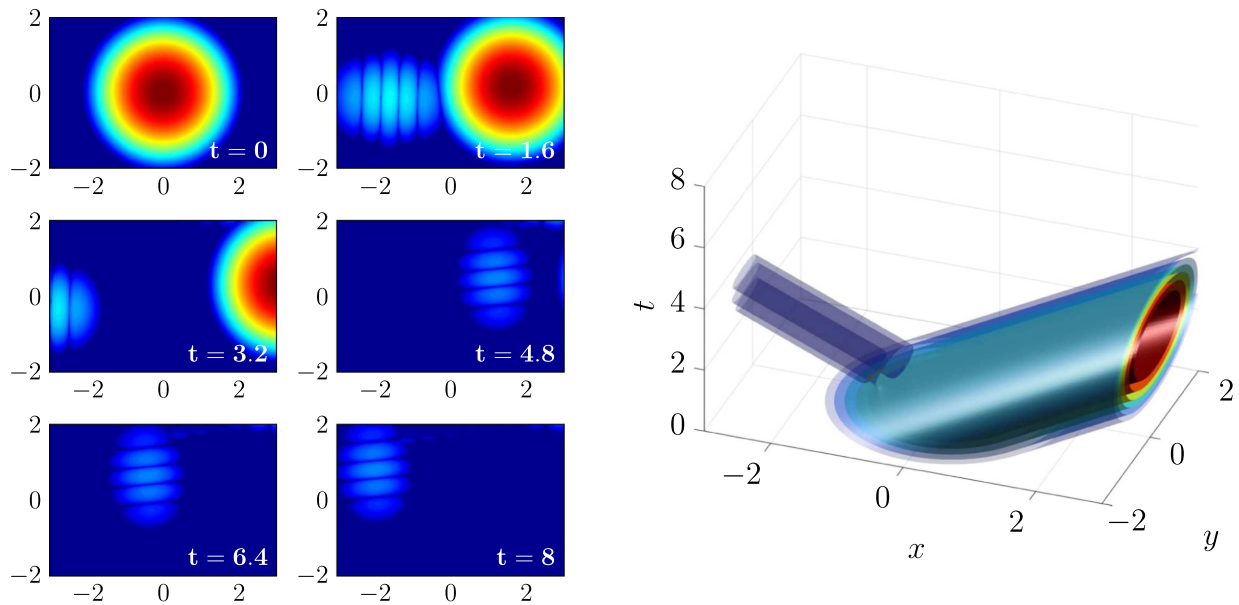


FIGURE 14. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 0.1)$ and DTBC of order 2 in the x -direction and order 1 in the y -direction.

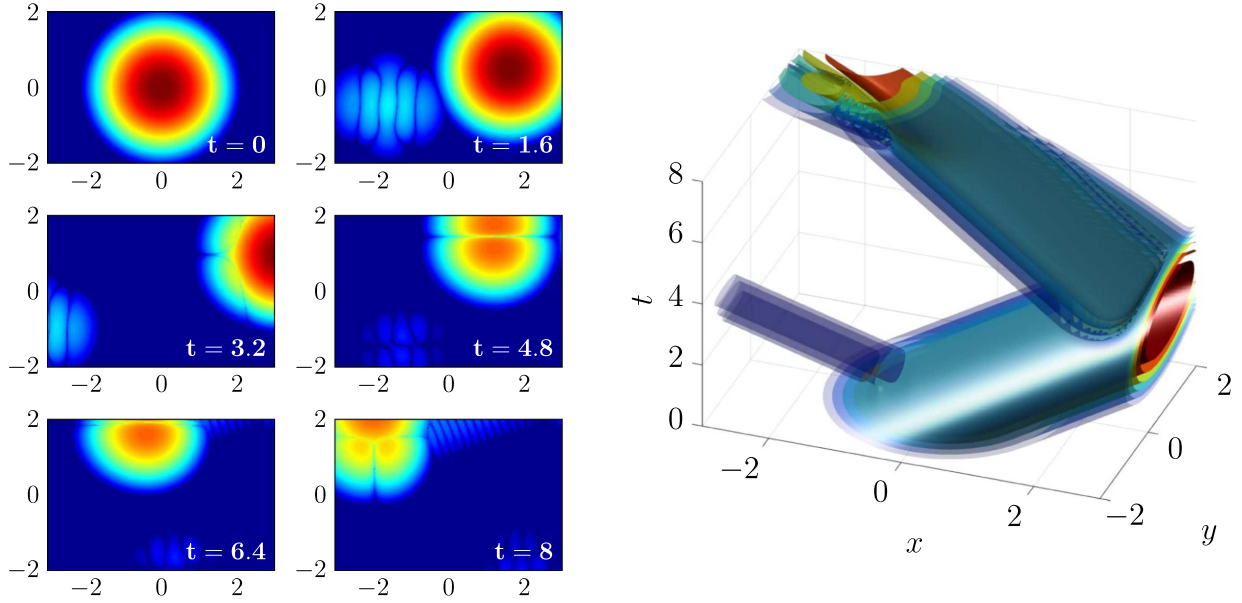


FIGURE 15. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 0.3)$ and DTBC of order 0.

- The second order DTBC (4.18c) and (4.17c) on the right and left boundaries with the first order DTBC (4.20b) and (4.19b) on the top and bottom boundaries. This case is shown in Figure 17; after the first reflection on the right face, the wave has magnitude 10^{-6} . The magnitude remains of order 10^{-6} after the second reflection on the top boundary. The overall accuracy is improved again.

Figures 18–20 represent similar simulations as Figures 15–17 with the velocity $\mathbf{c} = (1, 2/3)$. In that case, the initial condition is transported towards the upper right corner of the computational domain. There is again an improvement when passing from zero order DTBC to first order DTBC (compare the scales between Figs. 18 and 19). Figure 20 shows that coupling approximate DTBC with different tangential approximation orders on the two sides of a corner may affect the overall accuracy. Namely, for the velocity $\mathbf{c} = (1, 2/3)$, using DTBC of order 1 on the four sides of the rectangle yields better results than coupling DTBC of order 2 on two opposite sides with DTBC of order 1 on the two remaining sides.

Eventually, we show in Figure 21 the evolution of the ℓ^2 norm of $(u_{j,k}^n)$ as a function of the time t^n in the case $\mathbf{c} = (1, 0.3)$ when the second order DTBC (4.18c), (4.17c), (4.20c), and (4.19c) are implemented on the four sides of the rectangle. The computation exhibits two stages. In the first stage, the initial condition (mostly) exits through the right boundary. As long as the solution remains small on the boundary, the effect of the numerical boundary conditions is rather invisible and the ℓ^2 norm remains approximately constant. In a second stage, the trace of the solution is no longer small and an exponential instability takes place. The logarithmic scale in Figure 21 clearly shows the exponential behavior in time. The numerical solution displays a profile that is concentrated along the top and right boundaries (right of Fig. 21). It thus seems clear that the coupling between (4.18c) and (4.20c) gives rise to an unstable eigenvalue that is associated with a “surface wave” that has exponential decay with respect to the normal directions to both the right and top boundaries. We have unfortunately not been able to obtain an analytical proof of this fact, but our conclusion is that despite good stability properties of (4.18c) and (4.20c) when considered separately on each side of the rectangle, the coupling at the corner may yield strong instabilities.

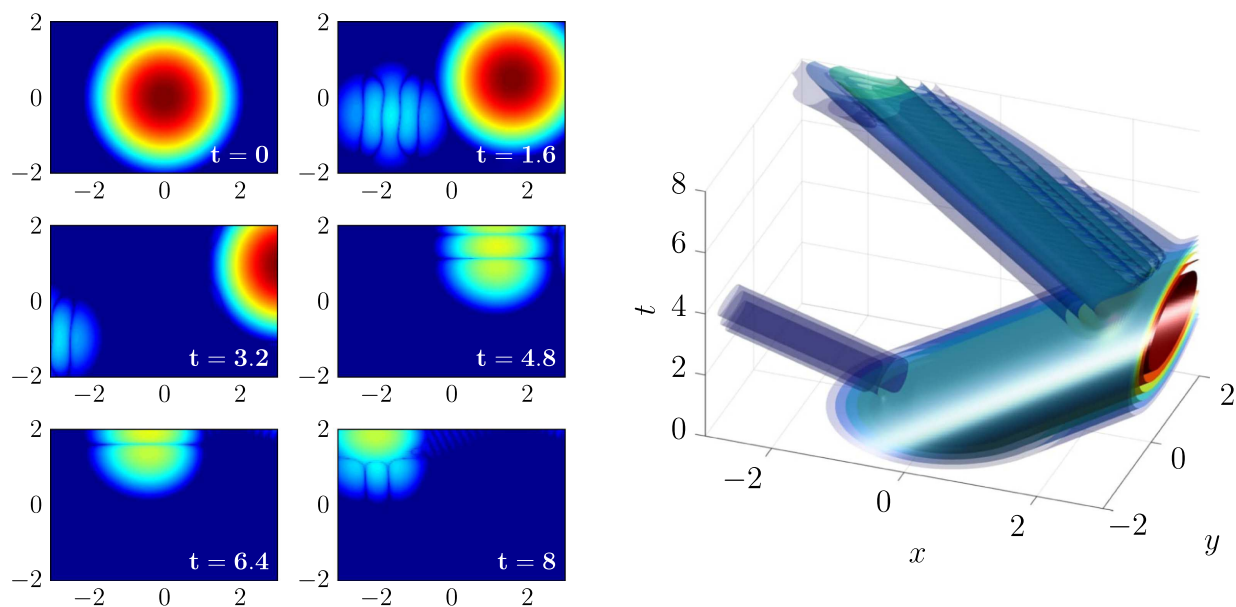


FIGURE 16. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 0.3)$ and DTBC of order 1.

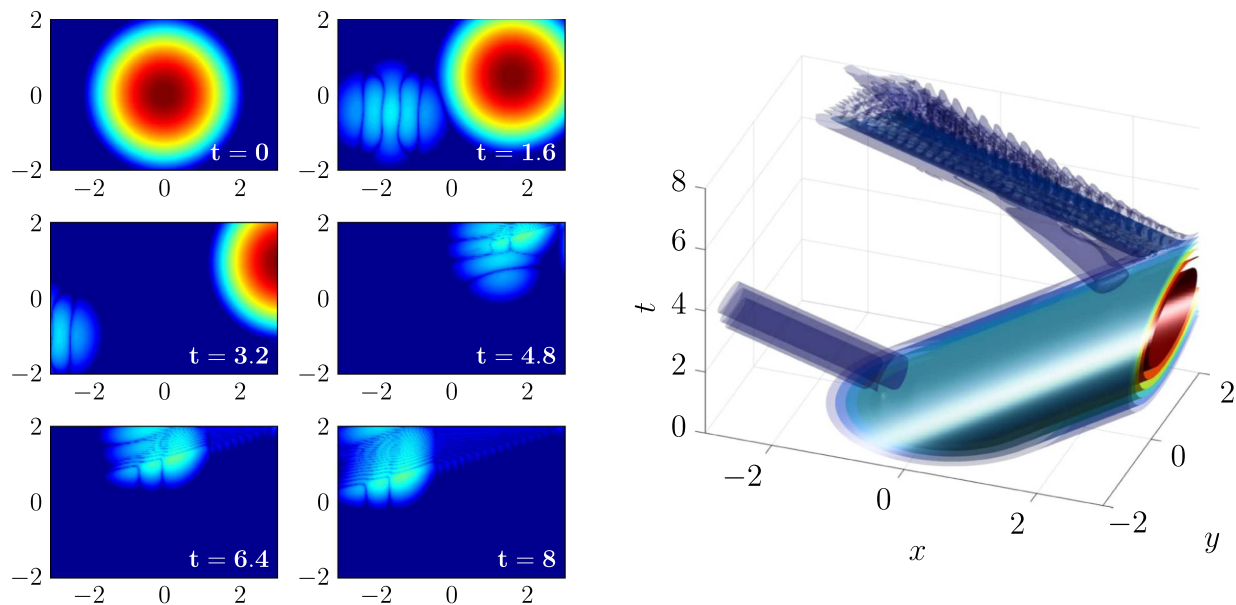


FIGURE 17. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 0.1)$ and DTBC of order 2 in the x -direction and order 1 in the y -direction.

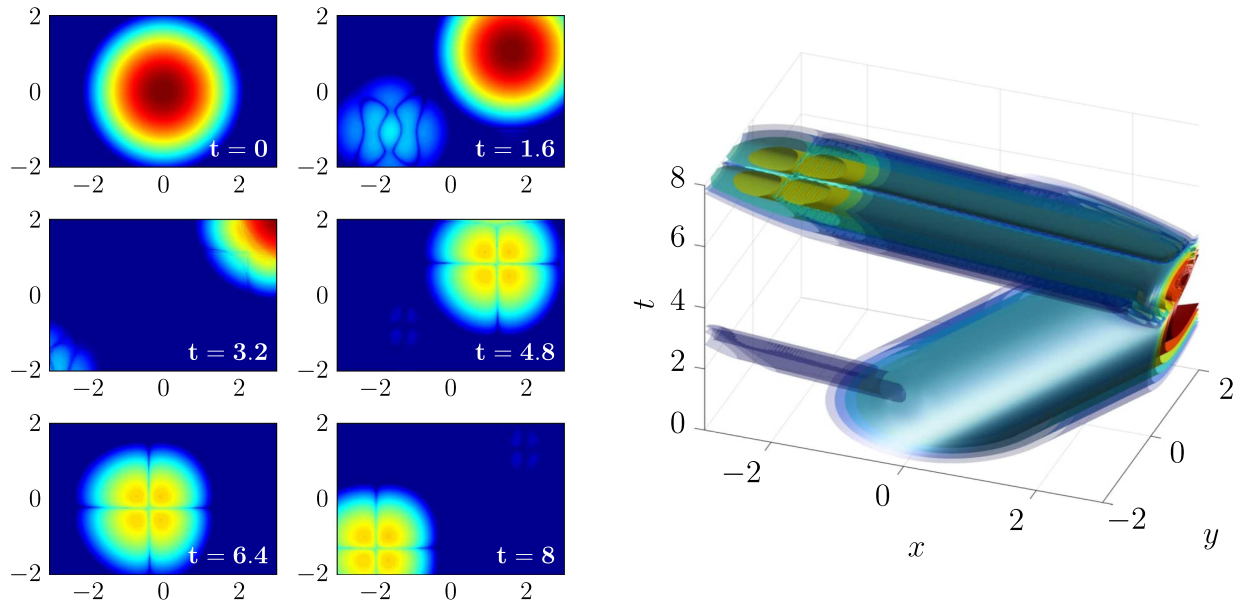


FIGURE 18. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 2/3)$ and DTBC of order 0.

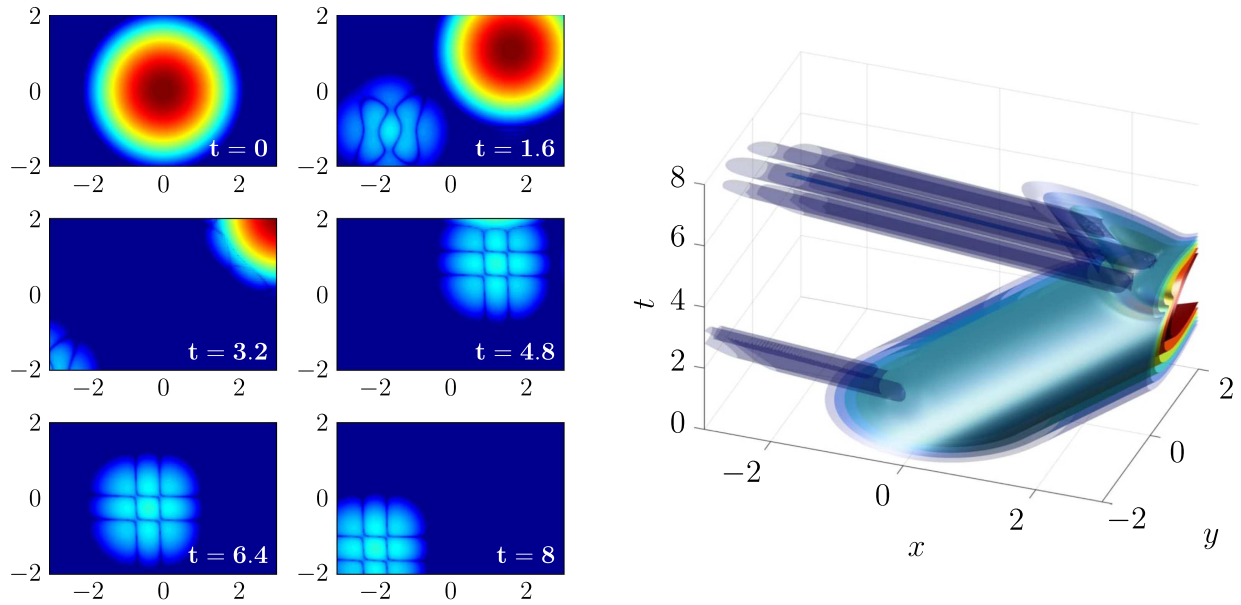


FIGURE 19. Evolution of $\log_{10} |u_{j,k}^n|$ (*left*: six successive times, *right*: full time evolution) for $\mathbf{c} = (1, 2/3)$ and DTBC of order 1.

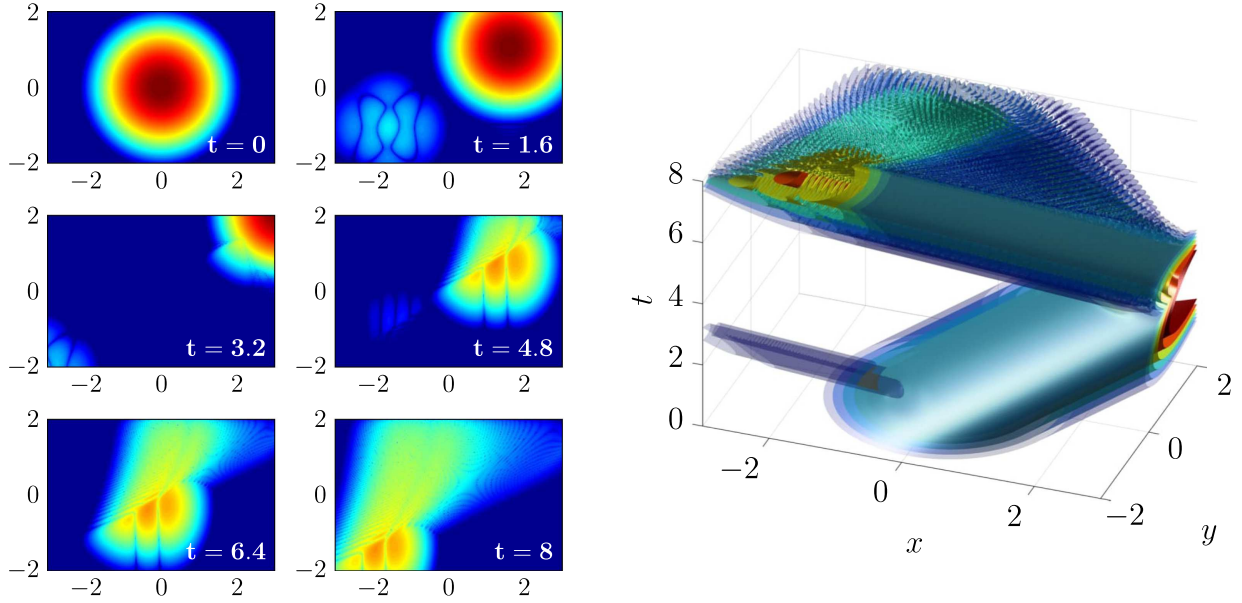


FIGURE 20. Evolution of $\log_{10} |u_{j,k}^n|$ (left: six successive times, right: full time evolution) for $\mathbf{c} = (1, 2/3)$ and DTBC of order 2 in the x -direction and order 1 in the y -direction.

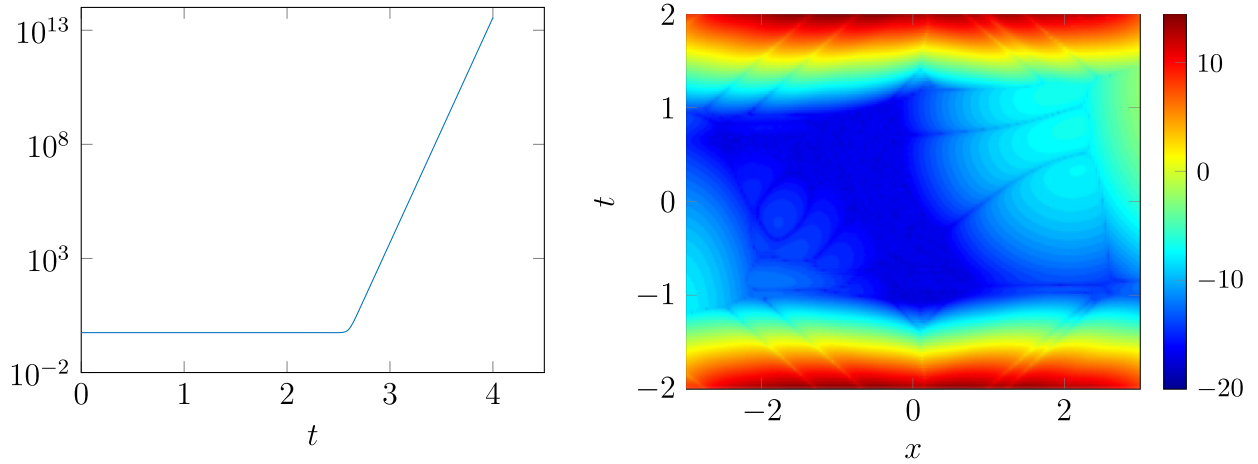


FIGURE 21. The instability for $\mathbf{c} = (1, 0.3)$ and DTBC of order 2 in both x and y directions. Left: time evolution of the logarithm of the ℓ^2 -norm of the solution. Right: the unstable profile at $t = 4$ (logarithmic scale).

At last, we have also tested the efficiency of the sum of exponential approximation compared to the standard convolution procedure in the case of DTBC of order 1 (4.18b), (4.17b), (4.20b), and (4.19b). Namely, we have compared the computational effort in that case with the sum of exponential approximation of both sequences (s_n^0) and (s_n^1) . We have chosen $(M, N) = (50, 20)$ for the degrees of the Padé approximant. The results are presented in Table 1. Though there is little gain for the sum of exponential approximation in one space dimension, there

TABLE 1. Comparison of cputime between the implementation of DTBC and the implementation of the sum of exponential approximation in two space dimensions.

	Minimal cputime	Mean cputime
DTBC	21.9850 s	22.2692 s
SumExp	9.7760 s	10.0946 s

is a factor 2 gain in terms of computational effort in two space dimensions for an overall accuracy that is comparable.

For the reader's convenience, all numerical codes are made available on the web page <http://nabuco.math.cnrs.fr>.

Acknowledgements. Research of all the authors was supported by ANR project NABUCO, ANR-17-CE40-0025.

REFERENCES

- [1] S. Abarbanel and D. Gottlieb, Stability of two-dimensional initial boundary value problems using leap-frog type schemes. *Math. Comput.* **33** (1979) 1145–1155.
- [2] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt and A. Schädle, A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations. *Commun. Comput. Phys.* **4** (2008) 729–796.
- [3] A. Arnold, M. Ehrhardt and I. Sofronov, Discrete transparent boundary conditions for the Schrödinger equation: fast calculation, approximation, and stability. *Commun. Math. Sci.* **1** (2003) 501–556.
- [4] A. Arnold, M. Ehrhardt, M. Schulte and I. Sofronov, Discrete transparent boundary conditions for the Schrödinger equation on circular domains. *Commun. Math. Sci.* **10** (2012) 889–916.
- [5] G.A. Baker, Jr., Essentials of Padé Approximants. Academic Press (1975).
- [6] G.A. Baker, Jr., Counter-examples to the Baker–Gammel–Wills conjecture and patchwork convergence. *J. Comput. Appl. Math.* **179** (2005) 1–14.
- [7] G.A. Baker, Jr. and P. Graves-Morris, Padé Approximants, 2nd edition. In: Vol. 59 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge (1996).
- [8] V.A. Baskakov and A.V. Popov, Implementation of transparent boundaries for numerical solution of the Schrödinger equation. *Wave Motion* **14** (1991) 123–128.
- [9] A. Benoit, Geometric optics expansions for hyperbolic corner problems, I: self-interaction phenomenon. *Anal. PDE* **9** (2016) 1359–1418.
- [10] S. Benzoni-Gavage and D. Serre, Multidimensional Hyperbolic Partial Differential Equations: First-Order Systems and Applications. Oxford University Press (2007).
- [11] C. Besse, P. Noble and D. Sanchez, Discrete transparent boundary conditions for the mixed KDV–BBM equation. *J. Comput. Phys.* **345** (2017) 484–509.
- [12] C. Besse, B. Mésognon-Gireau and P. Noble, Artificial boundary conditions for the linearized Benjamin–Bona–Mahony equation. *Numer. Math.* **139** (2018) 281–314.
- [13] G. Beylkin and L. Monzón, On approximation of functions by exponential sums. *Appl. Comput. Harmonic Anal.* **19** (2005) 17–48.
- [14] G. Beylkin and L. Monzón, Approximation by exponential sums revisited. *Appl. Comput. Harmonic Anal.* **28** (2010) 131–149. Special Issue on Continuous Wavelet Transform in Memory of Jean Morlet, Part I.
- [15] J.-F. Coulombel, Transparent numerical boundary conditions for evolution equations: derivation and stability analysis. *Ann. Fac. Sci. Toulouse Math.* **28** (2019) 259–327.
- [16] G. Dakin, B. Després and S. Jaouen, Inverse Lax–Wendroff boundary treatment for compressible Lagrange-remap hydrodynamics on Cartesian grids. *J. Comput. Phys.* **353** (2018) 228–257.
- [17] A. Dedner, D. Kröner, I.L. Sofronov and M. Wesenberg, Transparent boundary conditions for mhd simulations in stratified atmospheres. *J. Comput. Phys.* **171** (2001) 448–478.
- [18] M. Ehrhardt and A. Arnold, Discrete transparent boundary conditions for the Schrödinger equation. In: *Fluid dynamic processes with inelastic interactions at the molecular scale (Torino, 2000)*. *Riv. Mat. Univ. Parma* **4** (2001) 57–108.
- [19] B. Engquist and A. Majda, Absorbing boundary conditions for the numerical simulation of waves. *Math. Comput.* **31** (1977) 629–651.
- [20] F. Filbet and C. Yang, An inverse Lax–Wendroff method for boundary conditions applied to Boltzmann type models. *J. Comput. Phys.* **245** (2013) 43–61.
- [21] M. Goldberg, On a boundary extrapolation theorem by Kreiss. *Math. Comput.* **31** (1977) 469–477.

- [22] B. Gustafsson, H.-O. Kreiss and J. Oliger, Time Dependent Problems and Difference Methods. John Wiley & Sons, New York (1995).
- [23] L. Halpern, Absorbing boundary conditions for the discretization schemes of the one-dimensional wave equation. *Math. Comput.* **38** (1982) 415–429.
- [24] H.-O. Kreiss, Difference approximations for hyperbolic differential equations. In: *Numerical Solution of Partial Differential Equations (Proc. Sympos. Univ. Maryland, 1965)*. Academic Press (1966) 51–58.
- [25] P.D. Lax and B. Wendroff, Difference schemes for hyperbolic equations with high order of accuracy. *Commun. Pure Appl. Math.* **17** (1964) 381–398.
- [26] S. Osher, An ill posed problem for a hyperbolic equation near a corner. *Bull. Am. Math. Soc.* **79** (1973) 1043–1044.
- [27] S. Osher, Initial-boundary value problems for hyperbolic systems in regions with corners. I. *Trans. Am. Math. Soc.* **176** (1973) 141–164.
- [28] S. Osher, Initial-boundary value problems for hyperbolic systems in regions with corners. II. *Trans. Am. Math. Soc.* **198** (1974) 155–175.
- [29] R.D. Richtmyer and K.W. Morton, Difference Methods for Initial Value Problems. *Graduate Texts in Mathematics*. Interscience Publishers John Wiley & Sons (1967). Theory and applications.
- [30] L. Sarason and J.A. Smoller, Geometrical optics and the corner problem. *Arch. Ratio. Mech. Anal.* **56** (1974/75) 34–69.
- [31] C.-W. Shu and S. Tan, Inverse Lax–Wendroff procedure for numerical boundary treatment of hyperbolic equations. In: Vol. 18 of *Handbook of Numerical Methods for Hyperbolic Problems. Handbook of Numerical Analysis*. Elsevier, North-Holland (2017) 23–52.
- [32] G. Szegő, Orthogonal Polynomials. American Mathematical Society, Providence, RI (1975).
- [33] S. Tan and C.-W. Shu, Inverse Lax–Wendroff procedure for numerical boundary conditions of conservation laws. *J. Comput. Phys.* **229** (2010) 8144–8166.
- [34] L.N. Trefethen, Group velocity in finite difference schemes. *SIAM Rev.* **24** (1982) 113–136.
- [35] L.N. Trefethen, Instability of difference models for hyperbolic initial boundary value problems. *Commun. Pure Appl. Math.* **37** (1984) 329–367.
- [36] F. Vilar and C.-W. Shu, Development and stability analysis of the inverse Lax–Wendroff boundary treatment for central compact schemes. *ESAIM:M2AN* **49** (2015) 39–67.