

Article

Numerical Simulation of Conservation Laws with Moving Grid Nodes: Application to Tsunami Wave Modelling

Gayaz Khakimzyanov ¹, Denys Dutykh ^{2,3,*}, Dimitrios Mitsotakis ⁴ and Nina Yu. Shokina ⁵

¹ Institute of Computational Technologies of Siberian Branch of the Russian Academy of Sciences, 630090 Novosibirsk, Russia; Khak@ict.nsc.ru

² Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA, 73000 Chambéry, France

³ LAMA UMR 5127 CNRS, Université Savoie Mont Blanc, Campus Scientifique, 73376 Le Bourget-du-Lac CEDEX, France

⁴ School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, P.O. Box 600, Wellington 6140, New Zealand; dmitsot@gmail.com

⁵ Department of Radiology, Medical Physics, Medical Center, Faculty of Medicine, University of Freiburg, 79106 Freiburg, Germany; Nina.Shokina@gmail.com

* Correspondence: Denys.Dutykh@univ-smb.fr

Received: 22 March 2019; Accepted: 24 April 2019; Published: 30 April 2019



Abstract: In the present article, we describe a few simple and efficient finite volume type schemes on moving grids in one spatial dimension combined with an appropriate predictor–corrector method to achieve higher resolutions. The underlying finite volume scheme is conservative, and it is accurate up to the second order in space. The main novelty consists in the motion of the grid. This new dynamic aspect can be used to resolve better the areas with large solution gradients or any other special features. No interpolation procedure is employed; thus, unnecessary solution smearing is avoided, and therefore, our method enjoys excellent conservation properties. The resulting grid is completely redistributed according to the choice of the so-called *monitor function*. Several more or less universal choices of the monitor function are provided. Finally, the performance of the proposed algorithm is illustrated on several examples stemming from the simple linear advection to the simulation of complex shallow water waves. The exact well-balanced property is proven. We believe that the techniques described in our paper can be beneficially used to model tsunami wave propagation and run-up.

Keywords: conservation laws; finite volumes; conservative finite differences; moving grids; adaptivity; advection; shallow water equations; wave run-up

MSC: 74S10 (primary); 74J15; 74J30 (secondary)

1. Introduction

The main goal of the present manuscript consists in introducing to the tsunami modelling community some state-of-the-art numerical methods for solving nonlinear hyperbolic equations, which allow the use of the available degrees of freedom in the (nearly-) optimal way. Until the last 15 years, tsunami wave modelling has been done in the framework of Nonlinear Shallow Water Equations (NSWE). The numerical simulations have been performed by practitioners essentially with several well-established community codes. To give a few examples, let us mention the code TUNAMI [1] based on a conservative finite difference leap-frog scheme on staggered grids. This scheme approximates NSWE on real bathymetries. Another widely used code is MOST, which is based on the

dimensional splitting method (This technique is also known as the alternating directions) [2,3]. We would like to mention also the software COMCOT, which is based on a modified leap-frog scheme discretizing NSWE in Cartesian and spherical coordinates [4]. Finally, in the Institute of Computational Technologies (SB RAS) an in-house software, MGC, based on the finite difference MacCormack scheme in spherical coordinates was developed as well [5].

A big boost in tsunami research happened after Tsunami Boxing day [6] due to the availability of real world data. After this Sumatra 2004 event, several studies questioned the importance of dispersive effects for transoceanic tsunami propagation [7–9]. Their conclusions were supported by comparisons of numerical results with field and DART buoy data. We can only mention that the importance of dispersive effects has been already highlighted in 1982 by Mirchina and Pelinovsky [10]:

The considerations and estimates for actual tsunamis indicate that nonlinearity and dispersion can appreciably affect the tsunami wave propagation at large distances.

One of the first tsunami numerical models including spherical effects *and* linear dispersive terms was TUNAMI-N2. A sensitivity study was performed in Reference [11] on the example of the Sumatra 2004 event. The reported results were in favor of including dispersive effects. Another Weakly Nonlinear Weakly Dispersive (WNWD) model was presented in Reference [12,13]. It was basically a spherical counterpart of the classical Peregrine system [14]. Let us mention also the study by Glimsdal et al. [15] where the importance of dispersion has been investigated numerically and the use of the Fully Nonlinear Weakly Dispersive (FNWD) model was recommended. Such FNWD spherical equations (with the horizontal velocity defined on a certain surface inside the fluid bulk) have been derived in Reference [16]; however, only WNWD numerical results were presented. In a recent work [17], a systematic derivation of spherical rotating FNWD (and WNWD) models have been performed and the so-called *base model* was derived, which incorporates many other models as particular cases. The reduction of spherical FNWD to WNWD models was illustrated, and several known models were recovered in this way. Finally, the fully nonlinear numerical results have been presented in Reference [18]. In particular, the importance of sphericity, Coriolis, and dispersion effects was thoroughly discussed. The numerical algorithm employed in Reference [18] is based on the splitting approach: An elliptic equation is solved to determine the non-hydrostatic pressure component, and an evolution system of hyperbolic equations with source terms is solved to advance in time the velocity and water height variables. The algorithm was implemented as an explicit two-step predictor–corrector scheme. The volume of publication [18] did not allow the authors to describe all details of the employed numerical method. Henceforth, if we adopt dispersive models for tsunami simulations (It seems now that future generation tsunami modelling codes will include at least some dispersive effects), there is a clear need in robust numerical methods to solve hyperbolic balance equations. The present manuscript should be considered as a tutorial-style article to introduce moving grid algorithms to the developers of future generation tsunami propagation codes.

The numerical simulation of conservation laws is one of the most dynamic and central parts of modern numerical analysis. The systems of conservation laws appear in many fields ranging from traffic modeling [19] to shallow water modelling [20] and compressible fluid mechanics [21]. The finite volume method was proposed in a pioneering work of S. Godunov [22,23] and developed later by Ph. Roe [24] and many other researchers (see Reference [25] for the current state of the art).

Nowadays, the complexity of the problems which arise in practice is such that we have to think about the optimal usage of available computational resources. In this way, the community came up with the idea of developing numerical methods on adaptive grids in order to have a higher resolution only where it is needed [26]. Historically, *hp*-adaptive methods were first developed for the Finite Element-type (FEM) discretizations [27] (including the newest discontinuous Galerkin discretizations [28]). The main advantage of FEM is that this method is quite flexible with respect to local approximation spaces, thus allowing for a relatively easy *p*-adaptivity. FEM was applied to hyperbolic problems under the guise of relaxation schemes [29]. Later *h*-adaptive methods have been adapted for finite volume methods as well [30]. However, the widely used approach nowadays

consists in performing local grid refinement (or coarsening) with locally nested grids [31]. For instance, it has been successfully applied to tsunami propagation and run-up problems [32] as well as to more complex two-phase incompressible flows [33].

Adaptive mesh refinement techniques applied to the spatial variable after the computation of an approximate solution of the model equations at each time step usually consist of the following steps:

- Generation of a new spatial mesh according to prescribed adaptivity criteria;
- Reconstruction or interpolation of the numerical solution on the new mesh;
- Integration in time of the numerical solution on the new mesh.

The main disadvantages of this methodology is that the reconstruction of the numerical solution on the new mesh increases the complexity of the method and also reduces the accuracy of the method by adding more dissipation or dispersion to the solution. Moreover, the variable number of nodes represents some difficulties while implementing the algorithm on a computer. The underlying data structures have to be flexible enough to insert or to suppress some elements.

The approach we propose in this study is different and does not require an interpolation or reconstruction of the solution to the new mesh (a conservative interpolation is employed in Reference [34]). Although the proposed method is closer to the redistribution method proposed in References [34,35], it is simpler and more elegant at the level of the implementation details. Namely, it is based on the transformation of the model equations to some new equations of which the solutions are independent of the new mesh. Namely, the method uses the same number of discretization points of the spatial mesh during the simulation, and the adaptivity is achieved by moving the grid nodes to the places indicated by the so-called *monitor function*. The adapted grid is obtained as a solution of an elliptic (or parabolic) problem, which ensures the smoothness of the obtained grid. The main idea behind the equidistribution principle is to distribute the nodes over the computational domain so that the measure of a cell times the value of the monitor function on it is approximately constant [36,37]. The heart of the matter in the equidistribution method is the construction of a suitable monitor function. This method was described independently in the seminal paper of Reference [38] and later in Reference [39]. Numerous subsequent developments were published in recent references [40–42]. The review of earlier works on the adaptive grid generation can be found in References [43,44]. The “Holy Grail” in choosing the monitor function is to achieve ideally the reduction of the error in orders of magnitude when the discretization step $\Delta x \rightarrow 0$, as illustrated in Reference [45]. The main advantages of the proposed approach include

- Conservation in space;
- Second-order accuracy (in smooth regions);
- TVD property for scalar problems;
- Well-balanced character;
- Preservation of the discrete conservation law in cell centers and cell interfaces.

The present manuscript sheds also some light on the hyperbolic part of the splitting approach we used earlier to simulate the fully nonlinear and weakly dispersive wave propagations [46].

For simplicity, we focus on one-dimensional cases only, and even in this simple case, some open problems are outlined throughout the manuscript. According to our knowledge, the pioneers in 1-D cases were the authors of References [47] (for nonlinear shallow water equations) and [48] (for gas dynamics). The generation of 2-D curvilinear adaptive grids is a relatively classical topic [49]. A rigorous definition of a curvilinear grid was given in Reference [50]. The generalization of the equidistribution method to two spatial dimensions was performed for the first time in Reference [51]. There are a few studies which report recent modern implementations on *moving* redistributed two-dimensional grids (see, for example, References [34,52–54]) with generalized monitor functions formalized in Reference [41]. Some approaches for the intercomparison of generated grids are described in Reference [55]. We refer also to Reference [56] for a recent review of moving grid techniques.

The moving grid methods seem to be unavoidable in the modern numerical analysis. There are some classes of problems, which cannot be solved, in principle, with a fixed grid. To give an example, we can mention the class of problems involving moving boundaries such as free surface flows [57–59].

The discretization of the model equations is based on explicit, fully discrete predictor–corrector Finite Volume schemes that are accurate up to the second order in space. Predictor–corrector schemes have been introduced by MacCormack [60] and can be considered as a natural generalization of splitting schemes in higher spatial dimensions.

We have to mention that nonuniform grids can be used also to preserve some Lie symmetry group of the equation at the discrete level (see Section 3.2.2 in Reference [61] for a brief survey and the Burgers–Hopf equation example). Originally, this approach was proposed by Dorodnitsyn [62] and developed later in Reference [63]. In our study, the mesh motion is directed solely by the equidistribution principle as in Reference [34] without taking into account the symmetry considerations for the moment.

The present article is organized as follows. In Section 2, we discuss in all details the implementation of the algorithm for the linear advection equation. Then, we generalize it to the nonlinear case in Section 3 and the systems of conservation laws in Section 4. Some open problems are outlined in Section 5. Finally, the main conclusions and perspectives of our study are given in Section 6.

2. Linear Scalar Equation

In order to present the moving grid method, we shall start with the simplest scalar linear advection equation in this section and, then, we shall increase gradually the complexity by adding first the nonlinearities in the following section and moving to the systems by the end of our manuscript.

We pay special attention to the linear case for the following reasons:

- The main properties of the scheme are the most transparent in the linear case;
- The generalization to the nonlinear and vectorial cases will be easier once the linear case is fully understood. Therefore, it will allow us to go faster in the subsequent sections;
- An exhaustive error analysis is possible in the linear (and presumably only in the linear) case.

In order to construct an efficient finite volume scheme on a moving mesh, we have to choose first a robust and an accurate scheme on a fixed grid, which will be generalized later to incorporate the motion of mesh points. Such a scheme retained for our study is described in the next section.

2.1. A Predictor–Corrector Scheme on a Fixed Uniform Mesh

Consider the following linear advection equation with a constant propagation velocity:

$$u_t + a u_x = 0, \quad a \in \mathbb{R}. \tag{1}$$

The subscripts in this study denote the partial derivatives, i.e., $u_t = \frac{\partial u}{\partial t}$, $u_x = \frac{\partial u}{\partial x}$, etc.

We introduce a uniform discretization of the real space line \mathbb{R} with nodes $x_j = j \Delta x$, where $\Delta x > 0$; $j \in \mathbb{Z}$ is the spatial discretization step, and for simplicity, we do not pay attention to the boundary conditions here. The interval $C_j = [x_j, x_{j+1}]$ will be referred to as the cell C_j . The time step is denoted by τ , and the discrete solution is computed at $t^n = n\tau$, $n \in \mathbb{Z}^+$. Equation (1) is discretized in space and time using an explicit fully discrete predictor–corrector scheme.

$$\frac{u_{j+1/2}^* - u_{j+1/2}^n}{\tau_{j+1/2}^*} + a u_{x,j+1/2}^n = 0, \quad \frac{u_j^{n+1} - u_j^n}{\tau} + a \frac{u_{j+1/2}^* - u_{j-1/2}^*}{\Delta x} = 0, \tag{2}$$

where τ is the time step. The intermediate quantities $u_{j\pm 1/2}^*$ are evaluated in the middle of the cells at $x_{j\pm 1/2} = \frac{x_j + x_{j\pm 1}}{2} = x_j \pm \frac{1}{2}\Delta x$ and at time instances $t = t^n + \tau_{j\pm 1/2}^*$. Moreover, the following notations have been introduced in Equation (2):

$$u_{j+1/2}^n = \frac{u_{j+1}^n + u_j^n}{2}, \quad u_{x,j+1/2}^n = \frac{u_{j+1}^n - u_j^n}{\Delta x}, \quad \tau_{j+1/2}^* = \frac{\tau}{2} (1 + \theta_{j+1/2}^n).$$

We underline the fact that the parameter θ has not been fixed for the moment and that it may vary from one cell and one time layer to another one.

The predictor–corrector scheme in Equation (2) can be recast as a one-step scheme by combining two equations together:

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{\tau a^2}{2\Delta x} \left\{ ((1 + \theta)u_x)_{j+1/2}^n - ((1 + \theta)u_x)_{j-1/2}^n \right\} = 0. \tag{3}$$

The last fully discrete scheme is a canonical form of all two-step explicit schemes for Equation (1), since many well-known schemes can be obtained from Equation (3) by carefully choosing the parameter θ . For example,

Lax–Wendroff scheme [64] ($\theta = 0$)

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{\tau a^2}{2\Delta x} \left\{ u_{x,j+1/2}^n - u_{x,j-1/2}^n \right\} = 0.$$

First-order upwind [65] ($\theta = \frac{1}{C} - 1 > 0$) The number $C := |a| \frac{\tau}{\Delta x} < 1$ is the famous Courant–Friedrichs–Lewy number [66], which controls the stability of the upwind scheme.

$$\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{a + |a|}{2} u_{x,j-1/2}^n + \frac{a - |a|}{2} u_{x,j+1/2}^n = 0.$$

Lax–Friedrichs scheme [67] ($\theta = \frac{1}{C^2} - 1 > 0$)

$$\frac{u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)}{\tau} + a \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0.$$

However, the schemes listed above are not suitable for accurate simulations, since the first order upwind and Lax–Friedrichs schemes are too diffusive in practice and the Lax–Wendroff scheme is not monotonicity preserving. A good solution consists in choosing the parameter θ adaptively. In a previous work [68], the following strategy was proposed:

$$\theta_{j+1/2}^n = \begin{cases} 0, & \left| \frac{u_{x,j+1/2}^n}{u_{x,j+1/2}^n} \right| \leq \left| \frac{u_{x,j+1/2-s}^n}{u_{x,j+1/2-s}^n} \right| & \text{and } u_{x,j+1/2}^n \times u_{x,j+1/2-s}^n \geq 0, \\ \theta_0(1 - \zeta_{j+1/2}^n), & \left| \frac{u_{x,j+1/2}^n}{u_{x,j+1/2}^n} \right| > \left| \frac{u_{x,j+1/2-s}^n}{u_{x,j+1/2-s}^n} \right| & \text{and } u_{x,j+1/2}^n \times u_{x,j+1/2-s}^n \geq 0, \\ \theta_0, & u_{x,j+1/2}^n \times u_{x,j+1/2-s}^n < 0, \end{cases}$$

with $\theta_0 := \frac{1}{C} - 1 > 0$, $s := \text{sign}(a)$, $\zeta_{j+1/2}^n := \frac{u_{x,j+1/2-s}^n}{u_{x,j+1/2}^n}$. The first case corresponds to the classical

Lax–Wendroff scheme, the third case is the classical first order upwind scheme, and the second case is detailed in the following Remark 1. However, the proposed combination of schemes produces a robust TVD-type scheme, which ensures a very good trade-off between the solution accuracy and monotonicity properties. The scheme parameter θ plays somehow the role of the traditional minmod slope limiter in more conventional MUSCL-type finite volume methods [69]. Its performance on fixed

and mobile meshes will be illustrated in the numerical sections below. The complete justification and validation of this scheme can be found in Reference [68].

Remark 1. We note that the canonical form of Equation (3) contains not only three points schemes on a symmetric stencil. For instance, if we assume for simplicity $a > 0$, and set

$$\theta_{j+1/2}^n := \left(\frac{1}{C} - 1\right) \left(1 - \frac{u_{x,j-1/2}^n}{u_{x,j+1/2}^n}\right),$$

then we shall obtain the following second order upwind scheme on an asymmetric stencil:

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a \frac{3u_{x,j-1/2}^n - u_{x,j-3/2}^n}{2} - \frac{\tau a^2}{2} \frac{u_j^n - 2u_{j-1}^n + u_{j-2}^n}{\Delta x^2} = 0.$$

Of course, the numerical properties such as the accuracy, stability, and monotonicity depend crucially on the choice of the parameter θ and have to be studied carefully in each particular situation.

2.2. Predictor–Corrector Schemes on a Moving Grid

In this section, we shall restrict our attention to bounded domains only, since almost all numerical simulations in 1-D are made on finite intervals. To be more specific, we assume that the computational domain is $\mathcal{I} = [0, \ell]$. Consider also the reference domain $Q = [0, 1]$ and a smooth bijective time-dependent mapping $x(\cdot, t) : Q \times \mathbb{R}^+ \mapsto \mathcal{I}$, which satisfies the following boundary conditions

$$x(0, t) = 0, \quad x(1, t) = \ell. \tag{4}$$

For our purely numerical purposes, we do not even need to have the complete information about this mapping. Consider a uniform mesh of the reference domain $Q_h = \{q_j = j \Delta q\}_{j=0}^N$, with a constant step $\Delta q := \frac{1}{N}$. Strictly speaking, we need to know only the image of nodes q_j under the map $x(q, t)$, since they constitute the nodes of the moving mesh:

$$x(q_j, t^n) = x_j^n.$$

The linear advection of Equation (1) can be rewritten on domain Q if we introduce the composed function $v(q, t) := (u \circ x)(q, t) \equiv u(x(q, t), t)$:

$$v_t + \frac{\bar{a}}{J} v_q = 0, \tag{5}$$

where $\bar{a} := a - x_t$ is the advection speed relative to the mesh nodes velocity and $J := x_q > 0$ is the Jacobian of the transformation $x(q, t)$. Equation (5) does possess a conservative form as well

$$(Jv)_t + (\bar{a}v)_q = 0. \tag{6}$$

Now, we can write the predictor–corrector scheme for the linear advection equation on a moving mesh:

$$\frac{v_{j+1/2}^* - v_{j+1/2}^n}{\tau_{j+1/2}^*} + \left(\frac{\bar{a}}{J} v_q\right)_{j+1/2}^n = 0, \tag{7}$$

$$\frac{(Jv)_j^{n+1} - (Jv)_j^n}{\tau} + \frac{(\bar{a}^n v^*)_{j+1/2} - (\bar{a}^n v^*)_{j-1/2}}{\Delta q} = 0. \tag{8}$$

Note that the first step is based on Equation (5) and that the second one is based on the conservative form of Equation (6). Here, we introduced some new notations:

$$x_{t,j+1/2}^n = \frac{x_{t,j}^n + x_{t,j+1}^n}{2}, \quad x_{t,j}^n = \frac{x_j^{n+1} - x_j^n}{\tau},$$

$$J_{j+1/2}^n \equiv x_{q,j+1/2}^n = \frac{x_{j+1}^n - x_j^n}{\Delta q}, \quad J_j^n \equiv x_{q,j}^n = \frac{J_{j+1/2}^n + J_{j-1/2}^n}{2} = \frac{x_{j+1}^n - x_{j-1}^n}{2\Delta q}.$$

As before, one can recast Equations (7) and (8) in an equivalent one-stage form:

$$\frac{(Jv)_j^{n+1} - (Jv)_j^n}{\tau} + \frac{(\bar{a}v)_{j+1/2}^n - (\bar{a}v)_{j-1/2}^n}{\Delta q} - \frac{\tau}{2\Delta q} \left\{ \left((1+\theta) \frac{\bar{a}^2}{J} v_q \right)_{j+1/2}^n - \left((1+\theta) \frac{\bar{a}^2}{J} v_q \right)_{j-1/2}^n \right\} = 0.$$

A stability study using the method of frozen coefficients [70] leads to the following restrictions on the scheme parameters:

$$\theta_{j+1/2}^n \geq 0, \quad \max_{j \in \{0, \dots, N-1\}} (\sqrt{1+\theta} C)_{j+1/2}^n \leq 1, \tag{9}$$

where $C_{j+1/2}^n$ is the local CFL number [66] defined as

$$C_{j+1/2}^n = \frac{\tau}{\Delta q} \left(\frac{|\bar{a}|}{J} \right)_{j+1/2}^n.$$

In order to ensure the monotonicity of the numerical solution, it was shown in Reference [68] that it is sufficient to use the following choice of the scheme parameter θ :

$$\theta_{j+1/2}^n = \begin{cases} 0, & \left| \tilde{\delta}_{j+1/2}^n \right| \leq \left| \tilde{\delta}_{j+1/2-s}^n \right| \quad \text{and} \quad \tilde{\delta}_{j+1/2}^n \times \tilde{\delta}_{j+1/2-s}^n \geq 0, \\ (\theta_0(1-\zeta))_{j+1/2}^n, & \left| \tilde{\delta}_{j+1/2}^n \right| > \left| \tilde{\delta}_{j+1/2-s}^n \right| \quad \text{and} \quad \tilde{\delta}_{j+1/2}^n \times \tilde{\delta}_{j+1/2-s}^n \geq 0, \\ \theta_{0,j+1/2}^n, & \tilde{\delta}_{j+1/2}^n \times \tilde{\delta}_{j+1/2-s}^n < 0, \end{cases} \tag{10}$$

where $s = \text{sign}(\bar{a}_{j+1/2}^n)$ and

$$\theta_{0,j+1/2}^n = \frac{1}{C_{j+1/2}^n} - 1, \quad \tilde{\delta}_{j+1/2}^n = (|\bar{a}| (1-C) v_q)_{j+1/2}^n, \quad \zeta_{j+1/2}^n = \frac{\tilde{\delta}_{j+1/2-s}^n}{\tilde{\delta}_{j+1/2}^n}.$$

Using Equation (10), it can be shown that the stability requirements of Equation (9) are fulfilled under the following single restriction on local CFL numbers:

$$C_{j+1/2}^n \leq 1.$$

Finally, the global CFL number C is defined as

$$C := \max_j \{C_{j+1/2}^n\}.$$

It is this value which will be reported in the tables below.

There is an additional relation which is called the *geometric conservation law*. In the 1-D case, it takes a very simple form

$$J_t - (x_t)_q = 0,$$

since it translates a trivial fact that $x_{qt} \equiv x_{tq}$. However, it was demonstrated in Reference [71] that it is absolutely crucial to respect the geometric conservation law at the discrete level as well in order to have a consistent fully discrete scheme which preserves all constant solutions exactly. It can be shown

that the scheme proposed above satisfies the discrete counterpart of the geometric conservation law in integer and half-integer nodes:

$$\frac{J_j^{n+1} - J_j^n}{\tau} - \frac{x_{t,j+1/2}^n - x_{t,j-1/2}^n}{\Delta q} = 0,$$

$$\frac{J_{j+1/2}^{n+1} - J_{j+1/2}^n}{\tau} - \frac{x_{t,j+1}^n - x_{t,j}^n}{\Delta q} = 0.$$

2.3. Construction and Motion of the Grid

In the previous sections, we explained in some detail the predictor–corrector schemes on fixed and moving meshes. We saw that the mesh motion is parametrized by a bijective mapping $x(q, t) : Q \times \mathbb{R}^+ \mapsto \mathcal{I}$. Now, we explain how this map is effectively constructed, since so far, it was quite general. Moreover, the predictor–corrector schemes presented above are valid for other constructions of the mesh motion.

The first step consists in choosing a positive valued function $w(x, t) : \mathcal{I} \times \mathbb{R}^+ \mapsto \mathbb{R}^+$, the so-called *monitor function*, which controls the distribution of the nodes. Without entering into the details for the moment, we suggest as one choice for a monitor function the following expression:

$$w(x, t) = 1 + \alpha |u_x(x, t)|, \tag{11}$$

where $\alpha > 0$ is a real parameter. The monitor function of Equation (11) is a particular case of the class of monitor functions considered by Beckett et al. [53]. Other choices of the monitor function are discussed in References [54,72,73]. For instance, to give an example, another popular choice of the monitor function is

$$w(x, t) = \sqrt{1 + \alpha |u_x(x, t)|^2}.$$

In general, the choice of the monitor function and its parameters might be crucial for the accuracy of the scheme. The adaptive grid construction for a simple linear singularly perturbed Sturm–Liouville equation is deeply analysed in Reference [45]. This question seems to be rather open for unsteady nonlinear problems. The current state-of-the-art in this field is described in Reference [41].

It is noted that the monitor function varies in both space and time and takes large (positive) values in areas where the solution has important gradients. The monitor function $w(x, t)$ specified in Equation (11) can be readily discretized.

$$w_{j+1/2}^n = 1 + \alpha \frac{|u_{j+1}^n - u_j^n|}{x_{j+1}^n - x_j^n}.$$

The proposed algorithm is slightly different on the very first step when the initial grid is generated. It is important to obtain a high-quality initial mesh, since the numerical error committed in the beginning cannot be corrected afterwards. Therefore, the first step is explained in detail below and the algorithm for unsteady computations is presented in the following section.

2.3.1. Initial Grid Generation

It is of capital importance to produce a grid adapted to the initial condition as well, since no error committed initially can be corrected during dynamical simulations. At $t = 0$, we compute the monitor function $w(x, 0)$ of the initial condition $u(x, 0)$ and the mapping $x(q, 0)$ is determined as the solution to the following *nonlinear* elliptic problem, which degenerates to a simple second-order ordinary differential equation

$$(w(x, 0) x_q)_q = 0, \tag{12}$$

supplemented with Dirichlet boundary conditions (4). This elliptic problem can be readily discretized to produce the following finite difference approximation:

$$\frac{1}{\Delta q} \left\{ w_{j+1/2}^0 \frac{x_{j+1}^0 - x_j^0}{\Delta q} - w_{j-1/2}^0 \frac{x_j^0 - x_{j-1}^0}{\Delta q} \right\} = 0, \quad j = 1, \dots, N-1,$$

with boundary conditions $x_0^0 = 0, x_N^0 = \ell$. The last nonlinear system of equations is solved iteratively (usually with some fixed point-type iterations). The iterations are initialized with a uniform grid as the first guess. One can notice also that its solution satisfies the equidistribution principle:

$$w_{j+1/2}^0 \frac{x_{j+1}^0 - x_j^0}{\Delta q} = C_{\Delta q}, \quad j = 0, \dots, N-1, \tag{13}$$

where the constant C_h can be computed as

$$C_{\Delta q} = \sum_{j=0}^{N-1} (\Delta x \times w)_{j+1/2}^0, \quad \Delta x_{j+1/2}^0 := x_{j+1}^0 - x_j^0.$$

Therefore, $C_{\Delta q}$ is the discrete version of the monitor function integral. The equidistribution principle gives meaning to the monitor function: In the areas where $w_{j+1/2}^0$ takes large values, the spacing between two neighboring nodes x_{j+1}^0 and x_j^0 has to be inversely proportionally small.

2.3.2. Unsteady Computations

Assume that the grid $\{x_j^n\}_{j=0}^N$ at $t = t^n$ is known, and let us evolve it to the next time layer t^{n+1} . It is done by solving the following linear fully discrete problem:

$$\frac{1}{\Delta q} \left\{ w_{j+1/2}^n \frac{x_{j+1}^{n+1} - x_j^{n+1}}{\Delta q} - w_{j-1/2}^n \frac{x_j^{n+1} - x_{j-1}^{n+1}}{\Delta q} \right\} = \beta \frac{x_j^{n+1} - x_j^n}{\tau}, \quad j = 1, \dots, N-1, \tag{14}$$

with the same boundary conditions $x_0^{n+1} = 0, x_N^{n+1} = \ell$ as above. Here, the real positive parameter $\beta > 0$ plays the role of the diffusion coefficient, and it controls the smoothness of the nodes trajectories. The optimal choice of the parameter β is currently an open problem, and it has to be done empirically on a case-by-case basis. It can be easily seen that the scheme of Equation (14) is an implicit discretization of this nonlinear parabolic equation:

$$(w(x, t) x_q)_q = \beta x_t. \tag{15}$$

However, it has to be stressed that the fully discrete problem is linear, since the monitor function $w_{j\pm 1/2}^n$ is taken on the previous time layer. Thus, the overall complexity of the grid motion algorithm at every time step is equal to that of the tridiagonal matrix algorithm [74].

2.3.3. A Lyrical Digression

It is probably not completely clear to the reader why one has to use a different procedure initially, especially since the solution at every time step might be considered as an initial condition to the next time level. This question would be completely legitimate.

The moving grid technique based on the solution of a (nonlinear) parabolic equation as in Equation (15) was specifically designed for the solution of unsteady problems. It was proposed for the first time in Reference [38] to simulate gas dynamics problems, and it ensures the required smoothness of grid points trajectories (see the numerical examples below). The position of grid points on two consecutive time steps are related by a finite-difference analogue of Equation (15). This modification of the equidistribution principle allows an avoidance of any rapid changes in the location of grid

nodes. Basically, it was understood in the original work [38] and confirmed later in subsequent developments in late the 1980s and early 1990s [37,75–78]. Subsequent developments have been done in References [39,41,63]. Some recommendations against using the steady equation of Equation (12) to determine the position of grid nodes on each time step were formulated in References [78–80]. Some examples of grid nodes oscillations, and other instabilities were shown there as well. In Appendix A, we provide a few *analytical* examples to illustrate eventual problems which may arise if one uses the equidistribution principle of Equation (12) on every time step. Indeed, we demonstrate in Appendix A that even infinitesimal perturbations of the monitor function may yield finite perturbations in the solution $x(q, t)$.

2.3.4. Smoothing Step

In some problems where the solution contains a lot of oscillations (i.e., many local extrema), the monitor function $w(x, t)$ has to be filtered out to ensure the smoothness of the mesh motion. Moreover, a smoothing step allows an enlargement of the set of acceptable values of the parameter α in the definition of Equation (11) of the monitor function. Our numerical experiments show that the following implicit procedure (inspired by implicit discretizations of the heat equation) produces very robust results:

$$\bar{w}_{j+1/2} = w_{j+1/2} - \sigma \bar{w}_{j+1/2} + \frac{\sigma}{2} (\bar{w}_{j-1/2} + \bar{w}_{j+3/2}), \quad j = 1, \dots, N-2, \quad (16)$$

where $\sigma > 0$ is an ad hoc smoothing parameter to be set empirically. In our study, we do a smoothing similar to Asselin's filter proposed earlier for time integration problems [81]. The system of Equation (16) is completed with boundary conditions

$$\bar{w}_{1/2} = w_{1/2}, \quad \bar{w}_{N-1/2} = w_{N-1/2}.$$

The smoothed discrete monitor function $\{\bar{w}_j\}_{j=0}^N$ is obtained by solving the linear system of Equation (16) with any favorite method. In the present study, we used the simplest tridiagonal matrix algorithm [74]. In contrast, the authors of Reference [34] used an explicit smoothing method. We stress that one should apply a smoothing operator to the monitor function and not to the numerical solution.

2.3.5. On the Choice of the Monitor Function

In general, when one has a problem (i.e., a system of PDEs) to solve, there is a question of the best possible monitor function choice for the grid adaptation. One has to choose also the pertinent solution component as an input for the monitor function. It is extremely difficult to give some general recommendations which would work in all cases (It is probably as difficult as the general theory of PDEs). However, in choosing the monitor function, one has to take into account the following aspects:

Physical considerations: Any a priori knowledge on the solution can be used to underline important physical aspects (e.g., the presence of shock waves or the presence of multiple wave crests, etc.);

Geometrical considerations: The computational domain geometry (e.g., the presence of angles or fractal shorelines in wave run-up problems) and topology (i.e., the presence of islands or other obstacles);

Mathematical considerations: The nature of equations (differential, partial differential, integro-differential, singularly perturbed, etc.) but one also has to pick up the most pertinent variable to compute the monitor function on it (the free surface elevation or the vorticity field in Navier–Stokes simulations);

Numerical considerations: Particularities of the employed discretization scheme may also hint the choice of the monitor function.

Once the monitor function is chosen, one has to determine also the (nearly) optimal values of parameters which depend not only on the problem but also on the solution. Finally, the most important

guideline nowadays is the *experience* in solving practical problems using moving grid techniques. It cannot be replaced by any guidelines. We refer also to the review of Reference [56].

2.4. Numerical Results

In order to illustrate the performance of this algorithm, we choose a notorious test case, which consists in simulating the propagation of a discontinuous profile:

$$u(x,0) = \begin{cases} 1, & x \leq x_*, \\ 0, & x > x_*, \end{cases} \quad x_* \in [0, \ell].$$

The monitor function employed in this computation is defined in Equation (11). The numerical parameters used in this simulations are given in Table 1. The simulation results at time $t = T_f$ is shown in Figure 1. From the comparison of profiles obtained on fixed and moving meshes, one can see that the moving mesh reduces significantly the “width” of the numerical shock wave profile.

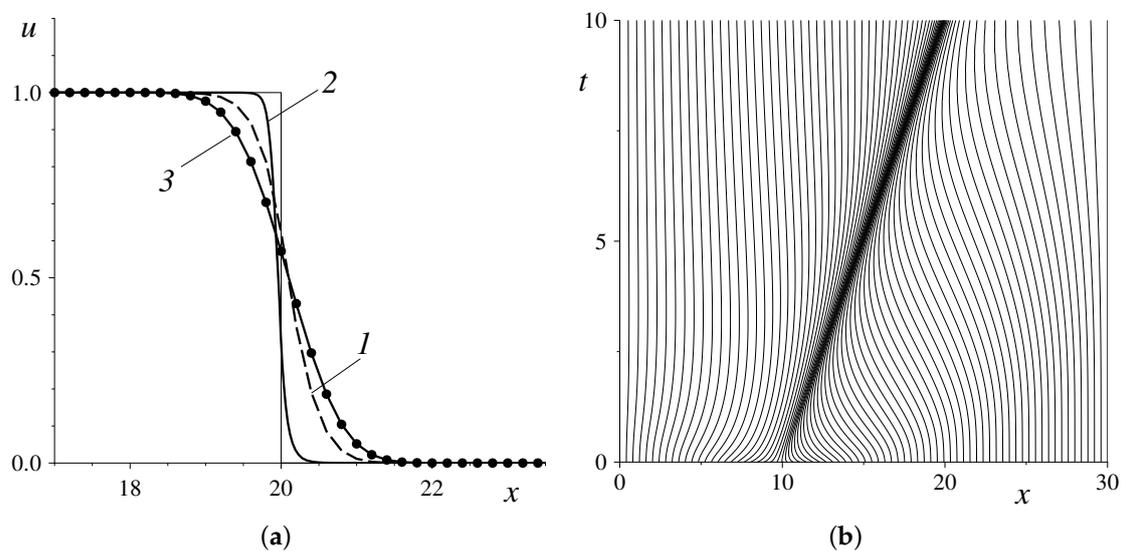


Figure 1. A discontinuous profile propagation under the linear advection of Equation (1): (a) A comparison of different profiles (The thin solid line is the exact solution, (1) is the numerical solution on the uniform mesh, (2) is the solution on the moving mesh, and (3) is the first order upwind scheme on the uniform mesh); (b) the trajectories of grid nodes in space–time. Note that the Figure 1a shows only a zoom of the computational domain.

Table 1. The numerical parameters used to simulate the discontinuous profile propagation under the linear advection of Equation (1).

Parameter	Value
Computational domain length, ℓ	30.0
Number of nodes, N	150
CFL number, C	0.8
Final simulation time, T_f	10.0
Propagation speed, a	1.0
Initial position of the discontinuity, x_*	10.0
Monitor function parameter, α	10.0
Grid diffusion parameter, β	150.0
Smoothing parameter, σ	100.0

The second test case is a smooth bell-shaped function advected by the dynamics of Equation (1). This test is used to measure the ability of the scheme to preserve the extrema (In other words, we assess the strength of the numerical diffusion on a given mesh). The exact solution is given by

$$u(x, t) = e^{-25(x - x_0 - at)^2}.$$

Thus, the initial condition is simply given by $u(x, 0) = e^{-25(x-x_0)^2}$. Another particularity of the present test case is that we use the monitor function specifically designed with an emphasis on the fine resolution of the local extrema (those with high positive or high negative values):

$$w(x, t) = 1 + \alpha |u(x, t)|. \quad (17)$$

It is noted that there is no differentiation operator in the definition of the last monitor function compared to the one defined in Equation (11).

The parameters used in the numerical simulation are provided in Table 2. The numerical results at $t = T_f$ are shown in Figure 2. For the sake of comparison, we show also the numerical solution obtained with the same scheme but on a uniform mesh. The use of a moving grid improves dramatically the preservation of the local maximum. One cannot even distinguish the exact solution from the numerical one up to the graphic resolution.

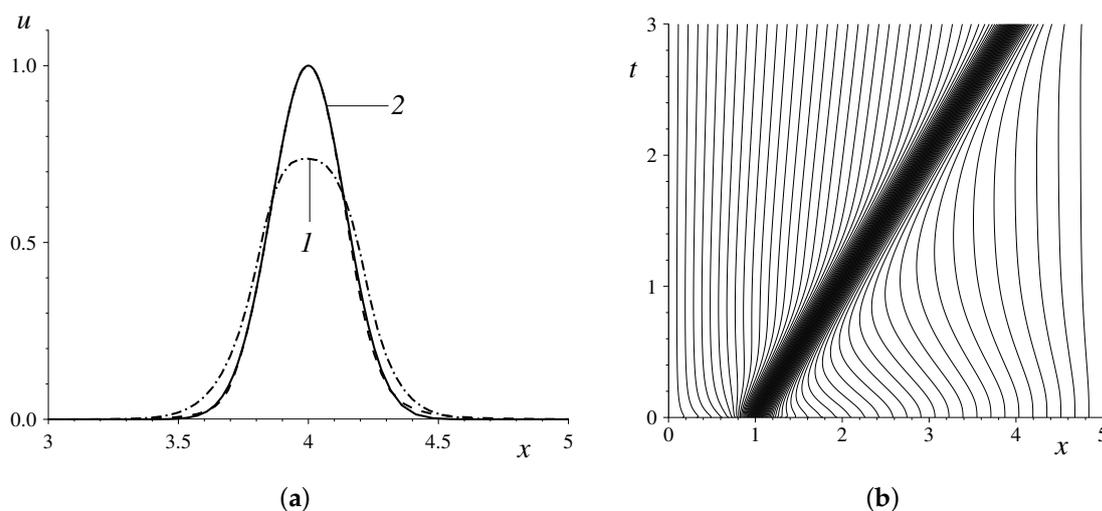


Figure 2. A smooth profile propagation under the linear advection of Equation (1): (a) A comparison of different profiles (The thin solid line is the exact solution, (1) is the numerical solution on the uniform mesh, and (2) is the solution on the moving mesh); (b) the trajectories of grid nodes in space–time. Note that Figure 2a shows only a zoom of the computational domain.

Table 2. The numerical parameters used to simulate the bell-shaped profile propagation under the linear advection of Equation (1).

Parameter	Value
Computational domain length, ℓ	5.0
Number of nodes, N	150
CFL number, C	0.8
Final simulation time, T_f	3.0
Initial position of the bell, x_0	1.0
Propagation speed, a	1.0
Monitor function parameter, α	20.0
Grid diffusion parameter, β	20.0
Smoothing parameter, σ	10.0

3. Nonlinear Scalar Equation

In this section, we consider a general hyperbolic scalar nonlinear conservation law of the form:

$$u_t + [f(u)]_x = 0.$$

As a chrestomathic example of such equations, we can mention the celebrated Burgers–Hopf equation [82]. As above, we make a change of variables to the fixed reference domain Q

$$v_t + \frac{1}{J} [f_q - x_t v_q] = 0,$$

or in the conservative form

$$(Jv)_t + [f - x_t v]_q = 0,$$

where $v := u \circ x$ and $J := x_q$, as before.

For nonlinear equations, we need an extra transport equation for the flux function:

$$f_t + a(u) f_x = 0,$$

where $a(u) = f_u(u)$. The last equation transforms to the following nonconservative form on the fixed domain Q :

$$f_t + \frac{a(v)}{J} [f_q - x_t v_q] = 0.$$

Now, we can write down the predictor–corrector scheme directly on a moving grid (Notice that, in the nonlinear case, three stages are required):

$$\begin{aligned} \frac{v_{j+1/2}^* - v_{j+1/2}^n}{\tau_{j+1/2}^*} + \left(\frac{f_q - x_t v_q}{J} \right)_{j+1/2}^n &= 0, \\ \frac{f_{j+1/2}^* - f_{j+1/2}^n}{\tau_{j+1/2}^*} + \left(\frac{a(f_q - x_t v_q)}{J} \right)_{j+1/2}^n &= 0, \\ \frac{(Jv)_j^{n+1} - (Jv)_j^n}{\tau} + \frac{(f^* - x_t^n v^*)_{j+1/2} - (f^* - x_t^n v^*)_{j-1/2}}{\Delta q} &= 0, \end{aligned}$$

where $v_{j+1/2}^n, f_{j+1/2}^n$ were defined above. The last scheme has the theoretical second-order approximation in space $\mathcal{O}(\Delta q^2)$ for smooth solutions provided that $\theta = \mathcal{O}(\Delta q)$ [68]. The only point which remains to be specified is the computation of the numerical Jacobian $a_{j+1/2}^n$. In order to ensure good properties of the scheme, one has to ensure the compatibility condition $f_q = a(v) v_q$ at the discrete level $f_{q,j+1/2}^n = (a v_q)_{j+1/2}^n$ for finite differences analogue of the derivatives:

$$f_{q,j+1/2}^n = \frac{f_{j+1}^n - f_j^n}{\Delta q}, \quad v_{q,j+1/2}^n = \frac{v_{j+1}^n - v_j^n}{\Delta q}.$$

This requirement can be satisfied with the following choice [83]:

$$a_{j+1/2}^n = \begin{cases} \frac{f_{j+1}^n - f_j^n}{v_{j+1}^n - v_j^n}, & v_{j+1}^n \neq v_j^n, \\ a(v_j^n), & v_{j+1}^n = v_j^n. \end{cases}$$

By noticing that the corrector step contains only the combination $f^* - x_t^n u^*$, we can write a compact two-stage form of the predictor–corrector scheme:

$$\begin{aligned} & \left(\frac{\hat{f} - f^n}{\tau^*}\right)_{j+1/2} + \left(\frac{\bar{a}^2}{J} v_q\right)_{j+1/2}^n = 0, \\ \frac{(Jv)_j^{n+1} - (Jv)_j^n}{\tau} + \frac{(\hat{f} - (x_t v)^n)_{j+1/2} - (\hat{f} - (x_t v)^n)_{j-1/2}}{\Delta q} &= 0, \end{aligned}$$

where $\bar{a}(v, q, t) = a(v) - x_t$. The parameter $\theta_{j+1/2}^n$ is computed according to Equation (10), which ensures the TVD property of the scheme. As a result, we obtain a generalization of the Harten scheme [83] to the case of moving meshes.

3.1. Numerical Results

In order to illustrate the performance of the generalized Harten scheme on moving meshes, we take the classical example of the inviscid Burgers–Hopf equation ($f(u) = \frac{u^2}{2}$). As the first illustration, let us take the following initial condition on the computational domain $x \in [0, \ell]$:

$$u_0(x) = \begin{cases} u_l, & x \leq x_l, \\ \frac{x - x_r}{x_l - x_r} u_l + \frac{x - x_l}{x_r - x_l} u_r, & x_l < x < x_r, \\ u_r, & x \geq x_r. \end{cases}$$

All numerical parameters are given in Table 3. The exact *weak* solution to this problem is given by the following formula:

$$u(x, t) = \begin{cases} u_l, & x \leq x_l + u_l t, \\ \frac{x - x_*}{t - t_*}, & x_l + u_l t < x < x_r + u_r t, \\ u_r, & x \geq x_r + u_r t, \end{cases} \quad t < t_*.$$

The gradient catastrophe takes place at $t = t^*$ and $x = x_*$, which can be computed exactly as well.

$$t^* = -\frac{x_r - x_l}{u_r - u_l}, \quad x_* = x_l + u_l t^* \equiv x_r + u_r t^*.$$

After $t = t^*$, we obtain a travelling shock-wave profile:

$$u(x, t) = \begin{cases} u_l, & x \leq x_* + \{u\}(t - t^*), \\ u_r, & x > x_* + \{u\}(t - t^*), \end{cases} \quad \{u\} := \frac{u_l + u_r}{2}.$$

For the parameters chosen in our numerical simulation (see Table 3), the shock wave remains stationary at $t = t^* = 5$ and $x = x_* = 15$ since $u_l = -u_r$. The numerical solution and the grid motion are shown in Figure 3. It can be seen that the solution monotonicity is preserved on both fixed and moving meshes. However, on the moving grid, we obtain a solution which cannot be distinguished from the exact one up to the graphical resolution. On the fixed grid, the shock wave has a finite width equal to two control volumes.

As another numerical illustration, we consider the same problem, but we change the value of $u_r := 0$ instead of -1 . In this case, the gradient catastrophe takes place at $t = t^* = 10$ and $x = x_* = 20$. The shock wave profile is not steady anymore since $\{u\} = \frac{u_l + u_r}{2} = \frac{1}{2} \neq 0$. Therefore, it will move in the rightward direction with a constant speed equal to $\frac{1}{2}$. The results of the numerical simulations are shown in Figure 4. It can be seen again that the adaptive solution is excellent while the fixed grid produces a shock wave smeared out on four cells only.

Table 3. The numerical parameters used to simulate a stationary shock-wave formation under the Burgers–Hopf equation dynamics.

Parameter	Value
Computational domain length, ℓ	30.0
Number of nodes, N	60
Final simulation time, T_f	10.0
CFL number, C	0.2
Left transition coordinate, x_l	10.0
Right transition coordinate, x_r	20.0
Left constant state, u_l	1.0
Right constant state, u_r	-1.0
Monitor function parameter, α	15.0
Grid diffusion parameter, β	80.0
Smoothing parameter, σ	60.0

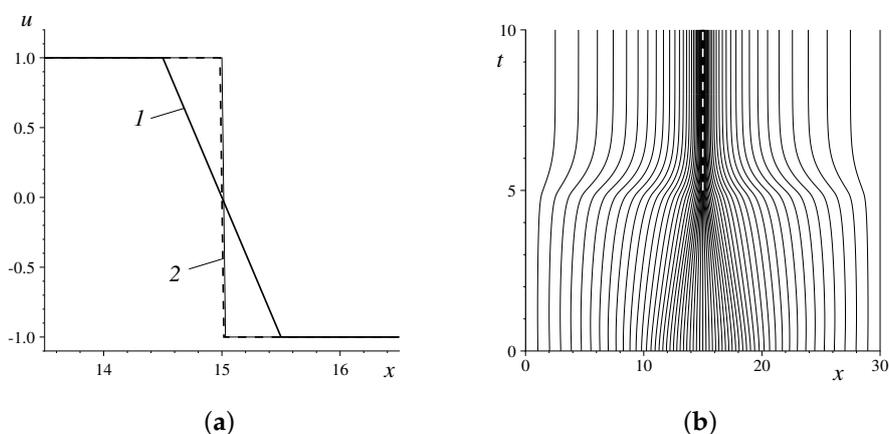


Figure 3. The stationary shock wave formation in the Burgers–Hopf equation: (a) A comparison of different profiles (The thin solid line is the exact solution, (1) is the numerical solution on the uniform mesh, and (2) is the solution on the moving mesh); (b) the trajectories of grid nodes in space–time. Note that Figure 3a shows only a zoom of the computational domain.

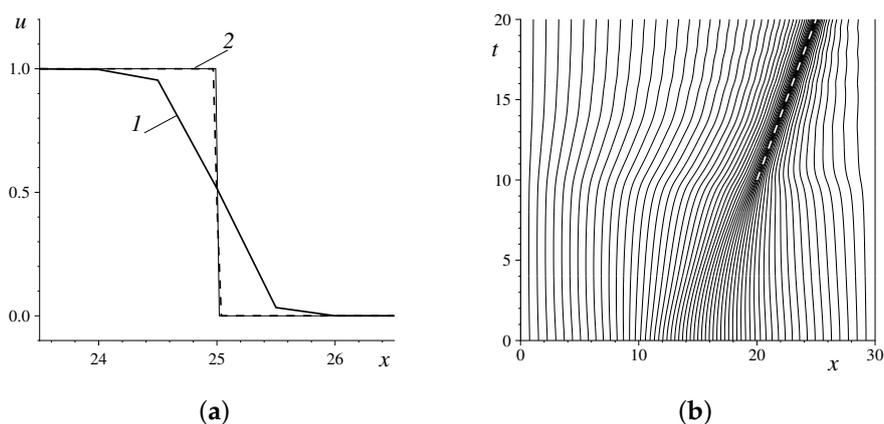


Figure 4. A stationary shock wave formation in the Burgers–Hopf equation: (a) A comparison of different profiles (The thin solid line is the exact solution, (1) is the numerical solution on the uniform mesh, and (2) is the solution on the moving mesh); (b) the trajectories of grid nodes in space–time. Note that Figure 4a shows only a zoom of the computational domain.

4. Nonlinear System of Equations

The predictor–corrector scheme for nonlinear systems on moving meshes will be presented on a practically important case of the Nonlinear Shallow Water Equations (NSWE) [84]. This choice reflects also the scientific interests of the authors of the present article. Moving grid techniques have been applied to other systems of conservation laws such as the Euler equations in Reference [73], pressureless gas dynamics [85] and magnetohydrodynamics [86].

Therefore, consider one-dimensional nonlinear shallow water equations written in a vectorial form:

$$\mathbf{u}_t + [f(\mathbf{u})]_x = G(\mathbf{u}, x), \tag{18}$$

where \mathbf{u} is the vector of conservative variables, $f(\mathbf{u})$ is the flux function, and $G(\mathbf{u}, x)$ contains the source terms:

$$\mathbf{u} = \begin{pmatrix} H \\ Hu \end{pmatrix}, \quad f(\mathbf{u}) = \begin{pmatrix} H u \\ Hu^2 + \frac{g}{2} H^2 \end{pmatrix}, \quad \mathbf{G}(\mathbf{u}, x) = \begin{pmatrix} 0 \\ gHh_x \end{pmatrix}.$$

Here, $u(x, t)$ is the depth-averaged velocity and $H(x, t) = \eta(x, t) + h(x)$ is the total water depth defined as the sum of the bathymetry function $h(x)$ and the free surface elevation $\eta(x, t)$ over the mean water level. The sketch of the physical domain with all these definitions is shown in Figure 5. The system of Equation (18) is considered on a finite segment $\mathcal{I} = [0, \ell]$ with appropriate boundary conditions. As before, we consider a smooth bijective time-dependent transformation $x = x(q, t)$ from the reference domain $Q = [0, 1]$ into \mathcal{I} . The NSWE on the reference domain are written in the form [87]:

$$\mathbf{v}_t + \frac{1}{J} [f_q - x_t \mathbf{v}_q] = \frac{1}{J} \mathbf{G}(\mathbf{v}, q),$$

where $\mathbf{v}(q, t) = \mathbf{u} \circ x = \mathbf{u}(x(q, t), t)$ and the source term contains a derivative with respect to q :

$$\mathbf{G}(\mathbf{v}, q) = \begin{pmatrix} 0 \\ gHh_q \end{pmatrix}, \quad h(q, t) = h(x(q, t)).$$

It is noted that the bathymetry becomes time-dependent under this change of variables, even if initially the bottom was fixed. There exists also a conservative form NSWE on the reference domain:

$$(J\mathbf{v})_t + [f - x_t \mathbf{v}]_q = \mathbf{G}(\mathbf{v}, q), \quad J := x_q > 0.$$

Finally, the flux vector satisfies the following equation:

$$f_t + \frac{1}{J} \mathcal{A} [f_q - x_t \mathbf{v}_q] = \frac{1}{J} \mathcal{A} \mathbf{G}(\mathbf{v}, q),$$

where \mathcal{A} is the advective flux’s Jacobian matrix, which can be easily computed:

$$\mathcal{A}(\mathbf{v}) = \frac{Df(\mathbf{v})}{D\mathbf{v}} = \begin{pmatrix} 0 & 1 \\ -v^2 + gH & 2v \end{pmatrix}.$$

Since the system of NSWE (18) is hyperbolic, the Jacobian \mathcal{A} can be decomposed into the product of left \mathcal{L} and right \mathcal{R} eigenvectors:

$$\mathcal{A} = \mathcal{R} \cdot \mathcal{D} \cdot \mathcal{L},$$

where the matrices \mathcal{L} , \mathcal{R} , and \mathcal{D} are defined as

$$\mathcal{L} = \frac{1}{c^2} \begin{pmatrix} -\lambda_2 & 1 \\ -\lambda_1 & 1 \end{pmatrix}, \quad \mathcal{R} = \frac{c}{2} \begin{pmatrix} -1 & 1 \\ -\lambda_1 & \lambda_2 \end{pmatrix}, \quad \mathcal{D} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

Here, $\lambda_{1,2} = v \mp c$ ($c = \sqrt{gH}$ being the speed of gravity waves) are eigenvalues of the Jacobian $\mathcal{A}(v)$. They are real and distinct provided that $H(q, t) > 0$. Mathematically, it implies the strict hyperbolicity property for the system of Equation (18).

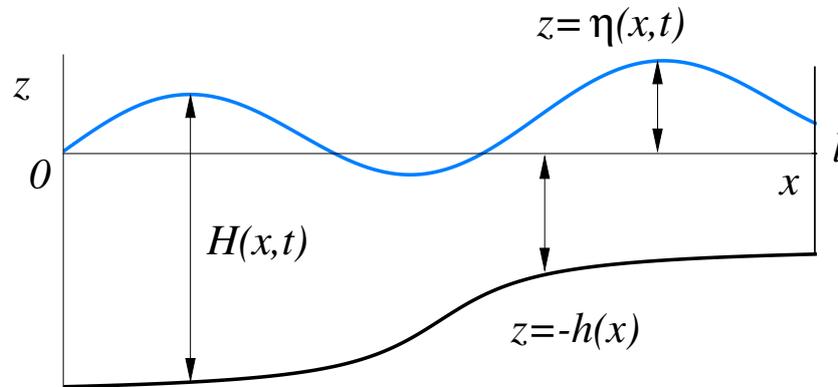


Figure 5. A sketch of the fluid domain in shallow water flows.

4.1. Finite Volume Discretization

The reference domain Q is discretized into N equal control volumes $[q_j, q_{j+1}]$ of the width $\Delta q \equiv \frac{1}{N}$. The nodes x_j^n are obtained as images of uniform nodes under the mapping $x(q, t)$:

$$x_j^n = x(q_j, t^n).$$

In the predictor–corrector scheme on the first step, we evaluate the flux vector in the cell centers:

$$\hat{f}_{j+1/2} = \frac{f_j^n + f_{j+1}^n}{2} - \frac{\tau}{2} \left[\frac{1}{j} \mathcal{R} \cdot \mathcal{D} \cdot \tilde{\mathcal{Q}} \cdot (\tilde{\mathcal{Q}} \cdot \mathcal{P} - \mathcal{L} \cdot \mathbf{G}) \right]_{j+1/2}^n, \quad \mathcal{P} := \mathcal{L} \cdot \mathbf{v}_q.$$

Vector \mathcal{P} is an analogue (The analogy is understood in the following sense: If we linearize our problem, the components of vector \mathcal{P} will provide *exactly* two Riemann’s invariants) of the Riemann’s invariants in the nonlinear case. The corrector step is based on the conservative form of the equation:

$$\frac{(J\mathbf{v})_j^{n+1} - (J\mathbf{v})_j^n}{\tau} + \frac{(\hat{f} - (x_t \mathbf{v})^n)_{j+1/2} - (\hat{f} - (x_t \mathbf{v})^n)_{j-1/2}}{\Delta q} = \mathbf{G}_j^*, \tag{19}$$

where τ is the time step and the following discretizations are used

$$\mathbf{G}_{j+1/2}^n = \begin{pmatrix} 0 \\ g(Hh_q)_{j+1/2}^n \end{pmatrix}, \quad \mathbf{G}_j^* = \begin{pmatrix} 0 \\ g(Hh_q)_j^{n+1/2} \end{pmatrix}, \quad \mathcal{D}_{j+1/2}^n = \begin{pmatrix} 1 + \theta_{j+1/2}^{1,n} & 0 \\ 0 & 1 + \theta_{j+1/2}^{2,n} \end{pmatrix}.$$

The total water depth H and local depth gradient h_q have to be computed in the following way:

$$H_j^{n+1/2} = \frac{H_{j+1}^{n+1} + H_{j-1}^{n+1} + H_{j+1}^n + H_{j-1}^n}{4}, \quad h_{q,j}^{n+1/2} = \frac{h_{j+1}^{n+1} - h_{j-1}^{n+1} + h_{j+1}^n - h_{j-1}^n}{4\Delta q}, \quad h_{q,j+1/2}^n = \frac{h_{j+1}^n - h_j^n}{\Delta q}.$$

Despite the fact that $H_{j\pm 1}^{n+1}$ is present in the formulas above, the scheme remains explicit, since we compute first $\{H_j^{n+1}\}_{j=1}^{N-1}$ from the mass conservation and, then, this value is used in the correction of the momentum conservation equation.

The matrix $\tilde{\mathcal{D}}$ plays the role of \bar{a} (the wave speed relative to the grid nodes) in the scalar case:

$$\tilde{\mathcal{D}} := \mathcal{D}_{j+1/2}^n - x_{t,j+1/2}^n \cdot \mathbb{I}, \quad \mathcal{D}_{j+1/2}^n = \begin{pmatrix} \lambda_{1,j+1/2}^n & 0 \\ 0 & \lambda_{2,j+1/2}^n \end{pmatrix}, \quad \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Here, $\lambda_{k,j+1/2}^n, k = 1, 2$ are eigenvalues of the Jacobian matrix $\mathcal{A}_{j+1/2}^n$:

$$\mathcal{A}_{j+1/2}^n = \begin{pmatrix} 0 & 1 \\ -v_j^n v_{j+1}^n + gH_{j+1/2}^n & 2v_{j+1/2}^n \end{pmatrix}, \quad (\lambda_{1,2})_{j+1/2}^n = (v \mp c)_{j+1/2}^n,$$

The local gravity wave speed $c_{j+1/2}^n$ can be computed analytically:

$$c_{j+1/2}^n = \sqrt{(v_{j+1/2}^n)^2 - v_j^n v_{j+1}^n + gH_{j+1/2}^n} \geq \sqrt{gH_{j+1/2}^n} > 0.$$

This choice of the Jacobian \mathcal{A} discretization is dictated by the requirement to preserve the compatibility condition $f_q = \mathcal{A}(v) v_q$ at the discrete level as well. It is analogous to a similar algebraic condition on the Jacobian in the Roe scheme [24] but substantially different in details.

Finally, the scheme parameter $\theta_{j+1/2}^k$ is computed according to the same strategy as above, with the only difference being that it is computed separately for each component $k = 1, 2$:

$$\theta_{j+1/2}^k = \begin{cases} 0, & \left| \tilde{g}_{j+1/2}^k \right| \leq \left| \tilde{g}_{j+1/2-s_k}^k \right| \quad \text{and} \quad \tilde{g}_{j+1/2}^k \times \tilde{g}_{j+1/2-s_k}^k \geq 0, \\ \theta_{0,j+1/2}^k (1 - \tilde{\zeta}_{j+1/2}^k), & \left| \tilde{g}_{j+1/2}^k \right| > \left| \tilde{g}_{j+1/2-s_k}^k \right| \quad \text{and} \quad \tilde{g}_{j+1/2}^k \times \tilde{g}_{j+1/2-s_k}^k \geq 0, \\ \theta_{0,j+1/2}^k, & \tilde{g}_{j+1/2}^k \times \tilde{g}_{j+1/2-s_k}^k < 0, \end{cases} \quad (20)$$

where $\tilde{\zeta}_{j+1/2}^k := \frac{\tilde{g}_{j+1/2-s_k}^k}{\tilde{g}_{j+1/2}^k}, s_k \equiv s_{k,j+1/2}^n := \text{sign}(\bar{\lambda}_{k,j+1/2}^n)$. Here, $\bar{\lambda}_{k,j+1/2}^n = (\lambda_k - x_t)_{j+1/2}^n$ is simply the diagonal elements of the matrix $\tilde{\mathcal{D}}$. The coefficient $\theta_{0,j+1/2}^k$ is related to the local CFL number as

$$\theta_{0,j+1/2}^k := \frac{1}{C_{k,j+1/2}^n} - 1, \quad C_{k,j+1/2}^n := \frac{\tau}{\Delta q} \left(\frac{|\bar{\lambda}_k|}{J} \right)_{j+1/2}^n.$$

Finally, we have to specify how to compute the indicators

$$\tilde{g}_{j+1/2}^k := (|\bar{\lambda}_k| (1 - C_k) \tilde{p}^k)_{j+1/2}^n,$$

where $\tilde{p}^k, k = 1, 2$ are components of the following vector:

$$\tilde{\mathcal{P}}_{j+1/2}^n := \frac{1}{(c_{j+1/2}^n)^2} \begin{pmatrix} -c \eta_q + H v_q \\ c \eta_q + H v_q \end{pmatrix}_{j+1/2}^n. \quad (21)$$

The fully discrete scheme is stable under the following CFL-type restriction [66] on the time step τ :

$$\max_{k,j} \{ C_{k,j+1/2}^n \} \leq 1.$$

Remark 2. We would like to explain better the motivation behind the choice of the vector $\tilde{\mathcal{P}}$ (cf. Equation (21)) in our scheme. This vector appears in the choice of the scheme parameter of Equation (20), of which the role consists in ensuring the monotonicity of the Riemann’s invariants. First of all, let us have a look at the components of vector \mathcal{P} :

$$\mathcal{P}_{j+1/2}^n = (\mathcal{L} \cdot \mathbf{v}_q)_{j+1/2}^n = \underbrace{\frac{1}{(c_{j+1/2}^n)^2} \left(\begin{matrix} -c \eta_q + H v_q \\ c \eta_q + H v_q \end{matrix} \right)_{j+1/2}^n}_{\equiv \tilde{\mathcal{P}}_{j+1/2}^n} + \left(\frac{h_q}{c} \right)_{j+1/2}^n \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

We can see that vector \mathcal{P} is a sum of two vectors: The first addend depends on the solution gradient (η_q, v_q) , and the second one depends on the bathymetry gradient h_q . We would like that the scheme parameters $\theta_{j+1/2}^k$ depend on the solution and not on the bathymetry. Thus, we made a choice to include only the first addend in the vector $\tilde{\mathcal{P}}$.

4.1.1. Well-Balanced Property

The following property can be shown for the scheme presented above:

Theorem 1. Assume that we have a fixed but possibly nonuniform grid $\{x_j^n\}$. If on the n^{th} time layer we have the lake at rest state, i.e.,

$$\eta_j^n \equiv 0, \quad v_j^n \equiv 0, \tag{22}$$

then the predictor–corrector scheme described above will preserve this state at the $(n + 1)^{\text{th}}$ time layer.

Proof. Since $x_{t,j} \equiv 0, \forall j$ and taking into account the definition of Equation (22) of the lake at rest state, we have

$$(\tilde{\mathcal{P}} \times \mathcal{P} - \mathcal{L} \times \mathbf{G})_{j+1/2}^n \equiv 0.$$

Consequently, the predictor step leads to

$$\hat{f}_{j+1/2} = \frac{f_j^n + f_{j+1}^n}{2}.$$

Then, from the obvious identities $h_j^{n+1} \equiv h_j^n$ and $J_j^{n+1} \equiv J_j^n$ and the way we compute \mathbf{G}_j^* , it follows that $\eta_j^{n+1} \equiv 0, v_j^{n+1} \equiv 0$. Thus, the lake at the rest state is preserved at the next time layer as well. \square

The well-balanced property is absolutely crucial to compute qualitatively correct numerical solutions to conservation (balance) laws [88]. Theorem 1 shows that the discretization proposed above does possess this property regardless of some complications coming from the grid motion.

4.1.2. Conservation Property on Moving Grids

For conservation laws with source terms (the so-called *balance laws*), the conservation property of a numerical scheme is understood in the sense that the *fully* discrete solution verifies some thoroughly chosen finite difference analogues of continuous balance laws. For the NSWE, we have the conservation of mass and the balance of a horizontal (depth-averaged) momentum. Please note that, on a flat bottom, the balance of momentum becomes a conservation law as well. The conservation property is crucial as well to compute qualitatively correct numerical solutions [88].

On moving grids, we have an additional difficulty: The computation of the speeds of mesh nodes has to be consistent with the calculation of the cell volumes. If it is done in the right way, the discrete geometric conservation law will be verified as well [71]. In 1-D, it can be written as

$$J_j^{n+1} = J_j^n + \tau x_{t,q,j}^n. \tag{23}$$

We shall show below that other schemes can be put into the conservative form similar to Equation (19). The peculiarity here consists in the fact that the Jacobian matrix \mathcal{A} is nonconstant in contrast to the linear advection equation for which any reasonable scheme is conservative (see also Remark 3 below about a nonlinear scalar equation). Moreover, the moving mesh is another

complication, which makes this task rather nontrivial. For example, the scheme parameter $\theta_{j\pm 1/2}^k$ can be chosen as

$$\theta_{j\pm 1/2}^k = \theta_{0,j\pm 1/2}^k, \quad k = 1, 2.$$

This choice corresponds to the first-order upwind scheme (as it was discussed above in Section 2.1). So, the scheme of Equation (19) can be rewritten as

$$\begin{aligned} & \frac{(Jv)_j^{n+1} - (Jv)_j^n}{\tau} + \\ & \frac{1}{h} \left[\left(\frac{f_{j+1}^n + f_j^n}{2} - \frac{(\mathcal{A}^+ - \mathcal{A}^-)_{j+1/2}^n}{2} (v_{j+1}^n - v_j^n) - (x_t v)_{j+1/2}^n + \frac{h}{2} (\mathcal{RSLG})_{j+1/2}^n \right) - \right. \\ & \left. \left(\frac{f_j^n + f_{j-1}^n}{2} - \frac{(\mathcal{A}^+ - \mathcal{A}^-)_{j-1/2}^n}{2} (v_j^n - v_{j-1}^n) - (x_t v)_{j-1/2}^n + \frac{h}{2} (\mathcal{RSLG})_{j-1/2}^n \right) \right] = \mathbf{G}_j^*, \end{aligned}$$

where $\mathcal{A}^\pm = \mathcal{R}\bar{\Lambda}^\pm \mathcal{L}$, $\bar{\lambda}_k^\pm = \frac{1}{2}(\bar{\lambda}_k \pm |\bar{\lambda}_k|)$ and

$$\mathcal{S} = \begin{pmatrix} \text{sign}(\lambda_1) & 0 \\ 0 & \text{sign}(\lambda_2) \end{pmatrix}, \quad \bar{\Lambda}^\pm = \begin{pmatrix} \bar{\lambda}_1^\pm & 0 \\ 0 & \bar{\lambda}_2^\pm \end{pmatrix}.$$

One can see also that the following identities hold:

$$\bar{\Lambda} \equiv \bar{\Lambda}^+ + \bar{\Lambda}^-, \quad \mathcal{A}^+ + \mathcal{A}^- \equiv \mathcal{A} \equiv \mathcal{A} - x_t \mathbb{I}.$$

By using these identities and the discrete geometric conservation law of Equation (23), one can derive also the following equivalent form of the scheme from Equation (19):

$$\begin{aligned} & \frac{v_j^{n+1} - v_j^n}{\tau} + \frac{1}{J_j^{n+1}} \left[(\mathcal{A}^+ v_q)_{j-1/2}^n + (\mathcal{A}^- v_q)_{j+1/2}^n \right] = \\ & \frac{1}{J_j^{n+1}} \left[\mathbf{G}_j^* - \frac{1}{2} ((\mathcal{RSLG})_{j+1/2}^n - (\mathcal{RSLG})_{j-1/2}^n) \right]. \end{aligned}$$

The choice of Equation (20) of the scheme parameter $\theta_{j\pm 1/2}^k$ leads to a conservative scheme as well, but the underlying computations (to show it) are less elegant.

Remark 3. In order to illustrate even better the interplay between conservative and nonconservative forms, let us take again the inviscid Burgers–Hopf equation [69]:

$$u_t + \left(\frac{u^2}{2} \right)_x = 0.$$

For the sake of simplicity, let us discretize this equation on a uniform grid using the following nonconservative scheme:

$$\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{u_j^n + u_{j-1}^n}{2} \times \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0.$$

It is not difficult to see that this nonconservative scheme is equivalent to the following conservative upwind scheme:

$$\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{1}{\Delta x} \left(\frac{(u_j^n)^2}{2} - \frac{(u_{j-1}^n)^2}{2} \right) = 0.$$

However, if one takes a different nonconservative scheme (The difference is in the estimation of propagation speed, cf. u_j^n vs. $\frac{1}{2}(u_j^n + u_{j-1}^n)$)

$$\frac{u_j^{n+1} - u_j^n}{\tau} + u_j^n \times \frac{u_j^n - u_{j-1}^n}{\Delta x} = 0, \tag{24}$$

it is straightforward to see that one cannot transform it into a conservative form. Moreover, one can even show that the scheme of Equation (24) is not converging. Therefore, the conservation and convergence properties are intrinsically related.

4.2. Numerical Results

In this section, we present two test cases of complex wave propagation in shallow water equations. The first test case is a validation against an implicit analytic solution, derived below specifically for this purpose, while the second test case is related to real-world applications.

4.2.1. Exact Solution Derivation

In order to validate the numerical scheme, we derive an exact solution to nonlinear shallow water equations which will serve as a reference solution. We already saw this particular solution in the literature [89]; however, the derivation procedure is published for the first time to our knowledge.

First of all, we assume that the bottom is flat, i.e., $h(x) = h_0 = \text{const} > 0$, and we consider an initial value problem on the real line:

$$\eta(x, 0) = \eta_0(x), \quad u(x, 0) = u_0(x).$$

The nonlinear shallow water equations are rewritten using the Riemann invariants [90]:

$$r_t + \frac{3r + s}{4} r_x = 0, \quad s_t + \frac{r + 3s}{4} s_x = 0,$$

where

$$r := u - 2c, \quad s := u + 2c, \quad c := \sqrt{gH}.$$

We seek for a particular solution in the form of an r -wave propagating in the leftward direction. In other words, we assume $s = s_0 = \text{const}$. Assuming that, at infinity, the water is at rest, we deduce that $s_0 = 2c_0 = 2\sqrt{gh_0}$. Then, from the formula $u + 2c = s_0$, we deduce that the speed $u(x, t)$ is related to the total depth:

$$u(x, t) = s_0 - 2c = 2c_0 - 2\sqrt{gH(x, t)}. \tag{25}$$

Consequently, taking the limit $t \rightarrow 0$, we have a similar relation between the initial conditions:

$$u_0(x) = 2c_0 - 2\sqrt{g(\eta_0(x) + h_0)}.$$

If we introduce a new variable

$$p(x, t) := \frac{3r(x, t) + s_0}{4},$$

the governing equation for the r -invariant becomes simply the Burgers–Hopf equation:

$$p_t + p p_x = 0.$$

The solution can be obtained using the method of characteristics by the following implicit formula [21,70]:

$$p(x, t) = p_0(x - p(x, t)t),$$

where $p_0(x)$ is the initial condition

$$p_0(x) = \frac{3r(x,0) + s_0}{4} = 2c_0 - 3\sqrt{g(\eta_0(x) + h_0)}.$$

Consequently, the solution $p(x, t)$ can be obtained at any space location x and time instance t by solving the following implicit equation

$$p(x, t) = 2c_0 - 3\sqrt{g(\eta_0(x - p(x, t)t) + h_0)}. \tag{26}$$

Once, $p(x, t)$ is found, the free surface elevation $\eta(x, t)$ can be reconstructed by using this formula

$$\eta(x, t) = \left(\frac{2c_0 - p(x, t)}{3\sqrt{g}} \right)^2 - h_0, \tag{27}$$

and the flow speed $u(x, t)$ from Equation (25).

Remark 4. Please note that this method works only until the gradient catastrophe occurs at some time instance $t = t^*$, which depends on the initial condition.

Remark 5. In order to guarantee the solvability of Equation (26), it is enough to check that the following function

$$f(p; x, t) = p + 3\sqrt{g(\eta_0(x - pt) + h_0)} - 2c_0$$

is monotonic in p for $0 < t < t^*$.

4.2.2. Validation

Using the method described above, we obtain the analytical solution for the following initial value problem:

$$\eta_0(x) = \begin{cases} \frac{a}{2} \left[1 + \cos\left(\frac{2\pi(x - x_w)}{\lambda}\right) \right], & |x - x_w| \leq \lambda/2, \\ 0, & |x - x_w| > \lambda/2, \end{cases}$$

where x_w is the wave crest position and a is the wave amplitude. For $0 < t < t^*$, the profile $\eta_0(x)$ will propagate to the left with the speed c_0 preserving its wavelength λ and amplitude a . However, the gradient catastrophe will inevitably take place since the characteristics $\frac{dx}{dt} = p_0(x)$ departing from the interval $(x_w - \lambda/2, x_w)$ form a converging beam. On the other hand, the characteristics from $(x_w, x_w + \lambda/2)$ diverges. Thus, the solution looks like a compression wave followed by a rarefaction wave. With the parameters specified in Table 4, the gradient catastrophe takes place at about $t = t^* \approx 5.57$. The numerical results are shown in Figure 6 right before the gradient catastrophe occurs. The adaptive grid was constructed using the following monitor function, which contains an additional term with respect to Equation (11)

$$w(x, t) = 1 + \alpha_0 |\eta(x, t)| + \alpha_1 |\eta_x(x, t)|, \tag{28}$$

which is discretized as

$$w_{j+1/2}^n = 1 + \alpha_0 \left| \eta_{j+1/2}^n \right| + \alpha_1 \frac{\left| \eta_{j+1}^n - \eta_j^n \right|}{x_{j+1}^n - x_j^n}. \tag{29}$$

The thin solid line in Figure 6 represents the exact solution computed by solving Equation (26) as it was explained in the previous section. The agreement between these two profiles is exemplary. On the right panel, one can see how the nodes follow the wave propagation and how their density increases around the steep gradient. On the other hand, the use of a fixed grid leads to a noticeable smearing of the numerical solution.

Table 4. The numerical parameters used in validations of the nonlinear shallow water equations solver.

Parameter	Value
Computational domain length, ℓ	40.0
Number of nodes, N	100
Final simulation time, T_f	5.0
CFL number, C	0.95
Gravity acceleration, g	9.81
Initial wave amplitude, a	0.2
Wave crest initial position, x_w	30.0
Initial wavelength, λ	10.0
Undisturbed water depth, h_0	1.0
Monitor function parameter, α_0	10.0
Monitor function parameter, α_1	10.0
Grid diffusion parameter, β	5.0
Smoothing parameter, σ	5.0

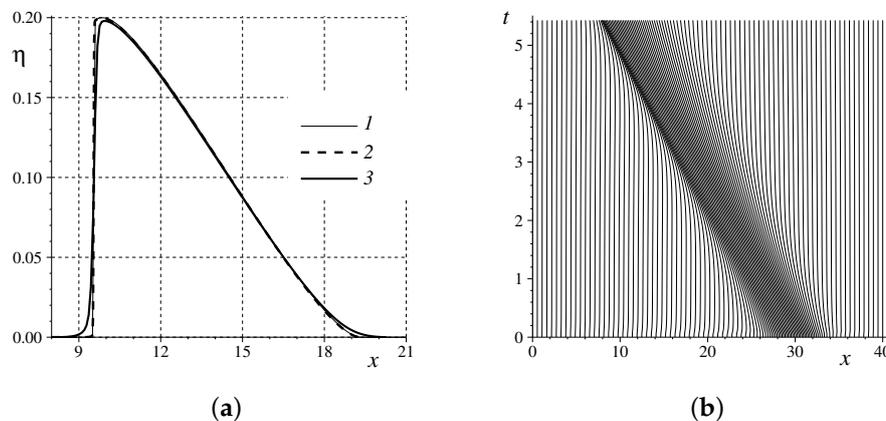


Figure 6. The shock wave formation in nonlinear shallow water equations: (a) A comparison of different profiles (The thin solid line (1) is the exact solution, (2) is the numerical solution on the moving mesh, and (3) is the solution on the fixed uniform mesh); (b) the trajectories of grid nodes in space–time. Notice that Figure 6a shows only a zoom of the computational domain.

The exact solution presented in the previous Section 4.2.1 can be used in order to study the absolute accuracy of proposed numerical methods. To this end, we introduce two discrete norms:

$$\|f_h - f_{\text{ref}}\|_{\infty} := \max_{1 \leq j \leq N-1} |f_j^n - f(x_j, T_f)|,$$

$$\|f_h - f_{\text{ref}}\|_2 := \left(\sum_{j=1}^{N-1} (f_j^n - f(x_j, T_f))^2 \cdot \frac{x_{j+1}^n - x_{j-1}^n}{2} \right)^2,$$

where n is the time step number corresponding to the final simulation time $t = T_f$. Here, f denotes the free surface elevation η or the depth-averaged horizontal velocity u . The reference solution for u and η is provided by Equations (25) and (27), correspondingly. We would like to mention that the above definition of the norm $\|\cdot\|_2$ is valid for the fixed and moving grids. The computational results are reported for the fixed and moving grids in Tables 5 and 6 respectively. All the parameters in these simulations were fixed and taken from Table 4 (except the parameter N , which was varied). The second order convergence can be observed in Table 5. In general, the results obtained on a moving grid are more accurate than those on a fixed one as it follows from the comparison of corresponding cells in Tables 5 and 6. In the last column of these tables, we report also the CPU times measured on a computer equipped with an Intel Pentium CPU 3.2 Ghz and 1Gb of RAM. The code was implemented in Fortran and compiled with Compaq Visual Fortran compiler (version 6.6c). In particular, for the

finest grid ($N = 6400$), the overhead related to the grid motion represents less than 7% of the overall CPU time and the error (measured in the most stringent norm $\|\cdot\|_\infty$) is reduced by a factor of four. Thus, we may conclude that the application of moving grid methods is very interesting from the practical point of view.

We underline also the fact that even better results could be obtained if one optimizes the monitor function parameters $\alpha_{0,1}$ along with mesh motion parameters β and σ . In our numerical simulations, we just took some *reasonable* values. At the current stage of the development of these methods, it is difficult to say something on the optimal values of these parameters a priori. Finally, we would like to mention that this test case is relatively difficult for both the analytical and numerical methods, since we deal with a nonlinear system of hyperbolic equations and at the final time $t = T_f$ the wave has almost developed a discontinuity (from a smooth initial condition). Thus, we operate at the edge of validity of the analytical solutions of Equations (25) and (27).

Table 5. The numerical solution errors measured at the final simulation time $t = T_f$ on a fixed grid. The reference solution is given by Equations (25) and (27). All numerical parameters for this simulation are reported in Table 4.

N	η		u		CPU Time, s
	$\ \eta_h - \eta_{\text{ref}}\ _\infty$	$\ \eta_h - \eta_{\text{ref}}\ _2$	$\ u_h - u_{\text{ref}}\ _\infty$	$\ u_h - u_{\text{ref}}\ _2$	
100	0.388×10^{-1}	0.289×10^{-1}	0.363×10^{-1}	0.270×10^{-1}	$< 10^{-2}$
200	0.245×10^{-1}	0.127×10^{-1}	0.229×10^{-1}	0.118×10^{-1}	0.03
400	0.104×10^{-1}	0.451×10^{-2}	0.977×10^{-2}	0.424×10^{-2}	0.20
800	0.418×10^{-2}	0.144×10^{-2}	0.394×10^{-2}	0.136×10^{-2}	1.36
1600	0.128×10^{-2}	0.393×10^{-3}	0.122×10^{-2}	0.373×10^{-3}	9.05
3200	0.353×10^{-3}	0.103×10^{-3}	0.337×10^{-3}	0.981×10^{-4}	71.08
6400	0.885×10^{-4}	0.257×10^{-4}	0.843×10^{-4}	0.245×10^{-4}	573.11

Table 6. The numerical solution errors measured at the final simulation time $t = T_f$ on a moving grid. The reference solution is given by Equations (25) and (27). All numerical parameters for this simulation are reported in Table 4.

N	η		u		CPU Time, s
	$\ \eta_h - \eta_{\text{ref}}\ _\infty$	$\ \eta_h - \eta_{\text{ref}}\ _2$	$\ u_h - u_{\text{ref}}\ _\infty$	$\ u_h - u_{\text{ref}}\ _2$	
100	0.125×10^{-1}	0.907×10^{-2}	0.123×10^{-1}	0.900×10^{-2}	0.03
200	0.591×10^{-2}	0.415×10^{-2}	0.586×10^{-2}	0.413×10^{-2}	0.20
400	0.266×10^{-2}	0.170×10^{-2}	0.264×10^{-2}	0.170×10^{-2}	1.13
800	0.897×10^{-3}	0.608×10^{-3}	0.894×10^{-3}	0.606×10^{-3}	5.76
1600	0.317×10^{-3}	0.215×10^{-3}	0.316×10^{-3}	0.214×10^{-3}	27.66
3200	0.881×10^{-4}	0.566×10^{-4}	0.878×10^{-4}	0.563×10^{-4}	130.00
6400	0.221×10^{-4}	0.142×10^{-4}	0.220×10^{-4}	0.142×10^{-4}	612.60

4.2.3. Solitary Wave Run-Up

As the last test case, we consider a practically important problem of a solitary wave run-up on a sloping beach. The main difficulty here consists in the fact that the fluid domain may vary in time and its evolution is not known a priori. It has to be determined dynamically during the simulation. The wetting/drying algorithm was described in detail in our previous study [91]. Here, we focus mainly on the mesh motion algorithm behaviour.

Consider a closed 1-D channel with the bathymetry prescribed by the following function:

$$z = -h(x) = \begin{cases} h_0 - x \tan \theta & 0 \leq x \leq x_s, \\ -h_0, & x_s \leq x \leq \ell, \end{cases}$$

where θ is the sloping beach angle. The variable bathymetry region is located leftwards from $x_s = 2 \cot \theta \approx 38.16$. At the initial time, the shoreline is located at $x_{00} = \cot \theta \approx 19.08$.

The initial condition is a solitary wave prescribed by the following formulas (which come from the analytical solitary wave solution to the Serre equations, see Reference [92], for example):

$$\eta(x, 0) = a \operatorname{sech}^2 \left[\frac{\sqrt{3ga}}{2h_0\sqrt{g(h_0+a)}} (x - x_{0,w}) \right],$$

$$u(x, 0) = -\sqrt{g(h_0+a)} \frac{\eta(x, 0)}{h_0 + \eta(x, 0)},$$

Variables a and $x_{0,w} = x_s + 30 \approx 68.16$ are the initial solitary wave amplitude and position, correspondingly. The right boundary is located at $\ell = x_{0,w} + 30 \approx 98.16$. All numerical parameters used in this experiment are provided in Table 7.

The considered situation is complex enough so that the analytical solution is not available. However, we can simulate it numerically with the predictor–corrector scheme on the moving grids presented above. The wave run-up process simulated in silico is depicted in Figure 7a, while the nodes motion is illustrated in the right panel Figure 7b. Please note that only every fifth node is depicted in Figure 7b in order to improve the visibility of the image. The rightmost node is fixed (i.e., its trajectory is a straight line), while the leftmost node’s trajectory coincides with the shoreline motion. One can see that, even on domains with variable spatial extent, the mesh motion algorithm is robust enough and it puts the resolution where it is needed.

The monitor function used in this computation is

$$w(x, t) = 1 + \alpha |\eta(x, t)|. \tag{30}$$

It is important to use the wave elevation $\eta(x, t)$ instead of the total water depth $H(x, t)$ because of solutions such as the lake at rest state. In these situations, the total depth H varies in space, while the free surface elevation $\eta(x, t) \equiv 0$. Thus, the usage of the monitor function of Equation (30) leads to the uniform grid (since $w(x, t) \equiv 1$) and, trivially, to the well-balanced property shown in Theorem 1.

Table 7. The numerical parameters used in a solitary wave run-up simulation in nonlinear shallow water equations.

Parameter	Value
Computational domain length, ℓ	98.16
Number of nodes, N	1000
Final simulation time, T_f	30.0
CFL number, C	0.95
Gravity acceleration, g	9.81
Initial wave amplitude, a/h_0	0.05
Wave crest initial position, $x_{0,w}$	68.16
Beach slope position, x_s	38.16
Beach slope, θ	3°
Undisturbed water depth, h_0	1.0
Monitor function parameter, α	60.0
Grid diffusion parameter, β	20.0
Smoothing parameter, σ	10.0

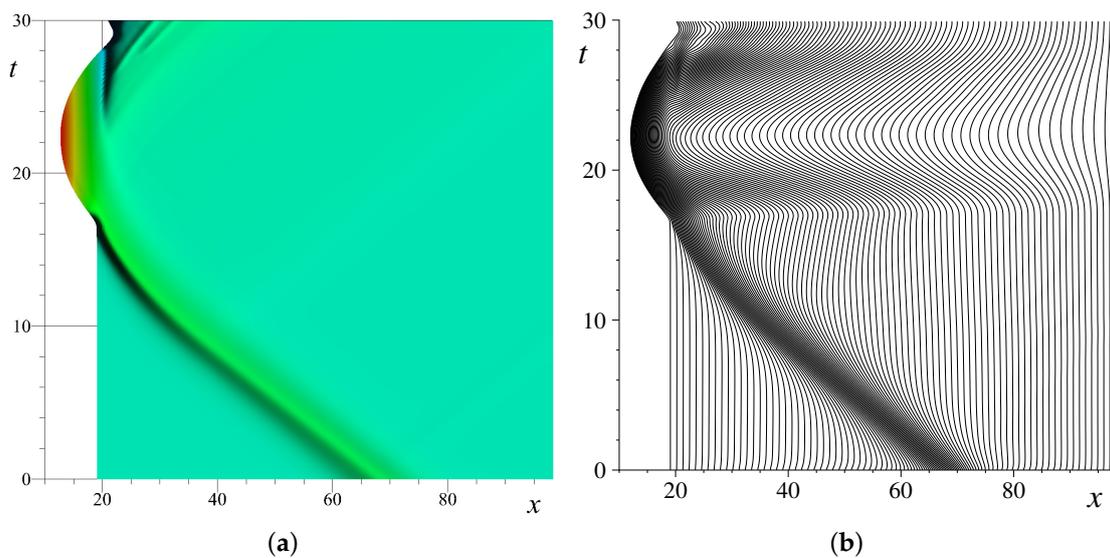


Figure 7. The solitary wave run-up simulation on a sloping beach with nonlinear shallow water equations: (a) The free surface elevation evolution and (b) the trajectories of grid nodes in space–time.

Some further validation tests and numerical experiments with this moving grid technique were presented in our preceding work [91] (without giving many details about the algorithm in use).

5. Some Open Problems

Only in very simple problems (linear and steady) one can study theoretically the choice of the optimal monitor function. Recently, we performed such an analysis for a singularly perturbed Sturm–Liouville problem [45]. In particular, we showed that a judicious choice of the monitor function allows an increase in the order of convergence from the 2nd to the 4th order (the so-called *supraconvergence* phenomenon), even if the scheme is just 2nd-order accurate on uniform meshes. Therefore, the first open problem consists in providing a similar analysis for, at least, the 1-D unsteady or nonlinear problems. The method employed in Reference [45] is suitable for smooth solutions. For discontinuous ones, a different technique has to be proposed. Even a linear transport equation is not completely understood from the perspective of constructing a suitable moving grid. It is even possible that, for every discretization scheme, there is its own “optimal” monitor function. Therefore, one choice clearly does not fit all possible schemes. Moreover, even if we choose the monitor function as in Equation (30), it was shown in Reference [45] that the error may depend non-monotonically on the coefficient α . Thus, even if the monitor function form is fixed, one has to set optimal values of coefficients in this function.

6. Discussion

The main conclusions and perspectives of this study are outlined below.

6.1. Conclusions

In the present work, our goal was to present a moving grid methodology that does not involve any interpolation procedure (thus avoiding the unnecessary smearing of the numerical solution). We showed some reasonable choices of the monitor function, which are currently used in practice, but these choices have to be refined further in the forthcoming works. The current state-of-the-art in this field is described in Reference [41].

In this manuscript, we presented a detailed description of a family of predictor–corrector schemes up to the second order in space, which were generalized to the case of nonuniform moving meshes, while having all these properties *simultaneously*:

- Conservation in space;
- Second-order accuracy (in smooth regions);
- TVD property for scalar problems;
- A well-balanced character;
- Preservation of the discrete conservation law in cell centers and cell interfaces;
- Adaptivity in space and in time (The time step is chosen the largest possible while meeting the stability requirement);
- No interpolation between the meshes; thus, we do not add any unnecessary numerical diffusion;
- Implementation is trivial, despite all good properties listed above.

This is the main novelty and contribution of our study to the growing field of adaptive numerical methods. This method was illustrated on a number of relatively serious tests coming from linear, nonlinear, and vectorial cases. Every time, the gain of using a moving grid (with respect to the fixed one) was clearly visible. Consequently, we can only encourage the community to use these (or similar) adaptive strategies to improve the quality and efficiency of their numerical simulations. In particular, the field of tsunami wave modelling could greatly benefit from the moving grid nodes to improve the accuracy and computational efficiency of operational tsunami wave predictions.

6.2. Perspectives

The adaptive moving mesh technology presented in this study relies on the choice of the monitor function along with several free real parameters. Numerical experiments show that the method performance may depend crucially on the choice of these real numbers. Consequently, it would be highly desirable to obtain some theoretical estimations of the optimal choice of these parameters at least in the case of the simplest linear advection Equation (1). It constitutes one of the main challenges raised by our study. In the absence of such theoretical indications, one might envisage extensive numerical studies in order to issue some practical recommendations on how to choose reasonable values of parameters. Most probably, the case of nonlinear systems of conservation laws will remain essentially empiric in the foreseeable future due to the important theoretical difficulties in its analysis.

From a more practical point of view, the predictor-corrector schemes *on moving grids* presented in this study have to be generalized at least to a two-dimensional case as it was done for the Godunov scheme in Reference [52]. We already reported an implementation of the moving grid technique in two space dimensions for an elliptic problem in Reference [58]. Moreover, the spatial adaptivity has to be coupled with time-adaptive strategies such as the PI step size control [93] in order to achieve the full adaptivity.

Finally, there is another goal behind this study. Ultimately, we would like to generalize moving grid methods to some dispersive wave equations [92], which are going to be addressed with the operator splitting approach along the lines sketched in Reference [46]. In this way, the hyperbolic part would be addressed with the methods outlined above. The particularity of dispersive wave equations is that they possess localised solitary wave solutions. In other words, all the dynamics are concentrated in small portions of space, which move perpetually. Consequently, the grid redistribution techniques seem to be the natural choice to simulate complex solitonic gas dynamics [94].

Author Contributions: All authors contributed equally to this study.

Funding: This research was supported by RSCF project No. 14-17-00219. D. Dutykh acknowledges the support of the CNRS under the PEPS InPhyNiTi project FARA and project No. EDC26179 “Wave interaction with an obstacle”, as well as the hospitality of the Institute of Computational Technologies of SB RAS during his visit in October 2015. D. Mitsotakis was supported by the Marsden Fund administered by the Royal Society of New Zealand.

Acknowledgments: We would like to thank Laurent Gosse (CNR, Italy) for the insightful discussions on the topics covered in our study. Moreover, we would like to thank the anonymous Referee on our previous manuscript [91] who incited us by his comments to produce a study which explains in more details our grid motion technique.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Illustrative Examples

Let us consider the “unsteady” version of the equidistribution principle expressed as the following nonlinear elliptic Boundary Value Problem (BVP):

$$(w(x, t) x_q)_q = 0, \quad q \in [0, 1], \quad 0 \leq t \leq T_f, \tag{A1}$$

with the following boundary conditions:

$$x(0, t) = 0, \quad x(1, t) = \ell \quad 0 \leq t \leq T_f. \tag{A2}$$

The first inherent problem with Equation (A1) is that small perturbations of the monitor function $w(x, t)$ may lead to finite changes in the solution $x(q, t)$ as it is illustrated in the following example.

Appendix A.1. Example 1

Let, for $t = t_n$, the monitor function w be constant, i.e., $w(x, t_n) \equiv 1$. Then, the solution to the BVPs of Equations (A1) and (A2) take the following simple form:

$$x(q, t_n) = q\ell, \quad q \in [0, 1]. \tag{A3}$$

This solution prescribes the uniform distribution of nodes on $[0, \ell]$. Let us also assume that, at the next time step $t = t_{n+1}$, the monitor function w was perturbed with quantity ε on a portion of the segment $[0, \ell]$:

$$w(x, t_{n+1}) = \begin{cases} 1 + \varepsilon, & x \leq x_*, \\ 1, & x > x_*, \end{cases} \quad x_* \in [0, \ell].$$

Then, the solution to the BVPs of Equations (A1) and (A2) will be given by the following piecewise linear function:

$$x(q, t_{n+1}) = \begin{cases} q \frac{x_*}{q_*}, & q \leq q_*, \\ x_* + (q - q_*) \frac{\ell - x_*}{1 - q_*}, & q > q_*, \end{cases} \tag{A4}$$

where $q_* = \frac{(1 + \varepsilon)x_*}{\ell + \varepsilon x_*}$. From here, the following estimation follows:

$$\|x(\cdot, t_{n+1}) - x(\cdot, t_n)\|_{C(Q)} = |x(q_*, t_{n+1}) - x(q_*, t_n)| = \frac{\varepsilon(\ell - x_*)x_*}{\ell + \varepsilon x_*}.$$

It is not difficult to check that this norm of the difference takes the maximal value for $x_*^{\max} = \frac{\ell(\sqrt{1 + \varepsilon} - 1)}{\varepsilon}$. For this given x_*^{\max} and for small values of ε , we have

$$\|x(\cdot, t_{n+1}) - x(\cdot, t_n)\|_{C(Q)} = \frac{\ell(2 + \varepsilon - 2\sqrt{1 + \varepsilon})}{\varepsilon} = \frac{\varepsilon\ell}{4} + \mathcal{O}(\varepsilon^2).$$

Hence, by choosing the domain length ℓ sufficiently large, we can obtain a finite (To obtain truly finite perturbations in $x(q, t)$, it is sufficient to take $\ell = \mathcal{O}(\varepsilon^{-1})$) perturbation in the solution to the BVPs of Equations (A1) and (A2).

In numerical practice, it means that the grid nodes in the vicinity of the point $x = x_*^{\max}$ will undergo a shift proportional to $\propto \varepsilon\ell$ when we advance the solution from the time layer $t = t_n$ to $t = t_{n+1}$. If, for some reason, on the time layer $t = t_{n+1}$ the monitor function w becomes constant (In this case, the monitor function may assume two possible constant values $w(x, t) \equiv 1$ or $w(x, t) \equiv 1 + \varepsilon$) again, the grid nodes will experience an opposite move towards a uniform grid as

in Equation (A3) again. Such oscillations have been observed numerically, and the present example provides a theoretical explanation of this phenomenon.

Appendix A.2. Example 2

In order to complete the description of possible shortcomings of the equidistribution principle for the mesh motion in evolutionary PDEs, in this section, we consider a simple unsteady problem where the scenario described in Example 1 is actually realized.

Consider the linear advection equation:

$$u_t + a u_x = f(x, t), \quad a = \text{const} > 0,$$

where the source term is chosen to be

$$f(x, t) = \begin{cases} \varepsilon(x_* - x - at), & x \leq x_*, \\ 0 & x > x_*. \end{cases}$$

In this section, ε is a small parameter and $x_* \in]0, \ell[$. An exact solution to the homogeneous Initial-Boundary Value Problem (IBVP) in domain $[0, \ell] \setminus \{x_*\}$ is given by the following formula:

$$u(x, t) = \begin{cases} \varepsilon t(x_* - x), & x < x_*, \\ 0, & x > x_*. \end{cases}$$

Let us choose the monitor function as

$$w(x, t) = 1 + \alpha |u_x(x, t)|$$

and take the parameter $\alpha \equiv 1$ for simplicity. Then, it is not difficult to check that

$$w(x, t) = \begin{cases} 1 + \varepsilon t, & x < x_*, \\ 1, & x > x_*. \end{cases}$$

The nonuniform grid given by the equidistribution principle is obtained as a solution to the BVPs of Equations (A1) and (A2). In this example, it takes precisely the form of Equation (A4), but the point q_* depends on time:

$$q_* = \frac{(1 + \varepsilon t)x_*}{\ell + \varepsilon t x_*}.$$

It is not difficult to show that the maximal speed of grid nodes $-x_t$ is achieved at $t = 0$ and $x_* = \frac{\ell}{2}$. Moreover, $\max\{-x_t\} = \frac{\varepsilon\ell}{4}$. The grid at $t = 0$ is uniform, and at the first time step, it will experience a brutal shift to the left. The maximal displacement being equal to $\frac{\varepsilon\ell}{4}$. This shift may provoke, at least, a loss of accuracy or, in the worst case scenario, a blowup of the numerical solution. The latter may routinely happen in some explicit schemes, where the time step does not take into account the grid nodes displacements.

Henceforth, we have just demonstrated analytically that the grid constructed with the equidistribution principle of Equation (A1) turns out to be unstable when the equidistribution principle is applied in unsteady problems. Indeed, arbitrary small perturbations in the monitor function or in the governing equation may lead to finite perturbations in the grid. Thus, we highly recommend the use of the unsteady parabolic Equation (15) in order to move grid nodes smoothly.

Appendix B. Abbreviations

- 1D One-dimensional
- 2D Two-dimensional

BVP Boundary Value Problem
CFL Courant–Friedrichs–Lewy
DART Deep-ocean Assessment and Reporting Tsunamis
FEM Finite Elements Method
FNWD Fully Nonlinear Weakly Dispersive
IBVP Initial-Boundary Value Problem
MOST Method Of Splitting Tsunami
NSWE Nonlinear Shallow Water Equations
PDE Partial Differential Equation
TVD Total Variation Diminishing
WNWD Weakly Nonlinear Weakly Dispersive

References

1. Imamura, F. Simulation of wave-packet propagation along sloping beach by TUNAMI-code. In *Long-Wave Runup Models*; Yeh, H., Liu, P.L.-F., Synolakis, C.E., Eds.; World Scientific: Singapore, 1996; pp. 231–241.
2. Titov, V.V.; González, F.I. *Implementation and Testing of the Method of Splitting Tsunami (MOST) Model*; Pacific Marine Environmental Laboratory: Seattle, WA, USA, 1997.
3. Titov, V.V.; Synolakis, C.E. Numerical modeling of 3-D long wave runup using VTCS-3. In *Long Wave Runup Models*; Yeh, H., Liu, P.L.-F., Synolakis, C.E., Eds.; World Scientific: Singapore, 1996; pp. 242–248.
4. Wang, X.; Powel, W.L. *COMCOT: A Tsunami Generation, Propagation and Run-Up Model*; GNS Science Report 2011/43; GNS Science: Lower Hutt, New Zealand, 2011; p. 121.
5. Shokin, Y.I.; Babailov, V.V.; Beisel, S.A.; Chubarov, L.B.; Eletsky, S.V.; Fedotova, Z.I.; Gusiakov, V.K. Mathematical Modeling in Application to Regional Tsunami Warning Systems Operations. In *Computational Science and High Performance Computing III, Proceedings of the 3rd Russian-German Advanced Research Workshop, Novosibirsk, Russia, 23–27 July 2007*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 52–68.
6. Synolakis, C.E.; Bernard, E.N. Tsunami science before and beyond Boxing Day 2004. *Phil. Trans. R. Soc. A* **2006**, *364*, 2231–2265.
7. Dalrymple, R.A.; Grilli, S.T.; Kirby, J.T. Tsunamis and challenges for accurate modeling. *Oceanography* **2006**, *19*, 142–151. [[CrossRef](#)]
8. Grilli, S.T.; Ioualalen, M.; Asavanant, J.; Shi, F.; Kirby, J.T.; Watts, P. Source Constraints and Model Simulation of the December 26, 2004, Indian Ocean Tsunami. *J. Waterw. Port Coast. Ocean Eng.* **2007**, *133*, 414–428. [[CrossRef](#)]
9. Murty, T.S.; Rao, A.D.; Nirupama, N.; Nistor, I. Numerical modelling concepts for tsunami warning systems. *Curr. Sci.* **2006**, *90*, 1073–1081.
10. Mirchina, N.R.; Pelinovsky, E.N. Nonlinear and dispersive effects for tsunami waves in the open ocean. *Int. J. Tsunami Soc.* **1982**, *2*, 1073–1081.
11. Dao, M.H.; Tkalich, P. Tsunami propagation modelling—A sensitivity study. *Nat. Hazards Earth Syst. Sci.* **2007**, *7*, 741–754. [[CrossRef](#)]
12. Løvholt, F.; Pedersen, G.; Gisler, G. Oceanic propagation of a potential tsunami from the La Palma Island. *J. Geophys. Res.* **2008**, *113*, C09026. [[CrossRef](#)]
13. Lovholt, F.; Pedersen, G.; Glimsdal, S. Coupling of Dispersive Tsunami Propagation and Shallow Water Coastal Response. *Open Oceanogr. J.* **2010**, *4*, 71–82. [[CrossRef](#)]
14. Peregrine, D.H. Long waves on a beach. *J. Fluid Mech.* **1967**, *27*, 815–827. [[CrossRef](#)]
15. Glimsdal, S.; Pedersen, G.K.; Harbitz, C.B.; Løvholt, F. Dispersion of tsunamis: Does it really matter? *Natl. Hazards Earth Syst. Sci.* **2013**, *13*, 1507–1526. [[CrossRef](#)]
16. Kirby, J.T.; Shi, F.; Tehranirad, B.; Harris, J.C.; Grilli, S.T. Dispersive tsunami waves in the ocean: Model equations and sensitivity to dispersion and Coriolis effects. *Ocean Model.* **2013**, *62*, 39–55. [[CrossRef](#)]
17. Khakimzyanov, G.S.; Dutykh, D.; Fedotova, Z.I. Dispersive shallow water wave modelling. Part III: Model derivation on a globally spherical geometry. *Commun. Comput. Phys.* **2018**, *23*, 315–360. [[CrossRef](#)]
18. Khakimzyanov, G.S.; Dutykh, D.; Gusev, O. Dispersive shallow water wave modelling. Part IV: Numerical simulation on a globally spherical geometry. *Commun. Comput. Phys.* **2018**, *23*, 361–407. [[CrossRef](#)]
19. Michalopoulos, P.G.; Yi, P.; Lyrintzis, A.S. Continuum modelling of traffic dynamics for congested freeways. *Transp. Res. Part B Methodol.* **1993**, *27*, 315–332. [[CrossRef](#)]

20. Dutykh, D.; Mitsotakis, D. On the relevance of the dam break problem in the context of nonlinear shallow water equations. *Discret. Contin. Dyn. Syst. Ser. B* **2010**, *13*, 799–818. [[CrossRef](#)]
21. Whitham, G.B. *Linear and Nonlinear Waves*; John Wiley & Sons Inc.: New York, NY, USA, 1999.
22. Godunov, S.K. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.* **1959**, *47*, 271–290.
23. Godunov, S.K. Reminiscences about Difference Schemes. *J. Comput. Phys.* **1999**, *153*, 6–25. [[CrossRef](#)]
24. Roe, P.L. Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.* **1981**, *43*, 357–372. [[CrossRef](#)]
25. Fuhrmann, J.; Ohlberger, M.; Rohde, C. *Finite Volumes for Complex Applications VII—Elliptic, Parabolic and Hyperbolic Problems*; Springer: Berlin, Germany, 2014.
26. Babuska, I.; Henshaw, W.D.; Oliger, J.E.; Flaherty, J.E.; Hopcroft, J.E.; Tezduyar, T. (Eds.) *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*; The IMA Volumes in Mathematics and Its Applications; Springer: New York, NY, USA, 1995; Volume 75.
27. Babuska, I.; Suri, M. The p and h-p Versions of the Finite Element Method, Basic Principles and Properties. *SIAM Rev.* **1994**, *36*, 578–632. [[CrossRef](#)]
28. Houston, P.; Süli, E. hp-Adaptive Discontinuous Galerkin Finite Element Methods for First-Order Hyperbolic Problems. *SIAM J. Sci. Comput.* **2001**, *23*, 1226–1252. [[CrossRef](#)]
29. Arvanitis, C.; Katsaounis, T.; Makridakis, C. Adaptive Finite Element Relaxation Schemes for Hyperbolic Conservation Laws. *ESAIM Math. Model. Numer. Anal.* **2010**, *35*, 17–33. [[CrossRef](#)]
30. Berger, M.J.; Oliger, J. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.* **1984**, *53*, 484–512. [[CrossRef](#)]
31. Berger, M.J.; Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comp. Phys.* **1989**, *82*, 64–84. [[CrossRef](#)]
32. George, D.L.; Leveque, R.J. Finite volume methods and adaptive refinement for global tsunami propagation and local inundation. *Sci. Tsunami Hazards* **2006**, *24*, 319.
33. Popinet, S. Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comp. Phys.* **2003**, *190*, 572–600. [[CrossRef](#)]
34. Tang, H.; Tang, T. Adaptive Mesh Methods for One- and Two-Dimensional Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.* **2003**, *41*, 487–515. [[CrossRef](#)]
35. Arvanitis, C.; Delis, A.I. Behavior of Finite Volume Schemes for Hyperbolic Conservation Laws on Adaptive Redistributed Spatial Grids. *SIAM J. Sci. Comput.* **2006**, *28*, 1927–1956. [[CrossRef](#)]
36. Degtyarev, L.M.; Drozdov, V.V.; Ivanova, T.S. The method of adaptive grids for the solution of singularly perturbed one dimensional boundary value problems. *Differ. Uravn.* **1987**, *23*, 1160–1169.
37. Degtyarev, L.M.; Ivanova, T.S. The adaptive-grid method in one-dimensional nonstationary convection-diffusion problems. *Differ. Uravn.* **1993**, *29*, 1179–1192.
38. Shokin, Y.I.; Urusov, A.I. On the construction of adaptive algorithms for unsteady problems of gas dynamics in arbitrary coordinate systems. In *Eighth International Conference on Numerical Methods in Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 1982; pp. 481–486.
39. Huang, W.; Ren, Y.; Russell, R.D. Moving Mesh Methods Based on Moving Mesh Partial Differential Equations. *J. Comp. Phys.* **1994**, *113*, 279–290. [[CrossRef](#)]
40. Gasparin, S.; Berger, J.; Dutykh, D.; Mendes, N. An innovative method to determine optimum insulation thickness based on non-uniform adaptive moving grid. *J. Braz. Soc. Mech. Sci. Eng.* **2019**, *41*, 173. [[CrossRef](#)]
41. Huang, W.; Russell, R.D. *Adaptive Moving Mesh Methods*; Applied Mathematical Sciences; Springer: New York, NY, USA, 2011.
42. Liseikin, V.D. *Grid Generation Methods*, 2nd ed.; Scientific Computation; Springer: Dordrecht, The Netherlands, 2010.
43. Godunov, S.K.; Zabrodin, A.; Ivanov, M.Y.; Kraiko, A.N.; Prokopov, G.P. *Numerical Solution of Multidimensional Problems of Gas Dynamics*; Nauka: Moscow, Russia, 1976.
44. Liseikin, V.D. The construction of structured adaptive grids - a review. *Comp. Math. Math. Phys.* **1996**, *36*, 1–32.
45. Khakimzyanov, G.; Dutykh, D. On supraconvergence phenomenon for second order centered finite differences on non-uniform grids. *J. Comp. Appl. Math.* **2017**, *326*, 1–14. [[CrossRef](#)]

46. Khakimzyanov, G.S.; Dutykh, D.; Gusev, O. Dispersive shallow water wave modelling. Part II: Numerical modelling on a globally flat space. *Commun. Comput. Phys.* **2018**, *23*, 30–92. [[CrossRef](#)]
47. Sudobicher, V.G.; Shugrin, S.M. Flow along a dry channel. *Izv. Akad. Nauk SSSR* **1968**, *13*, 116–122.
48. Alalykin, G.B.; Godunov, S.K.; Kireyeva, L.L.; Pliner, L.A. *Solution of One-Dimensional Problems in Gas Dynamics on Moving Grids*; Nauka: Moscow, Russia, 1970.
49. Serezhnikova, T.I.; Sidorov, A.F.; Ushakova, O.V. On one method of construction of optimal curvilinear grids and its applications. *Russ. J. Numer. Anal. Math. Model.* **1989**, *4*, 137–156. [[CrossRef](#)]
50. Shokin, Y.I.; Yanenko, N.N. *Method of Differential Approximation: Application to Gas Dynamics*; Nauka: Novosibirsk, Russia, 1985.
51. Darmaev, L.M.; Liseikin, V.D. A method of construction of multidimensional adaptive grids. *Model. Mech.* **1987**, *1*, 49–58.
52. Azarenok, B.N.; Ivanenko, S.A.; Tang, T. Adaptive Mesh Redistribution Method Based on Godunov's Scheme. *Commun. Math. Sci.* **2003**, *1*, 152–179. [[CrossRef](#)]
53. Beckett, G.; Mackenzie, J.A.; Ramage, A.; Sloan, D.M. Computational Solution of Two-Dimensional Unsteady PDEs Using Moving Mesh Methods. *J. Comp. Phys.* **2002**, *182*, 478–495. [[CrossRef](#)]
54. Cao, W.; Huang, W.; Russell, R.D. A Study of Monitor Functions for Two-Dimensional Adaptive Mesh Generation. *SIAM J. Sci. Comput.* **1999**, *20*, 1978–1994. [[CrossRef](#)]
55. Prokopov, G.P. About organization of comparison of algorithms and programs for 2D regular difference mesh construction. *Vopr. At. Nauk. Tekh. Ser. Mat. Model. Fiz. Prozess.* **1989**, *3*, 98–108.
56. Budd, C.J.; Huang, W.; Russell, R.D. Adaptivity with moving grids. *Acta Numer.* **2009**, *18*, 111–241. [[CrossRef](#)]
57. Bona, J.; Varlamov, V. Wave generation by a moving boundary. *Nonlinear Partial Differ. Equ. Relat. Anal.* **2005**, *371*, 41–71.
58. Khakimzyanov, G.S.; Dutykh, D. Numerical Modelling of Surface Water Wave Interaction with a Moving Wall. *Commun. Comput. Phys.* **2018**, *23*, 1289–1354. [[CrossRef](#)]
59. Knobloch, E. and Krechetnikov, R. Problems on Time-Varying Domains: Formulation, Dynamics, and Challenges. *Acta Appl. Math.* **2015**, *137*, 123–157. [[CrossRef](#)]
60. McCormack, R.W. The effect of viscosity in hypervelocity impact cratering. *AIAA Pap.* **1969**, 6.1969-354. [[CrossRef](#)]
61. Chhay, M.; Hoarau, E.; Hamdouni, A.; Sagaut, P. Comparison of some Lie-symmetry-based integrators. *J. Comp. Phys.* **2011**, *230*, 2174–2188. [[CrossRef](#)]
62. Dorodnitsyn, V.A. Finite Difference Models Entirely Inheriting Continuous Symmetry of Original Differential Equations. *Int. J. Mod. Phys. C* **1994**, *5*, 723–734. [[CrossRef](#)]
63. Huang, W.; Russell, R.D. Adaptive mesh movement—The MMPDE approach and its applications. *J. Comp. Appl. Math.* **2001**, *128*, 383–398. [[CrossRef](#)]
64. Lax, P.; Wendroff, B. Systems of conservation laws. *Commun. Pure Appl. Math.* **1960**, *13*, 217–237. [[CrossRef](#)]
65. Courant, R.; Isaacson, E.; Rees, M. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure Appl. Math.* **1952**, *5*, 243–255. [[CrossRef](#)]
66. Courant, R.; Friedrichs, K.; Lewy, H. Über die partiellen Differenzgleichungen der mathematischen Physik. *Math. Ann.* **1928**, *100*, 32–74. [[CrossRef](#)]
67. Breuss, M. About the Lax-Friedrichs scheme for the numerical approximation of hyperbolic conservation laws. *PAMM* **2004**, *4*, 636–637. [[CrossRef](#)]
68. Shokin, Y.I.; Sergeeva, Y.V.; Khakimzyanov, G.S. Construction of monotonic schemes by the differential approximation method. *Russ. J. Numer. Anal. Math. Model.* **2005**, *20*, 463–481. [[CrossRef](#)]
69. LeVeque, R.J. *Numerical Methods for Conservation Laws*, 2nd ed.; Birkhäuser: Basel, Switzerland, 1992.
70. Rozhdestvenskiy, B.L.; Yanenko, N.N. *Systems of Quasilinear Equations and Their Application to Gas Dynamics*; Nauka: Moscow, Russia, 1978.
71. Thomas, P.D.; Lombart, C.K. Geometric conservation law and its application to flow computations on moving grid. *AIAA J.* **1979**, *17*, 1030–1037. [[CrossRef](#)]
72. Budd, C.J.; Williams, J.F. Moving Mesh Generation Using the Parabolic Monge-Ampère Equation. *SIAM J. Sci. Comput.* **2009**, *31*, 3438–3465. [[CrossRef](#)]
73. Stockie, J.M.; Mackenzie, J.A.; Russell, R.D. A Moving Mesh Method for One-dimensional Hyperbolic Conservation Laws. *SIAM J. Sci. Comput.* **2001**, *22*, 1791–1813. [[CrossRef](#)]

74. Fedorenko, R.P. *Introduction to Computational Physics*; MIPT Press: Moscow, Russia, 1994.
75. Dar'in, N.A.; Mazhukin, V.I.; Samarskii, A.A. A finite-difference method for solving the equations of gas dynamics using adaptive nets which are dynamically associated with the solution. *USSR Comput. Math. Math. Phys.* **1988**, *28*, 164–174. [[CrossRef](#)]
76. Dorfi, E.A.; Drury, L.O. Simple adaptive grids for 1-D initial value problems. *J. Comp. Phys.* **1987**, *69*, 175–195. [[CrossRef](#)]
77. Zegeling, P.A.; Blom, J.G. An evaluation of the gradient-weighted moving-finite-element method in one space dimension. *J. Comp. Phys.* **1992**, *103*, 422–441. [[CrossRef](#)]
78. Zegeling, P.A.; Verwer, J.G.; Van Eijkeren, J. C.H. Application of a moving grid method to a class of 1D brine transport problems in porous media. *Int. J. Num. Meth. Fluids* **1992**, *15*, 175–191. [[CrossRef](#)]
79. Coyle, J.M.; Flaherty, J.E.; Ludwig, R. On the stability of mesh equidistribution strategies for time-dependent partial differential equations. *J. Comp. Phys.* **1986**, *62*, 26–39. [[CrossRef](#)]
80. Daripa, P. Iterative schemes and algorithms for adaptive grid generation in one dimension. *J. Comp. Phys.* **1992**, *100*, 284–293. [[CrossRef](#)]
81. Asselin, R. Frequency filter for time integrations. *Mon. Weather Rev.* **1972**, *100*, 487–490. [[CrossRef](#)]
82. Burgers, J.M. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.* **1948**, *1*, 171–199.
83. Harten, A. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.* **1983**, *49*, 357–393. [[CrossRef](#)]
84. Stoker, J.J. *Water Waves: The Mathematical Theory with Applications*; Interscience: New York, NY, USA, 1957.
85. Krejic, N.; Krunic, T.; Nedeljkov, M. Numerical verification of delta shock waves for pressureless gas dynamics. *J. Math. Anal. Appl.* **2008**, *345*, 243–257. [[CrossRef](#)]
86. van Dam, A.; Zegeling, P.A. A robust moving mesh finite volume method applied to 1D hyperbolic conservation laws from magnetohydrodynamics. *J. Comp. Phys.* **2006**, *216*, 526–546. [[CrossRef](#)]
87. Barakhnin, V.B.; Borodkin, N.V. The second order approximation TVD scheme on moving adaptive grids for hyperbolic systems. *Sib. Zhurnal Vychisl. Mat.* **2000**, *3*, 109–121.
88. Gosse, L. *Computing Qualitatively Correct Approximations of Balance Laws: Exponential-Fit, Well-Balanced and Asymptotic-Preserving*, 1st ed.; SIMAI Springer Series; Springer: Milan, Italy, 2013.
89. Voltsinger, N.E.; Pelinovsky, E.N.; Klevannyi, K.A. *Long Wave Dynamics of Coastal Regions*; Gidrometeoizdat: Leningrad, Russia, 1989.
90. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 2009.
91. Khakimzyanov, G.S.; Shokina, N.Y.; Dutykh, D.; Mitsotakis, D. A new run-up algorithm based on local high-order analytic expansions. *J. Comp. Appl. Math.* **2016**, *298*, 82–96. [[CrossRef](#)]
92. Dutykh, D.; Clamond, D.; Milewski, P.; Mitsotakis, D. Finite volume and pseudo-spectral schemes for the fully nonlinear 1D Serre equations. *Eur. J. Appl. Math.* **2013**, *24*, 761–787. [[CrossRef](#)]
93. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations: Nonstiff Problems*; Springer: Berlin, Germany, 2009.
94. Dutykh, D.; Pelinovsky, E. Numerical simulation of a solitonic gas in KdV and KdV-BBM equations. *Phys. Lett. A* **2014**, *378*, 3102–3110. [[CrossRef](#)]

