

Article

# Applications of Symmetry-Enhanced Physics-Informed Neural Networks in High-Pressure Gas Flow Simulations in Pipelines

Sultan Alpar \* , Rinat Faizulin, Fatima Tokmukhamedova  and Yevgeniya Daineko 

Mathematical and Computer Modeling Department, Faculty of Computer Technology and CyberSecurity, International Information Technology University (IITU), 34/1 Manas Street, 050040 Almaty, Kazakhstan

\* Correspondence: s.alpar@iitu.edu.kz

**Abstract:** This article presents a detailed examination of the methodology and modeling tools utilized to analyze gas flows in pipelines, rooted in the fundamental principles of gas dynamics. The methodology integrates numerical simulations with modern neural network techniques, particularly focusing on the PINN utilizing the continuous symmetry data inherent in PDEs, which is called the symmetry-enhanced Physics-Informed Neural Network. This innovative approach combines artificial neural networks (ANNs) integrating physical equations, which provide enhanced efficiency and accuracy when modeling various complex processes related to physics with a symmetric and asymmetric nature. The presented mathematical model, based on the system of Euler equations, has been carefully implemented using Python language. Verification with analytical solutions ensures the accuracy and reliability of the computations. In this research, a comparative and comprehensive analysis was carried out comparing the outcomes obtained using the symmetry-enhanced PINN method and those from conventional computational fluid dynamics (CFD) approaches. The analysis highlighted the advantages of the symmetry-enhanced PINN method, which produced smoother pressure and velocity fluctuation profiles while reducing the computation time, demonstrating its capacity as a revolutionary modeling tool. The estimated results derived from this study are of paramount importance for ensuring ongoing energy supply reliability and can also be used to create predictive models related to gas behavior in pipelines. The application of modeling techniques for gas flow simulations has the potential to improve the integrity of our energy infrastructure and utilization of gas resources, contributing to advancing our understanding of symmetry principles in nature. However, it is crucial to emphasize that the effectiveness of such models relies on continuous monitoring and frequent updates to ensure alignment with real-world conditions. This research not only contributes to a deeper understanding of compressible gas flows but also underscores the crucial role of advanced modeling methodologies in the sustainable management of gas resources for both current and future generations. The numerical data covered the physics of the process related to the modeling of high-pressure gas flows in pipelines with regard to density, velocity and pressure, where the PINN model was able to outperform the classical CFD method for velocity by 170% and for pressure by 360%, based on  $L_\infty$  values.

**Keywords:** physics-informed neural networks; artificial neural networks; CFD; gas flow; Euler equation



**Citation:** Alpar, S.; Faizulin, R.; Tokmukhamedova, F.; Daineko, Y. Applications of Symmetry-Enhanced Physics-Informed Neural Networks in High-Pressure Gas Flow Simulations in Pipelines. *Symmetry* **2024**, *16*, 538. <https://doi.org/10.3390/sym16050538>

Academic Editor: Qing Li

Received: 24 March 2024

Revised: 24 April 2024

Accepted: 25 April 2024

Published: 30 April 2024



**Copyright:** © 2024 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

The efficient transportation and distribution of natural gas play a pivotal role in the economic development and energy security of nations worldwide [1]. In Kazakhstan, a country abundant in oil and gas resources, the modeling of the gas flow in tubes holds significant importance across various sectors. Understanding and accurately predicting the behavior of gas within pipelines and distribution networks are paramount for optimizing infrastructure, ensuring energy supply reliability and promoting environmental sustainability [2,3]. Thus, modeling gas flow in tubes holds critical significance for Kazakhstan,

impacting its oil and gas industry, energy sector development, infrastructure planning, environmental situation, research and innovation endeavors, and safety assurance. Through advanced machine learning and computational fluid dynamics simulations, the country can optimize its energy resources, promote sustainable development, and bolster resilience in its energy infrastructure.

The comprehensive modeling of gas flow in tubes relies fundamentally on solving the partial differential equations (PDEs) governing fluid dynamics. PDEs, such as the Euler equations, form the mathematical framework for describing the conservation laws of mass, momentum, and energy in fluid flows. Solving these equations numerically using advanced computational techniques enables the accurate prediction of gas behavior, including shock wave propagation, turbulence effects, and flow characteristics within pipelines and tubes. In recent years, significant progress has been made in computational fluid dynamics (CFD) [4,5] for numerically solving PDEs, particularly in solving the Navier–Stokes equations [6–8], which has revolutionized the ability to simulate and analyze complex fluid flow phenomena using techniques like finite difference (FD) [9–11] and finite volume (FV) methods [12,13]. Even though models such as the Navier–Stokes equations [14–18] are able to describe the underlying problem, at this point, modeling still requires a high computation cost to achieve a high precision.

The rapid growth in data, the parallel emergence of computing, and the advent of graphics processors (GPUs), together with the advanced theoretical results in numerical analysis, have resulted in the explosive growth of machine learning (ML) [19,20], particularly in the area of physical problems and mathematical physics. Physics-Informed Neural Networks (PINNs) [21–25] are a type of artificial neural network that can dramatically reduce the computational complexity of modeling physical processes. The further need for developing the potential of these models is still relevant and they are still not a uniform method of CFD, but in the field of forward problems, they have a higher precision and a promising future. In recent advancements, PINNs have been employed to solve nuclear physics problems [26], astrophysics [27], and compressible flows governed by the Navier–Stokes equations [28,29], offering efficient simulations for different aspects of nature. However, despite these advancements, achieving optimal results in compressible flows remains a difficult process and requires skill.

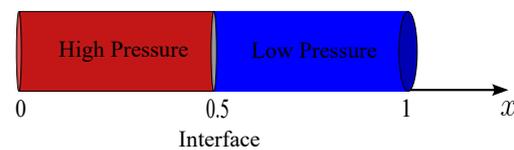
This research considers a transient gas flow in pipelines using symmetry-enhanced Physics-Informed Neural Networks, with manual selection of the necessary hyperparameters using a weighted loss function [30]. The PINN leverages the continuous Lie symmetry information inherent in PDEs, wherein the invariant surface conditions (ISCs) induced by these symmetries are integrated into the PINN's loss function [31]. When PDEs, along with their initial and boundary conditions, possess a Lie symmetry, the solutions remain invariant under such symmetry and also satisfy the ISCs. These ISCs, being intrinsic properties of the PDEs, impose crucial additional constraints on the solutions. Since the standard loss function in the PINN is the mean square error of the PDE residual, adding the ISCs to the loss function will definitely increase the number of constraints for the objective function during the optimization process and improve the accuracy of the solutions. In addition to this, after an evaluation of the PINN model's performance, the following problems will be solved to demonstrate the effectiveness of the method: Burgers' equation [32,33], viscous Burgers' equation [33,34] and Euler's equations of gas dynamics [35,36], where the main focus will be on the compressible flow problem.

The article is organized as follows. Section 2 introduces the physical problem in the general form of a PDE with initial and boundary conditions. The following Section 3 illustrates the numerical continuous PINN model with a forward propagation procedure for predictions of true solutions and the multi-objective loss for optimization, in addition to the architecture of the model. In addition to this, the discrete CFD model, which applies the four-step Runge–Kutta method with a general form and space derivative discretization. In Section 4, one-dimensional problems are solved with short discussions of their mathematical form: a PDE with initial and boundary conditions and physical

descriptions with graphical comparisons to analytical solutions and  $L$  norm error measures. It is also worth noting the modeling approach in comparison with other works; for example, the authors of [37] use the sequence-to-sequence (seq2seq) technique, which splits the domains into parts, while in our case, the model is trained on all time segments, which will be shown to be a valid approach.

## 2. Physical Problem

The physical problem involves the gas flow in a pipe, which is described by a system of Euler's equations. Equations govern the motion of adiabatic, inviscid, and compressible fluids, providing a powerful model for analyzing fluid dynamics in many practical scenarios. In order to capture the speed of the rarefaction wave, the discontinuity and the shock contacts, the problem is considered in a chamber that is separated into regions with high pressure and low pressure by interface points at  $x = 0.5$ , as shown in Figure 1. The physical process is observed for the time domain  $t \in [0, 0.1]$ . The space domain is defined as  $x \in [0, 1]$ .



**Figure 1.** Illustration of the pipe physical domain.

Euler's system of equations represents the conservation of mass, momentum, and energy. The conservation of mass in the system is described by the continuity equation. It determines how changes in the density affect the fluid flow and ensures that the total mass within a fluid volume remains constant over time:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0, \quad (1)$$

where the density of gas  $\rho$  [ $\text{kg} \cdot \text{m}^{-3}$ ] changes with respect to time and space as influenced by the flow velocity component  $u$  [ $\text{m} \cdot \text{s}^{-1}$ ].

The conservation of momentum is based on Newton's laws of motion:

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} = 0. \quad (2)$$

The momentum equation describes how the velocity of the fluid changes in response to convection and pressure  $p$  [ $\text{kg} \cdot \text{m}^{-1} \cdot \text{s}^{-2}$ ]. It influences the formation of shocks and determines the acceleration of the gas flow.

The general form of the energy equation is derived from the principles of the conservation of energy, which is expressed as:

$$\frac{\partial E}{\partial t} + \frac{\partial[u(E + p)]}{\partial x} = 0. \quad (3)$$

The energy equation describes the conservation of energy in fluid flows; it influences the system by determining how energy is transferred through advection and how work is done by pressure forces, where  $E$  [ $\text{J} \cdot \text{m}^{-3}$ ] is the total energy and is defined as,

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2, \quad (4)$$

with a specific heat ratio  $\gamma$  [–] equal to 1.4.

Taking into account shock phenomenon generation, the following initial conditions are used:

$$(\rho, u, p)|_{t=0} = \begin{cases} (1, 0, 1), & 0 \leq x \leq 0.5, \\ (0.125, 0, 0.1), & 0.5 \leq x \leq 1. \end{cases} \quad (5)$$

The Dirichlet boundary conditions are applied as boundary conditions, meaning that the quantities take on the values prescribed by the initial conditions at both boundaries, which ensures that the flow at the boundaries remains consistent with the initial conditions.

The transient gas flow in pipelines can be expressed in general form as:

$$\frac{\partial U}{\partial t} + \mathcal{N}[U; \lambda] = 0, \quad x \in \Omega, \quad t \in [0, t_f], \quad (6)$$

$$U(x, 0) = \mathcal{I}(x), \quad x \in \Omega, \quad (7)$$

$$U(x, t) = \mathcal{B}(t), \quad x \in \partial\Omega, \quad t \in [0, t_f]. \quad (8)$$

The solution is denoted by  $\hat{U}(x, t) \in \mathbb{R}$  with spatial coordinate  $x \in \mathbb{R}$ , time  $t$  and parameter of the PDE  $\lambda$ , where  $\Omega$  is the computational domain and  $\partial\Omega$  is the boundary. Here,  $\mathcal{N}(\cdot)$  is a linear or nonlinear differential operator, the initial condition operator is  $\mathcal{I}(\cdot)$  and the boundary condition operator is  $\mathcal{B}(\cdot)$ . Note that this form of representation of a parametrized PDE can be used for other problems as well.

### 3. Methodology

#### 3.1. Continuous PINN Time Model

The continuous symmetry-enhanced PINN time model is used as framework for solving PDEs; the left-hand side of the equation is defined as,

$$f(x, t) = \frac{\partial U}{\partial t} + \mathcal{N}[U; \lambda], \quad (9)$$

and the fully connected deep  $L$  neural network with inputs  $d_{in}$  and outputs  $d_{out}$  is defined as,

$$\mathcal{G} : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}, \quad (10)$$

The forward propagation procedure will be used to approximate solution  $\hat{U}(x, t)$  as follows,

$$\begin{aligned} \mathcal{G}^{[l]} &= (x, t) \in \mathbb{R}^{d_{in}}, l = 0 \\ \mathcal{G}^{[l]} &= \sigma(W^{[l]}\mathcal{G}^{[l-1]} + b^{[l]}) \in \mathbb{R}^{n_l}, 1 \leq l \leq L-1, \\ \mathcal{G}^{[L]} &= W^{[L]}\mathcal{G}^{[L-1]} + b^{[L]} \in \mathbb{R}^{d_{out}}, l = L, \end{aligned} \quad (11)$$

Here, activation value  $\mathcal{G}^{[l]}$  denotes the output of the  $l^{th}$  layer with weight matrix  $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$ , bias vector  $b^{[l]} \in \mathbb{R}^{n_l}$  and nonlinear activation function  $\sigma(\cdot)$ , where  $n_l$  denotes the number of neurons in the  $l^{th}$  layer. For the last layer  $L$ , a linear activation function with  $d_{out}$  outputs is used to approximate the solution, as  $\hat{U}(x, t) \approx U(x, t) = W^{[L]}\mathcal{G}^{[L-1]} + b^{[L]}$ .

The construction that converts the PDE into an optimization problem is the combined multi-objective weighted squared loss  $\mathcal{L}$ ,

$$\mathcal{L} = (1 - \alpha)(\mathcal{L}_{\mathcal{I}} + \mathcal{L}_{\mathcal{B}}) + \alpha\mathcal{L}_{\mathcal{F}}, 0 \leq \alpha \leq 1, \quad (12)$$

where the individual loss terms are defined as:

$$\mathcal{L}_{\mathcal{I}} = \frac{1}{N_I} \sum_{i=1}^{N_I} (\mathcal{I}(x_i^i) - U(x_i^i, t_i^i))^2, \quad (13)$$

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{N_B} \sum_{i=1}^{N_B} (\mathcal{B}(t_B^i) - U(x_B^i, t_B^i))^2, \quad (14)$$

$$\mathcal{L}_{\mathcal{F}} = \frac{1}{N_F} \sum_{i=1}^{N_F} f(x_F^i, t_F^i)^2, \quad (15)$$

Weights and biases denoted by  $\theta$  can be trained by minimizing,

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta), \quad (16)$$

This minimization process is performed via the gradient-based ADAM Algorithm 1.

---

**Algorithm 1** ADAM Optimization Algorithm

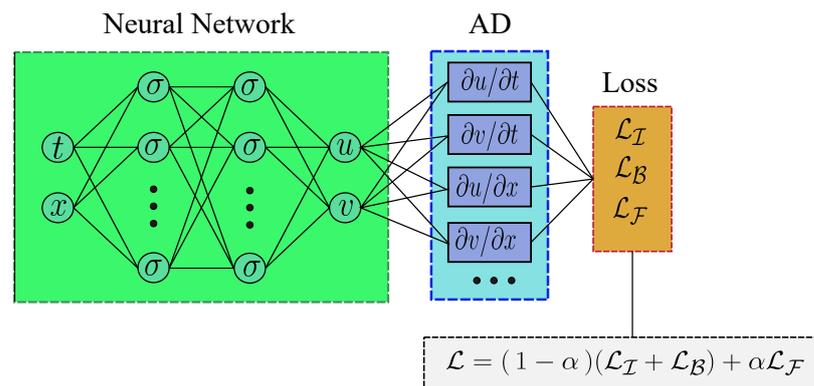
---

- 1: **Input:**  $\gamma$  (learning rate lr),  $\beta_1 = 0.9, \beta_2 = 0.999$  (coefficients),  $\epsilon = 10^{-8}$  (numerical stability term),  $\theta_0$  (parameters),  $\mathcal{L}(\theta)$  (objective)
  - 2: **Initialize:**  $m_0 \leftarrow 0$  (first moment),  $v_0 \leftarrow 0$  (second moment)
  - 3: **for**  $i = 1$  **to** ... **do**
  - 4:    $g_i \leftarrow \nabla_{\theta} \mathcal{L}_i(\theta_{i-1})$
  - 5:    $m_i \leftarrow \beta_1 m_{i-1} + (1 - \beta_1) g_i$
  - 6:    $v_i \leftarrow \beta_2 v_{i-1} + (1 - \beta_2) g_i^2$
  - 7:    $\hat{m}_i \leftarrow m_i / (1 - \beta_1^i)$
  - 8:    $\hat{v}_i \leftarrow v_i / (1 - \beta_2^i)$
  - 9:    $\theta_i \leftarrow \theta_{i-1} - \gamma \hat{m}_i / (\sqrt{\hat{v}_i} + \epsilon)$
  - 10: **end for**
- 

Here, to train the initial conditions,  $\{x_I^i, t_I^i\}_{i=1}^{i=N_I}$  is used as input and  $\{\mathcal{I}(x_I^i)\}_{i=1}^{i=N_I}$  is used as output; for the boundary conditions,  $\{x_B^i, t_B^i\}_{i=1}^{i=N_B}$  is the input and  $\{\mathcal{B}(t_B^i)\}_{i=1}^{i=N_B}$  is the output. For training, only the residual inputs are required  $\{x_F^i, t_F^i\}_{i=1}^{i=N_F}$ , where all data points are picked at random either from the entire computational domain or some subset without repetitions and  $N_I + N_B \ll N_F$ . The multi-objective optimization goal is to train both supervised  $\mathcal{L}_{\mathcal{I}}, \mathcal{L}_{\mathcal{B}}$  and unsupervised  $\mathcal{L}_{\mathcal{F}}$  losses, where the residual loss  $\mathcal{L}_{\mathcal{F}}$  plays a role as a regularization term that allows the model to generalize information outside of some local region and prevent overfitting by penalizing the physical constraints and  $\alpha$  is the balance between loss importances. It can be seen that with a higher number of initial and boundary data, the physics loss becomes less important  $\alpha \rightarrow 0$  and vice versa, i.e., with less available data, physics loss becomes more important  $\alpha \rightarrow 1$ .

For the calculations of the partial derivatives for  $\mathcal{L}_{\mathcal{F}}$ , automatic differentiation (AD) is applied, which means that outputs of the neural network  $u(x, t)$  are directly used to calculate its derivatives with respect to the inputs  $(x, t)$ , as  $[\partial u / \partial x, \partial u / \partial t, \dots]$ .

In Figure 2, a schematic representation of the PINN is shown with key elements (neural network, automatic differentiation AD, loss) and inputs  $(x, t)$  and outputs  $u$  and  $v$ . AD outputs are differentiated with respect to their inputs in order to construct the  $\mathcal{L}_{\mathcal{F}}$  loss; then, the outputs are directly used to compute  $\mathcal{L}_{\mathcal{I}}$  and  $\mathcal{L}_{\mathcal{B}}$  losses. Then, the final loss  $\mathcal{L}$  is computed.



**Figure 2.** Physics-Informed Neural Network.

### 3.2. Discrete-Time Model

The Runge–Kutta (RK4) method for time integration is used as a discrete time model in CFD; it is particularly effective in addressing nonlinear problems. This method is characterized by a fourth-order accuracy in time. The RK4 method is applied to the general form of the case problem:

$$\frac{\partial U}{\partial t} = \mathcal{N}(U). \tag{17}$$

The Runge–Kutta method’s four steps are written as:

$$\begin{aligned} U^{(1)} &= U^n + \frac{\Delta t}{2} \mathcal{N}^n \\ U^{(2)} &= U^n + \frac{\Delta t}{2} \mathcal{N}^{(1)} \\ U^{(3)} &= U^n + \Delta t \mathcal{N}^{(2)} \\ U^{n+1} &= U^n + \frac{\Delta t}{6} (\mathcal{N}^n + 2\mathcal{N}^{(1)} + 2\mathcal{N}^{(2)} + \mathcal{N}^{(3)}) \end{aligned} \tag{18}$$

Given the fact that the RK4 scheme only affects time, schemes are needed to approximate the space for convective and viscous terms in order to achieve second-order accuracy in space. Thus, the upwind scheme is used for the flux terms:

$$-u \frac{\partial \zeta_i^t}{\partial x} = -\frac{u_R \zeta_R - u_L \zeta_L}{\Delta x}, \tag{19}$$

where  $u_R$  and  $u_L$  are the average of the nodal point values and are defined as follows:

$$u_R = \frac{u_i^t + u_{i+1}^t}{2}, u_L = \frac{u_i^t + u_{i-1}^t}{2}, \tag{20}$$

The functions  $\zeta_R$  and  $\zeta_L$  depend on the signs of  $u_R$  and  $u_L$  and are defined as follows:

$$\zeta_R = \begin{cases} \zeta_i^t, u_R \geq 0 \\ \zeta_{i+1}^t, u_R < 0 \end{cases}, \zeta_L = \begin{cases} \zeta_{i-1}^t, u_L \geq 0 \\ \zeta_i^t, u_L < 0 \end{cases} \tag{21}$$

This approach, specifically employed for convective terms, calculates the average values of velocities at the cell boundaries of the spatial grid around each node point. It uses the directions of these velocities to establish at which grid node the values of  $\zeta$  need to be calculated, forming the differences in the flow. The central difference scheme is used for derivatives not involving non-linearity:

$$\frac{\Delta F(u)_i^t}{\Delta x} = \frac{F(u)_{i+1}^t - F(u)_{i-1}^t}{2\Delta x}, \tag{22}$$

The central difference scheme provides an approximation of spatial derivatives for the viscous part of the equations, where  $F$  is the flux function.

#### 4. Computational Experiments

First, the viscous and inviscid Burgers equations are considered as a reference method to verify both approaches. The computational experiments of the PINN and RK4 models will be compared to its analytical solutions graphically and with  $L$  norm measures. The following are the results of the case study problem.

The input data for the PINN model, such as the NN setup, the number of initial conditions points  $N_I [-]$ , the number of boundary conditions points  $N_B [-]$ , the number of interior points  $N_F [-]$ , the nonlinear activation function  $\sigma(\cdot)$ , the learning rate  $lr$  for the ADAM optimizer, value  $\alpha [-]$ , and the number of iterations epochs, are given in Table 1, where  $N_x [-]$ ,  $N_t [-]$  are the grid sizes for the space and time domain, respectively. In addition to this, input data for the CFD model, which are the grid size  $N_x$  and the Courant–Friedrichs–Lewy (CFL)  $[-]$  number, are presented in Table 2.

For the PINN model, all parameters were found either manually or through a grid search [21,30]. The RK4 model usually does not require any sophisticated parameter search; usually, a larger grid size is used to approximate the derivatives and a smaller CFL number is used to ensure the stability is sufficient.

**Table 1.** Input data for the PINN model.

Model	NN Setup	$N_x [-]$	$N_t [-]$	$N_I [-]$	$N_B [-]$	$N_F [-]$	$\sigma(\cdot)$	lr	$\alpha [-]$	Epochs
Burgers	$[3 \times 64]$	128	128	128	256	16,384	Tanh	$10^{-3}$	0.25	23,000
Viscous Burgers	$[3 \times 64]$	128	128	128	256	16,384	Tanh	$10^{-3}$	0.1	8000
Euler 1D	$[10 \times 20]$	256	256	30	50	13,600	Tanh	$3 \times 10^{-4}$	$\frac{5.3 \times}{10^{-4}}$	40,000

**Table 2.** Input data for the CFD model.

Model	$N_x [-]$	CFL $[-]$
Burgers	600	0.25
Viscous Burgers	200	0.25
Euler 1D	1000	0.01

The precision of the proposed models will be measured with the  $L_\infty$  norm computed for the time  $t_f$  and defined as follows:

$$L_\infty = \|\hat{u} - u\|_\infty, \quad (23)$$

where  $\hat{u}$  and  $u$  are analytical and numerical solutions, respectively.

##### 4.1. Verification

###### Inviscid Burgers' Equation

Despite the simplicity of Burgers' equation without a viscous part, it plays a significant role in physics due to its ability to capture essential fluid phenomena such as turbulence and wave propagation. This equation describes the behavior of non-linear waves and shock waves:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0, x \in [0, 1], t \in [0, 1], \quad (24)$$

with the initial condition:

$$u(x, 0) = \frac{1}{2\pi t_s} \sin(2\pi x), x \in [0, 1], \quad (25)$$

where the time at which a shock forms is indicated by the  $t_s$  [–] parameter and set equal to 1. Next, periodic boundary conditions are used:

$$u(0, t) = u(1, t) = 0, t \in [0, 1], \quad (26)$$

which lead to the next analytical solution  $\hat{u}(x, t)$ :

$$\hat{u}(x, t) = \frac{1}{2\pi t_s} \sin[2\pi(x - \hat{u}t)], \quad (27)$$

which can be solved numerically due to implicitity by a simple fixed-point iteration for time  $t_n$  and location point  $x_i$  with the initial condition as  $\hat{u}_i^0$ :

$$\hat{u}_i^{k+1} = \frac{1}{2\pi t_s} \sin[2\pi(x_i - \hat{u}_i^k t_n)], \quad k = 0, 1, \dots, \quad (28)$$

with the following convergence criterion:

$$|\hat{u}_i^{k+1} - \hat{u}_i^k| < 10^{-15}.$$

This solution can be used continuously to measure the numerical accuracy up to shock formation as well as to test the shock-capturing capability after shock formation.

$f(x, t)$  and  $\mathcal{N}(U)$  are given by:

$$f(x, t) := \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x},$$

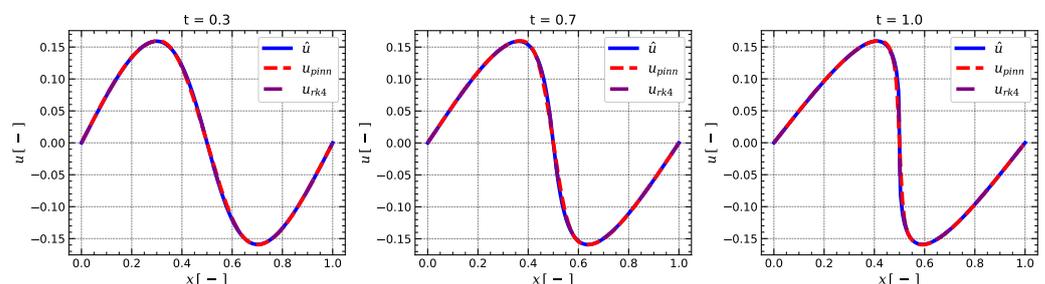
$$\mathcal{N}(U) := -u \frac{\partial u}{\partial x}.$$

The time step stability criterion for the CFD model is defined through the CFL condition and recalculated for each time iteration  $n$ :

$$s_{max}^{(n)} = \max_i |u_i^n|$$

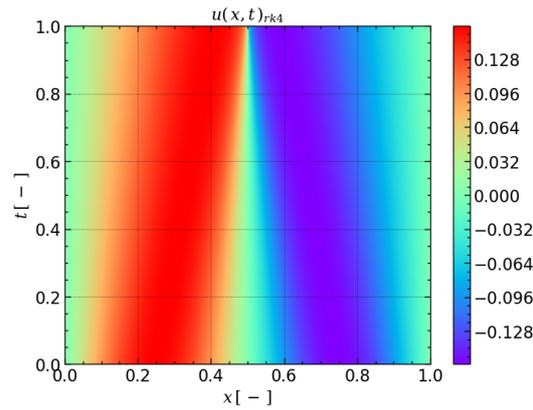
$$\Delta t^{(n)} = \frac{CFL \Delta x}{s_{max}^{(n)}} \quad (29)$$

Figure 3 illustrates the comparison of the numerical solutions of the inviscid Burgers equation with its analytical solution  $\hat{u}$ . The graph highlights the accuracy of both numerical approaches in capturing the complex behavior of this fluid dynamics problem. The  $L_\infty$  norm is computed for the time  $t_f = 1$  for both models.



**Figure 3.** Numerical  $u$  solutions and analytical  $\hat{u}$  solution for the 1D Burgers equation at time  $t = 0.3$ ,  $t = 0.7$ ,  $t = 1$  from left to right. The  $L_\infty$  error at time  $t_f = 1$  for the PINN and CFD models is 0.0449 and 0.0185, respectively.

Figure 4 shows a contour plot of the  $u$  distribution over the entire computational domain  $t \in [0, 0.1]$  and  $x \in [0, 1]$ . Here, from time  $t = 0$ , the smooth solution turns into a shock at  $t = 1.0$ , which is also in high accordance with the analytical solution. Thus, these methods capture all shock formations during the entire time period.



**Figure 4.** Contour plot for the entire computational domain for the CFD model for the Burgers equation.

### Viscous Burgers Equation

The viscosity introduces additional mathematical complexity, allowing for the study of phenomena such as the dissipation of kinetic energy, boundary layer separation, and the transition to turbulence. This transition provides valuable insights into the behavior of fluid flows near solid boundaries and in regions with a high velocity gradient. The viscous Burgers equation is described as:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, x \in [0, 1], t \in [0, 1], \quad (30)$$

where  $\nu [-]$  affects the diffusion term, which characterizes the viscosity in the equation, and is set to 0.01. Next, the initial conditions are prescribed:

$$u(x, 0) = \sin(\pi x), x \in [0, 1], \quad (31)$$

followed by the boundary conditions:

$$u(0, t) = u(1, t) = 0, t \in [0, 1]. \quad (32)$$

The analytical solution  $\hat{u}(x, t)$  is defined as follows:

$$\hat{u}(x, t) = \frac{2\pi\nu \sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2\nu t) n \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2\nu t) \cos(n\pi x)}, \quad (33)$$

where  $a_0$  and  $a_n$  are defined as follows:

$$\begin{aligned} a_0 &= \int_0^1 \exp\left[-\frac{1}{2\pi\nu}(1 - \cos(\pi x))\right] dx \\ a_n &= 2 \int_0^1 \exp\left[-\frac{1}{2\pi\nu}(1 - \cos(\pi x))\right] \cos(n\pi x) dx, \quad n = 1, 2, \dots \end{aligned} \quad (34)$$

$f(x, t)$  and  $\mathcal{N}(U)$  are given by:

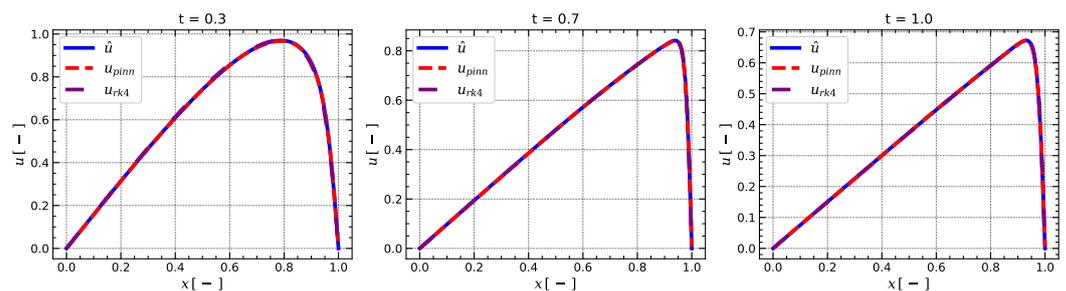
$$\begin{aligned} f(x, t) &:= \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2}, \\ \mathcal{N}(U) &:= -u \frac{\partial u}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2}, \end{aligned}$$

The time step stability criterion for time iteration  $n$  for the CFD model through the CFL condition is defined as follows:

$$s_{max}^{(n)} = \max_i |u_i^n|$$

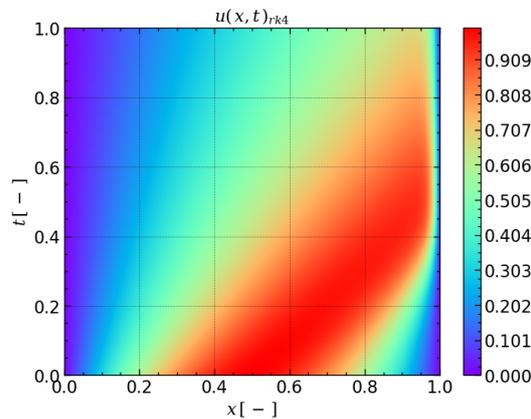
$$\Delta t^{(n)} = \min \left[ \frac{CFL \Delta x}{s_{max}^{(n)}}, \frac{\Delta x^2}{2\nu} \right] \quad (35)$$

Figure 5 demonstrates the comparison of the numerical solutions and the analytical solution of the Burgers equation with viscosity with the  $L_\infty$  norm measures at time  $t_f = 1$ . The presence of a viscous term in the equation has a distinct impact on the solution. Both methods show a high accuracy in capturing shock waves.



**Figure 5.** Numerical  $u$  solutions and analytical  $\hat{u}$  solutions for the 1D viscous Burgers equation at time 0.3, 0.7, 1 from left to right. The  $L_\infty$  error at time  $t_f = 1$  for the PINN and CFD models is 0.0019 and 0.0005, respectively.

Figure 6 demonstrates the behavior of the fluid flow for the entire computational domain. The flow exhibits the same pattern as in the previous case; from time  $t = 0$ , the smooth solution turns into a shock at  $t = 1.0$ . The shock formations are captured over the entire time period as well.



**Figure 6.** Contour plot for the entire computational domain for the CFD model for the viscous Burgers equation.

#### 4.2. Case Study

Since both methods have been verified, a case study is presented for a further analysis of their ability to capture the unique characteristics of gas flows in pipes, which involves contact discontinuity and shock waves. The analytical solutions  $(\hat{\rho}, \hat{u}, \hat{p})$  for the case study are taken from Section 2.4.1 in [35]. In Case 1, the rarefaction shock, the whole tube region/computational domain is divided into five regions and in each part, the solution is found through thermodynamical equations.

$f(x, t)$  and  $\mathcal{N}(U)$  are given by,

$$f(x, t) := \frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix},$$

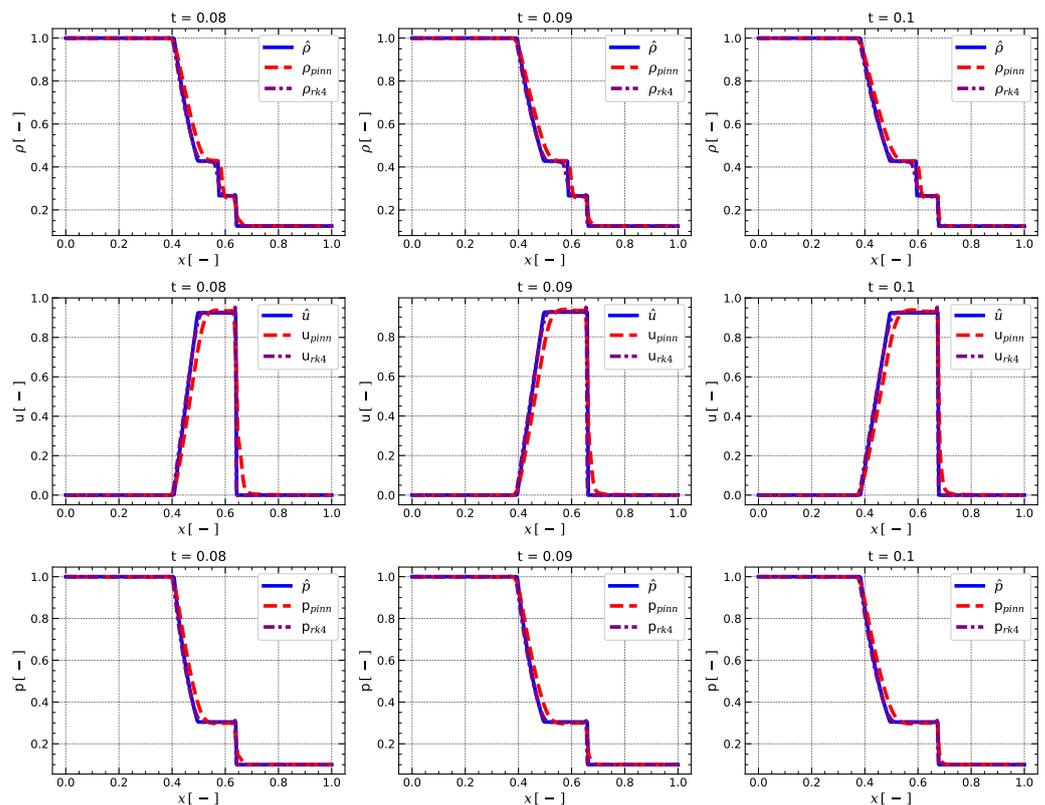
$$\mathcal{N}(U) := -\frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix},$$

In this case, the time stability condition for the CFD model for time iteration  $n$  is found through the CFL condition as follows:

$$s_{max}^{(n)} = \max_i \left[ |u_i^n| + \sqrt{\frac{\gamma p_i^n}{\rho_i^n}} \right] \quad (36)$$

$$\Delta t^{(n)} = \frac{CFL \Delta x}{s_{max}^{(n)}}$$

Figure 7 illustrates plots of the density  $\rho$ , velocity  $u$  and pressure  $p$  with comparisons to the analytical solutions.

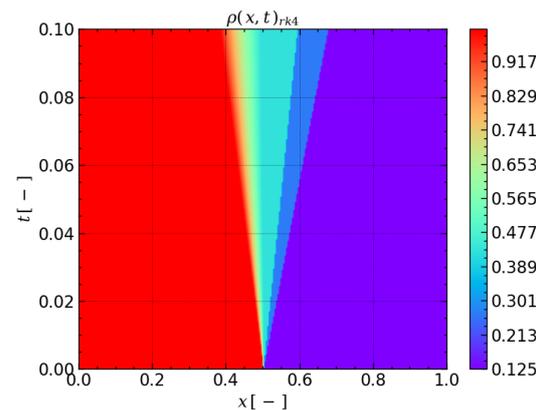


**Figure 7.** Numerical solutions ( $\rho$ ,  $u$ ,  $p$ ) and analytical solutions ( $\hat{\rho}$ ,  $\hat{u}$ ,  $\hat{p}$ ) for the 1D Euler's equations of gas dynamics at time 0.08, 0.09, and 0.1. The  $L_\infty$  norms at time  $t_f = 0.1$  for the PINN and CFD models are 0.018977 and 0.08876 for the density  $\rho$ , 0.163342 and 0.447174 for the velocity  $u$ , and 0.026128 and 0.121339 for the pressure  $p$ .

The significance of these plots lies in their ability to demonstrate the accuracy of the numerical methods in capturing the complex behavior of compressible flows. It can be seen that the RK4 model has some oscillations in velocity  $u$  and pressure  $p$  profiles, while the PINN model is more smooth. The  $L_\infty$  norms were computed at time  $t_f = 0.1$ . Given that  $L_\infty$  norms for the PINN and CFD models for the velocity  $u$  are 0.1633 and 0.4472 and

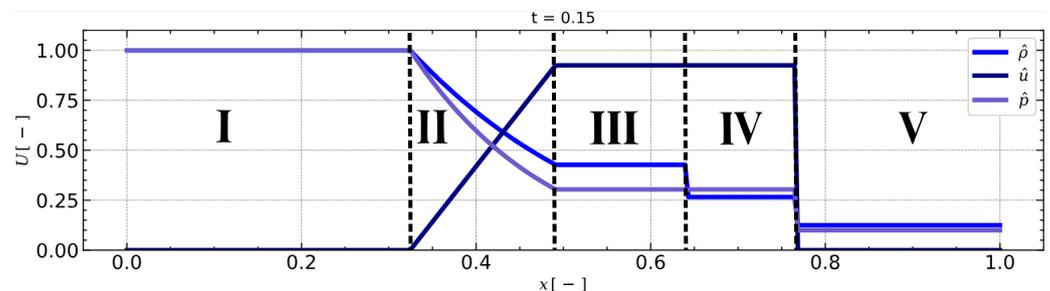
0.0261 and 0.1213 for the pressure  $p$ , the PINN model significantly outperformed the CFD model by 170 % for the velocity  $u$  and by 360 % for the pressure  $p$  relatively, which is the main advantage of the PINN model.

In Figure 8, a contour plot for the density  $\rho$  over the entire computational domain with the interface, a left nonlinear wave, a right nonlinear wave, rarefaction, and shock formations are shown, where a high pressure is present in the left zone from the discontinuity point  $x = 0.5$ , shown in red, and the right zone with a low pressure is shown in blue. The distribution looks similar at the initial time  $t = 0$ , as shown in the physical domain in Figure 1.



**Figure 8.** Contour plot for the entire computational domain for the CFD model for Euler's equations.

In order to understand the physics behind the estimated results, let us consider the case before any wave has reached the left or right boundary, for instance, at time  $t = 0.15$ . Figure 9 illustrates density, velocity and pressure formation in this case.



**Figure 9.** Analytical solutions ( $\hat{\rho}$ ,  $\hat{u}$ ,  $\hat{p}$ ) for the 1D Euler equations of gas dynamics at time  $t = 0.15$ .

Regions I and V show values of the physical quantities that are the same as the initial conditions. Region II denotes the position where the head and tail of the rarefaction wave move leftward. While the solution remains continuous within this region, certain derivatives of the fluid quantities might not exhibit continuity. Region III shows the element position of the fluid that is initially at interface point  $x = 0.5$  reached at time  $t = 0.15$ . The right end points of Region III and Region IV are called a contact discontinuity. It is observed that the pressure and the normal component of velocity remain continuous across the contact discontinuity. However, the density is not continuous across a contact discontinuity. The end point of Region IV is the location of the shock wave moving to the right. During the shock, all of the quantities will in general be discontinuous.

The pressure results reveal the locations and strengths of the shock waves and rarefaction waves as they propagate through the tube. The density results provide information about the compression and expansion of the fluid as the waves pass through it. The velocity results show how the fluid velocities change as waves propagate through the tube, including its acceleration due to compression and deceleration due to expansion.

All the results obtained from computational experiments are sufficient for both models in terms of comparing them with analytical plots and using L norms. The results show that the CFD model outperforms the PINN model, but this was achieved due to the higher number of points for space discretization. The CFD model starts to exhibit oscillations for velocity and pressure, while the PINN model gives a smoother solution. This effect could be explained by the PINN's nature, where it integrates physical laws into the numerical solution. Thus, it creates smooth and continuous solutions. Although the CFD model gives better computational and optimal results for the case of one-dimensional problems, as the dimensionality of the problem and the number of equations increase, the CFD model will require a large grid size and a long computational process and the obtained solutions may exhibit oscillations. Meanwhile, the PINN model can generalize the information using a subset of the entire computational domain in a relatively small number of iterations compared to the CFD model, resulting in a smoother and more optimal solution.

## 5. Discussion

In the realm of numerical techniques for solving differential equations, both classical CFD and PINN methods are unique and offer distinct advantages. Understanding the comparative strengths and limitations of these methods is crucial for selecting the most appropriate approach for a given problem.

### **Accuracy and Precision:**

Classical CFD methods are renowned for their ability to achieve a high accuracy, especially when employing fine grid resolutions. These methods excel in problems where the underlying physics is well understood and can be accurately represented by explicit equations. However, CFD methods may encounter challenges near discontinuities or sharp gradients, leading to accuracy degradations in such regions.

Conversely, PINNs adopt a data-driven approach, leveraging neural networks to approximate solutions based on available data. While PINNs may sacrifice some accuracy compared to FDMs, particularly in problems with well-defined physics, they offer advantages in handling complex or data-driven scenarios. PINNs can provide smoother and more continuous solutions, making them suitable for problems with irregular geometries or unstructured data.

### **Computational Efficiency:**

Classical CFD methods typically involve explicit grid generation and solving equations at each grid point, which can be computationally intensive for large-scale problems. Conversely, PINNs offer computational efficiency once trained, as they can rapidly generate solutions without the need for explicit grid structures. This efficiency is particularly advantageous for problems with complex geometries or time-dependent constraints.

### **Flexibility and Adaptability:**

Classical CFD methods are constrained by grids shapes and may struggle with problems involving complex or irregular geometries. In contrast, PINNs offer greater flexibility and adaptability, as they can handle unstructured data and irregular domains more effectively. PINNs have the potential to generalize well to new conditions or scenarios not explicitly included in the training data, enhancing their applicability to a wide range of problems.

In conclusion, the choice between classical CFD and PINN methods depends on the specific requirements of the problem, including the nature of the differential equations, the availability of data, the complexity of the geometry, and the computational resources. CFD methods excel in problems with well-understood physics and structured domains, offering a high accuracy with a fine grid resolution. On the other hand, PINNs provide flexibility, generalization, and efficiency advantages, especially for complex or data-driven problems with irregular geometries. By understanding the comparative strengths and limitations of these methods, researchers and engineers can make informed decisions to select the most suitable approach for their specific application.

## 6. Conclusions

This article investigates gas flow in a pipeline to capture the propagation speed of the rarefaction wave, the contact discontinuity and the shock discontinuity. First, the physical model is defined to provide gas characteristics. Initial and boundary conditions were specified according to the case problem. Then, two numerical models were proposed: a continuous symmetry-enhanced PINN model and a discrete CFD model (Runge–Kutta 4). The methods were verified through the one-dimensional inviscid and viscous Burgers equations. This was followed by case study problem estimation, which is described by a system of one-dimensional Euler equations. The results show a sufficient accuracy in the graphical comparison with the analytical solution and in the  $L$  norm value. According to the results of numerical studies, it can be seen that for relatively simple problems, the classical CFD method is still superior to PINNs, but with an increasing complexity of the problem, CFD suffers from increasing grid sizes and as a result may exhibit oscillations. In contrast, the PINN presents a smoother solution, which in turn leads to the expectation of a better result with an increasing dimensionality of the problem and an increasing number of input equations. Representing the main numerical simulation result, the PINN model was able to outperform the CFD model in terms of  $L_\infty$  norms, improving the velocity  $u$  by 170% and the pressure  $p$  by 360% in a relative comparison. Moreover, graphical results that are indistinguishable from the analytical solution for both models were obtained; in the case of the Burgers equations, plots at time 0.3, 0.7, and 1 and also contour plots for the whole computational domain were obtained, and for the Euler equations, plots at time 0.08, 0.09, and 0.1 were obtained, along with contour plots for the density over the whole computational domain. In conclusion, this study highlights the effectiveness of both Physics-Informed Neural Networks and classical CFD methods in modeling gas flows in pipelines. By accurately capturing the dynamic flow characteristics such as rarefaction waves and shock discontinuities, these numerical techniques provide valuable insights into the behavior of gas transportation systems. The comparison between PINN and CFD approaches underscores the importance of selecting the most appropriate method based on the specific requirements of the problem, ensuring reliable and efficient solutions for modeling gas flows in pipelines.

Future research could be aimed at improving both models: For the continuous PINN model, a way to find the most appropriate hyperparameters using, for instance, Bayesian hyperparameter optimization, should be found. For the discrete CFD model, the approximation of derivatives using the finite difference method should be replaced by more accurate methods with a higher order of approximation over space and a flexibility in complex domains, such as the finite volume method. In addition to this, the physical problem case study (Euler's equations) should be extended to two-dimensional forms and even further to two-dimensional Navier–Stokes equations.

**Author Contributions:** Conceptualization, S.A. and F.T.; methodology, S.A. and F.T.; software, R.F.; validation, R.F. and S.A.; formal analysis, Y.D. and F.T.; investigation, R.F.; resources, Y.D.; data curation, R.F.; writing—original draft preparation, S.A.; writing—review and editing, S.A. and R.F.; visualization, R.F.; supervision, S.A.; project administration, Y.D.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant AP14871641).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Roth, C.; Hartmann, J.; Schiewe, C.; Staudacher, S. Asymmetric Flow Phenomena Affecting the Characterization of the Control Plant of an Altitude Test Facility for Aircraft Engines. *Symmetry* **2023**, *15*, 1918. [[CrossRef](#)]
2. Zhao, X.; Cao, J.; Wang, B.; Yang, X. Experiment Study of Outburst Pulverized Coal-Gas Two-Phase Flow and Characteristic Analysis of Outburst Wave. *Geofluids* **2021**, *2021*, 8186230. [[CrossRef](#)]

3. Meleshko, S.V.; Kaptsov, E.I. Symmetry Analysis of the Two-Dimensional Stationary Gas Dynamics Equations in Lagrangian Coordinates. *Mathematics* **2024**, *12*, 879. [[CrossRef](#)]
4. Anderson, J.D.; Wendt, J. *Computational Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 1995; Volume 206.
5. Chung, T.J. *Computational Fluid Dynamics*; Cambridge University Press: Cambridge, UK, 2002.
6. Philip, N.J.; Mathew, E. Numerical Simulation of Supersonic Flow over a Flat Plate and Heat Flux Prediction. *Int. J. Adv. Engg. Tech.* **2014**, *6*, 8.
7. Hoque, S.; Kalita, P. Numerical Simulation of Supersonic Viscous Flow over a Flat Plate. Available online: [https://www.researchgate.net/publication/277475335\\_Numerical\\_Simulation\\_and\\_Analysis\\_of\\_Supersonic\\_flow\\_over\\_a\\_flat\\_plate](https://www.researchgate.net/publication/277475335_Numerical_Simulation_and_Analysis_of_Supersonic_flow_over_a_flat_plate) (accessed on 21 November 2023).
8. Daidzic, N.E. Unified Theory of Unsteady Planar Laminar Flow in the Presence of Arbitrary Pressure Gradients and Boundary Movement. *Symmetry* **2022**, *14*, 757. [[CrossRef](#)]
9. Rysbaiuly, B.; Sinitsa, A.; Capsoni, A. Analytical Inverse Analysis Methodological Approach for Thermo-Physical Parameters Estimation of Multilayered Medium Terrain with Homogenized Sampled Measurements. *Symmetry* **2022**, *14*, 2248. [[CrossRef](#)]
10. Alpar, S.; Berger, J.; Rysbaiuly, B.; Belarbi, R. Estimation of soils thermophysical characteristics in a nonlinear inverse heat transfer problem. *Int. J. Heat Mass Transf.* **2024**, *218*, 124727. [[CrossRef](#)]
11. Teixeira, F.; Sarris, C.; Zhang, Y.; Na, D.Y.; Berenger, J.P.; Su, Y.; Okoniewski, M.; Chew, W.; Backman, V.; Simpson, J. Finite-difference time-domain methods. *Nat. Rev. Methods Prim.* **2023**, *3*, 75. [[CrossRef](#)]
12. Satheesh Kumar Nair, V. High-Order Numerical Schemes for Compressible Flows. 2016. Available online: <http://resolver.tudelft.nl/uuid:e4297489-60e2-403e-a246-1b1ea4c4ea63> (accessed on 21 November 2023).
13. Droniou, J. Finite volume schemes for diffusion equations: Introduction to and review of modern methods. *Math. Models Methods Appl. Sci.* **2014**, *24*, 1575–1619. [[CrossRef](#)]
14. Fursikov, A. Stabilizability of Two-Dimensional Navier–Stokes Equations with Help of a Boundary Feedback Control. *J. Math. Fluid Mech.* **2001**, *3*, 259–301. [[CrossRef](#)]
15. Sritharan, S.; Sundar, P. Large deviations for the two-dimensional Navier–Stokes equations with multiplicative noise. *Stoch. Process. Their Appl.* **2006**, *116*, 1636–1659. [[CrossRef](#)]
16. Gottlieb, S.; Tone, F.; Wang, C.; Wang, X.; Wirosoetisno, D. Long time stability of a classical efficient scheme for two-dimensional Navier–Stokes equations. *SIAM J. Numer. Anal.* **2012**, *50*, 126–150. [[CrossRef](#)]
17. af Klinteberg, L.; Askham, T.; Kropinski, M.C. A fast integral equation method for the two-dimensional Navier–Stokes equations. *J. Comput. Phys.* **2020**, *409*, 109353. [[CrossRef](#)]
18. Ladyzhenskaya, O.A. Sixth problem of the millennium: Navier–Stokes equations, existence and smoothness. *Russ. Math. Surv.* **2003**, *58*, 251. [[CrossRef](#)]
19. Alamoudi, M.; Sattari, M.A.; Balubaid, M.; Eftekhari-Zadeh, E.; Nazemi, E.; Taylan, O.; Kalmoun, E.M. Application of Gamma Attenuation Technique and Artificial Intelligence to Detect Scale Thickness in Pipelines in Which Two-Phase Flows with Different Flow Regimes and Void Fractions Exist. *Symmetry* **2021**, *13*, 1198. [[CrossRef](#)]
20. Rabczuk, T.; Bathe, K.J. *Machine Learning in Modeling and Simulation: Methods and Applications*; Springer: Berlin/Heidelberg, Germany, 2023.
21. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
22. Markidis, S. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Front. Big Data* **2021**, *4*, 669097. [[CrossRef](#)] [[PubMed](#)]
23. Faroughi, S.A.; Soltanmohammadi, R.; Datta, P.; Mahjour, S.K.; Faroughi, S. Physics-informed neural networks with periodic activation functions for solute transport in heterogeneous porous media. *Mathematics* **2023**, *12*, 63. [[CrossRef](#)]
24. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics Informed Learning Machine. U.S. Patent 10,963,540, 30 March 2021.
25. Wang, Y.; Zhong, L. NAS-PINN: Neural architecture search-guided physics-informed neural network for solving PDEs. *J. Comput. Phys.* **2024**, *496*, 112603. [[CrossRef](#)]
26. Su, P.; He, W.B.; Fang, D.Q. Progress of machine learning studies on the nuclear charge radii. *Symmetry* **2023**, *15*, 1040. [[CrossRef](#)]
27. Krastev, P.G. A deep learning approach to extracting nuclear matter properties from neutron star observations. *Symmetry* **2023**, *15*, 1123. [[CrossRef](#)]
28. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [[CrossRef](#)]
29. Naderibeni, M.; Reinders, M.J.; Wu, L.; Tax, D.M. Learning solutions of parametric Navier–Stokes with physics-informed neural networks. *arXiv* **2024**, arXiv:2402.03153.
30. de Wolff, T.; Carrillo, H.; Martí, L.; Sanchez-Pi, N. Towards optimally weighted physics-informed neural networks in ocean modelling. *arXiv* **2021**, arXiv:2106.08747.
31. Zhang, Z.Y.; Zhang, H.; Zhang, L.S.; Guo, L.L. Enforcing continuous symmetries in physics-informed neural network for solving forward and inverse problems of partial differential equations. *J. Comput. Phys.* **2023**, *492*, 112415. [[CrossRef](#)]
32. Bonkile, M.P.; Awasthi, A.; Lakshmi, C.; Mukundan, V.; Aswin, V. A systematic literature review of Burgers’ equation with recent advances. *Pramana* **2018**, *90*, 69. [[CrossRef](#)]
33. Bec, J.; Khanin, K. Burgers turbulence. *Phys. Rep.* **2007**, *447*, 1–66. [[CrossRef](#)]

34. Inan, B.; Bahadir, A.R. Numerical solution of the one-dimensional Burgers' equation: Implicit and fully implicit exponential finite difference methods. *Pramana* **2013**, *81*, 547–556. [[CrossRef](#)]
35. Lora-Clavijo, F.; Cruz-Pérez, J.; Siddhartha Guzmán, F.; González, J. Exact solution of the 1D riemann problem in Newtonian and relativistic hydrodynamics. *Rev. Mex. Física E* **2013**, *59*, 28–50.
36. Endres, E.E.; Jenssen, H.K. Compressible 1D Euler Equations with Large Data: A Case Study. *J. Hyperbolic Differ. Equ.* **2009**, *6*, 389–406. [[CrossRef](#)]
37. Xiong, F.; Liu, L.; Liu, S.; Wang, H.; Yong, H. Gradient-weighted physics-informed neural networks for one-dimensional Euler equation. *Authorea Preprints* **2023**. [[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.