axioms

MDPI

Article

Numerical Treatment of Hyperbolic-Type Problems with Single and Double Interfaces via Meshless Method

Muhammad Asif [1,*], Naveed Akhtar [1], Farhan Khan [1], Faisal Bilal [1] and Ioan-Lucian Popa [2,3,*]

[1] Department of Mathematics, University of Peshawar, Peshawar 25120, Pakistan
[2] Department of Computing, Mathematics and Electronics, "1 Decembrie 1918" University of Alba Iulia, 510009 Alba Iulia, Romania
[3] Faculty of Mathematics and Computer Science, Transilvania University of Brasov, Iuliu Maniu Street 50, 500091 Brasov, Romania
* Correspondence: asif.tangi@uop.edu.pk (M.A.); lucian.popa@uab.ro (I.-L.P.)

Abstract

Hyperbolic interface problems frequently arise in a wide range of scientific and engineering applications, particularly in scenarios involving wave propagation or transport phenomena across media with discontinuous properties. These problems are characterized by abrupt changes in material coefficients or domain features, which pose significant challenges for numerical approximation. In this study, we propose an efficient and robust computational framework for solving one-dimensional hyperbolic interface problems with both single and double interfaces. The methodology combines the finite difference method (FDM) for time discretization with meshless radial basis functions (RBFs) for spatial approximation, enabling accurate resolution of interface discontinuities. This hybrid approach is adaptable to both linear and nonlinear models and is capable of handling constant as well as variable coefficients. Linear systems are solved using Gaussian elimination, while nonlinear systems are addressed through a quasi-Newton linearization method. To validate the performance of the proposed method, we compute the maximum absolute errors (MAEs) and root mean square errors (RMSEs) over various spatial and temporal discretizations. Numerical experiments demonstrate that the approach exhibits fast convergence, excellent accuracy, and ease of implementation, making it a practical tool for solving complex hyperbolic problems with interface conditions. Overall, the method provides a reliable and scalable solution for a class of problems where traditional numerical techniques often discontinuties.

Keywords: hyperbolic partial differential equation; interface models; radial basis functions; meshless method; finite difference method

MSC: 65Mxx

check for updates

Academic Editor: Roman Dmytryshyn

Received: 8 July 2025
Revised: 3 August 2025
Accepted: 6 August 2025
Published: 8 August 2025

Citation: Asif, M.; Akhtar, N.; Khan, F.; Bilal, F.; Popa, I.-L. Numerical Treatment of Hyperbolic-Type Problems with Single and Double Interfaces via Meshless Method. *Axioms* **2025**, *14*, 621. https://doi.org/10.3390/axioms14080621

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Mathematical models are widely used in science and engineering to solve a range of difficult problems, usually in the form of equations, especially partial differential equations (PDEs). From engineering and biological systems to electrostatics and chemical processes, electroshock therapy and medicine, and quantum physics and fluid dynamics, PDEs have been embraced as a common language in many different domains. Usually, PDEs fall into three main groups: elliptic, parabolic, and hyperbolic. Among the several uses of hyperbolic PDEs that have attracted great attention are aerospace engineering, atomic physics, industrial processes, structural vibrations, biological modeling, and mechanical

system analysis, including beams and structures. Many technical and scientific fields call for hyperbolic PDEs. An even more compact example is the wave equation, which describes not only the transmission of electromagnetic waves and sounds as well as longitudinal vibrations in bars but also the oscillation of strings with fixed ends. Furthermore, these equations are rather common in the field of computational fluid dynamics. In the realm of gas dynamics, Euler's equations are usually categorized as hyperbolic PDEs [1–3].

Domains are divided into numerous subregions, each of which is governed by specific mathematical or physical laws, and various interfaces are constructed within them. Composite materials, multiphase fluid dynamics, and similar applications demonstrate these phenomena. More classically, the problems of heat diffusion and wave propagation in non-uniform media are regarded as parabolic and hyperbolic interface problems, respectively. However, the achievement of precise analytical solutions is rare and conventional numerical methods often encounter significant challenges. The effectiveness of conventional techniques, such as finite difference and finite element methods, is significantly diminished, and they become unreliable when these interface problems produce discontinuous or weakly continuous solutions, as is frequently the case. The result has been an increase in the demand for numerical methods that are flexible and robust and can simulate the resolution of multifaceted subregions [4]. Fracture mechanics, fluid mechanics, electromagnetism, biology, and materials science are among the numerous disciplines in which interface models are applicable. In many disciplines, including materials engineering, biological sciences, fracture mechanics, fluid mechanics, and electromagnetic wave propagation, interface models are fundamental. The simulation of real-world systems depends on these models [5,6]. Usually, they correctly depict complicated interactions between several physical processes and materials using ordinary or partial differential equations with abrupt changes in values and specific conditions at the boundaries. Two materials in different physical states share a boundary [7–9], so examples include the coexistence of ice and liquid water or the interaction of oil and water. As in biological and physical models, these systems are characterized by coefficients that vary significantly or are not continuous; hence, they require sophisticated numerical approaches to identify true solutions.

Because of their inherent mathematical and physical complexity, a variety of computing techniques have been developed to effectively solve problems with both regular and complex geometric domains. This includes techniques such as the immersed method [10,11] and ghost fluid method [12–14], the immersed interface method [7,15], and several boundary-fitted approaches, including the matched interface technique [16–18]. Whereas Aziz et al. broadened this strategy to address parabolic interface equations, Haider et al. [19] studied the application of the Haar wavelet collocation method (HWCM) in conjunction with meshless techniques to resolve an elliptic equation with an interface [20]. Furthermore, finite element methods (FEMs) are applied in hyperbolic interface models [21]. Asif et al. improved the application of the HWCM for interface modeling [22] and then extended it to hyperbolic problems with double interfaces [23]. Although Zou used the finite element method (FEM) to handle both the heat and the elliptic interface problems [24], Meng presented the Galerkin technique for the matched interface and boundary (MIB) for elliptic interface equations [25]. Similar approaches have been used for elliptic interface equations such as well [26], and Siraj et al. used local meshless methods to solve the Stokes equations with interface discontinuities [27]. Using a mesh-free technique with a single interface [28], Asif and Faisal solved the nonlinear telegraph equation. Asif et al. numerically approximated the telegraph equation with double interfaces [29] using the HWCM. Asif et al. also numerically approximated the one-dimensional (1D) and three-dimensional (3D) telegraph equations with single interfaces [30,31] using the HWCM. Unlike existing methods, such as the HWCM, previously applied to hyperbolic problems with interfaces,

the present study introduces a novel hybrid approach that combines the meshless method based on RBFs with the FDM for temporal discretization. This combination leverages the flexibility of RBFs to handle complex geometries and discontinuities, while FDM ensures efficient and accurate time evolution. Compared to traditional mesh-based techniques, the proposed method avoids mesh generation and exhibits superior performance in capturing sharp gradients and interface discontinuities, as demonstrated by several benchmark problems involving both linear and nonlinear models with single and double interfaces. The method also proves to be highly stable, convergent, and effective even in the presence of strong discontinuities, offering a significant advancement over existing numerical schemes.

Mesh-free approximation approaches first inspired development in geodesy, geophysics, mapping, and meteorology. These techniques increasingly found uses over time in disciplines including learning theory, artificial intelligence, numerical solutions to partial differential equations, signal processing, sampling theory, neural networks, statistics (e.g., kriging), finance, and optimization. Mesh-free methods have several main benefits since they allow one to manage complicated geometries and huge deformations without the need for mesh generation or refinement. In 3D situations, including irregular geometries, free surfaces, or contact interfaces, where mesh production can be time-consuming and prone to errors [32], this becomes very valuable. However, mesh-free methods are not without limitations. They often suffer from poor system conditioning and can struggle in scenarios involving discontinuities or steep gradients, such as shock waves.

The foundation for a mesh-free approximation based on RBFs was laid by Hardy in 1971 with the development of the multiquadric method [33]. Later, Tarwater introduced the concept of a shape parameter in the mesh-free approximation for the first time [34]. Michelli contributed significantly by studying the uniqueness, convergence and existence of RBF approximations [35]. Kansa further advanced the approach by adapting the multiquadric method to solve various types of PDEs [36]. Madych and Nelson also made important contributions by examining the existence, convergence, and uniqueness of RBFs for different types of problems [37]. Golberg extended Kansa's work to a wider class of PDEs [38]. Hon and Mao applied RBFs to approximate solutions of the non-linear Burgers equation, which includes shock waves [39].

Our paper presents an innovative numerical technique that combines RBFs and FDM. We present this technique in our paper. This hybrid approach can be used to calculate the numerical solution for 1D linear and nonlinear hyperbolic interface models that have single and double interfaces. The following section provides an outline of the equations that are of interest:

$$u_{\tau\tau}(s,\tau) = (\beta(s)u_s(s,\tau))_s + f(s,\tau), \quad 0 < s < 1, \quad \tau > 0, \tag{1}$$

and nonlinear equation is

$$u_{\tau\tau}(s,\tau) = (\beta(s)u_s(s,\tau))_s + \psi(s,\tau) + f(s,\tau), \quad 0 < s < 1, \quad \tau > 0. \tag{2}$$

In Equations (1) and (2), $u(s,\tau)$ denotes the unknown solution function, where $s \in (0,1)$ is the spatial variable and $\tau > 0$ is the time variable. The function $\beta(s)$ represents a spatially variable wave propagation or diffusion coefficient, which may have discontinuities at the interface points. The term $f(s,\tau)$ is a known source or forcing function. The notation $u_{\tau\tau}$, u_s, and u_{ss} refer to the second-order time derivative, the first-order spatial derivative, and the second-order spatial derivative of u, respectively. The operator $\psi(s,\tau)$ in Equation (2) represents a general nonlinear term. In this study, we choose a nonlinear polynomial type term given by $\psi(s,\tau) = 2u^2$.

1.1. Single Interface

For models involving single-interface conditions at the interface point $s = \alpha$, the domain $I = [0,1]$ is divided into two subdomains $I_1 = [0,\alpha]$ and $I_2 = [\alpha,1]$. The functions involved in Equations (1) and (2) have the following forms:

$$(\beta(s), u(s,\tau), f(s,\tau)) = \begin{cases} \beta_1(s), u_1(s,\tau), f_1(s,\tau)), & s \in I_1, \\ \beta_2(s), u_2(s,\tau), f_2(s,\tau)), & s \in I_2. \end{cases}$$

The initial conditions are

$$\begin{aligned} u(s,0) &= u_0(s), \quad s \in I, \\ u_\tau(s,0) &= u_{\tau_0}(s). \end{aligned}$$

The Dirichlet boundary conditions are

$$\begin{aligned} u_1(0,\tau) &= g(0,\tau), \\ u_2(1,\tau) &= g(1,\tau) \quad \tau > 0. \end{aligned}$$

The interface conditions at $s = \alpha$ are

$$u_1(\alpha,\tau) - u_2(\alpha,\tau) = q_1(\tau) \quad \tau > 0, \tag{3}$$

$$\beta_1 u_{1s}(\alpha,\tau) - \beta_2 u_{2s}(\alpha,\tau) = q_2(\tau) \quad \tau > 0, \tag{4}$$

where the functions $f_1(s,\tau)$, $f_2(s,\tau)$ $\beta_1(s) \geq 1$, and $\beta_2(s) \geq 1$ are smooth functions defined on the corresponding sub-domains I_1 and I_2 of domain I.

1.2. Double Interface

In the presence of two interface points within the domain, specifically at $x = \alpha$ and $x = \beta$ with $0 < \alpha < 1$ and $0 < \beta < 1$, the interval $I = [0,1]$ is partitioned into three subintervals: $I_1 = [0,\alpha]$, $I_2 = [\alpha,\beta]$, and $I_3 = [\beta,1]$. The functions referenced in Equations (1) and (2) are then represented accordingly.

$$(\beta_1(s), u(s,\tau), f(s,\tau)) = \begin{cases} (\beta_1(s), u_1(s,\tau), f_1(s,\tau)), & s \in I_1, \\ (\beta_2(s), u_2(s,\tau), f_2(s,\tau)), & s \in I_2, \\ (\beta_3(s), u_3(s,\tau), f_3(s,\tau)), & s \in I_3. \end{cases} \tag{5}$$

The Dirichlet boundary conditions at the endpoints $s = 0$ and $s = 1$ are specified as

$$u(0,\tau) = \delta_3(\tau), \quad \text{and} \quad u(1,\tau) = \delta_4(\tau). \tag{6}$$

The interface conditions at the junctions $x = \alpha$ and $x = \beta$ are expressed by the following:

$$\sigma_1 u_1(\alpha,\tau) - \sigma_2 u_2(\alpha,\tau) = \varpi_1(\tau), \tag{7}$$

$$\sigma_3 u_{1s}(\alpha,\tau) - \sigma_4 u_{2s}(\alpha,\tau) = \varpi_2(\tau), \tag{8}$$

$$\sigma_5 u_2(\beta,\tau) - \sigma_6 u_3(\beta,\tau) = \varpi_3(\tau), \tag{9}$$

$$\sigma_7 u_{2s}(\beta,\tau) - \sigma_8 u_{3s}(\beta,\tau) = \varpi_4(\tau). \tag{10}$$

The initial condition for the solution is defined by

$$u(s,0) = u_0(s), \quad s \in I. \tag{11}$$

The coefficient functions $\beta_1(s)$, $\beta_2(s)$, and $\beta_3(s)$ are all greater than or equal to 1, i.e., $\beta_1(s) \geq 1$, $\beta_2(s) \geq 1$, and $\beta_3(s) \geq 1$. The source terms $f_1(s,\tau)$, $f_2(s,\tau)$, and $f_3(s,\tau)$ are assumed to be smooth enough on their respective subintervals of I. Furthermore, all coefficients, σ_1, σ_2, σ_3, σ_4, σ_5, σ_6,σ_7, and σ_8, are constants and do not vary over time.

The remainder of this article is structured as follows. Section 2 provides a brief overview of radial basis functions (RBFs). Section 3 outlines the formulation of the single-interface model, while Section 4 addresses the formulation of the double-interface model. Stability analysis is carried out in Section 5. Section 6 presents the numerical experiments that validate the proposed methodology. Section 7 presents the concluding remarks and highlights potential directions for future work.

2. Radial Basis Functions

The function $\phi : \mathbb{R}^d \longrightarrow \mathbb{R}$ is classified as a radial function if there arises a single-valued function $\psi : [0,\infty) \longrightarrow \mathbb{R}$ such that $\phi(y) = \psi(||y||)$ holds. Therefore, the value of $\phi(y)$ depends on the magnitude of $||y||$. The function $\psi(s)$, with $s = ||y||$, is designated as an RBF and is defined based on the Euclidean norm for $s \geq 0$. Every RBF possesses a parameter that exceeds zero. This is typically designated as the form parameter, with its value expressed as $\varepsilon = 1/\delta$. We articulate the mathematical definition of multiquadric (MQ) RBFs as follows:

$$\phi(y) = \sqrt{1 + (\delta y)^2},$$

where

$$\phi(||s - s_i||) = \sqrt{1 + (\delta(s - s_i))^2},$$

where y = $||s - s_i||$ and s is the center point [32].

In this approximation, the function $u(s,\tau)$ is expressed as a linear combination of radial basis functions centered at spatial points s_i:

$$u(s,\tau) = \sum_{i=1}^{F} \lambda_i \, \phi(\|s - s_i\|)$$

where λ_i are unknown weights, ϕ is the MQ-RBF, and s_i are the center nodes. Each RBF ϕ depends only on the Euclidean distance between the evaluation point s and its corresponding center s_i and does not depend on the weight λ_i.

By enforcing the RBF interpolation condition at all collocation points s_i, we obtain a linear system for the unknown coefficients λ_j. Specifically, we require that

$$u(s_i) = \sum_{j=1}^{F} \lambda_j \, \phi(\|s_i - s_j\|), \quad i = 1, 2, \ldots, F. \tag{12}$$

This leads to the following matrix system:

$$\underbrace{\begin{bmatrix} \phi(\|s_1 - s_1\|) & \cdots & \phi(\|s_1 - s_F\|) \\ \vdots & \ddots & \vdots \\ \phi(\|s_F - s_1\|) & \cdots & \phi(\|s_F - s_F\|) \end{bmatrix}}_{A_{F\times F}} \underbrace{\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_F \end{bmatrix}}_{\boldsymbol{\lambda}} = \underbrace{\begin{bmatrix} u_1 \\ \vdots \\ u_F \end{bmatrix}}_{\mathbf{u}} \tag{13}$$

In the above system, the matrix $A \in \mathbb{R}^{F\times F}$ contains the evaluations of the radial basis function ϕ between all pairs of spatial nodes. The vector $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_F]^T$ represents the unknown coefficients to be determined, while $\mathbf{u} = [u_1, u_2, \ldots, u_F]^T$ contains the known

values of the function u at the collocation points. Solving this system yields the RBF interpolation coefficients, which are used in the meshless approximation of the solution.

3. Meshless Collocation Method for Single-Interface Model

First of all, we partition the subintervals $I_1 = [0, \alpha]$ and $I_2 = [\alpha, 1]$. The partition of $I_1 = [0, \alpha]$ is given by $0 = x_1 < x_2 < x_3 \ldots x_{E_1+1} = \alpha$ and $\alpha = x_{E_1+1} < E_1 + 2 < E_1 + 3 \ldots x < x_F = 1$, respectively, where $F = E_1 + E_2 + 1$ is the total number of grid points and E_1 and E_2 are the numbers of subintervals of $I_1 = [0, \alpha]$ and $I_2 = [\alpha, 1]$, respectively.

We compute the time derivative by employing the backward difference technique.

$$u_\tau(s,\tau) = \frac{1}{d\tau}[u(s,\tau) - u(s,\tau_0)], \tag{14}$$

$$u(s,\tau_j) = u(s,\tau_0) + u_t(s,\tau_0)d\tau. \tag{15}$$

We compute the time derivative by employing the central difference technique.

$$u_{\tau\tau}(s,\tau) = \frac{1}{d\tau^2}[u(s,\tau_1) - 2u(s,\tau) + u(s,\tau_0)], \tag{16}$$

$$u_{\tau\tau}(s,\tau) = \frac{1}{d\tau^2}[u(s,\tau_1) - 2(u(s,\tau_0) + u_t(s,\tau_0)d\tau) + u(s,\tau_0)], \tag{17}$$

where τ_1, τ_0, and τ are the next, current and initial time intervals.

Then we approximate the unknown function $u_1(s,\tau)$ over the interval I = $[0, \alpha)$ using an MQ RBF:

$$u_1(s,\tau) = \sum_{i=1}^{E+1} \lambda_i \phi(||s - s_i||), \quad s \in I_1. \tag{18}$$

The result of differentiating with respect to s is

$$u_{1s}(s,\tau) = \sum_{i=1}^{E+1} \lambda_i \frac{d\phi}{ds}(||s - s_i||), \quad s \in I_1. \tag{19}$$

Again, the result of differentiating with respect to s is

$$u_{1ss}(s,\tau) = \sum_{i=1}^{E+1} \lambda_i \frac{d^2\phi}{ds^2}(||s - s_i||), \quad s \in I_1. \tag{20}$$

Similarly, approximating the unknown function $u_2(s,t)$ over the subinterval $I_2 = [\alpha, 1]$ using MQ RBFs, we obtain

$$u_2(s,\tau) = \sum_{i=E+1}^{F} \gamma_i \phi(||s - s_i||), \quad s \in I_2. \tag{21}$$

The result of differentiating with respect to s is

$$u_{2s}(s,\tau) = \sum_{i=E+1}^{F} \gamma_i \frac{d\phi}{ds}(||s - s_i||), \quad s \in I_2. \tag{22}$$

Again, the result of differentiating with respect to s is

$$u_{2ss}(s,\tau) = \sum_{i=E+1}^{F} \gamma_i \frac{d^2\phi}{ds^2}(||s - s_i||), \quad s \in I_2. \tag{23}$$

Now approximating the boundaries $s = 0, 1$ by using Equations (18) and (21) yields

$$\sum_{i=1}^{E+1} \lambda_i \phi(||0 - s_i||) = \Theta_1(t), \tag{24}$$

$$\sum_{i=E+1}^{F} \gamma_i \phi(||1 - s_i||) = \Theta_2(t), \tag{25}$$

where $\Theta_1(t) = u(0,t)$ and $\Theta_2(t) = u(1,t)$.

Similarly, the meshless estimation for the interface conditions given in Equations (3) and (4) becomes

$$\sum_{i=1}^{E+1} \lambda_i \phi(||s - s_i||) - \sum_{i=E+1}^{F} \gamma_i \phi(||s - s_i||) = q_1(t), \tag{26}$$

$$\beta_1 \left(\sum_{i=1}^{E+1} \lambda_i \frac{d\phi}{ds}(||s - s_i||) \right) - \beta_2 \left(\sum_{i=E+1}^{F} \gamma_i \frac{d\phi}{ds}(||s - s_i||) \right) = q_2(t). \tag{27}$$

We will solve the above systems for linear and nonlinear cases separately.

3.1. Linear Case

Substituting the MQ BRF estimations into Equation (1), we obtain the following expressions:

$$\begin{aligned} \sum_{i=1}^{E+1} \lambda_i \frac{d\phi}{ds}(||s - s_i||) - 2(u_1(s,\tau_0) - u_1(s,\tau_0)d\tau) + u_1(s,\tau_0) = \\ \beta_1 \sum_{i=1}^{E+1} \lambda_i \frac{d^2\phi}{ds^2}(||s - s_i||)d\tau^2 + f_1(s,\tau)d\tau^2, \quad s \in I_1. \end{aligned} \tag{28}$$

For $u_2(s,t)$, we have

$$\begin{aligned} \sum_{i=E+1}^{F} \gamma_i \frac{d\phi}{ds}(||s - s_i||) - 2(u_2(s,\tau_0) - u_2(s,\tau_0)d\tau) + u_2(s,\tau_0) = \\ \beta_2 \sum_{i=E+1}^{F} \gamma_i \frac{d^2\phi}{ds^2}(||s - s_i||)d\tau^2 + f_2(s,\tau)d\tau^2, \quad s \in I_2. \end{aligned} \tag{29}$$

After discretization, we obtain the following expressions for $u_1(s,\tau)$ and $u_2(s,\tau)$, respectively:

$$\begin{aligned} \sum_{i=1}^{E+1} \lambda_i \frac{d\phi}{ds}(||s_j - s_i||) - \beta_1 \sum_{i=1}^{E+1} \lambda_i \frac{d^2\phi}{ds^2}(||s_j - s_i||)d\tau^2 = 2(u_1(s_j,\tau_0) \\ -u_1(s_j,\tau_0)d\tau) - u_1(s_j,\tau_0) + f_1(s_j,\tau)d\tau^2, \quad s \in I_1, \end{aligned} \tag{30}$$

and

$$\begin{aligned} \sum_{i=E+1}^{F} \gamma_i \frac{d\phi}{ds}(||s_j - s_i||) - \beta_2 \sum_{i=E+1}^{F} \gamma_i \frac{d^2\phi}{ds^2}(||s_j - s_i||)d\tau^2 = 2(u_2(s_j,\tau_0) \\ -u_2(s_j,\tau_0)d\tau) - u_2(s_j,\tau_0) + f_2(s_j,\tau)d\tau^2, \quad s \in I_2. \end{aligned} \tag{31}$$

A linear system of equations with $F + 1$ unknowns is represented by Equations (26) and (27), as well as Equations (30) and (31). Both $\lambda_i(\tau), i = 1, 2, 3, \ldots, E_1 + 1$ and $\gamma_i(\tau), i =$

$E_1+1, E_1+2, E_1+3, \ldots, F$, respectively, are used in the expression. The following is an example of how the linear system can be written using matrix notation:

$$\mathbf{AX} = \mathbf{D}, \tag{32}$$

The coefficient systems are

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1,E_1+1} & 0 & \cdots & 0 \\ a_{21} & \cdots & a_{2,E_1+1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{E_1,1} & \cdots & a_{E_1,E_1+1} & 0 & \cdots & 0 \\ a_{E_1+1,1} & \cdots & a_{E_1+1,E_1+1} & a_{E_1+1,E_1+2} & \cdots & a_{E_1+1,F+1} \\ a_{E_1+2,1} & \cdots & a_{E_1+2,E_1+1} & a_{E_1+2,E_1+2} & \cdots & a_{E_1+2,F+1} \\ 0 & \cdots & 0 & a_{E_1+3,E_1+2} & \cdots & a_{E_1+3,F+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{F,E_1+2} & \cdots & a_{F,F+1} \\ 0 & \cdots & 0 & a_{F+1,E_1+2} & \cdots & a_{F+1,F+1} \end{bmatrix} \tag{33}$$

$$\mathbf{X} = [\lambda_1(\tau), \lambda_2(\tau), \ldots, \lambda_{E_1+1}(\tau), \gamma(\tau)_{E_1+1}(\tau), \gamma_{E_1+2}(\tau), \ldots, \gamma_F(\tau)]^T \tag{34}$$

and the entries in matrix **A** are given by

$$a_{j,i} = \begin{cases} \phi(||0 - s_i||), & \text{for} \quad j = 1,\ i = 1, 2, \ldots, 2F, \\ \phi(||s_j - s_i||) - \phi_{ss}(||s_j - s_i||) - \phi_s(||s_j - s_i||), & \text{for} \quad j = 2, \ldots E_1,\ i = 1, 2, \ldots, E_1 + 1, \\ \phi(||\alpha - s_i||), & \text{for} \quad j = E_1 + 1, i = 1, 2, \ldots, E_1 + 1, \\ -\phi(||\alpha - s_i||), & \text{for} \quad j = E_1 + 2, i = E_1 + 2, \ldots, E + 1, \\ \beta_1\phi(||\alpha - s_i||), & \text{for} \quad j = E_1 + 2, i = E_1 + 2, \ldots, E_1 + 1, \\ -\beta_2\phi(||\alpha - s_i||), & \text{for} \quad j = E_1 + 2, i = E_1 + 2, \ldots, F + 1, \\ \phi(||s_j - s_i||) - \phi_{ss}(||s_j - s_i||) - \phi_s(||s_j - s_i||), & \text{for} \quad j = E_1 + 3, \ldots, F, i = E_1 + 2, \ldots, F + 1, \\ \phi(||1 - s_i||), & \text{for} \quad j = E + 1, i = E_1 + 2, \ldots, F + 1, \end{cases} \tag{35}$$

whereas matrix **D** has the following entries:

$$b_j = \begin{cases} g(0, \tau), & \text{for} \quad j = 1, \\ 2(u_1(s_j, \tau_0) - u_1(s_j, \tau_0)d\tau) - u_1(s_j, \tau_0) & \\ +f_1(s_j, \tau)d\tau^2, & \text{for} \quad j = 2, ..., E_1, \\ q_1(\tau), & \text{for} \quad j = E_1 + 1, \\ q_2(\tau), & \text{for} \quad j = E_1 + 2, \\ 2(u_2(s_j, \tau_0) - u_2(s_j, \tau_0)d\tau) - u_2(s_j, \tau_0) & \\ +f_2(s_j, \tau)d\tau^2, & \text{for} \quad j = E_1 + 3, \ldots, F, \\ g(1, \tau), & \text{for} \quad j = F + 1, \end{cases} \tag{36}$$

From Equation (32), we obtain the following.

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{D}. \tag{37}$$

Finally, by solving the system described above using any linear solver, we can obtain the values of the MQ RBF coefficients that are unknown. Using the values of these coef-

ficients in Equations (18) and (21), we can effortlessly acquire the approximate solution necessary to the problem.

3.2. Nonlinear Case

The quasi-Newton linearization scheme is initially employed to transform the nonlinear Equation (2) into a linear form. The corresponding linearized expression is given by

$$(\psi\mu)^{n+1} = \varphi^n\mu^{n+1} + \varphi^{n+1}\mu^n - \varphi^n\mu^n + \mathcal{O}(\Delta\tau^2). \tag{38}$$

A comprehensive theoretical justification and analysis of this strategy may be found in [40].

The nonlinear term is given by

$$\psi(s,\tau) = 2u^2(s,\tau).$$

Let

$$u^n = u(s,\tau_n), \quad u^{n+1} = u(s,\tau_{n+1}).$$

Apply linearization:

$$u^2(s,\tau) \approx 2u^n u^{n+1} - (u^n)^2.$$

Multiply both sides by 2:

$$2u^2(s,\tau) \approx 4u^n u^{n+1} - 2(u^n)^2.$$

Replacing the nonlinear term in Equation (2), the fully linearized form of the equation becomes

$$u_{1\tau\tau}(s,\tau) \approx (\beta(s)u_{1s}(s,\tau))_s + 4u_1^n u_1^{n+1} - 2(u_1^n)^2 + f_1(s,\tau), \quad s \in I_1, \tag{39}$$

and

$$u_{2\tau\tau}(s,\tau) \approx (\beta(s)u_{2s}(s,\tau))_s + 4u_2^n u_2^{n+1} - 2(u_2^n)^2 + f_2(s,\tau), \quad s \in I_2. \tag{40}$$

The nonlinear system is first linearized using the quasi-Newton method. The resulting equations are then discretized by incorporating MQ-RBF approximations. There after, the resulting linear system is solved using a suitable numerical solver.

4. Meshless Collocation Method for Double-Interface Models

The intervals $I_1 = [0,\alpha]$, $I_2 = [\alpha,\beta]$, and $I_3 = [\beta,1]$ are subdivided with nodes such that the partitions satisfy $0 = x_1 < x_2 < \cdots < x_{F_1+1} = \alpha$, $\alpha = x_{F_1+1} < x_{F_1+2} < \cdots < x_{F_1+F_2+1} = \beta$, and $\beta = x_{F_1+F_2+1} < x_{F_1+F_2+2} < \cdots < x_F$, respectively. Here, $N = E_1 + E_2 + E_3 + 1$ denotes the total number of nodes, and F_1, F_2, and F_3 represent the number of subintervals in I_1, I_2, and I_3, respectively. The functions $u_1(s,\tau)$, $u_2(s,\tau)$, and $u_3(s,\tau)$ are approximated on these subintervals, I_1, I_2, and I_3, using MQ-RBFs as follows:

$$u_1(s,\tau) \approx \sum_{i=1}^{F_1+1} \lambda_i(\tau)\phi(\|s-s_i\|_2), \quad s \in I_1, \tag{41}$$

$$u_2(s,\tau) \approx \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i(\tau)\phi(\|s-s_i\|_2), \quad s \in I_2, \tag{42}$$

$$u_3(s,\tau) \approx \sum_{i=F_1+F_2+1}^{N} \rho_i(t)\phi(\|s-s_i\|_2), \quad s \in I_3, \tag{43}$$

respectively.

The first boundary condition at $s = 0$ in this case becomes

$$\sum_{i=1}^{F_1+1} \lambda_i(\tau)\phi(\|0 - s_i\|_2) \quad = \quad \delta_3(\tau), \tag{44}$$

whereas for the second boundary condition at $s = 1$, we have

$$\sum_{i=F_1+F_2+1}^{N} \rho_i(t)\phi(\|1 - x_i\|_2) \quad = \quad \delta_4(\tau). \tag{45}$$

The interface conditions yield the following:

$$\sigma_1 \sum_{i=1}^{F_1+1} \lambda_i(\tau)\phi(\|\alpha - s_i\|_2) - \sigma_2 \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i(\tau)\phi(\|\alpha - s_i\|_2) \quad = \quad \varpi_1(\tau), \tag{46}$$

$$\sigma_3 \sum_{i=1}^{F_1+1} \lambda_i(\tau)\phi_s(\|\alpha - s_i\|_2) - \sigma_4 \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i(\tau)\phi_s(\|\alpha - s_i\|_2) \quad = \quad \varpi_2(\tau), \tag{47}$$

$$\sigma_5 \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i(\tau)\phi(\|\beta - s_i\|_2) - \sigma_6 \sum_{i=F_1+F_2+1}^{N} \rho_i(\tau)\phi(\|\beta - s_i\|_2) \quad = \quad \varpi_3(\tau), \tag{48}$$

$$\sigma_7 \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i(\tau)\phi_s(\|\beta - s_i\|_2) - \sigma_8 \sum_{i=F_1+F_2+1}^{N} \rho_i(\tau)\phi_s(\|\beta - s_i\|_2) \quad = \quad \varpi_4(\tau). \tag{49}$$

The rest of the procedure will be discussed for linear and nonlinear cases separately.

4.1. Linear Case

Now we substitute the MQ-RBFs of the unknown function $u(s, \tau)$ and its derivatives into Equation (1). For $u_1(s, \tau)$, we obtain

$$\begin{aligned} &\sum_{i=1}^{F_1+1} \lambda_i \phi^*(\|s - s_i\|) - 2(u_1(s, \tau_0) - u_1(s, \tau_0)d\tau) + u_1(s, \tau_0) \\ &= \beta_1 \sum_{i=1}^{F_1+1} \lambda_i \frac{\partial^2}{\partial s^2}\phi(\|s - s_i\|)(d\tau)^2 + \beta_{1_s}(s, \tau) \sum_{i=1}^{F_1+1} \lambda_i \frac{\partial}{\partial s}\phi(\|s - s_i\|)(d\tau)^2 + f_1(s, \tau), \quad s \in I_1. \end{aligned} \tag{50}$$

For $u_2(s, \tau)$, we obtain

$$\begin{aligned} &\sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i \phi(\|s - s_i\|) - 2(u_2(s, \tau_0) - u_2(s, \tau_0)d\tau) + u_2(s, \tau_0) \\ &= f_2(s, \tau) + \beta_2 \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i \frac{d^2}{ds^2}\phi(\|s - s_i\|)(d\tau)^2 + C_{2_s}(s, \tau) \sum_{i=F_1+1}^{F_1+F_2+1} \gamma_i \frac{d}{ds}\phi(\|s - s_i\|)(d\tau)^2, \quad s \in I_2. \end{aligned} \tag{51}$$

For $u_3(s, \tau)$, we obtain

$$\begin{aligned} &\sum_{i=F_1+F_2+1}^{N} \rho_i \phi(\|s - s_i\|) - 2(u_3(s, \tau_0) - u_3(s, \tau_0)d\tau) + u_3(s, \tau_0) \\ &= f_3(s, \tau) + \beta_3 \sum_{i=F_1+F_2+1}^{N} \rho_i \frac{d^2}{ds^2}\phi(\|s - s_i\|)(d\tau)^2 + \beta_{3_s}(s, \tau) \sum_{i=F_1+F_2+1}^{N} \rho_i \frac{d}{ds}\phi(\|s - s_i\|)(d\tau)^2, \quad s \in I_3. \end{aligned} \tag{52}$$

Discretization yields the following system of equations:

For $u_1(s,\tau)$, we obtain

$$\begin{aligned}&\sum_{i=1}^{F_1+1}\lambda_i\phi(\|s_j-s_i\|)-\beta_1\sum_{i=1}^{F_1+1}\lambda_i\frac{d^2}{ds^2}\phi(\|s_j-s_i\|)(d\tau)^2\\&\quad-\beta_{1_s}(s_j,\tau)\sum_{i=1}^{F_1+1}\lambda_i\frac{d}{ds}\phi(\|s_j-s_i\|)(d\tau)^2\\&=2(u_1(s_j,\tau_0)-u_1(s_j,\tau_0)d\tau)-u_1(s_j,\tau_0)+f_1(s_j,\tau),\quad s_j\in I_1.\end{aligned}\tag{53}$$

For $u_2(s,\tau)$, we obtain

$$\begin{aligned}&\sum_{i=F_1+1}^{F_1+F_2+1}\gamma_i\phi(\|s_j-s_i\|)-\beta_2\sum_{i=F_1+1}^{F_1+F_2+1}\gamma_i\frac{d^2}{ds^2}\phi(\|s_j-s_i\|)(d\tau)^2\\&\quad-\beta_{2_s}(s_j,\tau)\sum_{i=F_1+1}^{F_1+F_2+1}\gamma_i\frac{d}{ds}\phi(\|s_j-s_i\|)(d\tau)^2\\&=2(u_2(s_j,\tau_0)-u_2(s_j,\tau_0)d\tau)-u_2(s_j,\tau_0)+f_2(s_j,\tau),\quad s_j\in I_2.\end{aligned}\tag{54}$$

For $u_3(s,\tau)$, we obtain

$$\begin{aligned}&\sum_{i=F_1+F_2+1}^{N}\rho_i\phi(\|s_j-s_i\|)-\beta_3\sum_{i=F_1+F_2+1}^{N}\rho_i\frac{d^2}{ds^2}\phi(\|s_j-s_i\|)(d\tau)^2\\&\quad-\beta_{3_s}(s_j,\tau)\sum_{i=F_1+F_2+1}^{N}\rho_i\frac{d}{ds}\phi(\|s_j-s_i\|)(d\tau)^2\\&=2(u_3(s_j,\tau_0)-u_3(s_j,\tau_0)d\tau)-u_3(s_j,\tau_0)+f_3(s_j,\tau),\quad s_j\in I_3.\end{aligned}\tag{55}$$

Equations (53)–(55) combined with Equations (46)–(49) represents a linear system of equations with N + 2 unknowns variables: λ_i, where $i=1,2,3\ldots,F_1+1$; γ_i, where $i=F_1+1,F_1+2\ldots,F_1+F_2+1$; and d, where $i=F_1+F_2+1,F_1+F_2+2\ldots,N$.

4.2. Nonlinear Case

The quasi-Newton linearization scheme is initially employed to transform the nonlinear Equation (2) into a linear form. The corresponding linearized expression is given by

$$(\psi\mu)^{n+1}=\varphi^n\mu^{n+1}+\varphi^{n+1}\mu^n-\varphi^n\mu^n+\mathcal{O}(\Delta\tau^2).\tag{56}$$

A comprehensive theoretical justification and analysis of this strategy may be found in [40].

The nonlinear term is given by

$$\psi(s,\tau)=2u^2(s,\tau).$$

Let

$$u^n=u(s,\tau_n),\quad u^{n+1}=u(s,\tau_{n+1}).$$

Apply linearization:

$$u^2(s,\tau)\approx 2u^nu^{n+1}-(u^n)^2.$$

Multiply both sides by 2:

$$2u^2(s,\tau)\approx 4u^nu^{n+1}-2(u^n)^2.$$

Replacing the nonlinear term in Equation (2), the fully linearized form of the equation becomes

$$u_{1\tau\tau}(s,\tau) \approx (\beta(s)u_{1s}(s,\tau))_s + 4u_1^n u_1^{n+1} - 2(u_1^n)^2 + f_1(s,\tau), \quad s \in I_1, \tag{57}$$

and

$$u_{2\tau\tau}(s,\tau) \approx (\beta(s)u_{2s}(s,\tau))_s + 4u_2^n u_2^{n+1} - 2(u_2^n)^2 + f_2(s,\tau), \quad s \in I_2, \tag{58}$$

and

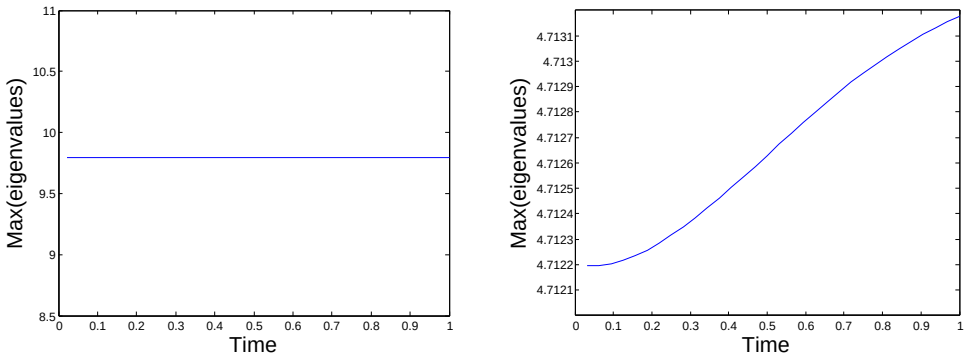
$$u_{3\tau\tau}(s,\tau) \approx (\beta(s)u_{3s}(s,\tau))_s + 4u_3^n u_3^{n+1} - 2(u_3^n)^2 + f_3(s,\tau), \quad s \in I_3. \tag{59}$$

Thus, we have successfully linearized the nonlinear system using the quasi-Newton approach, and we now proceed to discretize the resulting equations by substituting MQ-RBF values. Subsequently, the linear system will be solved using an appropriate linear solver.

5. Stability Analysis

Any numerical approach depends on stability since it guarantees that minor deviations, such data inconsistencies or rounding mistakes, do not cause uncontrollably increasing error development, therefore compromising the dependability of the results. We investigate in this part the stability properties of the suggested MQ-RBF method applied to the current interface issue. Generally, especially in time-stepping methods, the stability of numerical schemes is evaluated by studying the behavior of the matrices produced during the discretization process. Within the MQ-RBF framework, the matrix X represents the coefficient matrix generated at every time level from MQ-RBF weights. We track the eigenvalues of matrix X in order to evaluate the stability of the method, with a special focus on the highest eigenvalues since they provide an important understanding of the dynamic behavior of the method. Eigenvalues that approach zero or become too large could indicate instability; thus, it is important to monitor their size at all times during the simulation. Computed at every time step, the eigenvalues of matrix X help to assess computational stability.

As shown in Figure 1, our study reveals that the maximum eigenvalues regularly remain well away from zero during the whole simulation. This is a fundamental observation since eigenvalues close to zero could indicate almost singularity in the matrix, hence affecting the high condition numbers and increasing the risk of numerical error amplification. Figure 1 amply shows that the highest eigenvalues stay limited and keep a safe distance from zero all throughout the time domain. This implies that matrix X stays well-conditioned, thereby guaranteeing the stability of the numerical technique. Such behavior suggests that, even in the presence of small numerical disturbances or round-off errors, the suggested MQ-RBF technique is able to provide stable and accurate results over long time integrations. In this sense, tracking the maximum eigenvalues of matrix X at every time interval has confirmed, in general, the computational stability of the proposed scheme. Figure 1 shows that the fact that these eigenvalues remain constantly non-zero supports the stability of the MQ-RBF method and validates its robustness for solving interface problems.



Example 1 Example 6

Figure 1. The stability analysis of the proposed numerical method at $F = 16, 20$, $\Delta t = 0.1/20$, $\Delta t = 0.5/16$, $t = 1$, $a = 0$, and $b = 1$.

6. Numerical Studies

This part presents a comprehensive series of numerical tests meant to assess the performance of the proposed numerical scheme with regard to accuracy, computing efficiency, and practical application. The formulation created in Equation (56) is applied specially to address a range of nonlinear test problems, showing the method's capacity to handle complicated and realistic scenarios. To systematically assess the correctness of the scheme, we calculate the MAEs and RMSEs for different values of the discretization parameter F. These numerical measures offer a thorough assessment of the accuracy of the solution and the capacity of the approach to properly approximate the precise or benchmark solutions. The formulas used to calculate these error measures are stated as follows:

$$\text{MAEs} = \varphi^{\text{true}} - \varphi_{\infty}^{\text{approx}} \approx \max_{j} \left| \varphi_j^{\text{true}} - \varphi_j^{\text{approx}} \right|, \tag{60}$$

$$\text{RMSEs} \approx \sqrt{\left(\frac{1}{F} \sum_{j=1}^{F} \left(\varphi_j^{\text{true}} - \varphi_j^{\text{approx}} \right)^2 \right)}, \tag{61}$$

and

$$Order = \frac{\log(E_{\infty}(d\tau_1)) - \log(E_{\infty}(d\tau_2))}{\log 2}. \tag{62}$$

Example 1. *Consider the linear hyperbolic interface problem with constant coefficients:*

$$u_{\tau\tau}(s,\tau) = (\beta(s) u_s(s,\tau))_s + f(s,\tau), \tag{63}$$

with the following exact solutions:

$$u(s,\tau) = \begin{cases} u_1(s,\tau) = \frac{1}{3} s^3 cos(\tau), & 0 \le s \le 1/2, \\ u_2(s,\tau) = s^3 cos(\tau), & 1/2 \le s \le 1, \end{cases}$$

and

$$\beta(s) = \begin{cases} \beta_1(s) = 1, & 0 \le s \le 1/2, \\ \beta_2(s) = 2, & 1/2 \le s \le 1. \end{cases}$$

The exact solution involves defining the boundary, interface, and initial conditions. The test problem pertains to a linear scenario. The numerical method is applied to this problem and the resultant computational data are displayed in Table 1. This table presents the MAE and RMSE values for different quantities of GPs at various time intervals. The data in Table 1 indicate that the MAEs decrease to a magnitude of 10^{-4}*, despite a very limited number of GPs. This level of accuracy is typically sufficient for practical applications. Moreover, augmenting the density of the computing grid via an increased number of GPs can enhance accuracy. The behavior of the solution near the interface is further illustrated in Figure 2, which clearly highlights the observed discontinuities. Additionally, in Figure 3, we show a comparison of the estimated and actual solutions using 3D plots for different time step sizes, specifically* $\Delta\tau = 0.025, 0.005$*, with the number of control points at* $F = 4, 22$*. These visuals provide a clearer understanding of the performance and error characteristics of the suggested strategy. The graphical results depicted in Figure 3 demonstrate the robust concordance between the estimated and actual solutions. The main benefit of this hybrid system is that it can accurately capture sharp changes at the interface, as is clearly shown in the numerical results and 3D images.*

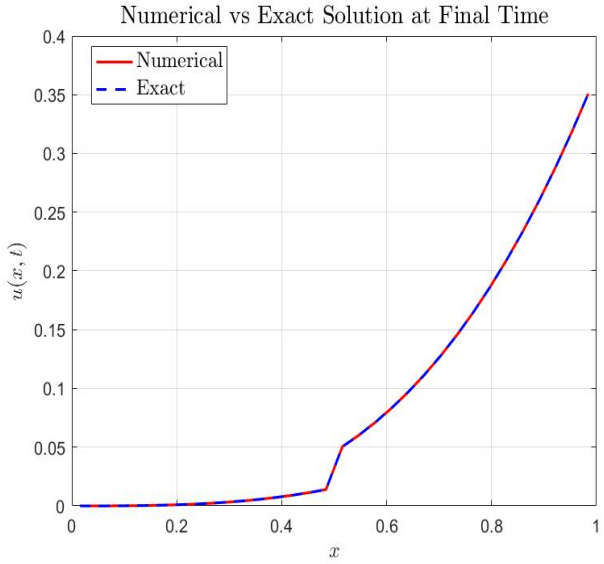
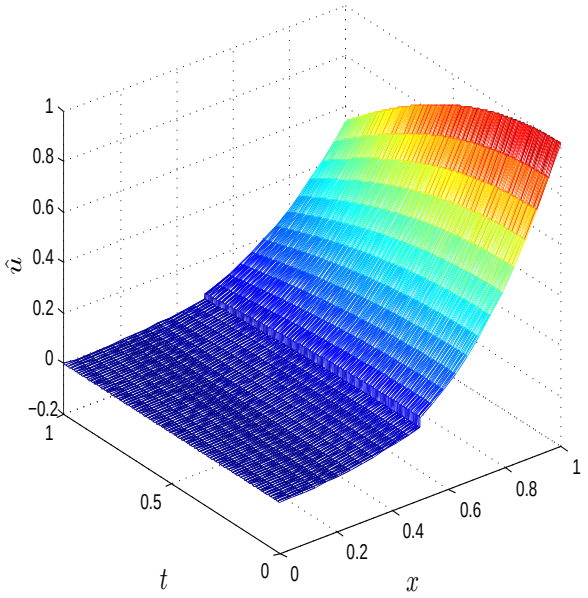


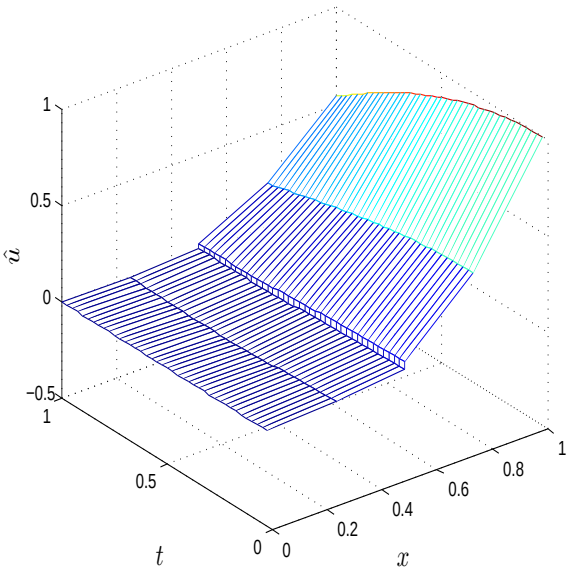
Figure 2. A comparison between the exact and estimated solutions to example 1 using the MCM method for $F = 22$ at $\tau = 1$ shows distinct jumps at the interface located at $s = 0.5$.

Table 1. Assessment of numerical errors in benchmark example 1, employing MQ-RBFs for $\Delta\tau = 0.1$ with $\tau = 1$.

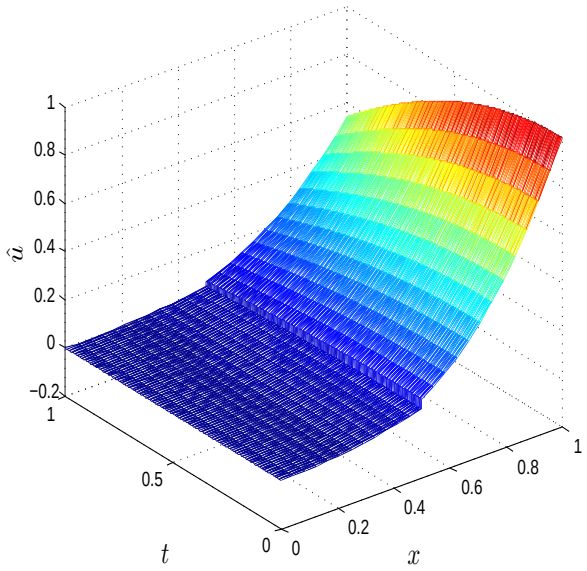
$\varepsilon = 2.5$			
F	$E_c(F)$	RMSEs	CPU Time (s)
4	8.5600×10^{-2}	6.0500×10^{-2}	0.017677
8	2.0700×10^{-2}	1.3100×10^{-2}	0.014940
12	8.4000×10^{-3}	5.5000×10^{-3}	0.002723
18	1.4000×10^{-3}	9.0722×10^{-4}	0.004070
20	7.3007×10^{-4}	4.3379×10^{-4}	0.002792



Exact solution



Computed solution, with $F = 4$ and $\Delta\tau = 0.025$



Computed solution, with $F = 22$ and $\Delta\tau = 0.005$

Figure 3. Assessment of the difference between the exact and computed solutions in eaxmple 1.

Example 2. *Let us examine the following linear hyperbolic model with a single interface and variable coefficients:*

$$u_{\tau\tau} = \begin{cases} (\beta_1(s)u_{1s}(s,\tau)) + f_1(s,\tau) & 0 \le s \le 1/2, \\ (\beta_2(s)u_{2s}(s,\tau))_s + f_2(s,\tau), & 1/2 \le s \le 1, \end{cases}$$

with the following exact solutions:

$$u(s,\tau) = \begin{cases} u_1(s,\tau) = scos(\tau), & 0 \le s \le 1/2, \\ u_2(s,\tau) = (s+\frac{1}{2})cos(\tau), & 1/2 \le s \le 1, \end{cases}$$

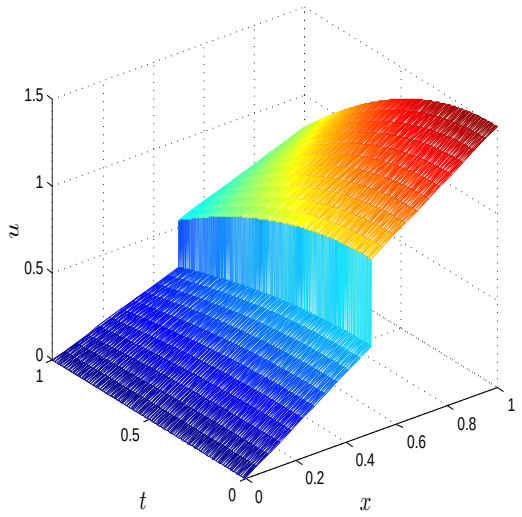
and

$$\beta(s) = \begin{cases} \beta_1(s) = \frac{s}{3}, & 0 \le s \le 1/2, \\ \beta_2(s) = s, & 1/2 \le s \le 1, \end{cases}$$

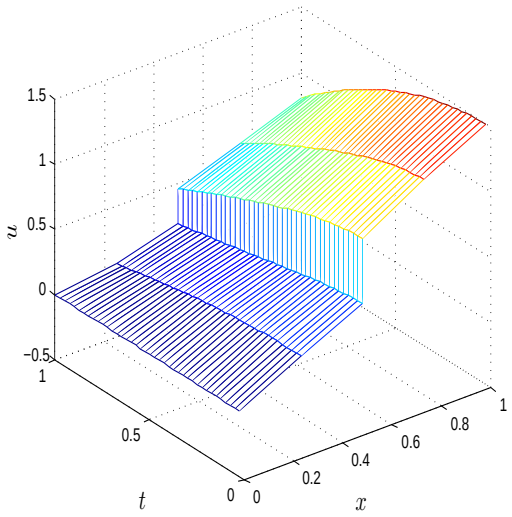
The exact solution is employed to delineate the boundary, interface, and initial conditions for this test problem, which pertains to a linear situation. The proposed numerical method is applied to this problem, and the obtained computational data are reported in Table 2. This table displays the MAEs and RMSEs for varying quantities of GPs and distinct time step sizes with the shape parameter $\varepsilon = 2$. *The data in Table 2 indicate that the MAEs diminish to a magnitude of* 10^{-4}, *even with a limited number of GPs, illustrating the efficacy of the strategy. This level of precision is typically deemed enough for practical applications. Enhanced precision in the results can be achieved by augmenting the density of the computing grid by utilizing a larger quantity of GPs. Furthermore, Figure 4 illustrates a visual comparison between the precise and computed solutions using 3D plots for different time step sizes, specifically* $\Delta\tau = 0.025, 0.005$, *while maintaining a constant number of control points at* $F = 4, 22$. *These visualizations offer additional insight into the performance and error characteristics of the proposed scheme. The graphical results depicted in Figure 4 demonstrate a robust concordance between the estimated and exact answers. The 3D plots distinctly illustrate that the proposed numerical solution has successfully encapsulated all the essential characteristics of this intricate interface problem. This method not only accurately resolves the principal solution behavior but also precisely monitors the complex fluctuations and discontinuities at the interface. These visual outcomes underscore the durability and dependability of the method in addressing complex contact dynamics, which are frequently problematic to describe using traditional numerical techniques. The behavior of the solution near the interface is further depicted in Figure 5, where the discontinuities at the interface are clearly visible. The corresponding absolute error between the exact and numerical solutions is presented in Figure 6.*

Table 2. Assessment of numerical errors in benchmark example 2, employing MQ-RBFs for $\Delta\tau = 0.1$ with $\tau = 1$.

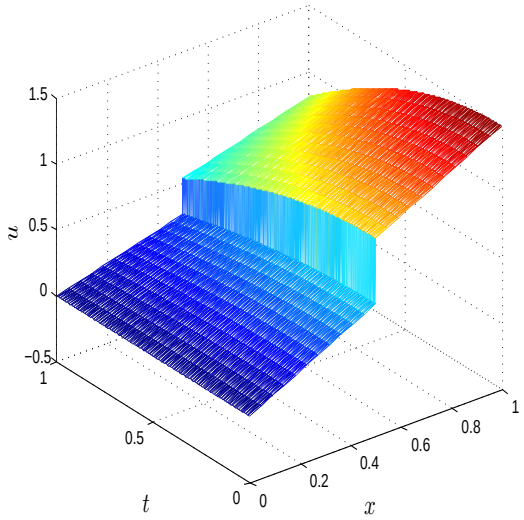
$\varepsilon = 2$			
F	$E_c(F)$	RMSEs	CPU Time (s)
2	1.5830×10^{-1}	1.1190×10^{-1}	0.014154
4	7.8600×10^{-2}	5.6500×10^{-2}	0.008559
10	1.2100×10^{-2}	7.6000×10^{-3}	0.006723
16	1.8000×10^{-3}	8.3680×10^{-4}	0.006121
22	9.1801×10^{-4}	4.0057×10^{-4}	0.009908



True solution



Computed solution, with $F = 4$ and $\Delta\tau = 0.025$



Computed solution, with $F = 22$ and $\Delta\tau = 0.005$

Figure 4. Assessment of the difference between the exact and computed solutions in test scenario 2.

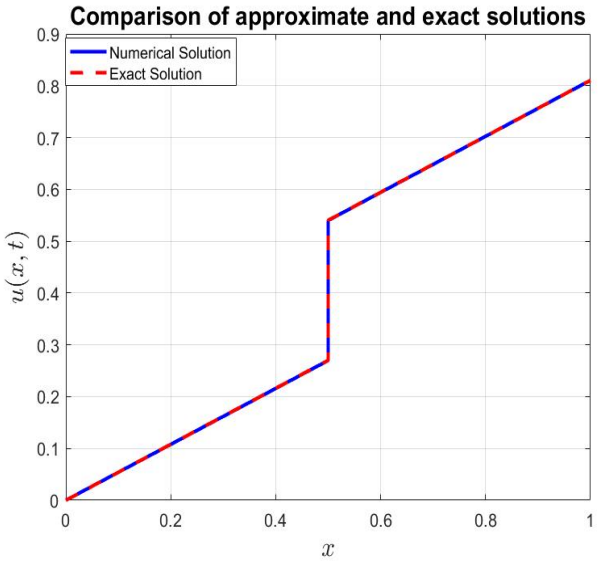


Figure 5. A comparison between the exact and estimated solutions to example 2 using the MCM method for $F = 22$ at $\tau = 1$ shows distinct jumps at the interface located at $s = 0.5$.

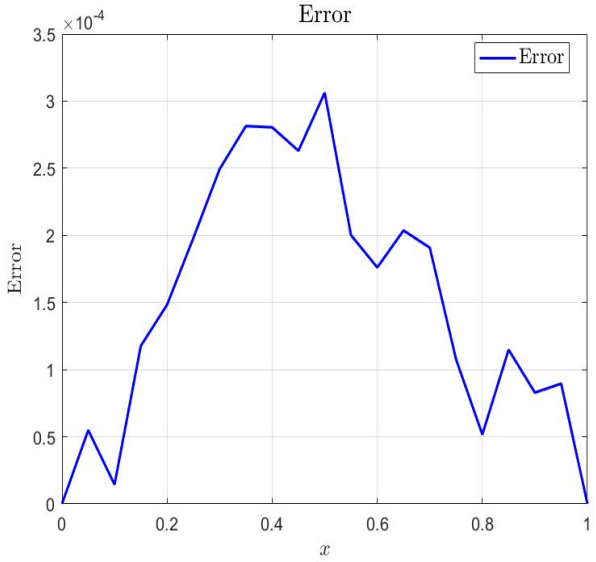


Figure 6. Absolute error between numerical and exact solutions at the final time for example 2.

Example 3. *Let us examine the following linear hyperbolic model with a single interface:*

$$u_{\tau\tau}(s,\tau) = \begin{cases} u_{1ss}(s,\tau) + s^8 e^{-\tau} + f_1(s,\tau), & 0 \le s \le 1/2, \\ 2u_{2ss}(s,\tau) + \frac{1}{2}(s^8 + \frac{1}{256})e^{-\tau} + f_2(s,\tau), & 1/2 \le s \le 1, \end{cases}$$

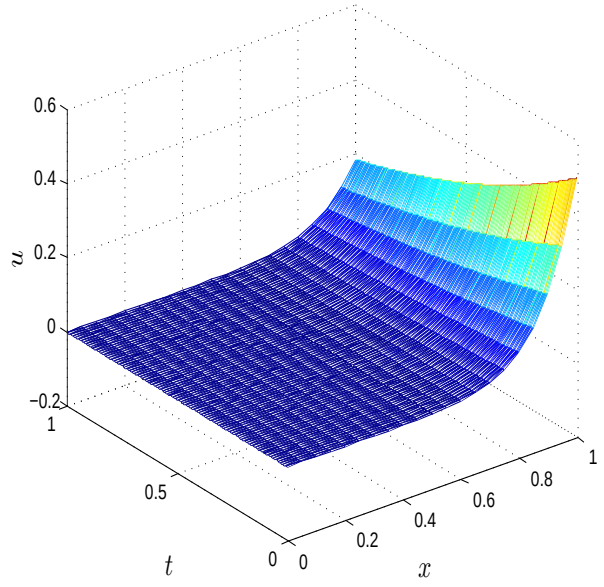
with the following exact solutions:

$$u(s,\tau) = \begin{cases} u_1(s,\tau) = s^8 e^{-\tau}, & 0 \le s \le 1/2, \\ u_2(s,\tau) = \frac{1}{2}(s^8 + \frac{1}{256})e^{-\tau}, & 1/2 \le s \le 1, \end{cases}$$

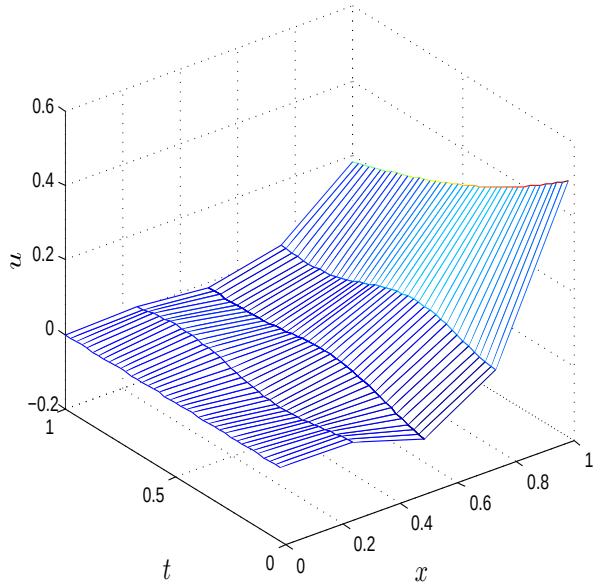
and

$$\beta(s) = \begin{cases} \beta_1(s) = 1, & 0 \le s \le 1/2, \\ \beta_2(s) = 2, & 1/2 \le s \le 1. \end{cases}$$

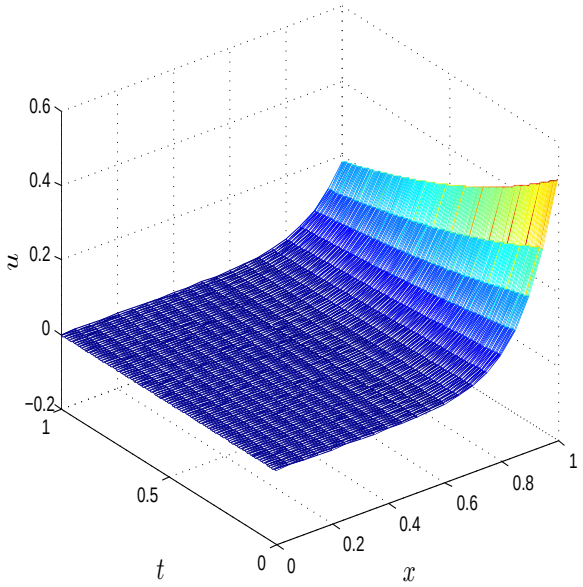
The boundary, the interface, and the initial conditions are all defined with the help of the exact solution provided. We present another linear hyperbolic interface problem here. The numerical method that was proposed is utilized to solve this problem. Table 3 provides the numerical results for the shape parameter $\varepsilon = 2.2$ and various numbers of GPs. You can also examine the comparison of the exact solution and the approximate solution by looking at the graph that is presented in Figure 7. Figure 8 displays the absolute error between the exact and numerical solutions.



True sol



Computed solution, with $F = 4$ and $\Delta\tau = 0.025$



Computed solution, with $F = 20$ and $\Delta\tau = 0.005$

Figure 7. Assessment of the difference between the computed and true solutions in example 3.

Table 3. Assessment of numerical errors in Benchmark Problem 3, employing MQ-RBFs for $\Delta\tau = 0.5$ with $\tau = 1$.

$\varepsilon = 2.2$			
F	$E_c(F)$	RMSEs	CPU Time (s)
2	8.5600×10^{-2}	6.0500×10^{-2}	0.011859
4	1.6800×10^{-2}	9.2000×10^{-3}	0.013385
12	7.3000×10^{-3}	4.5000×10^{-3}	0.004973
20	4.1613×10^{-4}	2.9514×10^{-4}	0.004792
24	4.9861×10^{-4}	2.4452×10^{-4}	0.006795

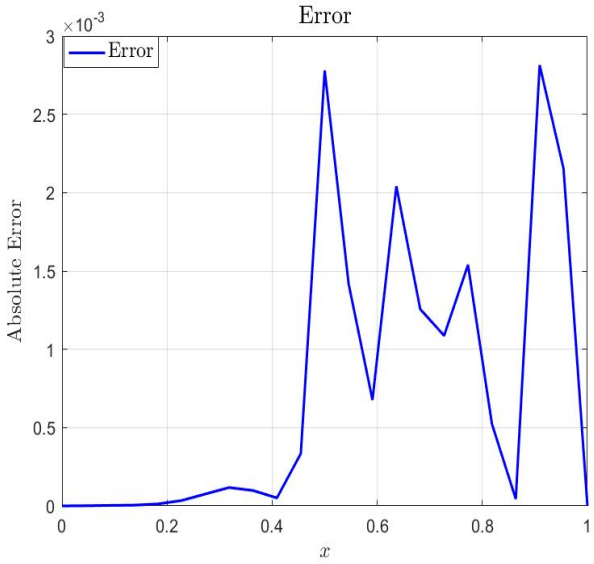


Figure 8. Absolute error between the numerical and exact solutions at the final time for example 3.

Example 4. *Let us examine the following nonlinear hyperbolic model with a single interface:*

$$u_{\tau\tau}(s,\tau) = \begin{cases} u_{1ss}(s,\tau) + 2u_1^2(s,\tau) + f_1(s,\tau), & 0 \le s \le 1/2, \\ 2u_{2ss}(s,\tau) + 2u_2^2(s,\tau) + f_2(s,\tau), & 1/2 \le s \le 1, \end{cases}$$

with the following exact solutions:

$$u(s,\tau) = \begin{cases} u_1(s,\tau) = s^4 e^{-\tau}, & 0 \le s \le 1/2, \\ u_2(s,\tau) = \frac{1}{2}(s^4 + \frac{1}{16})e^{-\tau}, & 1/2 \le s \le 1, \end{cases}$$

and

$$\beta(s) = \begin{cases} \beta_1(s) = 1, & 0 \le s \le 1/2, \\ \beta_2(s) = 2, & 1/2 \le s \le 1, \end{cases}$$

The source function, boundary, interface, and initial conditions are all defined with the help of the exact solution. A nonlinear hyperbolic interface problem is what we are dealing with here. This test challenge is being tackled with the recently established numerical approach. Presented in Table 4 are the numerical findings obtained for the shape parameter $\varepsilon = 2$. The MAEs and RMSEs are computed for varying numbers of grids at varying time increments. According to the table, the proposed method demonstrates an equally good performance in the nonlinear situation as it does in the linear one. Figure 9 show the MAEs for above four examples

Table 4. Assessment of numerical errors in Benchmark example 4, employing MQ-RBFs for $\Delta\tau = 0.001$ with $\tau = 1$.

$\varepsilon = 2$		
F	$E_c(F)$	RMSEs
4	1.1200×10^{-2}	7.5000×10^{-3}
8	4.3000×10^{-3}	2.3000×10^{-3}
16	2.5000×10^{-3}	1.3000×10^{-3}
20	2.2000×10^{-3}	1.2000×10^{-3}
24	2.0000×10^{-4}	1.0000×10^{-4}

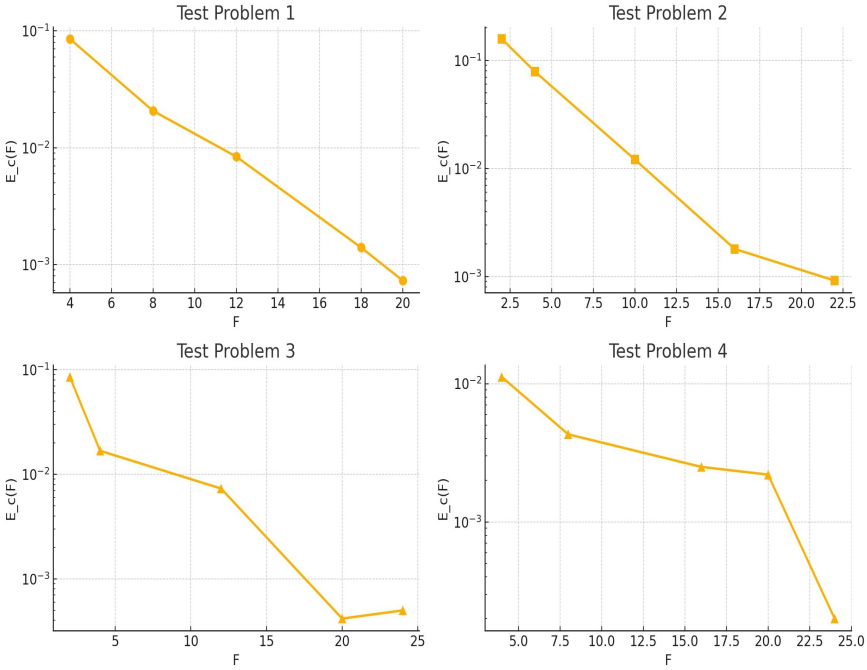


Figure 9. Maximum absolute error plots for different test problems.

Example 5. *Let us examine the following linear hyperbolic model with a single interface:*

$$u_{\tau\tau}(s,\tau) = (\beta(s)u_s(s,\tau))_s + f(s,\tau),$$

The coefficients, boundary conditions, interface conditions, and initial conditions are taken from test example 3. Since the exact solution to the problem is not available, the accuracy and convergence of the proposed numerical method are evaluated based on the residual errors. The results, summarized in Table 5, show that the residual errors remain extremely small across different discretization levels, demonstrating the stability and reliability of the method. The decrease in residual errors with mesh refinement suggests that the numerical solution approximates the true behavior of the system very closely. Furthermore, to provide additional verification, residual error plots and numerical solution profiles are presented in Figure 10, offering visual confirmation of the method's convergence and the solution's accuracy.

Table 5. Assessment of residual errors in benchmark example 4, employing MQ-RBFs with $\tau = 1$.

$\varepsilon = 2.2$			
F	$\Delta\tau$	**Residual Errors**	**CPU Time (s)**
2	0.5/2	4.6783×10^{-17}	0.055731
4	0.5/4	3.9034×10^{-16}	0.054083
8	0.5/8	1.0096×10^{-14}	0.056653
12	0.5/12	2.8139×10^{-13}	0.061229
16	0.5/16	4.8949×10^{-12}	0.060517

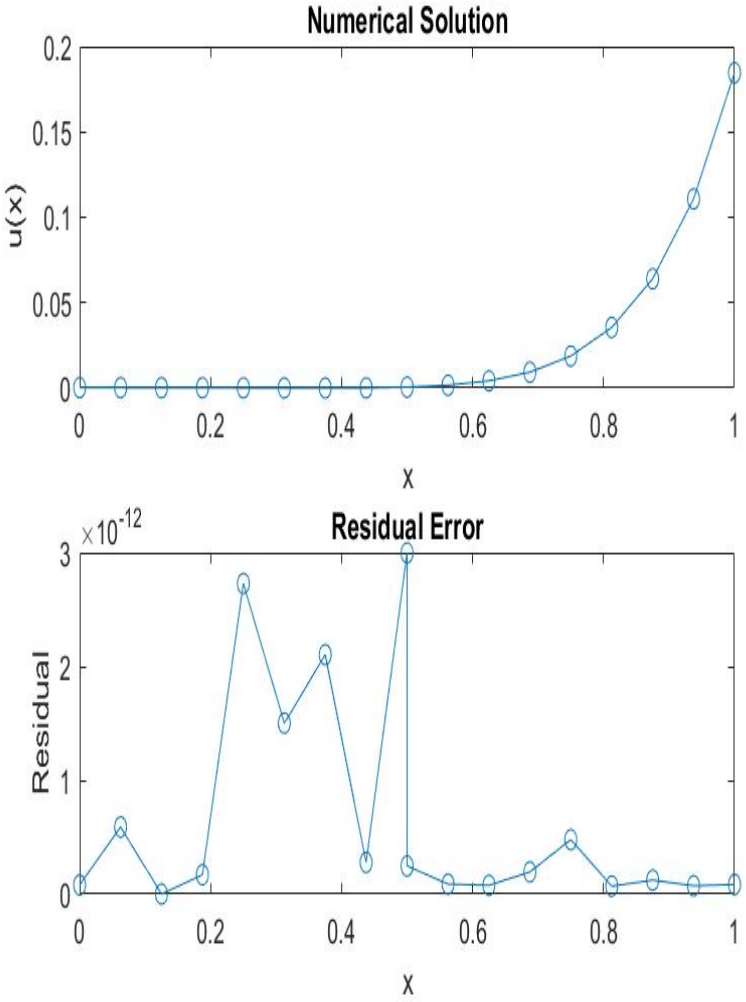


Figure 10. Numerical solution and residual error for test example 5.

Example 6. *Let us examine the following nonlinear hyperbolic model with a single interface:*

$$u_{\tau\tau} = \begin{cases} u_{1ss} + 2u_1^2(s,\tau) + f_1(s,\tau), & 0 \le s \le 1/2, \\ 2u_{2ss} + 2u_2^2(s,\tau) + f_2(s,\tau), & 1/2 \le s \le 1, \end{cases}$$

with the following exact solution:

$$u(s,\tau) = \begin{cases} u_1(s,\tau) = \frac{s}{3}cos(\tau), & 0 \le s \le 1/2, \\ u_2(s,\tau) = scos(\tau), & 1/2 \le s \le 1, \end{cases}$$

It is with the assistance of the exact solution that the source function, boundary, interface, and initial conditions are all determined. This is yet another problem with a nonlinear hyperbolic interface. In addition, the recommended solution is utilized for this test problem. The numerical findings obtained with the shape parameter $\varepsilon = 2.5$ *are presented in Table 6. The MAEs are reduced to a magnitude of* 10^{-4}*, even for a very tiny number grids, which is a level of accuracy that is considered satisfactory for practical purposes.*

Table 6. Assessment of numerical errors in benchmark example 6, employing MQ-RBFs for $\Delta\tau = 0.5$ with $\tau = 1$.

$\varepsilon = 2.5$		
F	$E_c(F)$	RMSEs
4	6.5400×10^{-2}	4.4200×10^{-2}
6	3.5500×10^{-2}	2.4000×10^{-2}
8	2.3000×10^{-3}	1.5700×10^{-3}
14	1.2600×10^{-4}	8.9000×10^{-4}
16	1.2300×10^{-4}	8.7000×10^{-4}

Example 7. *Let us examine the following linear hyperbolic model with a double interface:*

$$u_{\tau\tau}(s,\tau) = (\beta(s)u_s(s,\tau))_s + f(s,\tau),$$

with the following exact solution:

$$u(s,\tau) = \begin{cases} u_1(s,\tau) = e^{-\tau}s, & 0 \le p \le 1/5, \\ u_2(s,\tau) = e^{-\tau}e^{-s^2}, & 1/5 \le p \le 3/5, \\ u_3(s,\tau) = e^{-\tau}s^3, & 3/5 \le p \le 1. \end{cases}$$

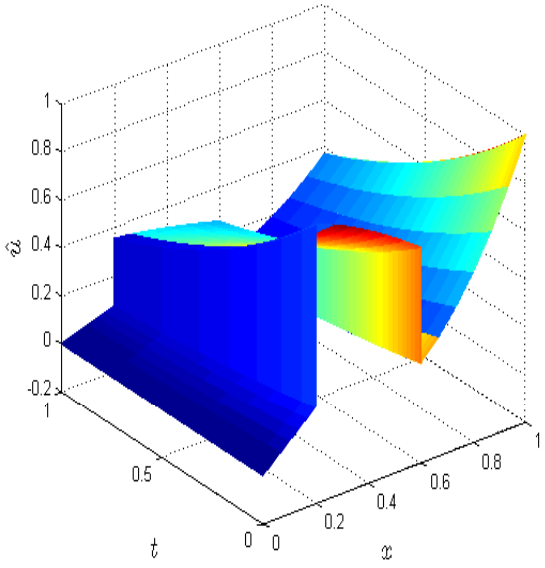
The initial, boundary, and interface conditions and the source function are all derived from the exact solution to ensure consistency in the numerical validation. This study looks at a 1D linear hyperbolic equation with double interfaces. We used the proposed numerical technique for this model to obtain the MAEs and RMSEs for different numbers of GPs. The results are shown in Tables 7 and 8. The findings show that the accuracy improves as the number of GPs increases because the inaccuracy decreases substantially.This method has also been tried with varying values for the time step size $\Delta\tau$, shape parameter ϵ, and coefficients. The results are shown in the Table 9. The graphical results depicted in Figure 11 demonstrate the robust concordance between the estimated and exact answers. The 3D plots distinctly illustrate that the proposed numerical solution has successfully encapsulated all the essential characteristics of this intricate interface problem. The method not only accurately resolves the principal solution behavior but also precisely monitors the complex fluctuations and discontinuities at the interface. These visual outcomes underscore the durability and dependability of the method in addressing complex contact dynamics, which are frequently problematic to describe using traditional numerical techniques. The behavior of the solution near the interface is further illustrated in Figure 12, which clearly highlights the observed discontinuities. Figure 13 displays the absolute error between the exact and numerical solutions.

Table 7. Assessment of numerical errors and CPU time in benchmark example 7, employing MQ-RBFs at $t = 1$ for $\epsilon = 2$ and $\epsilon = 1$ with $\Delta t = 0.001$.

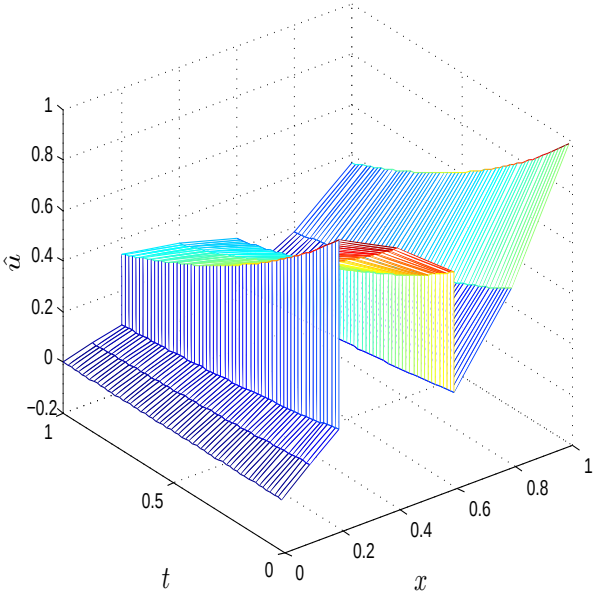
$(\beta_1(s), \beta_2(s), \beta_3(s)) = (1, 10, 100)$						
	$\epsilon = 2$			$\epsilon = 1$		
F	$E_c(F)$	RMSEs	CPU Time (s)	$E_c(F)$	RMSEs	CPU Time (s)
6	2.1900×10^{-2}	1.1300×10^{-2}	0.0326	6.9000×10^{-3}	3.3000×10^{-3}	0.0228
12	3.3000×10^{-3}	1.5000×10^{-3}	0.0431	2.4684×10^{-4}	9.9940×10^{-5}	0.0247
18	5.7700×10^{-4}	2.8682×10^{-4}	0.0368	2.8666×10^{-5}	1.7680×10^{-5}	0.0371
24	2.0081×10^{-4}	1.2822×10^{-4}	0.0472	-	-	-

Table 8. Assessment of numerical errors and CPU time in benchmark example 7, employing MQ-RBFs at $t = 1$ for $\epsilon = 2$ and $\epsilon = 1$ with $\Delta t = 0.001$.

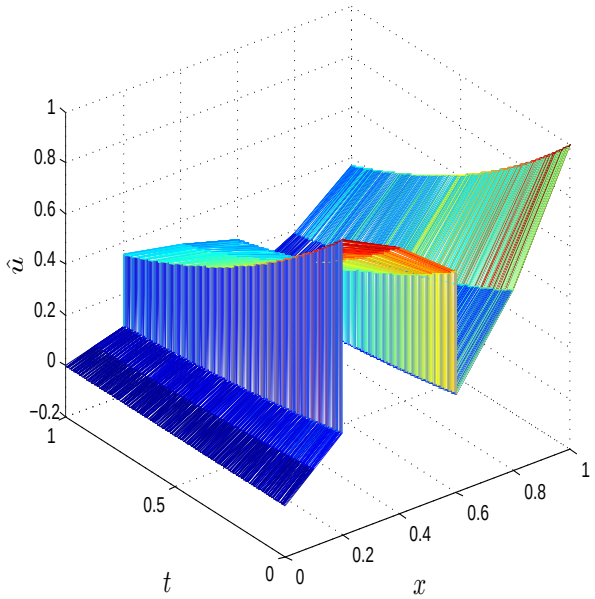
$(\beta_1(s), \beta_2(s), \beta_3(s)) = (1, 2, 3)$						
	$\epsilon = 2$			$\epsilon = 1$		
F	$E_c(F)$	RMSEs	CPU Time (s)	$E_c(F)$	RMSEs	CPU Time (s)
6	1.1200×10^{-2}	5.3000×10^{-3}	0.0324	3.4000×10^{-3}	1.6000×10^{-3}	0.0226
12	8.8058×10^{-4}	4.1460×10^{-4}	0.0412	8.1566×10^{-5}	3.8684×10^{-5}	0.0277
18	2.7313×10^{-4}	1.7421×10^{-4}	0.0321	2.8492×10^{-5}	1.7347×10^{-5}	0.0370
24	2.0862×10^{-4}	1.2705×10^{-4}	0.0387	-	-	-



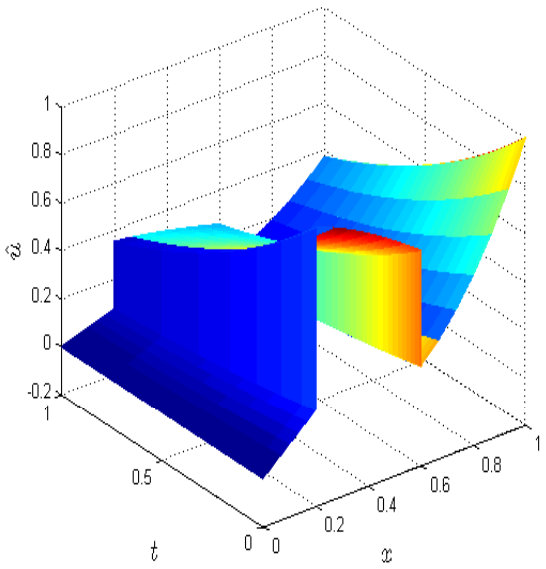
True solution



Computed solution, with $N = 6$ and $\Delta\tau = 0.1$



Computed solution, with $F = 6$ and $\Delta\tau = 0.01$



Computed solution, with $F = 18$ and $\Delta\tau = 0.001$

Figure 11. Assessment of the difference between the computed and true solutions in example 7.

Table 9. Assessment of errors in benchmark example 7 employing MQ-RBFs with $F = 18$ and $\tau = 1$.

$\varepsilon = 2.2$			
$\Delta\tau$	$E_c(F)$	Order	CPU Time (s)
0.1	3.0260×10^{-3}	-	0.001997
0.05	2.8202×10^{-3}	0.0964	0.002236
0.01	2.6450×10^{-3}	0.1066	0.004805
0.005	1.8994×10^{-3}	0.4523	0.008550
0.001	7.5180×10^{-4}	1.3372	0.037296

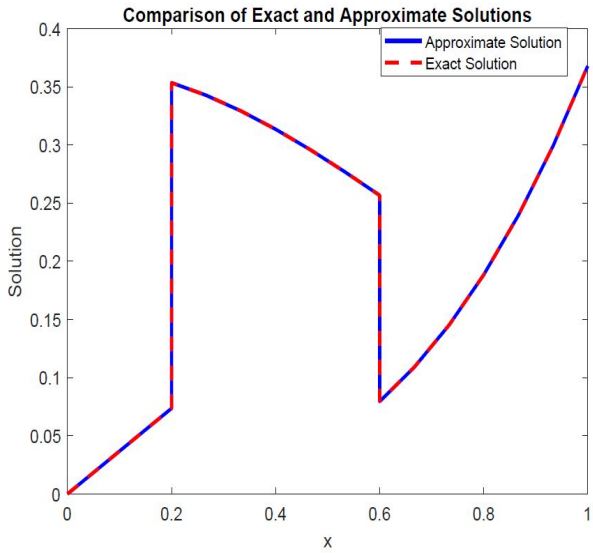


Figure 12. A comparison between the exact and estimated solutions to example 7 using the MCM method for $F = 18$ at $t = 1$ shows distinct jumps at the interfaces located at $s = 0.2$ and $s = 0.6$.

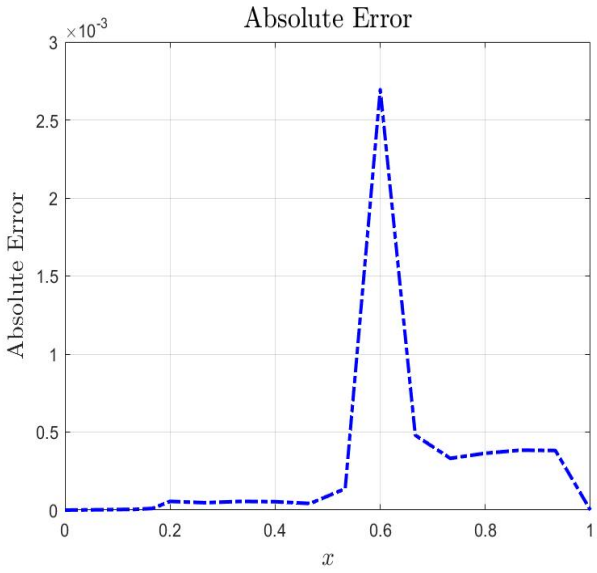


Figure 13. Absolute error between the numerical and exact solutions at the final timepoint for test scenario 7.

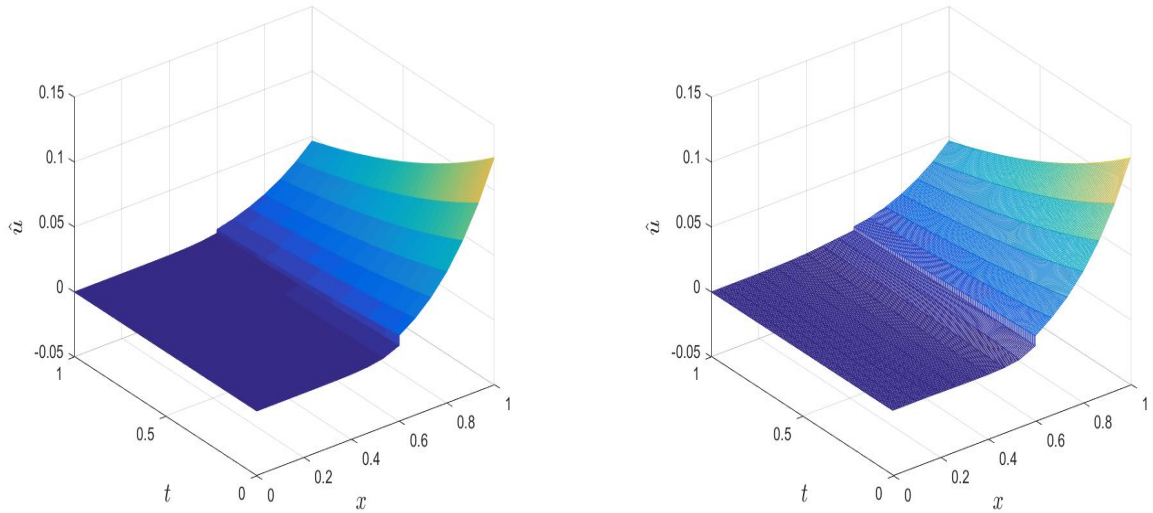
Example 8. *Let us examine the following linear hyperbolic model with a double interface:*

$$u_{\tau\tau}(s,\tau) = (\beta(s)u_s(s,\tau))_s + f(s,\tau),$$

with the following exact solution:

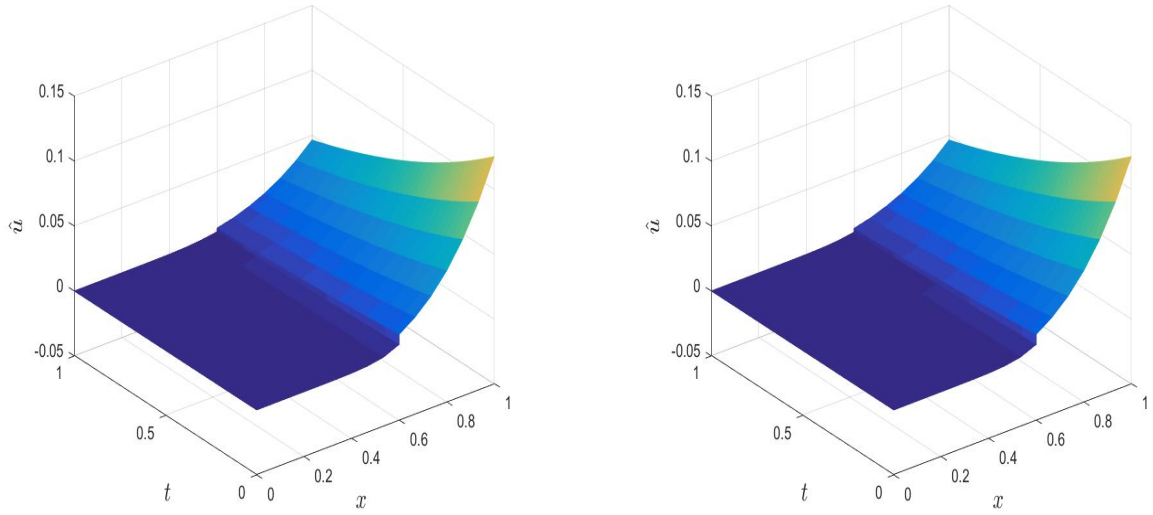
$$u(s,\tau) = \begin{cases} u_1(s,\tau) = s^{16}e^{-\tau}, & 0 \le s \le 1/5, \\ u_2(s,\tau) = \frac{1}{2}s^{8}e^{-\tau}, & 1/5 \le s \le 3/5, \\ u_3(s,\tau) = \frac{1}{8}s^{4}e^{-\tau}, & 3/5 \le s \le 1. \end{cases}$$

The initial, boundary, and interface conditions and the source function are all derived from the exact solution to ensure consistency in the numerical validation. This example also involves a 1*D linear hyperbolic equation with double interfaces and constant coefficients. It is used to assess the effectiveness, precision, and convergence behavior of the proposed numerical method. Table 10 reports the MAEs and RMSEs for various numbers of GPs. A comparison of the results for shape parameter values of* $\epsilon = 1$ *and* $\epsilon = 2$ *with a fixed time step of* 0.001 *is also provided in the same table. The data indicate that the numerical scheme yields more accurate results when* $\epsilon = 1$*. However, the method shows sensitivity to the choice of shape parameter, and inappropriate values can lead to unreliable outcomes. To further illustrate the method's performance, three-dimensional plots of both the exact and approximate solutions for different node counts are included, clearly showcasing the method's accuracy and robustness. The graphs in Figure 14 show that the estimated and exact answers are very close to each other. The 3D graphs make it very clear that the proposed numerical solution has captured all the significant features of this complicated interface problem. This approach not only accurately describes how the main solution behaves, but it also keeps a close eye on the complicated changes and breaks at the interface. These graphic results show how long-lasting and reliable the method is for dealing with complex contact dynamics, which are sometimes hard to describe using standard numerical methods. Figure 15 further illustrates the solution's behavior near the interface, clearly highlighting the discontinuities. The absolute error between the exact and numerical solutions is presented in Figure 16.*



True solution Computed solution, with $F = 6$ and $\Delta\tau = 0.1$

Figure 14. *Cont.*



Computed solution, with $F = 6$ and $\Delta\tau = 0.01$ Computed solution, with $F = 18$ and $\Delta\tau = 0.001$

Figure 14. Assessment of the difference between the computed and true solutions in test scenario 8.

Table 10. Assessment of numerical errors and CPU time in Benchmark Problem 8 employing MQ-RBFs at $\tau = 1$ and $\Delta\tau = 0.001$.

$\beta_1(s), \beta_2(s), \beta_3(s) = (1, 2, 3)$						
	$\epsilon = 1$			$\epsilon = 2$		
F	$E_c(F)$	RMSEs	CPU Time (s)	$E_c(F)$	RMSEs	CPU Time (s)
6	4.4713×10^{-4}	2.2672×10^{-4}	0.0410	4.0820×10^{-4}	1.9275×10^{-4}	0.0241
12	7.8330×10^{-5}	3.3879×10^{-5}	0.0400	8.6163×10^{-5}	3.7978×10^{-5}	0.0311
18	4.6535×10^{-5}	1.4454×10^{-5}	0.0239	4.7408×10^{-5}	1.5904×10^{-5}	0.0344

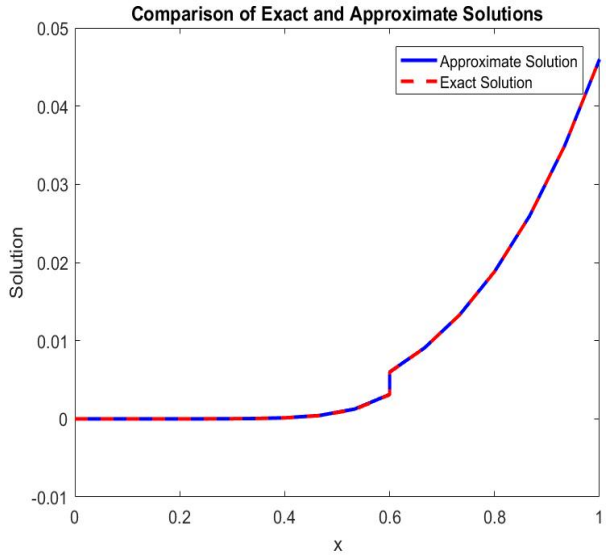


Figure 15. A comparison between the exact and estimated solutions to example 8 using the MCM method for $F = 18$ at $\tau = 1$ shows distinct jumps at the interfaces located at $s = 0.2$ and $s = 0.6$.

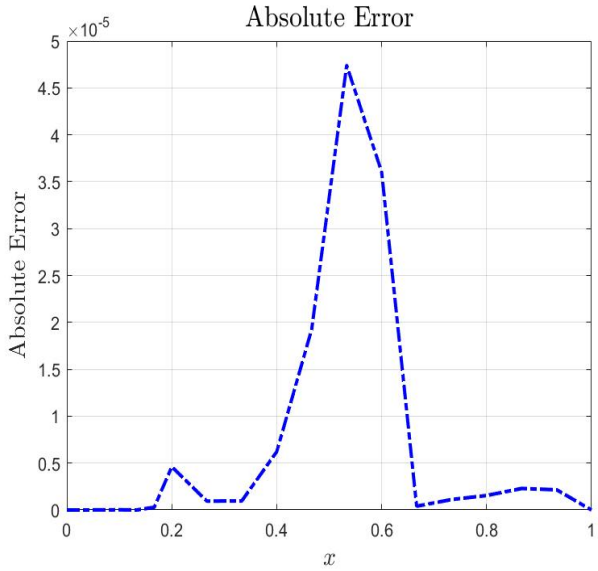


Figure 16. Absolute error between numerical and exact solutions at the final time for example 8.

7. Conclusions

This study presents an efficient and accurate numerical technique for solving linear and nonlinear hyperbolic interface problems involving both single and double interfaces. The proposed method combines MQ-RBFs with the FDM, leveraging the strengths of both approaches. To evaluate the performance of the scheme, benchmark problems with constant and variable coefficients are considered. Numerical results, validated against exact solutions, demonstrate that the accuracy of the method improves with an increasing number of GPs, as reflected by decreasing MAEs and RMSEs. The 3D plots further confirm that the method effectively captures the key features of the solution, even in the presence of sharp gradients across the interfaces. Theoretical stability analysis, supported by detailed numerical experiments, verifies the robustness and reliability of the scheme. Both theoretical and numerical evidence confirm that the method is stable, convergent, and capable of accurately resolving interface problems with discontinuities. In contrast, many existing methods exhibit reduced accuracy under such conditions. This approach has significant potential for extension to two- and three-dimensional telegraph equations that involve interface structures. Moreover, its application to fractional interface problems remains a promising direction for future research.

Author Contributions: Conceptualization, M.A.; methodology, M.A.; validation, N.A., F.K. and F.B.; formal analysis, N.A. and F.K.; investigation, N.A. and I.-L.P.; data curation, F.K. and F.B.; writing—original draft preparation, N.A. and F.K.; writing, review and editing, M.A., F.B. and I.-L.P.; visualization, F.B.; project administration, M.A. and I.-L.P.; funding acquisition, I.-L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No datasets were generated or analyzed during the course of this research; therefore, data sharing does not apply to this article.

Conflicts of Interest: The authors confirm that there are no financial or personal conflicts of interest that could have influenced the outcomes of this research.

References

1. Mohyud-Din, S.T.; Yildirim, A.; Kaplan, Y. Homotopy perturbation method for one-dimensional hyperbolic equation with integral conditions. *Z. Naturforsch. A* **2010**, *65*, 1077–1080. [CrossRef]
2. Aziz, I.; Nisar, M. On the numerical solution of some differential equations with nonlocal integral boundary conditions via Haar wavelet. *Proc. Est. Acad. Sci.* **2022**, *71*, 30–54. [CrossRef]
3. Zaman, S.S.; Amin, R.; Haider, N.; Akgül, A. Numerical solution of Fisher's equation through the application of Haar wavelet collocation method. *Numer. Heat Transf. Part B Fundam.* **2024**, *86*, 2746–2757. [CrossRef]
4. Fahim, M.; Asif, M.; Haider, N.; Amin, R. Hybrid Haar wavelet and meshfree methods for hyperbolic double interface problems: Numerical implementations and comparative performance analysis. *Part. Diff. Eqn. Appl. Math.* **2024**, *11*, 100773. [CrossRef]
5. Liu, W.K.; Liu, Y.; Farrell, D.; Zhang, L.; Wang, S.; Fukui, Y.; Patankar, N.; Zhang, Y.; Bajaj, C.; Hong, J.L. Immersed finite element method and its applications to biological systems. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 1722–1749. [CrossRef]
6. Adel, M.; Khader, M.M.; Babatin, M.M.; Youssef, M.Z. Numerical investigation for the fractional model of pollution for a system of lakes using the SCM based on the Appell type Changhee polynomials. *AIMS Math.* **2023**, *8*, 31104–31117. [CrossRef]
7. Li, Z.; Ito, K. *The Immersed Interface Methods: Numerical Solution of PDEs Involving Interfaces and Irregular Domains*; Frontiers in Applied Mathematics; SIAM: Philadelphia, PA, USA, 2006.
8. Ahmed, A.F.S.; Gamal, I.; Abdelrahman, M.; AbdEl-Bar, M. Derivation of an approximate formula of the Rabotnov fractional-exponential kernel fractional derivative and applied for numerically solving the blood ethanol concentration system. *Fractal Fract.* **2023**, *8*, 30704–30716.
9. Epshteyn, Y.; Phippen, S. High-order difference potentials methods for 1D elliptic type models. *Appl. Numer. Math.* **2015**, *93*, 69–86. [CrossRef]
10. Peskin, C.S. Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **1977**, *25*, 220–252. [CrossRef]
11. Peskin, C.S. The immersed boundary method. *Acta Numer.* **2002**, *11*, 479–517. [CrossRef]
12. Fedkiw, R.P.; Aslam, T.; Merriman, B.; Osher, S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method). *J. Comput. Phys.* **1999**, *152*, 457–492. [CrossRef]
13. Liu, X.-D.; Sideris, T.C. Convergence of the ghost fluid method for elliptic equations with interfaces. *Math. Comput.* **2003**, *72*, 1731–1746. [CrossRef]
14. Liu, X.-D.; Fedkiw, R.P.; Kang, M. A boundary condition capturing method for Poisson's equation on irregular domains. *J. Comput. Phys.* **2000**, *160*, 151–178. [CrossRef]
15. Li, J.; LeVeque, R.J. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* **1994**, *31*, 1019–1044. [CrossRef]
16. Zhou, S.Y.; Wei, G.W. Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *J. Comput. Phys.* **2007**, *224*, 729–756. [CrossRef]
17. Zhou, Y.C.; Liu, J.; Harry, D.L. A matched interface and boundary method for solving multi-flow Navier-Stokes equations with applications to geodynamics. *J. Comput. Phys.* **2012**, *231*, 223–242. [CrossRef]
18. Zhou, Y.C.; Zhao, S.; Feig, M.; Wei, G.W. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *J. Comput. Phys.* **2006**, *213*, 1–30. [CrossRef]
19. Aziz, I.; Islam, S.; Haider, N. Meshless and multi-resolution collocation techniques for parabolic interface models. *Appl. Math. Comput.* **2018**, *335*, 313–332. [CrossRef]
20. Aziz, I.; Islam, S.; Haider, N. Meshless and multi-resolution collocation techniques for steady state interface models. *Int. J. Comput. Methods* **2018**, *15*, 1750073. [CrossRef]
21. Matthew, A. On finite element method for linear hyperbolic interface problems. *J. Niger. Math. Soc.* **2018**, *37*, 41–55.
22. Rana, G.; Asif, M.; Haider, N.; Bilal, R.; Ahsan, M.; Al-Mdallal, Q.; Jarad, F. A modified algorithm based on Haar wavelets for the numerical simulation of interface models. *J. Funct. Spaces* **2022**, *2022*, 1541486. [CrossRef]
23. Asif, M.; Farooq, U.; Riaz, B.; Bilal, F.; Haider, N. Numerical assessment of hyperbolic type double interface problems via Haar wavelets. *Part. Differ. Equ. Appl. Math.* **2024**, *10*, 100665. [CrossRef]
24. Chen, Z.; Zou, Z. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numer. Math.* **1998**, *79*, 175–202. [CrossRef]
25. Kelin, X.; Meng, Z.; Wei, G.-W. MIB Galerkin method for elliptic interface problems. *J. Comput. Appl. Math.* **2014**, *272*, 195–220. [CrossRef]
26. Masood, A.; Elisabeth, L. Local meshless methods for second order elliptic interface problems with sharp corners. *J. Comput. Phys.* **2020**, *416*, 109599. [CrossRef]
27. Masood, A.; Baseer, A.; Islam, S. Local radial basis function collocation method for Stokes equations with interface conditions. *Eng. Anal. Bound. Elem.* **2020**, *119*, 246–256.
28. Asif, M.; Gul, T.; Riaz, M.B.; Bilal, F. Solution of nonlinear telegraph equation with discontinuities along the transmission line using meshless collocation method. *Part. Differ. Equ. Appl. Math.* **2025**, *14*, 101219. [CrossRef]

29. Asif, M.; Bilal, F.; Bilal, R.; Shakeel, M. An efficient algorithm for the numerical solution of telegraph interface model with discontinuous coefficients via Haar wavelets. *Alex. Eng. J.* **2023**, *72*, 275–285. [CrossRef]
30. Asif, M.; Bilal, F.; Haider, N.; Jarad, F. Robust numerical techniques for modeling telegraph equations in multi-scale and heterogeneous environments. *J. Appl. Math. Comput.* **2025**, *70*, 1–36. [CrossRef]
31. Ul Haq, K.S.; Asif, M.; Faheem, M.; Popa, I.-L. Capturing discontinuities with precision: A numerical exploration of 3D telegraph interface models via multi-resolution technique. *Mathematics* **2025**, *13*, 2391. [CrossRef]
32. Fasshauer, G.E. *Meshfree Approximation Methods with MATLAB*; World Scientific Publishing: Hackensack, NJ, USA, 2007.
33. Hardy, R. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **1971**, *76*, 1905–1915. [CrossRef]
34. Tarwater, A.E. *A Parameter Study of Hardy's Multiquadric Method for Scattered Data Interpolation*; Lawrence Livermore National Lab.: Livermore, CA, USA, 1985.
35. Micchelli, C.A. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. In *Approximation Theory and Spline Functions*; Springer: Dordrecht, The Netherlands, 1984; Volume 2, pp. 143–145.
36. Kansa, E.J. Multiquadrics—A scattered data approximation scheme with applications to computational fluid dynamics—II: Solution to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.* **1990**, *19*, 147–161. [CrossRef]
37. Madych, W.R.; Nelson, S.A. Multivariate interpolation and conditionally positive definite functions. *Math. Comput.* **1990**, *54*, 211–230. [CrossRef]
38. Golberg, M.A.; Chen, C.S.; Karur, S.R. Improved multiquadric approximation for partial differential equations. *Eng. Anal. Bound. Elem.* **1996**, *18*, 9–17. [CrossRef]
39. Hon, Y.C.; Cheung, K.F.; Mao, X.Z.; Kansa, E.J. An efficient numerical scheme for Burger's equation. *Appl. Math. Comput.* **1998**, *95*, 37–50.
40. Ahsan, M.; Tran, T.; Hussain, I.; Shakeel, M. A multiresolution collocation method and its convergence for Burgers' type equations. *Math. Meth. Appl. Sci.* **2023**, *46*, 11702–11725. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.