An Adaptive Algorithm for the Direct and Inverse Problems of Burgers' Equation

LEA SAFETLY*, TONI SAYAH
Laboratoire de Mathématiques et Applications,
Unité de recherche Mathématiques et Modélisation,
Faculté des Sciences, Université Saint-Joseph de Beyrouth,
B.P. 11-514 Riad El Solh, Beyrouth 1107 2050,
LEBANON

**Corresponding Author*

Abstract: This paper presents adaptive methods for approximating the solution of the direct problem and for solving the inverse problem of reconstructing smooth initial data from boundary measurements, both for the one-dimensional inviscid Burgers' equation. The discretization is based on a Godunov upwind scheme implemented on a dynamically evolving space-time mesh that undergoes refinement and coarsening in response to the solution features and local variations. For both direct and inverse problems, we introduce an adaptive algorithm based on local quantities called indicators, and perform a space-time adaptive technique to improve computational efficiency. Numerical experiments illustrate the performance of the adaptive algorithm, validating its accuracy and robustness.

Key-Words: Burgers' equation; Godunov scheme; adaptive mesh; direct problem; inverse problem; residual error indicators.

Received: April 11, 2026. Revised: May 26, 2026. Accepted: June 28, 2026. Published: November 26, 2026.

1 Introduction

Hyperbolic conservation laws appear in a wide range of physical and engineering applications, including fluid dynamics, traffic flow, and gas dynamics [1], [2]. They form the basis of many concepts, describing how physical quantities are preserved and redistributed within closed systems. Scalar conservation laws represent a simplified class of conservation laws that describe the evolution of a single conserved quantity, typically denoted by $u(x,t)$. They play a key role in modeling wave phenomena, including propagation and shock formation, and their relatively simple form allows more focused theoretical analysis and facilitates the design of efficient numerical methods for their solution [3], [4]. Among them, the one-dimensional inviscid Burgers' equation

$$\partial_t u(x,t) + \partial_x \left(\frac{u^2(x,t)}{2} \right) = 0, \ (x,t) \in \mathbb{R} \times (0,T), \tag{1}$$

where $T > 0$, serves as a fundamental model for understanding nonlinear wave propagation and shock formation [5].

This equation was first introduced in [6], where its stationary solution was obtained, and was later studied in [7] for the mathematical modeling of turbulence. Since then, Burgers' equation has been widely used in various fields. Even for smooth initial data, solutions may develop discontinuities in finite time, and therefore several weak solutions have been introduced, among which is the entropy solution in the sense of Kružkov [8]. The well-posedness theory for transport and conservation laws has been studied in the works of [9], [10], [11], among others.

For the Burgers' equation, considered on the spatial domain $\mathbb{R}^+$, we refer to [12] for the existence and uniqueness of weak entropy solutions. Note that results on the existence and uniqueness of entropy solutions on a bounded domain were established in [13], [14], with appropriate boundary conditions.

For the inverse problem related to Burgers' equation (2), the objective is to recover the smooth initial data u_0 in a bounded domain from the boundary measurements over time. In the context of inverse problems for partial differential equations, it is common to consider the inverse backward problem, where the aim is to recover the initial state of the direct problem from the solution at the final time. Such problems are known to be ill-posed [15], [16].

Classical finite volume methods, when combined with monotone or upwind numerical fluxes, provide robust schemes capable of capturing shock waves without producing spurious oscillations [4], [9], [17]. However, when a uniform spatial mesh is employed, the computational cost may become unnecessarily large. Adaptive mesh refinement (AMR) methods were developed precisely to overcome this limitation. The main idea is to dynamically adjust the mesh according to the local behavior of the solution: the grid is refined where sharp features such as shocks,

rarefaction fronts, or boundary layers emerge, and coarsened where the solution remains smooth. This strategy significantly reduces the number of degrees of freedom while maintaining and enhancing the accuracy of the numerical approximation (see among others [18], [19], [20]). Indeed, adaptive methods rely on an *a posteriori* error estimate that bounds the error between the exact and numerical solutions, locally by a quantity known as a local error indicator [21], [22], [23]. Moreover, optimal mesh adaptation has been applied to nonlinear conservation laws, in particular to the Burgers' equation, as in [24].

In this work, we consider the following Problem in the bounded domain $(0, 1)$:

$$\begin{cases} \partial_t u(x,t) + \partial_x \left(\dfrac{u^2(x,t)}{2} \right) = 0, & \text{in } (0,1) \times (0,T), \\ u(x,0) = u_0(x), & \text{in } [0,1], \\ u(0,t) = \alpha(t), & \text{in } [0,T], \end{cases} \tag{2}$$

where $u_0 \in C^1([0,1])$, $u_0(x) > 0$, for $x \in [0,1]$,, $\alpha \in C^1([0,T])$, and the boundary value at $x = 0$ satisfies $u(0,t) > 0$, for $t \in [0,T]$. The existence and uniqueness of solutions to Problem (1), under appropriate assumptions, have been studied in [25], whereas in the present work, we focus on the numerical approximation of the direct and inverse problems. Although the present work and our previous one [5] both concern inverse problems for Burgers' equation, their objectives and contributions are different. In [5], we considered Riemann initial data and established stability results for two inverse problem settings, and numerical simulations illustrated these theoretical results. As for this work, we consider numerical methods for both the direct and inverse problems based on space–time residual indicators computed directly from the discrete solution. The inverse problem aims to reconstruct smooth initial data from boundary measurements over a fixed time interval. We begin with the direct problem, where, for each space–time cell $(x_i^n, x_{i+1}^n) \times (t^n, t^{n+1})$, we apply the Godunov upwind scheme [4] on a dynamically refined space–time mesh. The refinement and coarsening procedures are driven by local residual-based indicators computed from the numerical solution. We then consider the inverse problem, which aims to recover the initial data from boundary measurements at $x = 1$ over time, using a backward space–time numerical method. A similar residual-based adaptive strategy is employed for the inverse problem. For both the direct and inverse problems, we present the space–time adaptive numerical algorithms and illustrate their performance through numerical simulations for smooth and non-smooth initial data.

This paper is organized as follows. Section 2 presents the space–time adaptive framework for the direct and inverse problems. In particular, we describe the Godunov upwind discretization, the space–time reconstruction of the numerical solutions, and the residual-based space and time indicators used. Section 3 presents the adaptive algorithm based on refinement and coarsening criteria and the numerical results comparing adaptive and uniform meshes, for smooth and non-smooth initial data in both problem settings.

2 Adaptive mesh method for a Godunov Upwind Scheme for Burgers' Equation

In this section, we present a space-time adaptive method for a discrete problem associated with Problem (2). This approach is applied to both the direct problem and to the inverse problem of recovering smooth initial data from boundary measurements at $x = 1$ over time. We describe the Godunov upwind scheme on a dynamically refined space-time mesh, then we evaluate the local residual errors associated with the discrete solutions for both problem settings.

2.1 Discrete Scheme

In this work, we are interested in the numerical simulation of the problem (2). In order to compute an approximate solution to the Burgers' equation, the interval $[0, 1]$ is initially partitioned into M^0 uniform cells. The initial nodes are

$$X^0 = \{x_i^0\}_{i=0}^{M^0}, \qquad x_i^0 = i\frac{1}{M^0} \tag{3}$$

and the initial time step is Δt^0. As we use adaptive space-time, both the time step and the spatial mesh change during the time iterations. Therefore, at each time iteration t^n, we denote by Δt^n the corresponding time step and by

$$X^n = \{x_i^n\}_{i=0}^{M^n} \tag{4}$$

the spatial nodes, where:

- M^n is the **number of cells at time step** n (changing due to refinement/coarsening);
- The cell widths are

$$\Delta x_i^n = x_i^n - x_{i-1}^n, \qquad i = 1, \ldots, M^n - 1. \tag{5}$$

Several schemes have been proposed for Burgers' equation, such as Lax-Friedrichs, Lax-Wendroff, and Engquist–Osher (see [2], [26]). In this work, we use

the Godunov scheme due to its monotonicity and consistency with the entropy condition. Moreover, this scheme is first-order accurate, which is simple and efficient, especially when used with the adaptive mesh method. For a given mesh X^n, the exact solution $u(.,t)$ is approximated in each cell (x_i^n, x_{i+1}^n) of the mesh by $u_i^n \approx u(\cdot, t^n)$. At each time iteration n and having u_i^n, we compute u_i^{n+1} using the following first–order **Godunov** upwind scheme:

$$\begin{cases} u_i^{n+1} = u_i^n - \dfrac{\Delta t^n}{\Delta x_i^n}\left(F(u_i^n) - F(u_{i-1}^n)\right), \\ \qquad i = 1, \ldots, M^n, \\ u_i^0 = \hat{u}_i^0, \qquad i = 1, \ldots, M^0, \\ u_0^n = \hat{\alpha}_0^n, \end{cases} \tag{6}$$

where $F(\cdot,\cdot)$ denotes the Godunov numerical flux for Burgers' equation, $\hat{u}_i^0$ is an approximation of the initial condition $u_0(x_i)$ and $\hat{\alpha}_0^n$ is an approximation of $\alpha(t^n)$. These numerical fluxes are given by

$$F(u_i^n, u_{i+1}^n) = \max\left(\frac{(\max(u_i^n, 0))^2}{2}, \frac{(\min(u_{i+1}^n, 0))^2}{2}\right), \tag{7}$$

$$F(u_{i-1}^n, u_i^n) = \max\left(\frac{(\max(u_{i-1}^n, 0))^2}{2}, \frac{(\min(u_i^n, 0))^2}{2}\right). \tag{8}$$

Since the numerical states considered in this work are nonnegative, the standard Godunov flux for the convex Burgers flux reduces to

$$F(u_i^n, u_{i+1}^n) = \frac{(u_i^n)^2}{2}, \qquad F(u_{i-1}^n, u_i^n) = \frac{(u_{i-1}^n)^2}{2}. \tag{9}$$

To ensure stability, we impose at each time step t^n the following CFL condition (see [4])

$$\max_{1 \le i \le M^n} |u_i^n| \, \frac{\Delta t^n}{\min\limits_{1 \le i \le M^n} \Delta x_i^n} \le \frac{1}{2}. \tag{10}$$

2.2 Direct Problem

In this part, we introduce the numerical framework used for the direct problem. Starting from the discrete solution obtained by the Godunov upwind scheme (6), we construct a space–time reconstruction of the numerical solution and define the corresponding residual error. This residual is then used to construct the local space and time error indicators that drive the adaptive refinement and coarsening procedure.

At each time step t^n, having the discrete numerical solution u_i^n, we denote by u_h^n the space interpolated solution such that: for $x \in [x_i^n, x_{i+1}^n]$,

$$u_h^n(x) = \frac{x - x_i^n}{\Delta x_i^n} u_{i+1}^n + \frac{x_{i+1}^n - x}{\Delta x_i^n} u_i^n, \tag{11}$$

where $\Delta x_i^n = x_{i+1}^n - x_i^n$. Furthermore, we denote by u_{ht} the space–time reconstructed solution based on the numerical solution u_h^n defined as: for $t \in [t^n, t^{n+1}]$,

$$u_{ht}(x,t) = \frac{t - t^n}{\Delta t^n} u_h^{n+1}(x) + \frac{t^{n+1} - t}{\Delta t^n} u_h^n(x), \tag{12}$$

where $\Delta t^n = t^{n+1} - t^n$. The resulting function u_{ht} is continuous and piecewise linear in both variables and will be used in the evaluation of the error indicators. The above space-time interpolated solution u_{ht} allows us to introduce the following residual error

$$\varepsilon_{ht}(x,t) = \partial_t u_{ht}(x,t) + u_{ht}(x,t)\partial_x(u_{ht}(x,t)), \tag{13}$$

which gives an indication of the error in both time and space.

We note that *a posteriori* error estimates for one-dimensional scalar conservation laws have been studied extensively in the literature; see, for instance, [27], [28]. In particular, for the one-dimensional Burgers equation, an L^2-type *a posteriori* error estimate based on a space–time reconstruction and its associated residual has been established under periodic boundary conditions; see, e.g., [29].

Remark 1. *The problem considered in this work is posed on the bounded interval* $(0,1)$ *and is associated with the non-periodic boundary condition* $u(0,t) = \alpha(t)$, $t \in [0,T]$. *A rigorous derivation of an a posteriori error estimate adapted to the present problem is beyond the scope of this paper and will be the subject of forthcoming work.*

In order to use the adaptive space-time method, we decompose the residual ε_{ht}, given by (13), into two parts: the space indicators and the time indicators. By adding and subtracting $u_h^{n+1}\partial_x u_h^{n+1}$, we obtain

$$\begin{aligned} \varepsilon_{ht} = &\left(\frac{u_h^{n+1} - u_h^n}{\Delta t^n} + u_h^{n+1}\partial_x u_h^{n+1}\right) \\ &+ \left(u_{ht}\partial_x u_{ht} - u_h^{n+1}\partial_x u_h^{n+1}\right). \end{aligned} \tag{14}$$

We therefore write

$$\varepsilon_{ht} = \varepsilon_{s,i}^n + \varepsilon_{t,i}^n, \tag{15}$$

where the discrete space indicators are given by

$$\varepsilon_{s,i}^n = \frac{1}{\Delta t^n}(u_h^{n+1} - u_h^n) + u_h^{n+1}\partial_x(u_h^{n+1}) \tag{16}$$

and the discrete time indicators are given by

$$\varepsilon_{t,i}^n = u_{ht}\partial_x(u_{ht}) - u_h^{n+1}\partial_x(u_h^{n+1}). \tag{17}$$

The first quantity is used as the space indicator for the mesh refinement and coarsening, while the second one is used for the time adaptation.

2.3 Inverse Problem

In this section, we introduce the inverse problem that recovers the initial data u_0 from boundary measurements at $x = 1$ over time. We begin by presenting the space-time backward problem. Let $s = 1 - x$, $\tau = T - t$ and $v(s, \tau) = u(1 - s, T - \tau) = u(x, t)$, such that the above Problem (2) becomes:

$$\partial_t v(x,t)+\partial_x \left(\frac{v^2(x,t)}{2}\right) = 0, \qquad \text{in } (0,1)\times(0,T). \tag{18}$$

We assume that at the final time T, the solution becomes constant, equal to its value at the boundary $x = 1$, denoted by $\alpha_0 > 0$. Indeed, if the solution varies slowly with respect to time, that is $\partial_t v(x, T) = 0$, we obtain from the equation $\partial_x \left(\frac{v^2(x,T)}{2}\right) = 0$, which leads to $v(x, T) = v(1, T) = \alpha_0$. Hence, we set the initial data for Problem (18) as $v(x, 0) = u(1 - x, T) = \alpha_0$, $x \in [0, 1]$. In practice, α_0 corresponds to the measurement of the solution at $x = 1$ at time T. For the boundary condition, we have $v(0, t) = u(1, T - t)$ in $[0, T]$. We assume that $u(1, T - \cdot) \in L^\infty(0, T)$. Thus, the backward problem becomes the following:

$$\begin{cases} \partial_t v(x,t) + \partial_x \left(\frac{v^2(x,t)}{2}\right) = 0, & \text{in } (0,1)\times(0,T), \\ v(x,0) = \alpha_0, & \text{in } [0,1], \\ v(0,t) = u(1,T-t), & \text{in } [0,T]. \end{cases} \tag{19}$$

We remark that solving the problem (19) allows us to get the initial data u_0 of the direct problem (2) as $u_0(x) = v(1 - x, T)$.

In general, for the inverse problem, we consider Problem (19) directly with given measurements $\beta(t)$ at $x = 1$ and over time as follows:

$$\begin{cases} \partial_t v(x,t) + \partial_x \left(\frac{v^2(x,t)}{2}\right) = 0, & \text{in } (0,1)\times(0,T), \\ v(x,0) = \alpha_0, & \text{in } [0,1], \\ v(0,t) = \beta(t), & \text{in } [0,T], \end{cases} \tag{20}$$

where $\beta \in L^\infty(0, T)$.

Assumption 1. *In the following, we assume that $\beta \in C^0([0, T)]$.*

Remark 2. *We note that, in practice, $\beta(t)$ represents discrete measurements at $x = 1$ over time. It can then be interpolated using a high-order polynomial interpolation to obtain a continuous function on $[0, T]$, also denoted by β.*

The existence, uniqueness, and stability of the inverse problem, under appropriate assumptions, have been established in [25], whereas in this work we focus on the associated numerical approximation following the discretization presented in the previous part.
At each time step t^n, having the discrete numerical solution v_i^n, we compute v_i^{n+1} using the following first–order **Godunov** upwind scheme:

$$\begin{cases} v_i^{n+1} = v_i^n - \dfrac{\Delta t^n}{\Delta x_i^n}\left(F(v_i^n) - F(v_{i-1}^n)\right), \\ i = 1, \ldots, M^n, \\ v_i^0 = \hat{\alpha}_0, \qquad i = 1, \ldots, M^n, \\ v_0^n = \hat{\beta}^n, \end{cases} \tag{21}$$

where $\hat{\alpha}_0$ is the initial condition assumed to be constant and equal to the measurement at $x = 1$ and at time T, and $\hat{\beta}^n$ are the measurements at $x = 1$ and time t^n, $n = 1, \ldots, N$. At each time step t^n, having the discrete numerical solution v_i^n, we denote by v_h^n the space interpolated solution such that: for $x \in [x_i^n, x_{i+1}^n]$,

$$v_h^n(x) = \frac{x - x_i^n}{\Delta x_i^n}\, v_{i+1}^n + \frac{x_{i+1}^n - x}{\Delta x_i^n}\, v_i^n. \tag{22}$$

Furthermore, we denote by v_{ht} the space–time reconstructed solution based on the numerical solution v_h^n of Problem (20), and defined as: for $t \in [t^n, t^{n+1}]$,

$$v_{ht}(x,t) = \frac{t - t^n}{\Delta t^n}\, v_h^{n+1}(x) + \frac{t^{n+1} - t}{\Delta t^n}\, v_h^n(x). \tag{23}$$

The resulting function v_{ht} is continuous and piecewise linear in both variables and is used in the evaluation of the error indicators. The above space-time interpolated solution v_{ht} allows us to introduce the following residual error

$$\hat{\varepsilon}_{ht}(x,t) = \partial_t v_{ht}(x,t) + v_{ht}(x,t)\partial_x(v_{ht}(x,t)), \tag{24}$$

which gives an indication of the error in both time and space.

Remark 3. *A rigorous derivation of an a posteriori error estimate adapted to the present inverse problem, is beyond the scope of this paper and will be the subject of forthcoming work.*

As in the previous part, to use the adaptive space–time method, we decompose $\hat{\varepsilon}_{ht}$, in this case, into two parts, the space indicators and the time indicators, as follows: for $t \in [t^n, t^{n+1})$ and $x \in [x_i^n, x_{i+1}^n)$,

$$\hat{\varepsilon}_{ht} = \hat{\varepsilon}_{s,i}^n + \hat{\varepsilon}_{t,i}^n, \tag{25}$$

where the discrete space indicators are given by

$$\hat{\varepsilon}_{s,i}^n = \frac{1}{\Delta t^n}(v_h^{n+1} - v_h^n) + v_h^{n+1}\partial_x(v_h^{n+1}) \tag{26}$$

and the discrete time indicators are given by

$$\hat{\varepsilon}_{t,i}^{n} = v_{ht}\partial_x(v_{ht}) - v_h^{n+1}\partial_x(v_h^{n+1}). \quad (27)$$

In the present numerical study, the residual quantities defined are used as error indicators for the space-time adaptive algorithm.

3 Numerical Simulations

In this section, we present numerical simulations, using MATLAB, to illustrate the performance of the adaptive space–time method for the direct and inverse problems, with smooth and non-smooth initial data.

3.1 Adaptive Algorithm in Space and Time: Direct Problem

In the following, we present numerical results for the direct problem (2) with both smooth and non-smooth initial data. In both cases, the numerical approximation of the solution is obtained using a fully discrete adaptive space–time scheme based on the Godunov (6) upwind method introduced above.

3.1.1 Smooth Initial data

In this part, we consider the following smooth initial data

$$u_0(x) = a + b\sin^6(m\pi x)\,, \quad (28)$$

where a, b and m are positive constants. In the numerical experiments, we take

$$M^0 = 10, \qquad a = 0.5, \quad b = 0.5, \quad m = 1. \quad (29)$$

Since Burgers' equation is a nonlinear conservation law, the solution may develop shocks. To efficiently resolve this, we use a space-time adaptive method based on local error indicators. At each time step, following the indicators $\varepsilon_{s,i}^n$ and $\varepsilon_{t,i}^n$, we may adapt the time step or the mesh accordingly. The mesh is refined in regions where the solution presents fluctuations and coarsened in regions where it remains smooth. This strategy concentrates the computational effort where it is most needed and avoids unnecessary resolution elsewhere. At each time step t^n, we compute the space and time indicators given by (16) and (17), namely $\varepsilon_{s,i}^n$ and $\varepsilon_{t,i}^n$, and use them to adapt the mesh and the time step. We define the following local time and space indicators by:

$$\begin{aligned}
(I_{t,i}^n)^2 &= \int_{t^n}^{t^{n+1}} \int_{x_i^n}^{x_{i+1}^n} (\varepsilon_{t,i}^n)^2\, dx\, dt,\\
(I_{s,i}^n)^2 &= \int_{t^n}^{t^{n+1}} \int_{x_i^n}^{x_{i+1}^n} \left(\varepsilon_{s,i}^n\right)^2 dx dt,\\
(\mathcal{U}_i^n)^2 &= \Delta t^n \int_{x_i^n}^{x_{i+1}^n} (u_h^{n+1})^2\, dx.
\end{aligned} \quad (30)$$

We also define the mean space indicator

$$\overline{I}^{\,n} = \frac{1}{M^n}\sum_{i=0}^{M^n-1} I_{s,i}^n, \quad (31)$$

the global space indicator

$$\mathcal{E}_s^n = \left(\frac{\sum_{i=0}^{M^n-1}(I_{s,i}^n)^2}{\sum_{i=0}^{M^n-1}(\mathcal{U}_i^n)^2}\right)^{1/2}, \quad (32)$$

and the global time indicator

$$\mathcal{E}_t^n = \left(\frac{\sum_{i=0}^{M^n-1}(I_{t,i}^n)^2}{\sum_{i=0}^{M^n-1}(\mathcal{U}_i^n)^2}\right)^{1/2}. \quad (33)$$

These indicators guide the space–time adaptive method. For time adaptation, the time step Δt^n is increased or decreased according to the global time indicator $\mathcal{E}_t^n$, while for space adaptation, refinement or coarsening is performed by classifying the cells based on the size of their local residual $I_{s,i}^n$ relative to the average residual $\overline{I}^n$. We mention that at each time step t^n, and after performing space-time adaptive mesh, we adjust the time step Δt^n to satisfy the stability condition given by (10). Before introducing the adaptive algorithm, we briefly present the corresponding strategy: Starting from an initial mesh and time step, the solution is advanced in time using the Godunov upwind scheme under the CFL stability condition (10). At each time iteration, we compute the global indicators $\mathcal{E}_t^n$ and $\mathcal{E}_s^n$. If $\mathcal{E}_s^n + \mathcal{E}_t^n \le \text{tol}_\varepsilon$, where tol_ε is a given tolerance, we advance to the next time step, else we have one of the two following situations: if the global time indicator $\mathcal{E}_t^n$ is greater than the global spatial indicator $\mathcal{E}_s^n$, we adapt the time-step; Elsewhere, we perform local mesh adaption. The adaptive algorithm can be summarized as follows:

Algorithm: Adaptive Upwind Scheme with Local Space–Time Refinement

(1) **Initialization.**

- Set the parameters a, b, m, T, initial number of cells M^0, and the CFL coefficient (10).
- Construct the initial mesh $X^0 = \{x_i\}_{i=0}^{M^0}$ and define the initial condition $u^0(x) = u_0(x)$, $x \in [0,1]$.

- Choose an initial time step Δt^0 and a tolerance tol_ε for the adaptive loop. Set $t^n = 0$.

(2) While $t^n < T$ and having u_h^n do,

(a) We compute u_h^{n+1} by solving Problem (6).

(b) We compute the local and global indicators: $\varepsilon_{s,i}^n$, $\varepsilon_{t,i}^n$, $\mathcal{E}_s^n$ and $\mathcal{E}_t^n$.

(c) If $\mathcal{E}_s^n + \mathcal{E}_t^n \leq \text{tol}_\varepsilon$, set $t^n = t^n + \Delta t^n$, $u_h^n = u_h^{n+1}$ and go to step (2) $\Big($with an eventual larger time step when $\mathcal{E}_s^n + \mathcal{E}_t^n \leq 0.8 * \text{tol}_\varepsilon$ such that $\Delta t^n = \min\Big(1.2\,\Delta t^n,\ \text{CFL} \cdot \frac{\min_i(\Delta x_i^n)}{\max_i|u_i^n|}\Big)\Big)$.

(d) If $\mathcal{E}_s^n + \mathcal{E}_t^n > \text{tol}_\varepsilon$ perform:

- If $\mathcal{E}_s^n < \mathcal{E}_t^n$, set $\Delta t^n = \min\Big(\frac{\Delta t^n}{1.2},\ \text{CFL} \cdot \frac{\min_i(\Delta x_i^n)}{\max_i|u_i^n|}\Big)$ and go to step (2).
- If $\mathcal{E}_s^n \geq \mathcal{E}_t^n$, adapt the mesh as follows:
  - Cut each cell i in two that satisfies

$$I_{s,i}^n > 1.2\,\overline{I}^n \quad \text{and} \quad \max_n(u^n)\frac{\Delta t^n}{\min_i \Delta x_i^n} \leq 0.49. \tag{34}$$

  - Remove the vertex j for which the two adjacent cells ℓ satisfy

$$I_{s,\ell}^n < 0.4\,\overline{I}^n \quad \text{and} \quad \max_n(u^n)\frac{\Delta t^n}{\min_\ell \Delta x_\ell} \leq 0.99. \tag{35}$$

Go to step (2) without changing t^n.

The constants used in the refinement and coarsening criteria are chosen as numerical thresholds for the adaptive procedure. The constant 1.2 identifies cells whose local space indicator is sufficiently larger than the mean indicator and therefore require refinement, whereas the factor 0.4 identifies cells with a sufficiently small local indicator and allows coarsening. The constants 0.49 and 0.99 are used as CFL thresholds in the refinement and coarsening steps, respectively.

To illustrate our work, we present numerical simulations showing the efficiency of the proposed algorithm. Figure 1, Figure 2, Figure 3 and Figure 4 show the comparison between the numerical solution computed with the above algorithm and the exact solution at $t = 0.1$, $t = 0.2$, $t = 0.3$ and $t = 0.4$, respectively, with a tolerance $tol_\varepsilon = 0.04$. We can clearly see that the numerical solution is very close to the exact one and that the mesh is concentrated in critical regions; it is heavily refined near the peak of the solution, while significantly fewer refinement points are allocated in other regions (i.e. between $x \approx 0$ and $x \approx 0.2$ in Figure 4), due to the absence of strong variations in the solution. We notice that Figure 4 highlights the advantage of the adaptive method.

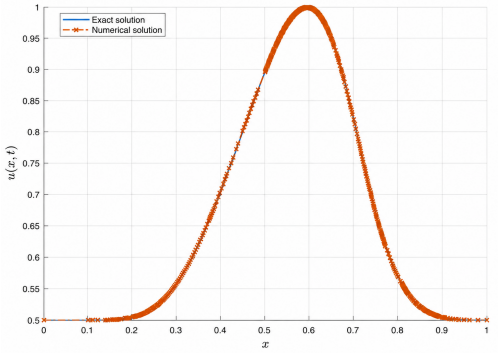

Fig. 1: Comparison of the numerical and exact solution at $t = 0.1$, with $\text{tol}_\varepsilon = 0.04$ and $M = 388$. (Source: Created by the authors).

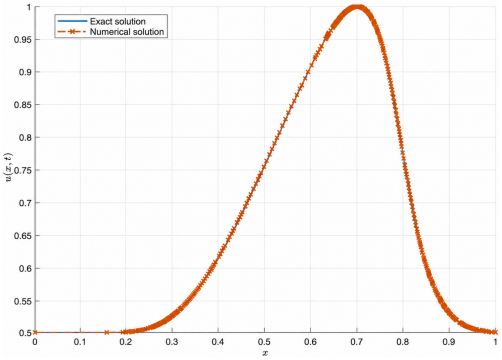

Fig. 2: Comparison of the numerical and exact solutions at $t = 0.2$, with $\text{tol}_\varepsilon = 0.04$ and $M = 344$. (Source: Created by the authors).

Figure 5 illustrates the propagation of the numerical and exact solutions at $T = 2$, showing that they are almost constant and equal to $a = 0.5$. Indeed, the numerical values are between 0.5 and 0.508, which is numerically close to 0.5.

A quantitative comparison is presented next using the global relative error Err defined in (36), with respect to the total number of degrees of freedom for both the adaptive and uniform mesh methods which allows us to study the behavior of the error as the number of degrees of freedom increases and to compare the numerical performance of the two mesh methods. For the adaptive case, we consider $T = 0.4$ and the tolerance $tol_\varepsilon \in \{0.12, 0.10, 0.09, 0.08, 0.07\}$. For each tolerance, the solution is computed using the adaptive algorithm described above, where the initial mesh is a uniform

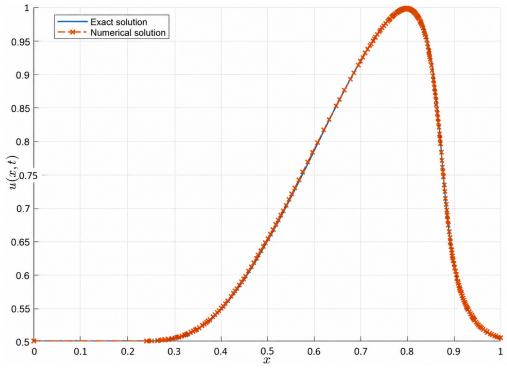

Fig. 3: Comparison of the numerical and exact solutions at $t = 0.3$, with $\text{tol}_\varepsilon = 0.04$ and $M = 233$. (Source: Created by the authors).

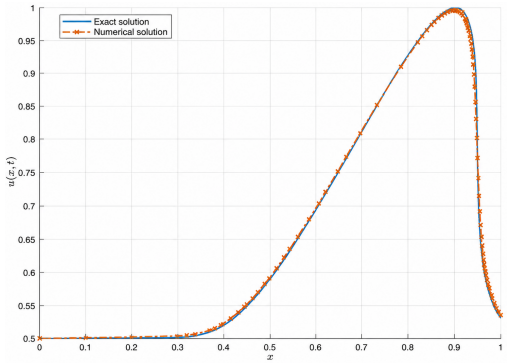

Fig. 4: Comparison of the numerical and exact solutions at $t = 0.4$, with $\text{tol}_\varepsilon = 0.04$ and $M = 84$. (Source: Created by the authors).

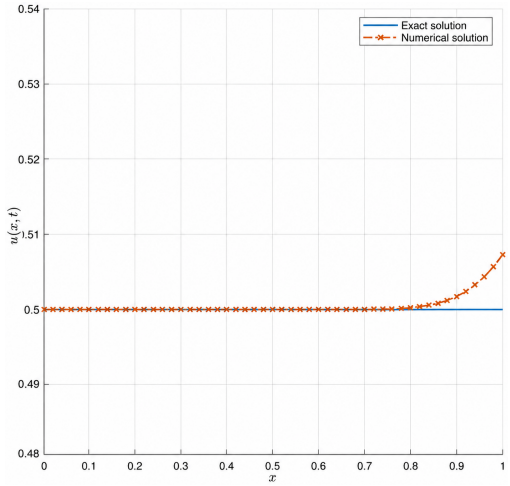

Fig. 5: Comparison of the numerical and exact solutions, with $\text{tol}_\varepsilon = 0.04$ at $t = 2$. (Source: Created by the authors).

mesh with $M^0 = 50$ cells and the initial time step is $\Delta t^0 = 10^{-3}$. We record the total number of degrees of freedom accumulated over time, as well as the global relative error

$$\text{Err} = \left(\frac{\sum_{n=0}^{N-1} \Delta t^n \sum_{i=0}^{M^n-1} \int_{x_i^n}^{x_{i+1}^n} \left(u_{ht}(t^{n+1},x) - u_e(t^{n+1},x)\right)^2 dx}{\sum_{n=0}^{N-1} \Delta t^n \sum_{i=0}^{M^n-1} \int_{x_i^n}^{x_{i+1}^n} \left(u_e(t^{n+1},x)\right)^2 dx} \right)^{1/2}, \tag{36}$$

where u_{ht} and u_e are, respectively, the numerical and exact solutions. Figure 6 compares the error between the uniform and adaptive methods on a logarithmic scale, showing the variation of $\log_{10}(Err)$ with respect to $\log_{10}(DOF)$, where DOF is the total number of degrees of freedom during all the adapted iterations. For both methods, the error decreases as the number of DOF increases and the adaptive method gives a smaller error than the uniform mesh method for a comparable number of DOF. Thus, this figure highlights the advantage of adaptive method versus the uniform method.

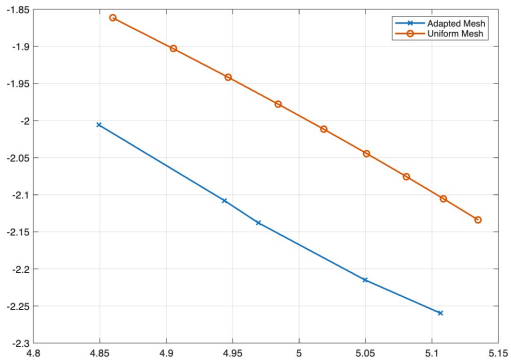

Fig. 6: Error-versus-DOF comparison for the direct problem: Variation of $\log_{10}(Err)$ with respect to $\log_{10}(DOF)$ for the adapted and uniform mesh. (Source: Created by the authors).

3.1.2 Non Smooth Initial Data

In this part, we consider the following non-smooth initial data

$$u_0(x) = a + b \cdot \begin{cases} x, & \text{if } x \le 0.5, \\ 1 - x, & \text{if } x > 0.5, \end{cases} \tag{37}$$

where a and b are positive constants. We apply the same previous adaptive space–time method and algorithm, with $a = 0.5$ and $b = 0.5$. Figure 7 (zoomed view in Figure 8) and Figure 9 (zoomed view in Figure 10) show the comparison between the numerical solution computed with the above algorithm and the exact solution at $t = 0.1$ and

$t = 0.3$, respectively, with a tolerance $tol_\varepsilon = 0.01$. Once again, we can clearly see that the numerical solution is very close to the exact one and that the mesh is concentrated in critical regions. We observe two regions where the refinement is concentrated. It is heavily refined near the peak of the solution, while significantly fewer refinement points are allocated in other regions (i.e., between $x \approx 0$ and $x \approx 0.1$ in Figure 9). While in the remaining parts of the domain, no additional refinement is performed due to the lack of significant variations.

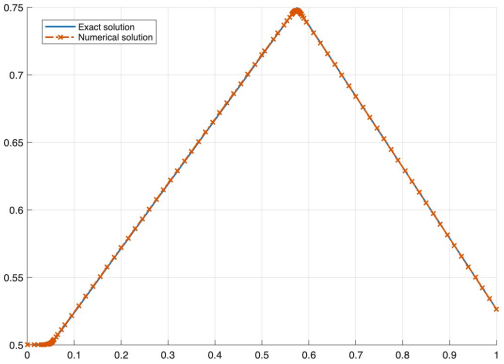

Fig. 7: Comparison of the numerical and exact solution, with $tol_\varepsilon = 0.01$ at $t = 0.1$ and $M = 108$. (Source: Created by the authors).

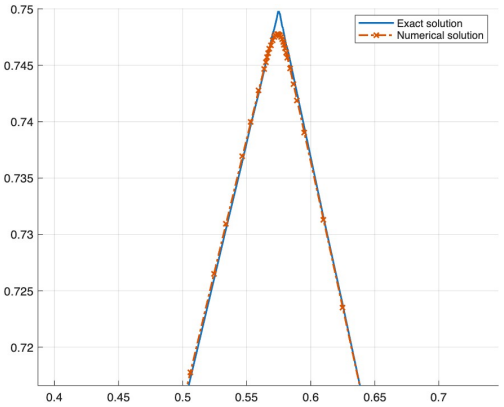

Fig. 8: Zoomed comparison of the numerical and exact solutions, with $tol_\varepsilon = 0.01$ at $t = 0.1$ and $M = 108$. (Source: Created by the authors).

As for the comparison between the adaptive and uniform mesh methods, we consider $T = 0.4$ and the tolerance $tol_\varepsilon \in \{0.04, 0.03, 0.02, 0.01\}$. For each tolerance, the solution is computed using the above adaptive algorithm, where the initial mesh is a uniform mesh with $M^0 = 50$ cells and the initial time step is $\Delta t^0 = 10^{-3}$. Figure 11 compares the error between the uniform and adaptive mesh methods for the inverse problem on a logarithmic scale, showing the variation of $\log_{10}(\text{Err})$ with respect to $\log_{10}(\text{DOF})$. For both methods, the error decreases as the number of degrees of freedom increases. Moreover, the adaptive mesh gives a smaller error

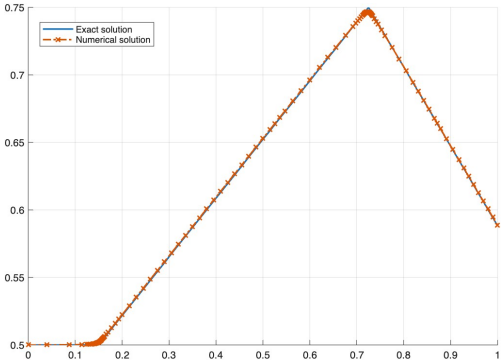

Fig. 9: Comparison of the numerical and exact solutions, with $tol_\varepsilon = 0.01$ at $t = 0.3$ and $M = 118$. (Source: Created by the authors).

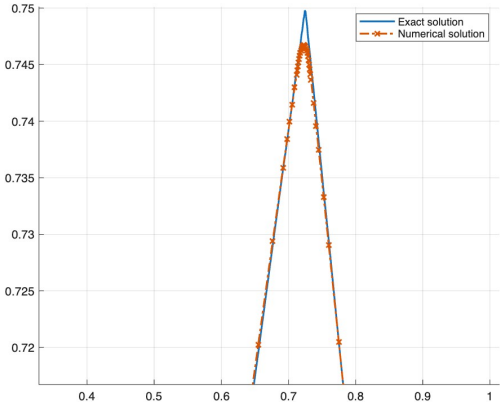

Fig. 10: Zoomed comparison of the numerical and exact solutions, with $tol_\varepsilon = 0.01$ at $t = 0.3$ and $M = 118$. (Source: Created by the authors).

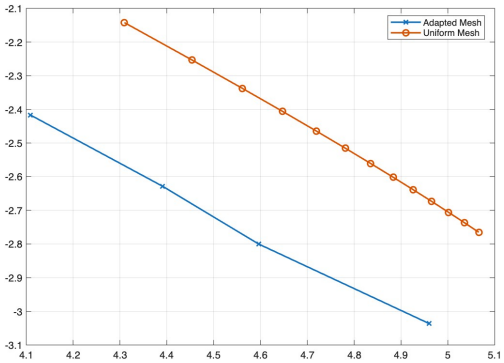

Fig. 11: Error-versus-DOF comparison for the direct problem: Variation of $\log_{10}(Err)$ with respect to $\log_{10}(DOF)$ for the adapted and uniform mesh. (Source: Created by the authors).

than the uniform mesh for a similar number of degrees of freedom. This shows the advantage of the adaptive method versus the uniform method.

3.2 Adaptive Algorithm in Space and Time: Inverse Problem

In this part, using the same adaptive approach in space and time as in the previous section, we present numerical results for the inverse problem, where the goal is to reconstruct the initial data from boundary measurements over time, for both smooth and non-smooth initial data.

3.2.1 Smooth Initial Data

In this part, we consider Problem (20) completed with the smooth initial data (28). The inverse problem involves reconstructing the initial condition u_0 of the direct problem (2) given the data at the boundary $x = 1$ over time. To perform numerical simulations, we begin by solving the direct discrete problem (6) with the initial data (28). We assume that the final time T is sufficiently large so that u becomes constant in the domain $[0, 1]$ at the time T. Therefore, we compute $u^n_{M^n}$, $n = 0, \cdots, N$, which will be the boundary condition $\hat{\beta}^n = v^n_0 = u^{N-n}_{M^n}$ of the inverse problem (21). We note that the initial condition $\hat{\alpha}_0$ will be given as $\hat{\alpha}_0 = u^N_{M^N}$. Then we solve the problem (21), to recover the initial data $u^0_i = v^N_{M^N-i}$, $i = 0, \cdots M^N$, which will be compared to the exact initial data $u(x_i, 0) = u_0(x_i)$, $i = 0, \cdots M^N$.

In practice, the boundary measurements $\hat{\beta}^n$ used in the inverse problem are obtained from the numerical solution of the direct problem $u^{N-n}_{M^n}$ at the boundary $x = 1$. As in the direct problem, since the adaptive strategy modifies the time step during the iteration, the boundary data are interpolated in time using a high-order polynomial interpolation to obtain a continuous function also denoted by $\hat{\beta}(t)$.

Since the inverse problem for recovering the initial data from boundary measurements of the Burgers' equation may develop shocks, we resolve it by using a space-time adaptive method based on local error indicators, as already explained in the previous section. At each time step, following the indicators $\hat{\varepsilon}^n_{s,i}$ and $\hat{\varepsilon}^n_{t,i}$, we may adapt the time step or the mesh accordingly.

We consider $M^0 = 50$, $a = 0.5$, $b = 0.4$, $m = 1$ and $T = 2.5$. At each time step t^n, we compute the space and time indicators given by (26) and (27), namely $\hat{\varepsilon}^n_{s,i}$ and $\hat{\varepsilon}^n_{t,i}$, and use them to adapt the mesh and the time step. We define the local time and space indicators by:

$$
\begin{aligned}
(I^n_{t,i})^2 &= \int_{t^n}^{t^{n+1}} \int_{x^n_i}^{x^n_{i+1}} (\hat{\varepsilon}^n_{t,i})^2 \, dx \, dt, \\
(I^n_{s,i})^2 &= \int_{t^n}^{t^{n+1}} \int_{x^n_i}^{x^n_{i+1}} \left(\hat{\varepsilon}^n_{s,i}\right)^2 dx \, dt, \\
(\mathcal{V}^n_i)^2 &= \Delta t^n \int_{x^n_i}^{x^n_{i+1}} (v^{n+1}_h)^2 \, dx.
\end{aligned}
\tag{38}
$$

We also define the mean space indicator $\overline{I}^n$, the global space indicator $\mathcal{E}^n_s$ and the global time indicator $\mathcal{E}^n_t$ as in the previous section, and use the same adaptive algorithm established above.

To illustrate our work, we present numerical simulations showing the efficiency of the proposed algorithm and we begin with an initial time step $\Delta t^0 = 10^{-3}$ and an initial uniform mesh $M^0 = 50$. Figure 12 (zoomed view in Figure 13) shows the comparison between the exact and recovered initial data computed with the above algorithm with a tolerance $tol_\varepsilon = 0.01$. We can clearly see that the numerical solution is close to the exact one and that the mesh is concentrated in critical regions; it is heavily refined near the peak of the solution, while significantly fewer refinement points are allocated in other regions (i.e. between $x \approx 0$ and $x \approx 0.1$ in Figure 12), due to the absence of strong variations in the solution. Next, we aim to show the comparison between the adaptive mesh method and the uniform mesh method applied to the inverse problem. For the adaptive case, we consider $T = 2.5$ and the tolerance $tol_\varepsilon \in \{0.02, 0.01, 0.009, 0.008, 0.007, 0.006\}$. For each tolerance, the solution is computed using the adaptive algorithm described above, where the initial mesh is a uniform mesh with $M^0 = 50$ cells and the initial time step is $\Delta t^0 = 10^{-3}$. We record the total number of degrees of freedom accumulated over time, as well as the global relative error

$$
\text{Err} = \left(\frac{\displaystyle\sum_{n=0}^{N-1} \Delta t^n \sum_{i=0}^{M^n-1} \int_{x^n_i}^{x^n_{i+1}} \left(v_{ht}(t^{n+1}, x) - v_e(t^{n+1}, x)\right)^2 dx}{\displaystyle\sum_{n=0}^{N-1} \Delta t^n \sum_{i=0}^{M^n-1} \int_{x^n_i}^{x^n_{i+1}} \left(v_e(t^{n+1}, x)\right)^2 dx} \right)^{1/2},
\tag{39}
$$

where v_{ht} and v_e are the numerical and exact solutions of the inverse problem, respectively. Figure 14 compares the error between the uniform and adaptive mesh methods for the inverse problem on a logarithmic scale, showing the variation of $\log_{10}(\text{Err})$ with respect to $\log_{10}(\text{DOF})$. For both methods, the error decreases as the number of DOF increases. This shows the advantage of the adaptive method, which

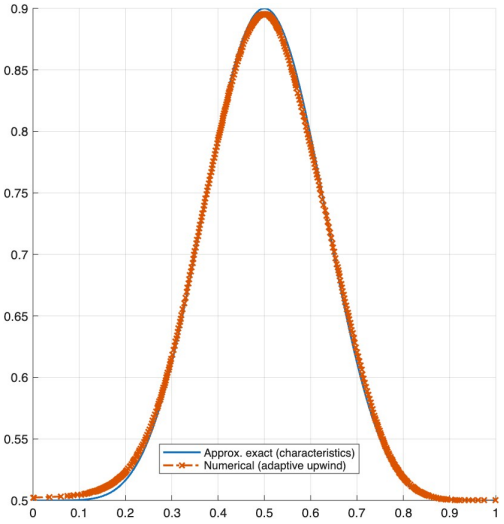

Fig. 12: Comparison of the recovered and exact initial data with $tol_\varepsilon = 0.01$ and $M = 438$. (Source: Created by the authors).

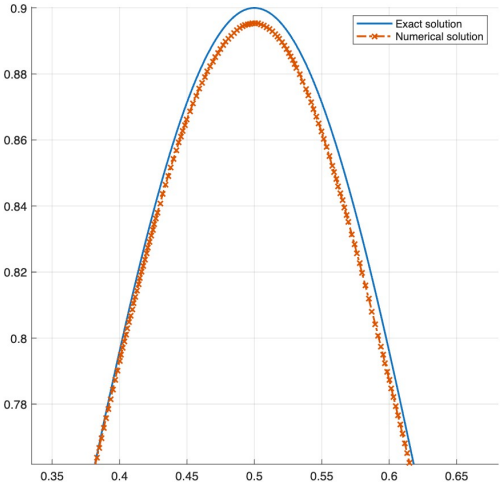

Fig. 13: Zoomed comparison of the recovered and exact initial data, with $tol_\varepsilon = 0.01$ and $M = 438$. (Source: Created by the authors).

concentrates the mesh in the critical regions where the solution presents significant variations.

3.2.2 Non-Smooth Initial Data

In this part, we apply the same previous adaptive space–time method for the inverse problem of recovering the initial data from boundary measurements at $x = 1$, considering the non-smooth initial data (37), with $a = 0.5$ and $b = 0.4$. Figure 15 shows the comparison between the exact and recovered non-smooth initial data computed with the above adaptive algorithm, with a tolerance $\text{tol}_\varepsilon = 0.01$. We note that the recovered initial data are close to the exact ones. The main difference appears near the peak at $x = 0.5$, where the initial data are non-smooth. The mesh is concentrated in this critical region, while fewer refinement points are

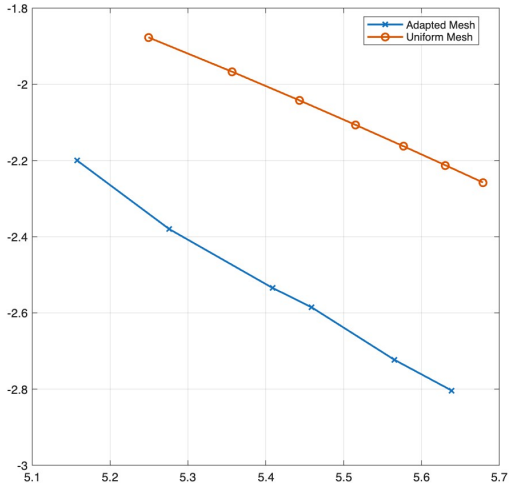

Fig. 14: Error-versus-DOF for the inverse problem: Variation of $\log_{10}(Err)$ with respect to $\log_{10}(DOF)$ for the adapted and uniform mesh. (Source: Created by the authors).

allocated in the other regions where the initial data present fewer variations. Figure 16 shows a zoomed view around the peak, where the difference between the exact and recovered initial data can be seen more clearly. The recovered initial data remain close to the exact ones.

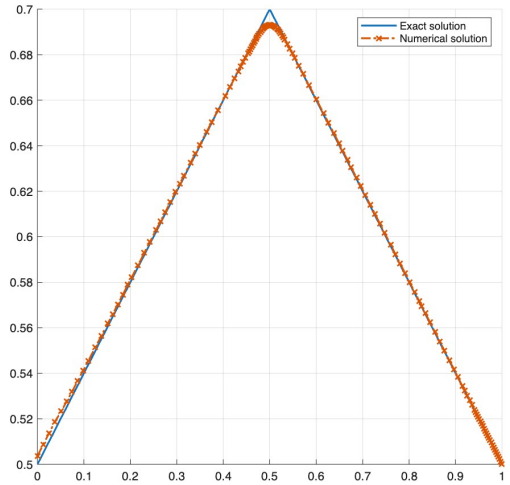

Fig. 15: Comparison of the exact and recovered initial data for $tol_\varepsilon = 0.01$ ($M = 140$). (Source: Created by the authors).

As for the comparison between the adaptive and uniform mesh methods for the inverse problem, we consider $T = 2.5$ and the tolerance $tol_\varepsilon \in \{0.02, 0.01, 0.009, 0.008, 0.007, 0.006\}$. For each tolerance, the solution is computed using the above adaptive algorithm, where the initial mesh is a uniform mesh with $M^0 = 50$ cells and the initial time step is $\Delta t^0 = 10^{-3}$. Figure 17 compares the errors of the uniform and adaptive mesh methods for the inverse problem on a logarithmic scale. For both methods, the error decreases as the DOF increases,

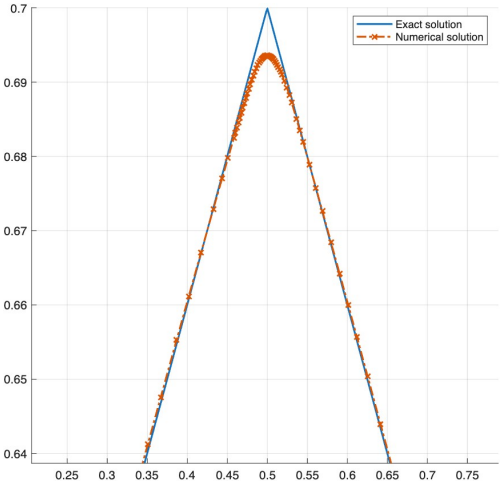

Fig. 16: Zoomed comparison of the exact and recovered initial data around $x = 0.5$ ($tol_\varepsilon = 0.01, M = 140$). (Source: Created by the authors).

while the adaptive mesh gives a smaller error for a similar DOF. This shows the advantage of the adaptive method, which concentrates the mesh in critical regions where the solution presents significant variations.

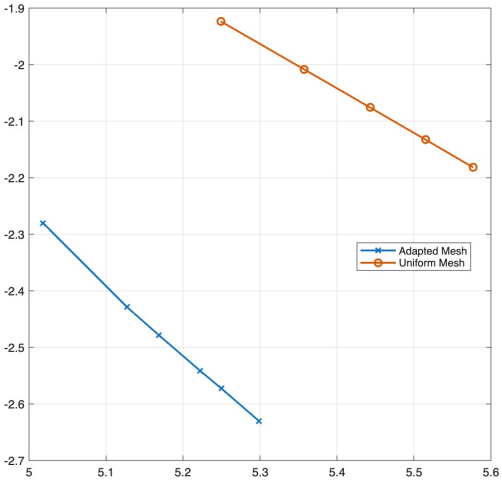

Fig. 17: Error-versus-DOF for the inverse problem: Variation of $\log_{10}(Err)$ with respect to $\log_{10}(DOF)$ for the adaptive and uniform mesh. (Source: Created by the authors).

References:

[1] L. C. Evans, *Partial Differential Equations*, 2nd ed., Graduate Studies in Mathematics, Vol. 19, American Mathematical Society, Providence, RI, 2010. ISBN: 978-0-8218-4974-3. DOI: 10.1090/gsm/019.

[2] R. J. LeVeque, *Numerical Methods for Conservation Laws*, 2nd ed., Lectures in Mathematics ETH Zurich, Birkhäuser, Basel, 1992. ISBN: 978-3-7643-2723-1. DOI: 10.1007/978-3-0348-8629-1.

[3] R. J. DiPerna and P. L. Lions, Ordinary differential equations, transport theory and Sobolev spaces, *Inventiones Mathematicae*, Vol. 98, No. 3, 1989, pp. 511–547. DOI: 10.1007/BF01393835.

[4] S. Mishra, U. S. Fjordholm, and R. Abgrall, *Numerical Methods for Conservation Laws and Related Equations*, Lecture Notes, 2019. `https://metaphor.ethz.ch/x/2019/hs/401-4671-00L/literature/mishra_hyperbolic_pdes.pdf`. (Accessed date: June 12, 2026).

[5] L. Safetly, T. Sayah, and F. Triki, Inverse problem for Burgers' equation: recovering Riemann initial data, *Applicable Analysis*, Vol. 105, No. 12, 2026, pp. 2616–2661. DOI: 10.1080/00036811.2026.2618981.

[6] J. M. Burgers, A mathematical model illustrating the theory of turbulence, *Advances in Applied Mechanics*, Vol. 1, 1948, pp. 171–199. DOI: 10.1016/S0065-2156(08)70100-5.

[7] H. Bateman, Some recent researches on the motion of fluids, *Monthly Weather Review*, Vol. 43, No. 4, 1915, pp. 163–170. DOI: 10.1175/1520-0493(1915)43<163:SRROTM>2.0.CO;2.

[8] S. N. Kruzkov, First order quasilinear equations in several independent variables, *Mathematics of the USSR-Sbornik*, Vol. 10, No. 2, 1970, pp. 217–243. DOI: 10.1070/SM1970v010n02ABEH002156.

[9] E. Godlewski and P. A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Applied Mathematical Sciences, Vol. 118, Springer, New York, 1996. ISBN: 978-0-387-94529-3. DOI: 10.1007/978-1-4612-0713-9.

[10] O. A. Oleinik, Uniqueness and stability of the generalized solution of the Cauchy problem for a quasi-linear equation, *Uspekhi Matematicheskikh Nauk*, Vol. 14, No. 2(86), 1959, pp. 165–170 (text in Russian); English translation, *American Mathematical Society Translations*, Ser. 2, Vol. 33, 1963, pp. 285–290. `https://www.mathnet.ru/eng/rm7292`. (Accessed date: July 29, 2026).

[11] P. D. Lax, Hyperbolic systems of conservation laws II, *Communications on Pure and Applied Mathematics*, Vol. 10, No. 4, 1957, pp. 537–566. DOI: 10.1002/cpa.3160100406.

[12] F. Dubois and P. G. Le Floch, Boundary conditions for nonlinear hyperbolic systems of conservation laws, *Journal of Differential Equations*, Vol. 71, No. 1, 1988, pp. 93–122. DOI: 10.1016/0022-0396(88)90040-X.

[13] C. Bardos, A. Y. Leroux, and J. C. Nedelec, First order quasilinear equations with boundary conditions, *Communications*

in Partial Differential Equations, Vol. 4, No. 9, 1979, pp. 1017–1034. DOI: 10.1080/03605307908820117.
[14] K. Ammar, P. Wittbold, and J. Carrillo, Scalar conservation laws with general boundary condition and continuous flux function, *Journal of Differential Equations*, Vol. 228, No. 1, 2006, pp. 111–139. DOI: 10.1016/j.jde.2006.05.002.
[15] T. Liard and E. Zuazua, Initial data identification for the one-dimensional Burgers equation, *IEEE Transactions on Automatic Control*, Vol. 67, No. 6, 2022, pp. 3098–3104. DOI: 10.1109/TAC.2021.3096921.
[16] T. Liard and E. Zuazua, Analysis and numerical solvability of backward-forward conservation laws, *SIAM Journal on Mathematical Analysis*, Vol. 55, No. 3, 2023, pp. 1949–1968. DOI: 10.1137/22M1478720.
[17] F. Boyer, Analysis of the upwind finite volume method for general initial- and boundary-value transport problems, *IMA Journal of Numerical Analysis*, Vol. 32, No. 4, 2012, pp. 1404–1439. DOI: 10.1093/imanum/drr031.
[18] M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *Journal of Computational Physics*, Vol. 82, No. 1, 1989, pp. 64–84. DOI: 10.1016/0021-9991(89)90035-1.
[19] K. Çar, *Adaptive Mesh Refinement for One-Dimensional Scalar Conservation Laws*, Master's Thesis, Middle East Technical University, Ankara, Turkey, 2024. https://tezara.org/theses/892626. (Accessed date: July 20, 2026).
[20] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, Introduction to adaptive methods for differential equations, *Acta Numerica*, Vol. 4, 1995, pp. 105–158. DOI: 10.1017/S0962492900002531.
[21] D. Kröner and M. Ohlberger, A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions, *Mathematics of Computation*, Vol. 69, No. 229, 2000, pp. 25–39. DOI: 10.1090/S0025-5718-99-01158-8.
[22] T. J. Barth and M. G. Larson, A Posteriori Error Estimates for Higher Order Godunov Finite Volume Methods on Unstructured Meshes, NASA Technical Report NAS-02-001, NASA Ames Research Center, 2002. https://www.nas.nasa.gov/assets/nas/pdf/techreports/2002/nas-02-001.pdf. (Accessed date: August 15, 2026).
[23] Ch. Arvanitis and A. I. Delis, Behavior of finite volume schemes for hyperbolic conservation laws on adaptive redistributed spatial grids, *SIAM Journal on Scientific Computing*, Vol. 28, No. 5, 2006, pp. 1927–1956. DOI: 10.1137/050632853.
[24] T. S. Phillips, C. J. Roy, E. J. Alyanak, and C. F. Ollivier-Gooch, Optimal mesh adaption for Burgers' equation, *42nd AIAA Fluid Dynamics Conference and Exhibit*, New Orleans, Louisiana, USA, June 25–28, 2012, AIAA Paper 2012-2710. DOI: 10.2514/6.2012-2710.
[25] L. Safetly and T. Sayah, On the Inverse Problem for the Burgers' Equation with Smooth Initial Data in a Bounded Domain, *Journal of Applied Analysis and Computation*, Vol. 17, No. 2, 2027, pp. 1014–1037. DOI: 10.11948/20260119.
[26] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, 2nd ed., SIAM, Philadelphia, 2004. ISBN: 978-0-89871-567-5. DOI: 10.1137/1.9780898717938.
[27] B. Cockburn and H. Gau, A posteriori error estimates for general numerical methods for scalar conservation laws, *Matemática Aplicada e Computacional*, Vol. 14, No. 1, 1995, pp. 37–47. https://conservancy.umn.edu/handle/11299/2654 (Accessed date: April 30, 2026).
[28] J. Giesselmann, C. Makridakis, and T. Pryer, A posteriori analysis of discontinuous Galerkin schemes for systems of hyperbolic conservation laws, *SIAM Journal on Numerical Analysis*, Vol. 53, No. 3, 2015, pp. 1280–1303. DOI: 10.1137/140970999
[29] G. Sialounas, *Reconstruction Based Error Detection for Robust Approximation of Partial Differential Equations*, Doctoral Dissertation, University of Reading, 2022. DOI: 10.48683/1926.00110385

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

The authors declare that no funding was received for conducting this research or for the preparation of this article.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US