Computer Methods in Applied Mechanics and Engineering 440 (2025) 117933

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

RoePINNs: An integration of advanced CFD solvers with Physics-Informed Neural Networks and application in arterial flow modeling

J. Orera [a,*], J. Ramírez [a,b], P. García-Navarro [a], J. Murillo [a]

[a] Aragon Institute of Engineering Research, University of Zaragoza, C. de Mariano Esquillor Gómez, s/n, Zaragoza, 50018, Spain
[b] William Harvey Research Institute, Queen Mary University of London, John Vane Science Centre, Charterhouse Square, London, EC1M 6BQ, United Kingdom

ARTICLE INFO

Keywords:
Roe finite volume method
Burger's equation
Arterial blood flow
Arterial wall stiffness
Physics-Informed Neural Networks
Source term reconstruction
Solution discontinuities

ABSTRACT

The characterization of forward and inverse problems describing blood flow dynamics plays a decisive role in numerous biomedical applications. These systems can be modeled using one-dimensional (1D) approaches leading to a hyperbolic system of equations with source terms. Their numerical discretization, associated to the spatial variation of mechanical and geometrical properties, requires advanced numerical solvers that ensure both stability and an accurate description of the dynamics of the system. In this work, we present RoePINNs, a hybrid framework for the embedding of advanced Computational Fluid Dynamics (CFD) solvers into Physics-Informed Neural Networks (PINNs), and give examples of application to Burgers' equation as well as the propagation of nonlinear waves in elastic arteries, both under the presence of geometric-type source terms, for forward and inverse problems. We demonstrate that Augmented Riemann solvers can be incorporated into the PINN framework with straightforward adjustments to the hyperparameters, providing a promising alternative to automatic differentiation (AD), especially in cases where the solution exhibits strong nonlinearities and physical constraints are required. Benefits of the proposed RoePINN compared with the *vanilla* PINN based in AD are twofold: on the one hand, this hybrid approach employs numerical differentiation by means of support points in the surroundings of the collocation points, hence the robustness, generalization capacity and tunability of the PINNs are, in most cases, largely enhanced. On the other hand, the RoePINN incorporates the numerical solver, hence it is also capable of capturing sharp discontinuities with an order-of-magnitude improvement in accuracy compared with the vanilla version.

1. Introduction

In the context of fluid dynamics, partial differential equations (PDEs) can mathematically represent a broad array of physical situations, including blood flow in vessels [1]. In many cases, the governing PDEs of these phenomena are hyperbolic [2].

Numerical methods make use of the discretization of space and time to find physical solutions for these equations, and have found enormous success in the field of Computational Fluid Dynamics (CFD) over the past fifty years [3,4], specially when predicting flow variables from parameters and supplied with complete knowledge of initial and boundary conditions [5]. Following this appraisal,

* Corresponding author.
E-mail address: jorera@unizar.es (J. Orera).

https://doi.org/10.1016/j.cma.2025.117933
Received 23 January 2025; Received in revised form 7 March 2025; Accepted 12 March 2025
Available online 25 March 2025
0045-7825/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

commonly known as *forward problem* approach, numerical methods exhibit high accuracy, but at the expense of high computational demands, especially in terms of time, hampering the development of real-time solvers for fluid problems [5]. Yet, even to this day, traditional numerical CFD methods struggle enormously with inverse problems [6], where usually part of the data in the initial and boundary conditions is missing, despite the numerous efforts made in this direction with methods such as the adjoints [7].

Alternatively, the rapid advance of deep learning methodologies over the past ten years has opened new possibilities for the development of solvers with enhanced capabilities for extracting hidden information out of measurable variables and incorporating noisy measurements [8]. In particular, Physics Informed Neural Networks (PINNs) [9] have showcased how the PDEs describing the fluid problems can be solved by turning the iterative method that numerical methods employ into a constrained optimization problem where the specifics of the available data can be seamlessly integrated with the optimization framework, without having to resort to a discretization of the domain, commonly known as *meshing* [9]. While having also shown very promising results in forward problems, it is still not clear if PINNs will be able to outperform traditional numerical methods in terms of accuracy of the solution and computational overhead [10], especially for more complex real-world problems where costs associated with the training of the model might significantly surge [11].

This work concentrates on a particular formulation of the discretized loss function of the PINN, with particular emphasis on 1D hyperbolic nonlinear problems illustrated by means of the 1D Burgers' equation with discontinuous geometrical source terms [12] and blood flow inside vessels. The proposed framework allows for the approximation of solutions of the Burgers' equation with sharp discontinuities without having to resort to smooth approximations of the source terms, which turn unavoidable if computation of gradients is performed via traditional automatic differentiation [13].

Physics-Informed Neural Networks (PINNs) [9] originally appeared as optimization-based mesh-less solvers that, while not being necessarily more accurate than traditional CFD solvers, are able to circumvent some of their drawbacks. These are (1) the complexity present in the discretization of derivatives in numerical methods, (2) the inability to compute solutions when data on the initial and boundary conditions are partially or completely missing (this being the case in most practical real-world applications), (3) the inability to incorporate multi-fidelity data in the algorithm and solve ill-posed problems without resorting to unsustainable computational costs, and (4) the impossibility of dealing with parametrized equations of high dimensionality [14]. Moreover, CFD codes be, in some cases, extremely long and unmanageable from one generation to the next [14].

Classical numerical methods, such as Finite Volume Method (FVM), require the discretization of the domain of interest into a mesh where the solutions of the PDEs are approximated with an accuracy that is usually proportional to the granularity of the grid [15]. PINNs, however, do not require a pre-defined mesh or grid. Instead, they use scattered data points across the domain and boundaries, known as collocation points, making them especially suitable for complex geometries where meshing is cumbersome.

When centering on real-world applications, constants related to the interaction of the system with the boundaries, describing dependencies between internal variables, or even functions depending on time and space might be partially or completely unknown, either because it is not possible to measure them or the tests that need to be conducted are too costly. This is where PINNs show a great inference potential and provide simple shortcuts in favor of the predictability of all the variables in the system.

The field of Physics-Informed Machine Learning (PIML) offers the possibility of partial system identification (model calibration) or complete system identification (model discovery) when a significant amount of data is available. Extensions of PINNs such as Physically Guided Neural Networks with Internal Variables (PGNNIVs) or Efficient Unsupervised Constitutive Law Identification and Discovery (EUCLID), have shown the possibility to discover complete constitutive models without stress–strain data in solid mechanics, while predicting macroscopic properties for arbitrary boundary conditions [11,16].

PINNs usually exhibit poor performance in problems involving strong discontinuities [17]. When the viscous term is absent or the viscosity coefficient is significantly low, training a PINN becomes extremely challenging, resulting in its inability to accurately represent shock waves [17]. This issue usually concerns the use of PINNs as forward PDE solvers with no available data of the solution in the domain. However, if some domain data is available, the prediction improves drastically.

In particular, [18] deals with the approximation of solutions of the Euler equations considering various configurations of contact shock waves and proposed several workarounds to mitigate the effect of such discontinuities, such as the clustering of collocation points around the shock wave. This, however, requires an approximate knowledge of the exact solution that in most practical application cannot be acquired. It is concluded that, even though numerical methods outperform PINNs in capturing discontinuities, the potential for solving inverse problems is considerably higher in PINNs.

Furthermore, [19] provides a comprehensive description of the possible sources of error of PINNs in the learning of hyperbolic equations, including oblique shocks, expansions and bow waves in one and two-dimensional domains. The entropy constraint is enforced by means of a formulation of the problem based on a volume scheme, giving rise to a particular type of PINNs coined *cvPINNs*.

Since the advent of PINNs, extensive research has been lead towards the construction of algorithms extending the vanilla PINN framework, initially formulated in [8], with the aim of solving some of its deficiencies in terms of accuracy, training speed, etc. The term *vanilla* refers to the baseline algorithm that uses automatic differentiation [13] for the purpose of computing the derivatives involved in the loss function (built in most deep learning libraries), comprises a Multi Layer Perceptron (MLP) architecture (usually 5–10 layers of 10–200 neurons each) with common activation functions (sigmoid, hyperbolic tangent, ReLU).

In the context of PINNs, automatic differentiation (AD) has played an essential role in the definition of PDE losses, provided that the functions involved in the differential equation are continuous and differentiable. CAN-PINNs [20] propose a hybrid approach based on a combination of AD and finite differences for the formulation of the PDE loss combining AD. [20] provides interesting insights into how the number of collocation points affects the inferred solution. In particular, it is argued that AD is able to perform at least as well as the proposed CAN-PINN if the number of collocation points is large. However, if the number of collocation points

is small, then AD leads to overfitting and prediction of nonphysical solutions, while the CAN-PINN shows more robustness. They conclude that a mixed approach yields optimal results.

Although some authors have put together compilations on good practices for training PINNs, the lack of a unifying set of principles that transcends the specifics of a particular set or type of PDEs has given rise to a diverse but fragmented landscape where almost every distinct particular PDE setting is associated with a certain type of singularities in the solution and leads to a new different implementation of the PINN framework, with usually new hyperparameters that ought be tuned, and a new optimization algorithm that most likely depends on the intricacies of the particular research field. Examples of these are countless [19–38].

Besides, hyperparameters related to the convergence and optimization of the weights and biases of the network can drastically change from one framework to another, which in turn hampers the subsequent transferability, reproducibility and applicability of these models to real-world or academic problems.

Alternatively, in this work we put forward a different approach for the construction of the PINN based on the exploitation of the properties of numerical methods, which offer accuracy, usually guarantee stability in and out of the domain, and normally have more flexibility than PINNs, specially because the latter are limited to the domain that they were trained on.

We advocate that, by plugging in the numerical scheme into the PINN framework, it is possible to circumvent complex training optimization schemes whose applicability is limited to a set of PDEs, while introducing prior knowledge about the dynamics of the systems gathered over the last 50 years.

The structure of the work is as follows:

First, in Section 2 we describe the numerical treatment of scalar hyperbolic equations, particularizing to the application of the Augmented Roe Finite Volume Method [12] (RoeFV) to the scalar Burgers' equation with geometrical source terms. The RoeFV is an example of advanced numerical method able to tackle strong discontinuities and instabilities in hyperbolic equations. We consider source terms that contain abrupt discontinuities.

Then, we describe the *vanilla* PINN, a MultiLayer Perceptron (MLP) architecture that employs simple activation functions such as *ReLU*, *sigmoid*, or *tanh*. Also, in the *vanilla* version, the loss function is constructed using automatic differentiation (AD) and the weighting coefficients of the different terms of the loss function are chosen manually via trial-and-error before the training process.

Later, the details of the implementation of RoePINNs. The RoeFV is introduced in the loss function of the PINN are provided. The integration of this numerical scheme within the optimization framework of PINNs will allow for the better prediction of discontinuous solutions as compared with classical *vanilla* PINNs. Results of the comparison of both frameworks are shown by means of a hyperparameter sensitivity analysis across several test cases that involve forward problems and inverse problems.

In Section 4, we scale up in complexity by considering a more realistic PDE system of hyperbolic equations describing blood flow in human vessels. We build upon the mathematical principles in Section 2 to further extend the formulation of the vanilla and RoePINN in both forward and inverse problems, involving the reconstruction of non-measurable and internal variables such as the vessel's luminal area and stiffness, and its space–time dependencies. Importantly, the wall stiffness plays here the role of the geometrical source term of Burgers' equation. Results comparing the implementations are provided, supported by a sensitivity analysis that explores various hyperparameter combinations for forward and inverse problems, particularly in the context of vessel stiffness estimation.

2. Scalar conservative laws with geometrical source terms

The presence of source terms accounting for variations in material properties and their role in the propagation of haemodynamic waves can be first analyzed using a simplified dynamical model that nevertheless comprises the essential features of hyperbolic PDE systems leading to nonlinear unsteady solutions. The one-dimensional (1D) inviscid Burgers' scalar equation with geometrical source terms [39] can be written:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = s\,, \tag{1}$$

where $f(u)$ is a convex non linear flux and $s = s(u, x)$ is a source term of the form:

$$f(u) = \frac{1}{2}u^2, \quad s = -u\frac{\partial z}{\partial x}. \tag{2}$$

where z depends on the position x, $z = z(x)$, can be discontinuous and does not depend on time, $\partial_t z = 0$. From $f(u)$ it is possible to find an advection, or transport velocity λ:

$$\lambda = \frac{df}{du} = u \tag{3}$$

The domain of interest is discretized into a number of computational cells, N_c that have equal length size Δx, which is constant over time. The interval associated with the ith cell is expressed as $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, where x_i denotes the center of cell i. The initial conditions are specified by piecewise constant values u_i at each cell i at the starting time t.

2.1. RoeFV method in the scalar case

The RoeFV method used here [12,40] is derived as a special case of the reconstruction, evolution and averaging steps method proposed originally by Godunov [41], and includes the effect of the discontinuity in the source term, providing updated cell averages values for the conserved variables u_i^{n+1}. This method provides an approximate solution or an evolved state to the following Riemann Problem (RP) or initial value problem:

$$\partial_t u + \partial_x f - s = 0\,, \qquad u(x, t=0) = \begin{cases} u_i & if \quad x < 0 \\ u_{i+1} & if \quad x > 0 \end{cases} \tag{4}$$

that starts from piecewise constant data reconstructions at the initial time $t = 0$. The RP represents a fundamental initial value problem involving hyperbolic PDEs with special relevance in the study of waves propagation [42].

The RoeFV method can be expressed in fluctuation form as

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}[(\delta m)^-_{i+\frac{1}{2}} + (\delta m)^+_{i-\frac{1}{2}}]\,, \tag{5}$$

with

$$(\delta m)^-_{i+\frac{1}{2}} = (\widetilde{\lambda}^- \theta \delta u)_{i+\frac{1}{2}}\,, \quad (\delta m)^+_{i-\frac{1}{2}} = (\widetilde{\lambda}^+ \theta \delta u)_{i-\frac{1}{2}}\,. \tag{6}$$

where u_i^{n+1} is the solution at cell i of the domain at time $n+1$, u_i^n is the solution at cell i of the domain at time n, Δt and Δx are the time and space discretization respectively and δm is the fluctuation term computed with the intercell fluxes. Eq. (6) encapsulates the dynamic evolution of the update scheme. Further details can be found in [12].

The time step is defined considering that the celerity of the propagation of the information between cells in each RP is provided by $\lambda_{i+\frac{1}{2}}$. Considering that the time step in this explicit method must be constant in each computational cell, the time step is defined in general as

$$\Delta t = CFL \min\left\{\Delta t_{i+\frac{1}{2}}\right\}, \quad \Delta t_{i+\frac{1}{2}} = \frac{\Delta x}{|\widetilde{\lambda}_{i+\frac{1}{2}}|} \tag{7}$$

where CFL is the Courant–Friedrichs–Lewy or CFL condition. It is possible to relax the CFL condition over the time step when using explicit schemes, leading to values of $CFL < 1$, even in the presence of large variations of the source terms, provided that a suitable discretization of the source term is enforced [1].

2.2. PINN formulation with automatic differentiation: vanilla PINN

The vanilla PINN employs a fixed distribution of collocation points in the space–time plane. By minimizing the PDE residual at each collocation point, the variables and its derivatives are driven towards the fulfillment of the PDE. In practice, collocation points are sampled uniformly across the computational domain [9,43]. Reasonable values for the number of collocation points (n_{coll}) range between 10^4 and 10^5 for domains where $x, t \in [0, 1]$ [44]. Collocation points will be denoted with tuples (x_c, t_c), where x_c and t_c are the spatial and temporal coordinates of the collocation points, respectively.

2.2.1. Forward vanilla PINN

For the Burgers' equation, the vanilla PDE loss $\mathcal{L}^{v,F}_{PDE,B}$ associated with the minimization of the PDE residual is

$$\mathcal{L}^{v,F}_{PDE} = \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(\frac{\partial u}{\partial t}\bigg|_{(x_c^i, t_c^i)} + u(x_c^i, t_c^i) \frac{\partial u}{\partial x}\bigg|_{(x_c^i, t_c^i)} + u(x_c^i, t_c^i) \frac{\partial z}{\partial x}\bigg|_{(x_c^i, t_c^i)} \right)^2 \tag{8}$$

with $u = \mathcal{PINN}(x, t; \boldsymbol{w})$ the predicted solution at some training iteration, $\boldsymbol{w}$ the set of weights and biases of the neuronal network, n_{coll} the number of collocation points and $z(x)$ a known function. The initial condition loss $\mathcal{L}_o$ related to the fitting of the initial condition $u(x, t=0)$ with datapoints $u_o(x_o, t_o)$ becomes

$$\mathcal{L}_o = \frac{1}{n_o} \sum_{i=1}^{n_o} \left(u(x_o^i, t_o^i) - u_o^i(x_o^i, t_o^i) \right)^2, \tag{9}$$

with n_o the number of initial condition datapoints at $t = 0$. Equivalently, the boundary condition loss $\mathcal{L}_{bc}$ related to the fitting of the boundary condition $u(x = \partial\Omega, t > 0)$ with datapoints $u_{bc}(x_{bc}, t_{bc})$ becomes

$$\mathcal{L}_{bc} = \frac{1}{n_{bc}} \sum_{i=1}^{n_{bc}} \left(u(x_{bc}^i, t_{bc}^i) - u_{bc}^i(x_{bc}^i, t_{bc}^i) \right)^2 \tag{10}$$

with n_{bc} the number of points at the boundary conditions. It is possible to include other available data inside the domain as training data, in the form of a loss function

$$\mathcal{L}_{data} = \frac{1}{n_d} \sum_{i=1}^{n_d} \left(u(x_d^i, t_d^i) - u_d^i(x_d^i, t_d^i) \right)^2, \tag{11}$$

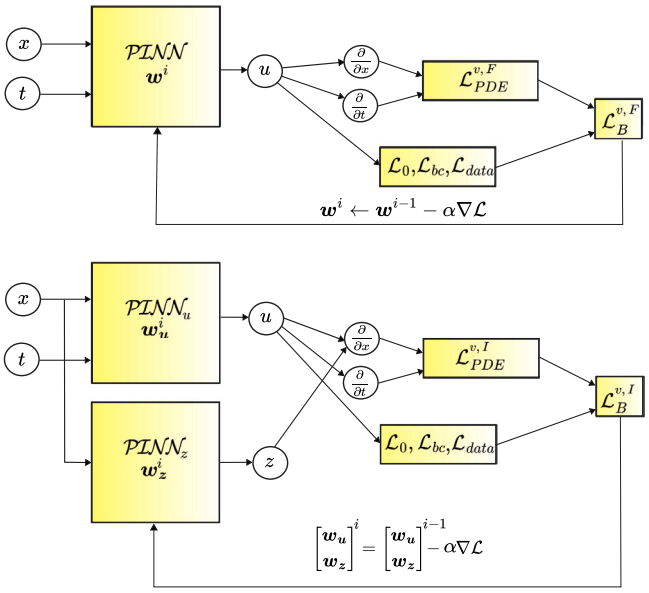


Fig. 1. Scheme of the forward vanilla PINN solver for the scalar Burgers' equation (top). Scheme of the ISVanPINN solver for the scalar Burgers' equation (bottom).

provided in the form of solution snapshots, scattered data or time series, defined as $D_d : \{u_d, x_d, t_d\}_{i=1}^{n_d}$ where n_d is the number of data points.

The global loss function of the vanilla PINN $\mathcal{L}_B^{v,F}$ is constructed as a weighted sum of the individual losses, and has the following form for the forward setting:

$$\mathcal{L}_B^{v,F} = l_1 \mathcal{L}_{PDE}^{v,F} + l_2 \mathcal{L}_o + l_3 \mathcal{L}_{bc} \tag{12}$$

where constants l_i are weighting coefficients that have to be tuned so as to balance the magnitudes of the PDE loss $\mathcal{L}_{PDE}$ (or *collocation* loss) and the data losses $\mathcal{L}_o$, $\mathcal{L}_{bc}$. They are also known as Lagrange multipliers [45]. The configuration of a training step of the vanilla PINN for the forward problem is plotted in Fig. 1 (left). Note that variables (x, t) fed into the PINN can represent training or test data, as well as collocation points (x_c, t_c). All derivatives are evaluated at the collocation points (x_c, t_c).

2.2.2. Inverse vanilla PINN with separated models (ISVanPINN)

For the inverse problem, the loss associated with the minimization of the PDE residual, $\mathcal{L}_{PDE}^{v,I}$, has the same form as the forward loss $\mathcal{L}_{PDE}^{v,F}$, except that in the inverse case, we assume $z = z(x)$ is the output of a second model that is only fed with x, that is, $z = \mathcal{PINN}_z(x; \boldsymbol{w_z})$, and $u = \mathcal{PINN}_u(x, t; \boldsymbol{w_u})$.

Therefore, the global loss function of the vanilla PINN $\mathcal{L}_B^{v,I}$ has the following form:

$$\mathcal{L}_B^{v,I} = l_1 \mathcal{L}_{PDE}^{v,I} + l_2 \mathcal{L}_o + l_3 \mathcal{L}_{bc} + l_4 \mathcal{L}_{data} \tag{13}$$

with $l_4 > 0$. The configuration of a training step of the vanilla PINN for the inverse problem is plotted in Fig. 1 (right).

2.3. PINN formulation with RoeFV method: RoePINN

Instead of computing the derivatives of the loss function using automatic differentiation [13], we bring forward the numerical scheme formulation replacing AD. Notice that derivatives with respect to the weights are still being computed using traditional backpropagation [46].

2.3.1. Forward RoePINN

Using the update scheme comprised in Eq. (5), the PDE loss function of the RoePINN has the following form for the forward problem:

$$\mathcal{L}_{PDE}^{Roe,F} = \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(u_{x_c^i}^{t_c^i + \Delta t^i} - u_{x_c^i}^{t_c^i} + \frac{\Delta t^i}{\Delta x} \left[(\delta m)^-_{x_c^i + \frac{1}{2}\Delta x} + (\delta m)^+_{x_c^i - \frac{1}{2}\Delta x} \right] \right)^2 \tag{14}$$

where

$$(\delta m)^-_{x_c + \frac{1}{2}\Delta x} = (\widetilde{\lambda}^- \theta \delta u)_{x_c + \frac{1}{2}\Delta x}, \quad (\delta m)^+_{x_c - \frac{1}{2}\Delta x} = (\widetilde{\lambda}^+ \theta \delta u)_{x_c - \frac{1}{2}\Delta x}. \tag{15}$$

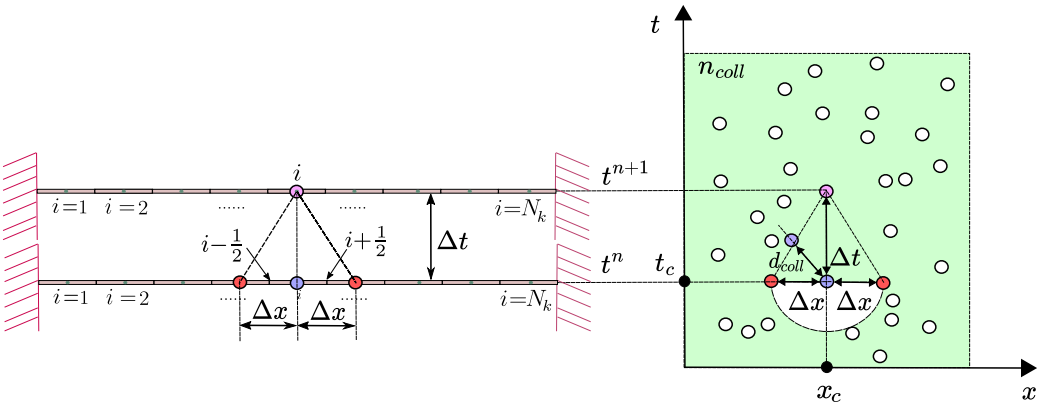


Fig. 2. An illustrative analogy between the discretized vessel (left), at two consecutive time steps, and the RoePINN's collocation points and local support points inside the PDE computational domain (right).

and where $u = \mathcal{PINN}(x,t;\boldsymbol{w})$. The equivalent of the cell size in the actual RoeFV, dubbed *local support* distance, Δx, and the time step parameter, CFL, are hyperparameters concerning the discretization of the computational domain and can be tuned accordingly to improve learning and prediction. Furthermore, for each collocation point $i \equiv (x_c, t_c)$, Δt^i is chosen adaptively as a function of Δx, hence

$$\Delta t^i = CFL\frac{\Delta x}{|\widetilde{\lambda}_i|}, \tag{16}$$

as it is done in the actual RoeFV numerical scheme. Further details on the adaptive time step and choice of discretization in space are provided in Appendix 8.1. Datapoints $\{(x_c,t_c),(x_c+\Delta x,t_c),(x_c-\Delta x,t_c),(x_c,t_c+\Delta t^i)\}$ are called *local support points*, as they represent auxiliary locations in the computational domain surrounding the collocation points that are used to approximate the space and time derivatives of the PDE.

Fig. 2 shows a representation of the construction of the RoePINN PDE loss, and the adaptive time step Δt used during the training of the model. In the RoeFV method, the spatial domain is divided using a mesh with a constant value of Δx. The updating step of each cell in Eq. (15) requires the information of the initial conditions in two neighboring cells and in the updated cell itself. Also, all cells are updated globally using the same time step Δt.

When this method is applied to construct the RoePINN algorithm, the updating process is limited to the randomly predefined collocation points. As a result, the distance between any two pairs of collocation points does not preserve $\widetilde{\lambda}$, hence Δt is no longer global. For that reason, the cloud of collocation points (x_c,t_c) may be sparser than the actual number of nodes in the mesh of the RoeFV.

Finally, the global loss function of the RoePINN has the following form in the forward setting:

$$\mathcal{L}_B^{Roe,F} = l_1\mathcal{L}_{PDE}^{Roe,F} + l_2\mathcal{L}_o + l_3\mathcal{L}_{bc}. \tag{17}$$

2.3.2. Inverse RoePINN with separated models (ISRoePINN)

For the inverse problem, the PDE loss function of the RoePINN, $\mathcal{L}_{PDE}^{Roe,I}$, is the same as the forward loss $\mathcal{L}_{PDE}^{Roe,F}$, but in the inverse case $u = \mathcal{PINN}_u(x,t;\boldsymbol{w_u})$ and $z = \mathcal{PINN}_z(x;\boldsymbol{w_z})$.

The loss function of the RoePINN for the Burgers' equation has the following form for the inverse problem:

$$\mathcal{L}_B^{Roe,I} = l_1\mathcal{L}_{PDE}^{Roe,I} + l_2\mathcal{L}_o + l_3\mathcal{L}_{bc} + l_4\mathcal{L}_{data} \tag{18}$$

with

$$\mathcal{L}_{data} = \frac{1}{n_d}\sum_{i=1}^{n_d}\left(u(x_d^i,t_d^i) - u_d^i(x_d^i,t_d^i)\right)^2 \tag{19}$$

Fig. 3 represents the configuration training step for the forward RoePINN (left) and for the ISRoePINN (right). In this figure, box denoted $\mathcal{L}_{PDE}^{Roe}$ encapsulates the intermediate variables computed by the numerical scheme, as well as specific constrains in the selection of the adaptive time step to ensure the stability of the solution. Estimates of the solution in future time points $(u(x,t+\Delta t))$ predicted by the RoePINN during the training are computed using an adaptive time step calculated as in Eq. (16).

3. Numerical results for the scalar Burgers' equation with geometrical source terms

Predictions of the RoeFV, described in Eq. (1), along with the RoePINN and the vanilla PINN, are compared against a reference solution that corresponds to either the analytical solution or the RoeFV solution with an finer discretization. L_2, L_1 and L_{inf} error definitions can be found in Appendix C 8.3.

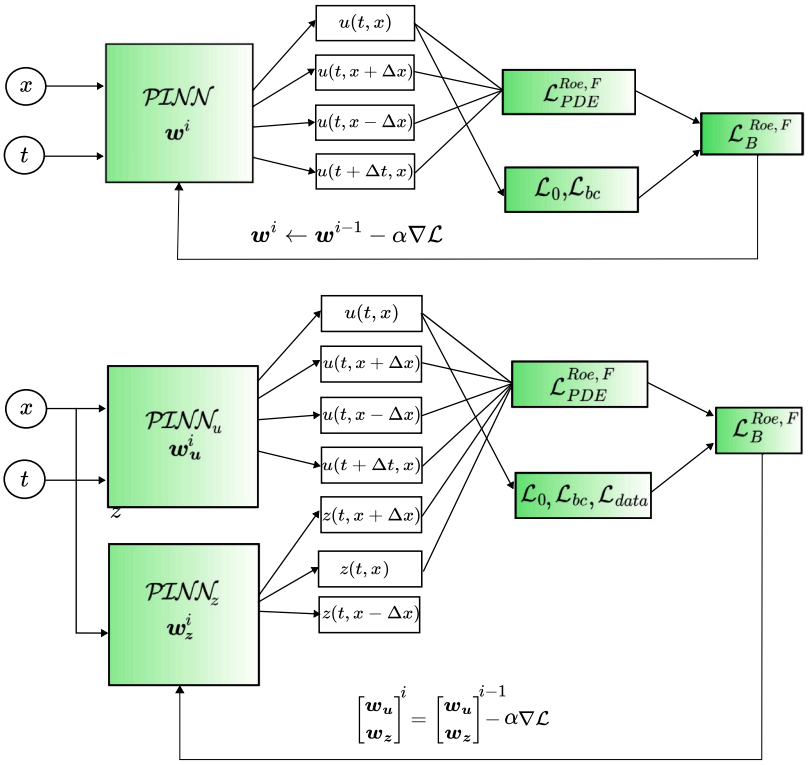


Fig. 3. Scheme of the forward RoePINN solver for the scalar Burgers' equation (top). Scheme of the ISRoePINN with separated model for the source term (bottom).

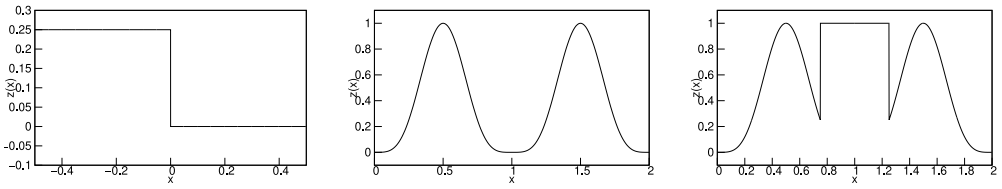


Fig. 4. Source term employed in Test case F1 (left), F2 (middle) and F3 (right).

The test cases analyzed in this section will be named F1, F2, F3 and I1, where -F- stands for *Forward* problem and -I- for *Inverse* problem. Test case F1 involves a discontinuous source term of the form (Fig. 4, left):

$$z(x) = \begin{cases} z_L & if \quad x < 0 \\ z_R & if \quad x > 0 \end{cases} . \tag{20}$$

Although discontinuous functions with perfect discontinuities rarely appear in real-world applications, evaluating the method's performance in such scenarios provides a robust test of its capabilities.

In Test case F2, a smooth differentiable function is used, with $z(x) = sin^4(4\pi x)$ (Fig. 4, middle). In Test case F3, we introduce a piece-wise source term that combines both discontinuous and smooth functions:

$$z(x) = \begin{cases} \sin^4(4\pi x) & \text{if} \quad 0 < x < 0.75 \\ 1 & \text{if} \quad x > 1.25 \\ \sin^4(4\pi x) & \text{if} \quad 0.75 < x < 1.25 \end{cases} , \tag{21}$$

plotted in Fig. 4 (right).

In the following Test cases F1, F2, F3 and I1, a thorough analysis of the RoePINN and the vanilla PINN predicted solution against the RoeFV solution and the reference solution is provided. The motivation for the subsequent extensive analysis is to provide a clear description of all the parameters and hyperparameters involved, as well as to introduce several conceptual details that are required for the understanding of the application to vessels in Section 4. Importantly, the transition from a strong discontinuous source term (F1) into a smooth one (F2), and a combination of both (F3 and I1), helps showcase the main difference between the features and performance of the RoePINN compared with the vanilla PINN.

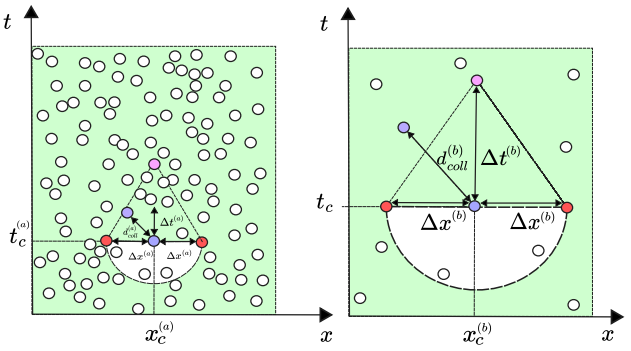


Fig. 5. Dense cloud of collocation points sampled across the computational domain (left). Sparse cloud of collocation points sampled across the computational domain (right).

Source terms and initial conditions in Test case I1 are equal to Test case F3, but now feeding not only initial and boundary data to the algorithm, but also domain data $\mathcal{D}_d : \{u_d, x_d, t_d\}_{d=1}^{n_d}$. However, we proceed on the assumption that $z(x)$ is unknown and must be reconstructed, hence we are dealing with an inverse problem where some domain data of the solution is available.

The geometrical source term of interest can be directly integrated in the RoePINN. However, in the case of the vanilla PINN, most AD algorithms are not able to explicitly tackle discontinuities and compute derivatives at these singularities [47], thus the need for smooth approximations that can lead to either undesirable exploding gradients, or simply to wrong solutions. In order to overcome this limitation, the discontinuous function $z(x)$ is approximated in Test case F1 when using the vanilla PINN, by a combination of sigmoid functions as follows:

$$z_s(a, x) = z_L(1 - \sigma(a, x)) + z_R\sigma(a, x) \tag{22}$$

where

$$\sigma(a, x) = \frac{1}{1 + e^{-ax}} \tag{23}$$

with a the slope of z_s at $x = 0$.

When comparing the RoePINN with the vanilla PINN, it is clear that the RoePINN introduces two extra hyperparameters, the *local support* distance Δx and the time stability condition parameter CFL. Considering that the RoePINN requires the use of these two hyperparameters that define the *local support* points, it is convenient to explore the performance of the RoePINN in terms of a descriptive ratio, namely $r = \frac{\Delta x}{d_{coll}}$, with

$$d_{coll} = \frac{1}{n_{coll}} \sum_{i=1}^{N} \min_{j \neq i} \left(d_{ij}\right) \tag{24}$$

a characteristic length defined as the mean of the minimum distances from the ith collocation point to any other collocation point, and

$$d_{ij} = \sqrt{(x_c^{(i)} - x_c^{(j)})^2 + (t_c^{(i)} - t_c^{(j)})^2} \tag{25}$$

the distance between any two collocation points $(x_c^{(i)}, t_c^{(i)})$ and $(x_c^{(j)}, t_c^{(j)})$ within set $\{(x_c^{(i)}, t_c^{(i)})\}$, $i = 1, 2, \ldots, n_{coll}$.

The ratio r compares the spatial discretization Δx to the mean distance d_{coll}, which represents how densely the collocation points are distributed in space and time. If $r \gg 1$, this implies that the collocation points are very closely spaced compared to the grid resolution Δx. In this case, the model could potentially overfit the collocation points due to their high density. If $r \ll 1$, the collocation points are sparsely distributed compared to the grid resolution, meaning that the RoePINN may struggle to capture fine details of the solution because there are too few collocation points in each grid cell.

Ideally, $r \approx 1$ indicates a balance between the density of collocation points and the spatial resolution, leading to efficient learning by the model without overfitting or underfitting. Fig. 5 represents a comparison between a dense (left) and sparse (right) cloud of collocation points, with $r \approx 1$ in both cases, where $n_{coll}^{(a)} \gg n_{coll}^{(b)}$ and $d_{coll}^{(a)} \ll d_{coll}^{(b)}$, hence to attain $r \approx 1$ we have to set $\Delta x^{(a)} \gg \Delta x^{(b)}$. As is logical, increasing the value of Δx results in a decrease in accuracy, but improves robustness and generalization. This trade-off allows for a better performance of the RoePINN across a broader range of conditions.

3.1. Forward test case F1. Discontinuous source term

In this section we study the solution of the Burgers' equation with a discontinuous initial condition for $u(x, t = 0)$, with $u_L = 1.0$ for $x < 0$ and $u_R = 0.5$ and with a discontinuous geometrical source term, with $z_L = 0.25$ and $z_R = 0.0$, setting $a = 500.0$ in the case of the vanilla PINN, to ensure that the approximation of the source term $z_s(x)$ is accurate.

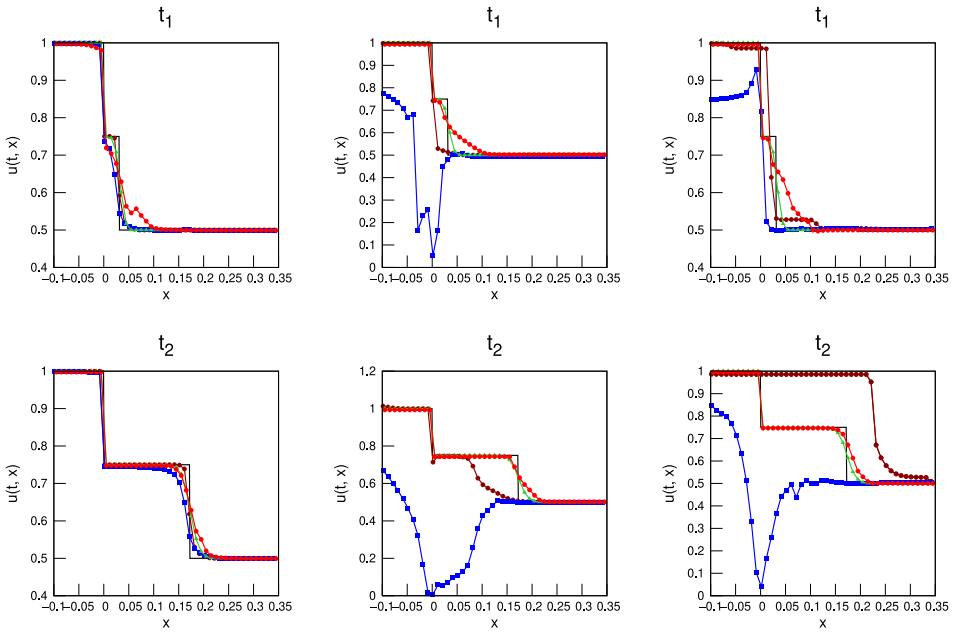


Fig. 6. Test case F1: Solution of the *vanilla* PINN (-■-), RoePINN with $\Delta x = 10^{-3}$ (-•-), RoePINN with $\Delta x = 10^{-2}$ (-•-), RoeFV with $\Delta x = 10^{-2}$ (-▲-), and the reference solution (—). Time snapshots are ordered from upper to lower ($t_1 = 5 \times 10^{-2} < t_2 = 0.275$). The first, second, and third columns correspond to different numbers of collocation points $n_{coll} = \{10^5, 10^4, 10^3\}$, respectively. The plotted discretization is $10 \times \Delta x$.

The computational domain is defined $[x_o, x_f] \times [t_o, t_f]$, with $t_o = 0$ s, $t_f = 0.5$ s, $x_o = -0.5$cm and $x_f = 0.5$ cm, and $CFL = 1.0$. A summary of all the different parameters employed in this and subsequent cases can be found in Appendix D 8.4. It is worth noting that the imposition of an initial discontinuous function for u does not require the use of approximate solutions, as the model simply fits the initial condition datapoints.

Fig. 6 shows the graphical representation of the solution for Test case F1 at $t_1 = 5 \times 10^{-2} < t_2 = 0.275$ (from upper to lower) for different values of $n_{coll} = \{10^5, 10^4, 10^3\}$ (from left to right) and Δx.

The performance of the RoeFV with $\Delta x = 10^{-3}$ (-▲-), RoeFV with $\Delta x = 10^{-2}$ (-▲-), the RoePINN with $\Delta x = 10^{-3}$ (-•-), RoePINN with $\Delta x = 10^{-2}$ (-•-), and the vanilla PINN (-■-), with the reference cell-averaged solution (—) are compared. The RoeFV provides the most accurate solutions and accuracy improves with mesh refinement. This aligns with the general observation that PINNs are typically less accurate in forward problems compared to traditional numerical methods [48]. PINNs rely on an optimization process where the PDE residual is minimized but not strictly enforced to be zero. In Fig. 6, the solution of the *vanilla* PINN does not change with hyperparameter Δx, since Δx is only defined for the RoePINN. The solution discontinuity is accurately reconstructed by the RoePINN, even when drastically reducing n_{coll} from 10^5 to 10^4, while the vanilla PINN completely collapses when decreasing n_{coll} to 10^4. This tendency remains as the Δx decreases, specially for $\Delta x = 10^{-2}$, where the RoePINN clearly outperforms the vanilla PINN despite the small value of n_{coll}. As opposed to the RoeFV, it is evident that increasing the hyperparameter Δx improves the performance of the RoePINN, particularly when fewer collocation points are used. In contrast, the *vanilla* PINN lacks this hyperparameter and cannot be further fine-tuned.

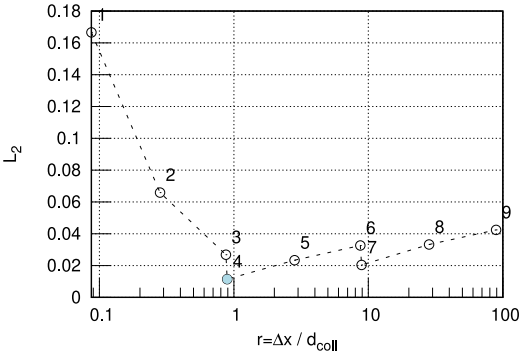
The reason why the RoePINN remains robust against the decrease in the value of n_{coll} is two-fold. First, as n_{coll} is reduced, collocation points become more surrounded by empty areas in the computational domain, with no nearby collocation points. To compensate for this empty areas, we fine-tune the RoePINN hyperparameter Δx by increasing its value. Secondly, by accounting for predictions in regions of the domain that lie further away from the collocation points, we not only improve the PINNs approximation of the numerical PDE derivatives at intermediate iterations, but also avoid under-fitting of the residuals in the empty areas surrounding the collocation points.

Table 1 shows the L_2 error for the different algorithms. The error in the predicted solution of the RoeFV is measured by comparing the numerical result in each computational cell $\{x_e - \Delta x, x_e + \Delta x\}$, with the average of the reference solution $u_e(x_e, t_e)$ in the same cell at different times t_e. The latter, called reference solution, is given in the form of *snapshots* that are equally distributed across the computational domain in time and space. Data snapshot e can be expressed as $u_e(x = \{x_o, x_o + \Delta x, \dots, x_f\}, t_e)$ with $t_e = e\Delta t_e$, $e = 1, \dots, n_e$, Δt_e the sampling period over the total simulation time t_f and $n_e = 100$. Values of t_e are reported for each of the subsequent cases in Appendix D 8.4. As observed in the aforementioned Table 1, the error in the RoeFV decreases with the reduction of the cell size as expected.

The errors of the RoePINN and vanilla PINN have been also compared with the reference solution for three levels of mesh refinement, that is, varying hyperparameter Δx. The RoePINN and vanilla PINN predictions are less accurate than those of the

Table 1
Test case F1: Comparison of L_2 errors for the different algorithms and values of Δx and n_{coll}.

		L_2 Error	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	RoePINN ($\Delta x = 10^{-3}$)	Vanilla PINN
10^5	–	0.011426	0.024421
10^4	–	0.065886	0.327288
10^3	–	0.166603	0.199043
–	0.005831	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	RoePINN ($\Delta x = 10^{-2}$)	Vanilla PINN
10^5	–	0.007119	0.022155
10^4	–	0.011333	0.328712
10^3	–	0.009039	0.200007
–	0.01683	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	RoePINN ($\Delta x = 10^{-1}$)	Vanilla PINN
10^5	–	0.042468	0.026006
10^4	–	0.033287	0.315904
10^3	–	0.032591	0.232633
–	0.033532	–	–



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-3}	10^3	0.0116	0.09
2	10^{-3}	10^4	0.0036	0.28
4	$\mathbf{10^{-3}}$	10^3	**0.0011**	**0.89**
3	10^{-2}	10^3	0.0116	0.86
5	10^{-2}	10^4	0.0036	2.80
7	10^{-2}	10^5	0.0011	8.92
6	10^{-1}	10^3	0.0116	8.59
8	10^{-1}	10^4	0.0036	27.96
9	10^{-1}	10^5	0.0011	89.17

Fig. 7. Test case F1: (*Left*) Evolution of the RoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r.

RoeFV for any combination of n_{coll} and Δx. For the RoePINN, when $\Delta x = 10^{-3}$, the solution improves with the increase of n_{coll}, but for $\Delta x = 10^{-2}$ or $\Delta x = 10^{-1}$ there is not a general trend and the error is less sensitive to n_{coll}.

Even though the vanilla PINN is not dependent on Δx, numerical errors in Table 1 slightly change with Δx, because the reference solution used to compare is discretized with the selected Δx.

We observe that the RoePINN method clearly outperforms the vanilla PINN because errors in the vanilla PINN are approximately one order of magnitude larger than those of the RoePINN. When decreasing n_{coll}, the RoePINN remains low while the vanilla PINN error increases for any value of Δx.

Fig. 7 (left) illustrates the evolution of the RoePINN prediction L_2 error as r increases. An optimal value for the error is found at $r = 0.86$ and corresponds to point 4, which is marked in blue in Fig. 7 (right). We note that this optimal value of r is close to 1, meaning that optimal robustness and prediction capacity in the presence of sharp discontinuities is achieved when d_{coll} is similar to Δx. The equilibrium between these two hyperparameters ensures therefore optimal balancing between the different terms involved in the computation of the derivatives in the PDE.

Fig. 8 shows the solution in Test case F1 for different values of the *CFL* hyperparameter. It is observed that the RoePINN inherits this feature from the numerical solver, because solutions with high *CFL* factor show more numerical diffusion than those with lower *CFL*. Furthermore, it allows for values of *CFL* greater than 1, where explicit finite volume numerical solvers fail [49]. In particular, we observe that the prediction for $CFL = 5$ shows more numerical diffusion than the rest.

3.2. Forward test case F2. Sinusoidal source term

In Test case F2 the initial condition is $u(x, t = 0) = 1$. Fig. 9 shows the graphical representation of the solution for Test case F2 at $t_1 = 0.03 < t_2 = 0.177$ (from upper to lower), different number of collocation points $n_{coll} = \{10^5, 10^4, 10^3\}$ (from left to right) and Δx. The performance of the RoeFV with $\Delta x = 10^{-4}$ (-▲-), RoeFV with $\Delta x = 10^{-2}$ (-▲-), the RoePINN with $\Delta x = 10^{-4}$ (-●-), RoePINN with $\Delta x = 10^{-2}$ (-●-), the vanilla PINN (-■-) and the reference solution (—) are compared. In Test cases F2, F3 and I1, we will use, as a reference solution, the solution provided by the RoeFV with a finer discretization ($\Delta x = 10^{-5}$). The RoePINN and the vanilla PINN have comparable performance, being the vanilla PINN more accurate in the approximation to the solution in the first peak of the

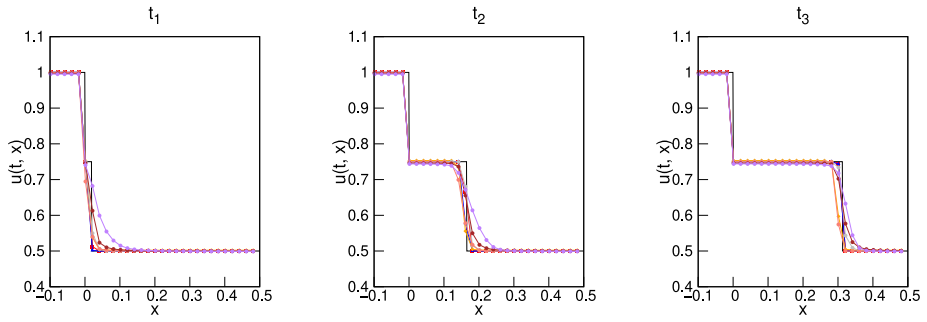


Fig. 8. Test case F1: Solution of the RoePINN with $\Delta x = 10^{-3}$ and $n_{coll} = 10^5$ for different values of $CFL = \{0.1$ (-■-), 0.5 (-▾-), 1 (-•-), 1.5 (-•-), 2 (-•-), 5 (-•-)$\}$, RoeFV with $\Delta x = 10^{-3}$ (-•-) and the reference solution (—). Time snapshots are ordered from left to right ($t_1 = 5 \times 10^{-2} < t_2 = 0.275 < t_3 = 0.5$). The plotted discretization is $10 \times \Delta x$.

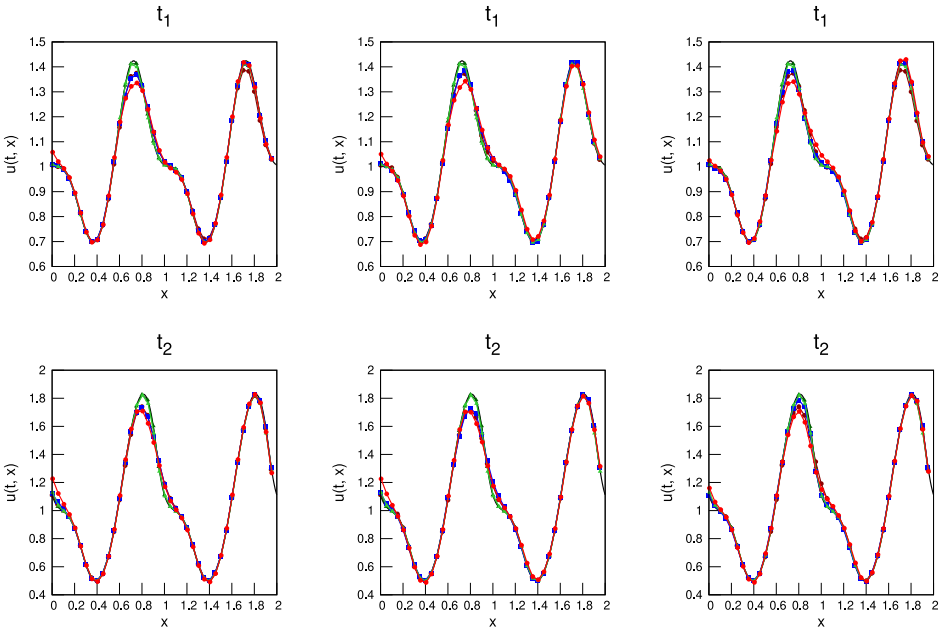


Fig. 9. Test case F2: Solution of the *vanilla* PINN (-■-), RoePINN with $\Delta x = 10^{-4}$ (-•-), RoePINN with $\Delta x = 10^{-2}$ (-•-), RoeFV with $\Delta x = 10^{-4}$ (-▲-), RoeFV with $\Delta x = 10^{-2}$ (-▲-), and the reference solution (—). Time snapshots are ordered from upper to lower ($t_1 = 0.03 < t_2 = 0.177$). The first, second, and third columns correspond to different numbers of collocation points $n_{coll} = \{10^5, 10^4, 10^3\}$, respectively. The plotted discretization is $30 \times \Delta x$.

solution and in the boundary condition at $x = 0$. The RoePINN with $\Delta x = 10^{-4}$ shows better performance than the RoePINN with $\Delta x = 10^{-2}$, for $n_{coll} = 10^3$, as opposed to Test case F1, where a coarser refinement of Δx leads to more accuracy.

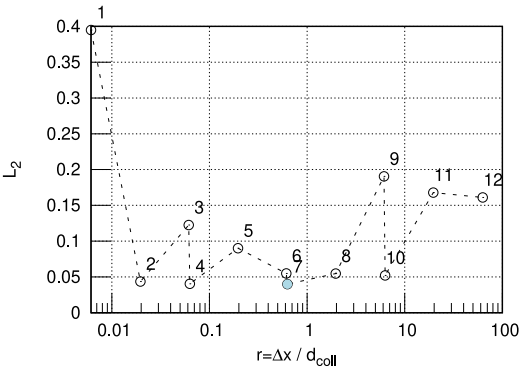
Table 2 shows the L_2 error for the different algorithms and combinations of hyperparameters Δx and n_{coll}. As expected, the error in the RoeFV decreases with the reduction of the cell size. The RoePINN predictions and the vanilla PINN predictions are less accurate than the RoeFV for all combinations of n_{coll} and Δx. From the table, a general trend for the evolution the predictive error against n_{coll} for both PINNs cannot be inferred. In the case of the RoePINN, a decrease of the error is observed when Δx is reduced. Even though the vanilla PINN is not dependent on Δx, numerical errors in Table 2 change with the reference solution used, dependent on the Δx selected. In particular, the vanilla PINN is more accurate than the RoePINN in this test case, for almost any combination of n_{coll} and Δx.

Fig. 10 shows the evolution of the RoePINN prediction L_2 error as r increases. An optimal value for the error is found at point 7 with $r = 0.63$. In this case, low errors are achieved also in the surroundings of $r = 1$.

Fig. 11 shows the solution in Test case F2 for different values of the *CFL*. The RoePINN with $CFL = 0.1$ struggles capturing the peaks in the solution. The prediction for $CFL = 5$ shows the best performance, as opposed to Test case F1, due to the absence of discontinuities in the source term and the solution.

Table 2
Test case F2: Comparison of L_2 errors for the different algorithms and values of Δx and n_{coll}.

		L_2 Error	
n_{coll}	RoeFV ($\Delta x = 10^{-4}$)	RoePINN ($\Delta x = 10^{-4}$)	Vanilla PINN
10^5	–	0.040472	0.038177
10^4	–	0.043630	0.040031
10^3	–	0.394987	0.040101
–	0.000203	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	RoePINN ($\Delta x = 10^{-3}$)	Vanilla PINN
10^5	–	0.039888	0.038177
10^4	–	0.090164	0.040031
10^3	–	0.122556	0.040101
–	0.00494	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	RoePINN ($\Delta x = 10^{-2}$)	Vanilla PINN
10^5	–	0.052041	0.040416
10^4	–	0.054733	0.041396
10^3	–	0.054833	0.047445
–	0.020796	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	RoePINN ($\Delta x = 10^{-1}$)	Vanilla PINN
10^5	–	0.160984	0.119479
10^4	–	0.167909	0.117774
10^3	–	0.190509	0.133435
–	0.118985	–	–



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	1×10^{-4}	10^3	0.0166	0.0060
2	1×10^{-4}	10^4	0.0051	0.0196
4	1×10^{-4}	10^5	0.0016	0.0625
3	10^{-3}	10^3	0.0166	0.0601
5	10^{-3}	10^4	0.0051	0.1963
7	$\mathbf{10^{-3}}$	$\mathbf{10^5}$	**0.0016**	**0.6300**
6	10^{-2}	10^3	0.0166	0.6010
8	10^{-2}	10^4	0.0051	1.9630
10	10^{-2}	10^5	0.0016	6.3000
9	10^{-1}	10^3	0.0166	6.0100
11	10^{-1}	10^4	0.0051	19.6300
12	10^{-1}	10^5	0.0016	63.0000

Fig. 10. Test case F2: (*Left*) Evolution of the RoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r.

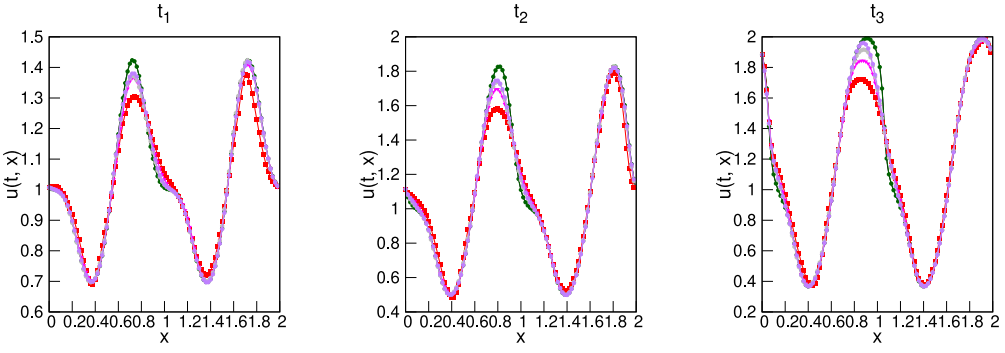


Fig. 11. Test case F2: Solution of the RoePINN with $\Delta x = 10^{-4}$ and $n_{coll} = 10^5$ for different values of $CFL = \{0.1$ (-■-), 0.5 (-▾-), 1 (-•-), 5 (-◦-)$\}$, RoeFV with $\Delta x = 10^{-3}$ (-•-) and the reference solution (—). Time snapshots are ordered from left to right ($t_1 = 5 \times 10^{-2} < t_2 = 0.275 < t_3 = 0.5$). The plotted discretization is $20 \times \Delta x$.

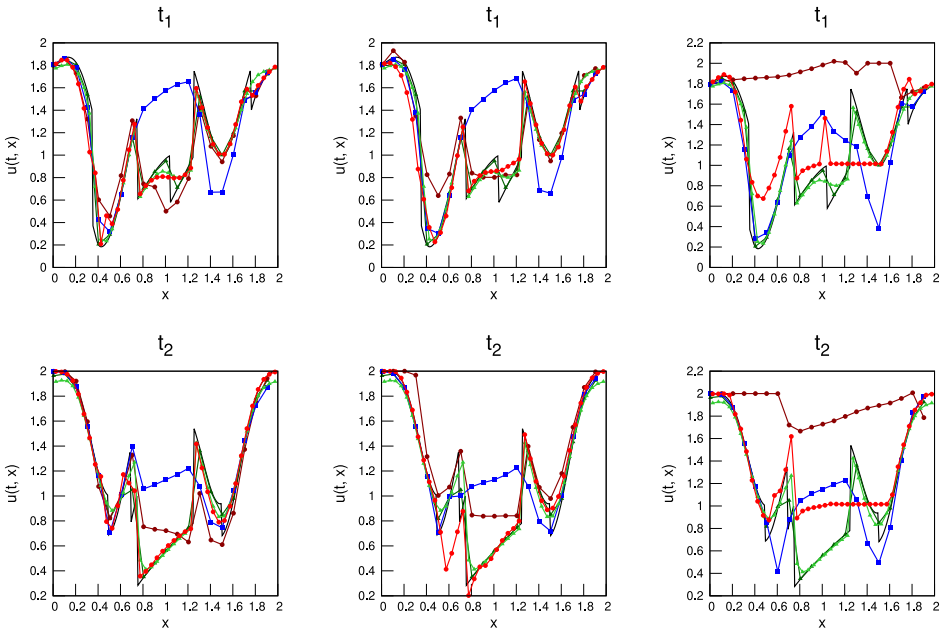


Fig. 12. Test case F3: Solution of the *vanilla* PINN (-■-), RoePINN with $\Delta x = 10^{-2}$ (-•-), RoePINN with $\Delta x = 5 \times 10^{-2}$ (-•-), RoeFV with $\Delta x = 10^{-2}$ (-▲-), RoeFV with $\Delta x = 5 \times 10^{-2}$ (-▲-), and the reference solution (—). Time snapshots are ordered from upper to lower ($t_1 = 0.5 < t_2 = 1$). The first, second, and third columns correspond to different numbers of collocation points $n_{coll} = \{2 \times 10^5, 10^4, 10^3\}$, respectively. The plotted discretization is Δx for RoePINN with $\Delta x = 5 \times 10^{-2}$, and $10 \times \Delta x$ for the rest.

3.3. Forward test case F3. Piecewise source term

In Test case F3, the solution of the Burgers' equation with initial condition $u(x, t = 0) = 1$ is considered. Fig. 12 shows the graphical representation of the solution for the Test case F3 at $t_1 = 0.5 < t_2 = 1$ (from upper to lower) and different values of $n_{coll} = \{2 \times 10^5, 10^4, 10^3\}$ (from left to right) and Δx.

The RoePINN clearly outperforms the vanilla PINN, which is unable to reproduce the jagged solution pattern in the middle region of the spatial domain for any number of collocation points. Also, for $n_{coll} = 10^3$, we are able to improve the prediction of the RoePINN by increasing Δx, as observed in Test case F1.

Table 3 shows the L_2 error for the different algorithms and combinations of hyperparameters Δx and n_{coll}. As expected, the error in the RoeFV decreases with the reduction of the cell size. The RoePINN predictions and the vanilla PINN predictions are less accurate than those of the RoeFV for any combination of n_{coll} and Δx. Errors in Test case F3 for both PINNs are approximately one order of magnitude larger than the errors in cases F1 and F2. This is explained by the complexity of the source term, which leads to a very ragged solution with multiple shocks and rarefactions.

The evolution of the error with n_{coll} closely resembles that of Test case F1 for the RoePINN. When $\Delta x = 10^{-3}$ or $\Delta x = 10^{-2}$, the solution improves with the increase of n_{coll}. For $\Delta x = 5 \times 10^{-2}$ or $\Delta x = 10^{-1}$ there is not a general trend and the error is less sensitive to n_{coll}. For values of $\Delta x = 10^{-2}$ and $\Delta x = 5 \times 10^{-2}$, it is clear that the RoePINN significantly outperforms the vanilla PINN. For the vanilla PINN, slight variations in the error appear with the variation of n_{coll}.

Fig. 13 shows the evolution of the RoePINN prediction L_2 error as r increases. An optimal value for the error is found in this case for values $r > 1$, a similar tendency to the one observed in Test case F1.

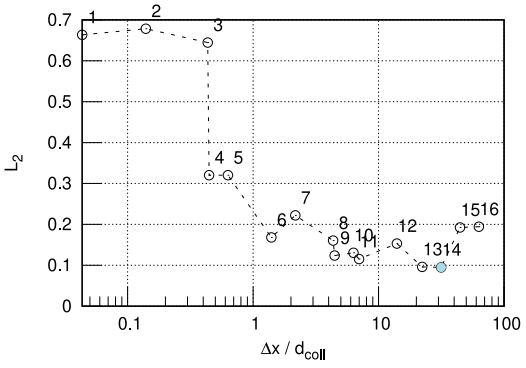
Fig. 14 shows the prediction of the RoePINN in Test case F3 for different values of the *CFL* hyperparameter, the prediction of the RoeFV and the reference solution. The prediction with $CFL = 0.1$ shows more numerical diffusion than those with higher values. For $CFL > 2$, we observe a decrease in accuracy not only in the middle part of the solution, but also at the boundaries. As in Test case F1, the worse predictions occurs with $CFL = 5$, because of the presence of discontinuities. Predictions with $CFL = \{0.3, 0.5, 0.9, 1.0, 1.5\}$ show better performance, similar to that of the RoeFV. As in Test cases F1 and F2, values of CFL > 1 do not destabilize the solution, unlike in explicit solvers.

3.4. Inverse test case I1. Unknown piecewise source term

In Test case I1, the solution of the Burgers' equation in Test case F3 is considered again, but now we aim to reconstruct $z(x)$, which we assume to be unknown. We also assume that domain data is now available as a set of snapshots equally distributed across the computational domain. Data snapshot d can be expressed as $u_d(x = \{x_o, x_o + \Delta x_d, \ldots, x_f\}, t_d)$ with $d = 1, \ldots, n_d$, where $\Delta x_d = 10^{-3}$ is the discretization of the data snapshots, $\Delta t_d = 0.005$ corresponds to the sampling period over the overall simulation time $t_f = 0.5$

Table 3
Test case F3: Comparison of L_2 errors for the different algorithms and values of Δx and n_{coll}.

		L_2 Error	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	RoePINN ($\Delta x = 10^{-3}$)	Vanilla PINN
2×10^5	–	0.320328	0.271013
10^4	–	0.678309	0.269595
10^3	–	0.663378	0.258027
–	0.010593	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	RoePINN ($\Delta x = 10^{-2}$)	Vanilla PINN
2×10^5	–	0.130634	0.272991
10^4	–	0.167937	0.271707
10^3	–	0.644376	0.260902
–	0.033306	–	–
n_{coll}	RoeFV ($\Delta x = 5 \times 10^{-2}$)	RoePINN ($\Delta x = 5 \times 10^{-2}$)	Vanilla PINN
2×10^5	–	0.094993	0.284320
10^4	–	0.115285	0.284125
10^3	–	0.222092	0.276194
–	0.082732	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	RoePINN ($\Delta x = 10^{-1}$)	Vanilla PINN
2×10^5	–	0.194443	0.298741
10^4	–	0.153167	0.302047
10^3	–	0.160325	0.295879
–	0.158252	–	–



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-3}	10^2	0.0234	0.04
2	10^{-3}	10^3	0.0072	0.14
3	10^{-3}	10^4	0.0022	0.45
6	10^{-3}	10^5	0.0016	0.63
5	10^{-2}	10^2	0.0234	0.43
4	10^{-2}	10^3	0.0072	1.39
7	10^{-2}	10^4	0.0022	4.48
10	10^{-2}	10^5	0.0016	6.31
8	5×10^{-2}	10^2	0.0234	2.14
10	5×10^{-2}	10^3	0.0072	6.95
11	5×10^{-2}	10^4	0.0022	22.41
14	$\mathbf{5 \times 10^{-2}}$	10^5	**0.0016**	**31.55**
13	10^{-1}	10^2	0.0234	4.28
12	10^{-1}	10^3	0.0072	13.90
15	10^{-1}	10^4	0.0022	44.83
16	10^{-1}	10^5	0.0016	63.10

Fig. 13. Test case F3: (*Left*) Evolution of the RoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r.

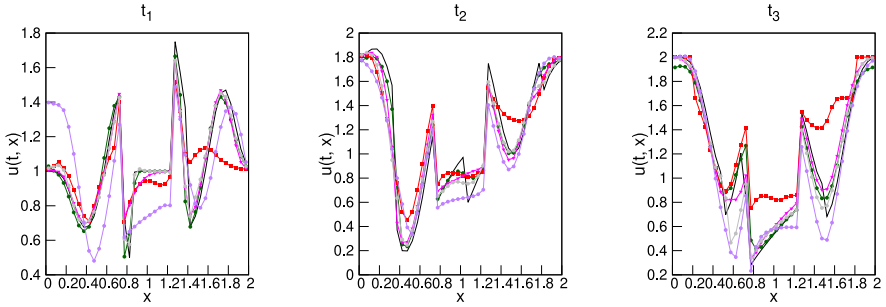


Fig. 14. Test case F3: Solution of the RoePINN with $\Delta x = 10^{-3}$ and $n_{coll} = 2 \times 10^5$ for different values of $CFL = \{0.1$ (-■-), 0.5 (-▾-), 1 (-•-), 5 (-•-)$\}$, RoeFV with $\Delta x = 10^{-3}$ (-•-) and the reference solution (—). Time snapshots are ordered from left to right ($t_1 = 5 \times 10^{-2} < t_2 = 0.275 < t_3 = 0.5$). The plotted discretization is $10 \times \Delta x$.

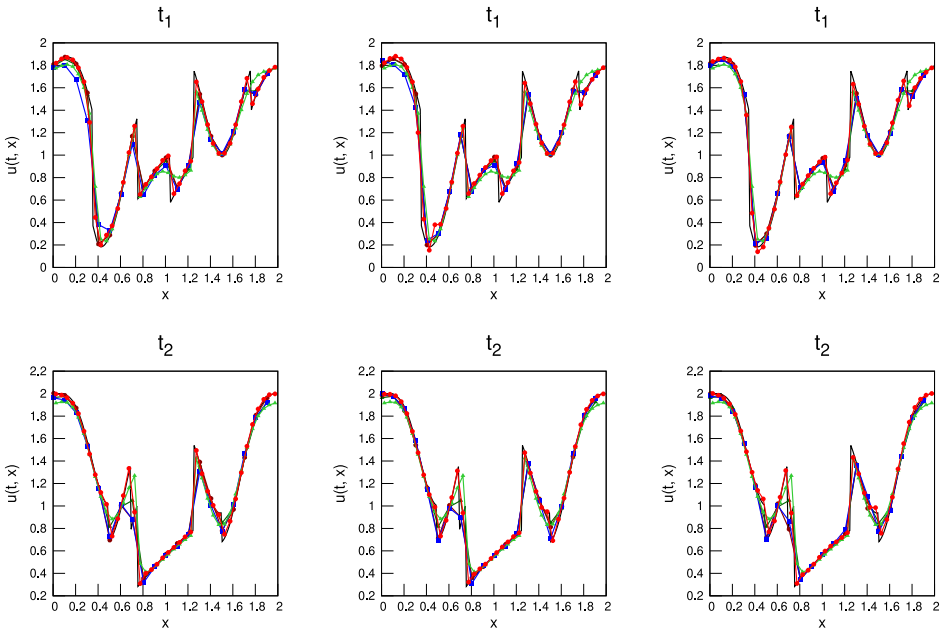


Fig. 15. Test case I1. Forward predictions: Solution of the *vanilla* PINN (-■-), RoePINN with $\Delta x = 10^{-2}$ (-•-), RoePINN with $\Delta x = 5 \times 10^{-2}$ (-•-), RoeFV with $\Delta x = 10^{-2}$ (-▲-), RoeFV with $\Delta x = 5 \times 10^{-2}$ (-▲-), and the reference solution (—). Time snapshots are ordered from upper to lower ($t_1 = 0.5 < t_2 = 1$). The first, second, and third columns correspond to different numbers of collocation points $n_{coll} = \{2 \times 10^5, 10^4, 10^3\}$, respectively. The plotted discretization is Δx for RoePINN with $\Delta x = 5 \times 10^{-2}$, and $10 \times \Delta x$ for the rest.

and $n_d = t_f / \Delta t = 50$ the number of data snapshots sampled across the domain. A summary of the multiple parameters can be found in Appendix D 8.4.

Fig. 15 shows the graphical representation of the solution for Test case I1 at $t_1 = 0.5 < t_2 = 1$ (from upper to lower) and different values of $n_{coll} = \{2 \times 10^5, 10^4, 10^3\}$ (from left to right) and Δx.

As expected, predictions improve significantly with respect to the results of Test case F3, due to the provision of domain data, specially for the RoePINN, that yields the best approximation. It is observed that, even though we provide domain data, the vanilla PINN still struggles to capture sharp discontinuities across the solution.

Table 4 presents the L_2 error for the different algorithms and combinations of hyperparameters, Δx and n_{coll}. The RoePINN yields errors one order of magnitude lower than those obtained in Test case F3, across all values of Δx and n_{coll}, except for $\Delta x = 10^{-1}$. For the RoePINN with $\Delta x = 10^{-1}$, the trade-off between fitting the domain data and minimizing the residuals shifts in favor of the latter, leading to a higher error metric because Δx is comparably larger. We also observe that the errors for the RoePINN only slightly increase when reducing the value of n_{coll}, because now the fitting of the data compensates for the loss of robustness when reducing the value of n_{coll}. RoeFV and RoePINN errors are evenly matched. For the vanilla PINN, errors also considerably decrease as compared with those in Test case F3, but they are larger than the errors for the RoePINN in all cases.

Fig. 16 shows the evolution of the RoePINN prediction L_2 error as r increases. An optimal value for the error is found at $r = 6.32$, but the error is also lower for $r = 1.40$, which is closer to 1, similar to what we observed in previous test cases. When feeding the RoePINN with domain data, the general trend of the error is different from previous test cases, because the nature of the problem is also different.

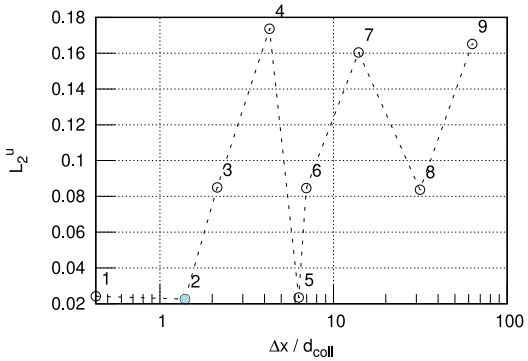
Parallel to the fitting of the available domain data, the algorithm's main objective is to reconstruct the geometrical source term $z(x)$. The IVanPINN and the IRoePINN were unable to infer the source term function. On the contrary, the ISVanPINN and the ISRoePINN models, which enforce $z = z(x)$ without time dependency, provided satisfactory results that are presented here.

Fig. 17 shows the graphical representation of the reconstructed source term function for three different values of n_{coll} (from left to right) and Δx. Now, we compare the ISRoePINN prediction and the ISVanPINN predictions against the real value of the source term, which we refer to as reference source term, discretized with Δx. The ISRoePINN shows a more accurate reconstruction of the peaks and troughs present in the source term than the ISVanPINN, for $n_{coll} = 10^5$ and $n_{coll} = 10^4$, but is less accurate for $n_{coll} = 10^3$, particularly in the central part of the source term. Both PINNs struggle to capture the rightmost sinusoidal edge.

Table 5 presents the L_2 error for the different algorithms and combinations of hyperparameters, Δx and n_{coll}. Most of the patterns observed do not align with the trends obtained from earlier test cases. This inconsistency arises because we are reconstructing a source term, which is static in time and fundamentally differs from the dynamics of the solution in forward problems. The ISRoePINN infers $z(x)$ more accurately than the ISVanPINN when using the lowest value of Δx for any number of collocation points. When increasing the value of Δx, the ISRoePINN provides more accurate results for the largest value of collocation points.

Table 4
Test case I1. Forward predictions: Comparison of L_2^u errors for the different algorithms and values of Δx and n_{coll}.

		L_2^u Error	
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	RoePINN ($\Delta x = 10^{-2}$)	Vanilla PINN
2×10^5	–	0.023509	0.072045
10^4	–	0.022568	0.062119
10^3	–	0.024150	0.055884
–	0.033306	–	–
n_{coll}	RoeFV ($\Delta x = 5 \times 10^{-2}$)	RoePINN ($\Delta x = 5 \times 10^{-2}$)	Vanilla PINN
2×10^5	–	0.083710	0.090860
10^4	–	0.084609	0.080398
10^3	–	0.084941	0.086343
–	0.082732	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	RoePINN ($\Delta x = 10^{-1}$)	Vanilla PINN
2×10^5	–	0.165149	0.162552
10^4	–	0.160403	0.161056
10^3	–	0.173654	0.160668
–	0.158252	–	–



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-2}	10^3	0.0235	0.43
2	$\mathbf{10^{-2}}$	10^4	**0.0071**	**1.40**
5	10^{-2}	10^5	0.0016	6.32
3	5×10^{-2}	10^3	0.0235	2.13
6	5×10^{-2}	10^4	0.0071	7.01
8	5×10^{-2}	10^5	0.0016	31.62
4	10^{-1}	10^3	0.0235	4.26
7	10^{-1}	10^4	0.0071	14.05
9	10^{-1}	10^5	0.0016	63.22

Fig. 16. Test case I1. Forward predictions: (*Left*) Evolution of the RoePINN L_2^u error against $r = \frac{\Delta x}{d_{coll}}$ for u. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r ordered by r for u.

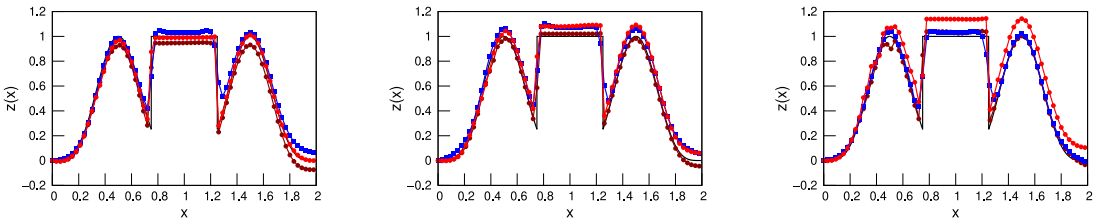


Fig. 17. Test case I1. Source term discovery: Source term reconstruction of the vanilla PINN (-■-), ISRoePINN with $\Delta x = 10^{-2}$ (-•-), ISRoePINN with $\Delta x = 5 \times 10^{-2}$ (-•-), and the reference source term (—). The first, second, and third columns correspond to different numbers of collocation points $n_{coll} = \{2 \times 10^5, 10^4, 10^3\}$, respectively. The plotted discretization is $30 \times \Delta x$.

Fig. 18 shows the evolution of the ISRoePINN L_2 prediction error as r increases. An optimal value for the error is found at $r = 31.62$. The error for $r = 1.40$ is also low, in accordance with the observation of previous test cases. This is again due to the inverse data-driven character of the Test case I1. The error curve is again rather oscillatory for this same reason.

4. 1D mathematical models of elastic arteries

In this section, we seek to extend the techniques in Section 2 to the modeling of blood flow inside elastic vessels.

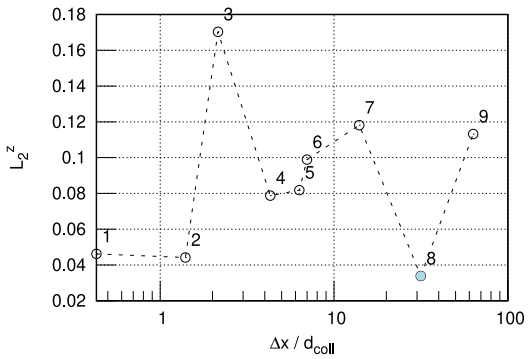
4.1. Tube laws and elastic mechanical properties of vessels

A 1D model describing blood flow inside a vessel can be formulated as a nonlinear hyperbolic system of equations as follows:

$$\partial_t \mathbf{U} + \partial_x \mathcal{F}(\mathbf{U}) = \mathbf{G}_\chi, \tag{26}$$

Table 5
Test case I1. Source term discovery: Comparison of L_2^z errors for the different algorithms and values of Δx and n_{coll}.

	L_2^z Error	
n_{coll}	ISRoePINN ($\Delta x = 10^{-2}$)	ISVanPINN
2×10^5	0.082207	0.099430
10^4	0.073642	0.116845
10^3	0.045797	0.088686
n_{coll}	ISRoePINN ($\Delta x = 5 \times 10^{-2}$)	ISVanPINN
2×10^5	0.051876	0.139587
10^4	0.129688	0.146917
10^3	0.220912	0.123095
n_{coll}	ISRoePINN ($\Delta x = 10^{-1}$)	ISVanPINN
2×10^5	0.028769	0.062046
10^4	0.094526	0.091054
10^3	0.169126	0.038268



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-2}	10^3	0.0235	0.43
2	10^{-2}	10^4	0.0071	1.40
5	10^{-2}	10^5	0.0016	6.32
3	5×10^{-2}	10^3	0.0235	2.13
6	5×10^{-2}	10^4	0.0071	7.01
8	$\mathbf{5 \times 10^{-2}}$	$\mathbf{10^5}$	**0.0016**	**31.62**
4	10^{-1}	10^3	0.0235	4.26
7	10^{-1}	10^4	0.0071	14.05
9	10^{-1}	10^5	0.0016	63.22

Fig. 18. Test case I1. Source term discovery: (*Left*) Evolution of the ISRoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$ for z. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r for z.

with $\mathbf{U} = \mathbf{U}(x,t)$ and

$$\mathbf{U} = \begin{bmatrix} A \\ Q \end{bmatrix}, \quad \mathcal{F}(\mathbf{U}) = \begin{bmatrix} Q \\ \kappa \frac{Q^2}{A} \end{bmatrix}, \mathbf{G}_\chi = \begin{bmatrix} 0, -\frac{A}{\rho}\frac{\partial p}{\partial x} - \frac{f}{\rho} - gA\frac{\partial \eta}{\partial x} \end{bmatrix}, \tag{27}$$

where x is the axial coordinate along the vessel and t represents time, A denotes the cross-sectional area. The volume flow rate, $Q = Au$, depends on u, the cross-sectional average axial velocity. The internal pressure averaged over the cross section is $p(x,t)$, and f is the friction force per unit length. Additionally, ρ represents blood density, while η is the coordinate perpendicular to the Earth's surface, accounting for gravitational forces due to acceleration g, which is required to account for postural changes. The parameter κ, related to the velocity profile, is set to 1 in this study, assuming a blunt velocity profile.

An explicit algebraic relationship between p and A (or tube law) is also required to close the system of equations in (26) and account for the fluid–structure interaction of the problem

$$p(x,t) - p_e(x,t) = p_{tr}, \tag{28}$$

where p_e is the external pressure, p_{tr} is the elastic transmural pressure that can be expressed as:

$$p_{tr} = K_o \sigma + p_o, \quad \sigma(\alpha) = \alpha^{m_\sigma} - \alpha^{n_\sigma}, \tag{29}$$

with $K_o = K_o(x)$ the vessel stiffness, σ represents the dimensionless transmural pressure difference, $\alpha = A/A_o$, $A_o = A_o(x)$ the reference area, $p_o = p_o(x)$ the reference pressure and the exponents m_σ and n_σ, are dependent on the specific type of vessel. For arteries, our case, the common parameters are $m_\sigma = 1/2$ and $n_\sigma = 0$. In this work, as a first approximation we will consider a frictionless case, $f = 0$ with $\partial p_o = 0$, $\partial p_e = 0$ and $\partial \eta = 0$. Then, the system can be written in quasilinear form as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{J}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} = \mathbf{S}_\chi, \tag{30}$$

where

$$\mathbf{J}(\mathbf{U}) = \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix}, \quad \mathbf{S}_\chi = \begin{bmatrix} 0 \\ -\left(\frac{A}{\rho}\sigma \partial_x K_o + \frac{A}{\rho} K_o \sigma_{A_o} \partial_x A_o\right) \end{bmatrix}, \tag{31}$$

The matrix $\mathbf{J}$ has two eigenvalues, $\lambda^1 = u - c$ and $\lambda^2 = u + c$, as well as two real eigenvectors, $\mathbf{e}^1 = \left(1, \lambda^1\right)^T$ and $\mathbf{e}^2 = \left(1, \lambda^2\right)^T$, where c is often known as the pulse wave velocity (PWV), given by

$$c = \sqrt{\frac{A}{\rho} K_o \sigma_A} = \sqrt{\frac{K_o}{\rho}(m\alpha^m - n\alpha^n)}, \tag{32}$$

for a given vessel stiffness K_o, or alternatively as

$$c^2 = \frac{c_o^2}{m-n}(m\alpha^m - n\alpha^n), \quad c_o = \sqrt{\frac{K_o}{\rho}(m-n)}, \tag{33}$$

with c_o as the *reference* PWV. Further details can be found in [49,50]. In this work, we will use the stiffness K_o, noting that its conversion to c_o is straightforward.

Neural network models operate at their best when input and output variables, and their correspondent derivatives, lie in the same order of magnitude, usually 10^0. Therefore, nondimensionalization is crucial, especially in our case, as the magnitudes of the variables involved in the description of the flow inside the vessel vary significantly across time and space, differ several orders of magnitudes with each other, and their variations span across very low order of magnitudes ($10^{-4}, 10^{-5}$). For higher orders of magnitude ($> 10^2$), nondimensionalization is recommended but not always necessary.

The above equations can be nondimensionalized by choosing $a = \frac{A}{A_d}$, $a_o = \frac{A_o}{A_d}$, with $A_d = \pi r_d^2$, $x' = \frac{x}{X}$, $t' = \frac{t}{T}$, $U = \frac{L}{T}$, $q = \frac{Q}{A_d U}$ and $p' = \frac{p}{\rho U^2}$, with $X = x_f - x_o$ the length of the vessel and T a characteristic time, in this case the total simulation time $T = t_f - t_o$, as elaborated in [51,52]. Also, we have that $k_o = \frac{K_o}{\rho U^2}$, $c' = \sqrt{a k_o \sigma_a}$ and $u' = \frac{q}{a}$, $p'_o = \frac{p_o}{\rho U^2}$, $a = a/a_o = \alpha$. To keep the notation uncluttered, we remove the tilde from the dimensionless variables.

4.2. RoeFV method in systems of equations

The system of equations in (30) can be integrated in time and space according to the Godunov first order method, so that the updating can be written in fluctuation form as follows:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\delta\mathbf{M}^{ARoe-}_{i+\frac{1}{2}} + \delta\mathbf{M}^{ARoe+}_{i-\frac{1}{2}}\right), \tag{34}$$

where

$$\delta\mathbf{M}^{ARoe-}_{i+\frac{1}{2}} + \delta\mathbf{M}^{ARoe+}_{i+\frac{1}{2}} = \delta\mathbf{M}_{i+\frac{1}{2}}. \tag{35}$$

with $\delta\mathbf{M}^{ARoe\mp}_{i\pm\frac{1}{2}}$ constructed using a set of constant coefficients

$$\delta\mathbf{M}^{ARoe-}_{i+\frac{1}{2}} = \sum_r \left(\widetilde{\lambda}^- \alpha\theta\widetilde{\mathbf{e}}\right)^r_{i+\frac{1}{2}}, \quad \delta\mathbf{M}^{ARoe+}_{i-\frac{1}{2}} = \sum_r \left(\widetilde{\lambda}^+ \alpha\theta\widetilde{\mathbf{e}}\right)^r_{i-\frac{1}{2}}, \tag{36}$$

expressing relations for conserved variables and consistency preservation of the approximate Jacobian used, provided in [1,49]. The time step is defined in general as

$$\Delta t = CFL \min\left\{\Delta t_{i+\frac{1}{2}}\right\}, \quad \Delta t_{i+\frac{1}{2}} = \frac{\Delta x}{\max\left\{|\widetilde{\lambda}^1_{i+\frac{1}{2}}|, |\widetilde{\lambda}^2_{i+\frac{1}{2}}|\right\}} \tag{37}$$

with $CFL < 1$.

4.3. PINN formulation with automatic differentiation: vanilla PINN

4.3.1. Forward vanilla PINN

The vanilla PDE loss $\mathcal{L}^{v,F}_{PDE}$ associated with the minimization of the PDE residual is

$$\begin{aligned}\mathcal{L}^{v,F}_{PDE} = {} & l_1 \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(\left.\frac{\partial a}{\partial t}\right|_{(x_c^i, t_c^i)} + \left.\frac{\partial q}{\partial x}\right|_{(x_c^i, t_c^i)}\right)^2 \\ & + l_2 \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(\left.\frac{\partial q}{\partial t}\right|_{(x_c^i, t_c^i)} + \left.\frac{\partial (q^2/a)}{\partial x}\right|_{(x_c^i, t_c^i)} + a \left.\frac{\partial p(k_o, a)}{\partial x}\right|_{(x_c^i, t_c^i)}\right)^2\end{aligned} \tag{38}$$

with $(a, q) = \mathcal{PINN}(x, t; \boldsymbol{w})$ the predicted solutions at some training iteration, and pressure p computed with the predictions for the area using the closure model. The initial condition loss $\mathcal{L}_o$ related to the fitting of the initial conditions $q(x, t = 0)$ and $a(x, t = 0)$ to available initial datapoints $q_o(x_o, t_o)$ and $a_o(x_o, t_o)$ becomes

$$\mathcal{L}_o = l_3 \frac{1}{n_o} \sum_{i=1}^{n_o} \left(q(x_o^i, t_o^i) - q_o^i(x_o^i, t_o^i)\right)^2 + l_4 \frac{1}{n_o} \sum_{i=1}^{n_o} \left(a(x_o^i, t_o^i) - a_o^i(x_o^i, t_o^i)\right)^2, \tag{39}$$

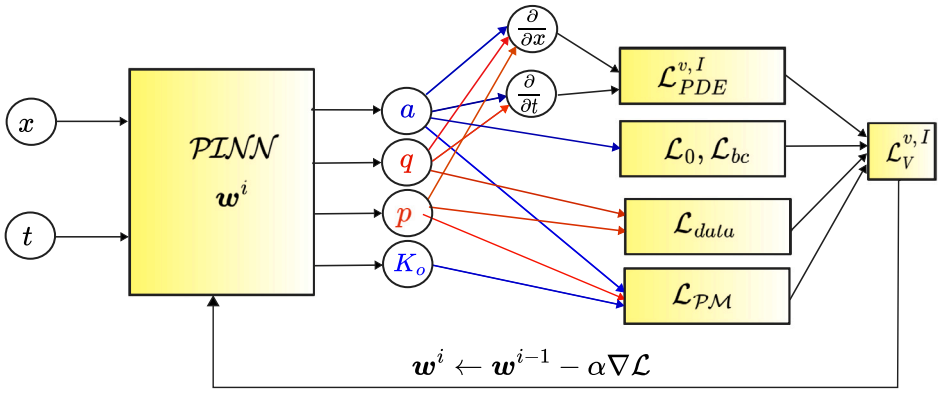


Fig. 19. Scheme of a IVanPINN solver for vessels. Again, note the difference between variables q and p, associated with available data (red color), and discovered variables a and K_o, where no data is available or measurable (blue color).

associated with the fitting of the initial conditions. The boundary condition loss $\mathcal{L}_{bc}$ related to the fitting of the boundary conditions $q(x = \partial\Omega, t > 0)$ and $a(x = \partial\Omega, t > 0)$ to available boundary datapoints $q_{bc}(x_{bc}, t_{bc})$ and $a_{bc}(x_{bc}, t_{bc})$ becomes

$$\mathcal{L}_{bc} = l_5 \frac{1}{n_{bc}} \sum_{i=1}^{n_{bc}} \left(q(x^i_{bc}, t^i_{bc}) - q^i_{bc}(x^i_{bc}, t^i_{bc})\right)^2 + l_6 \frac{1}{n_{bc}} \sum_{i=1}^{n_{bc}} \left(a(x^i_{bc}, t^i_{bc}) - a^i_{bc}(x^i_{bc}, t^i_{bc})\right)^2. \tag{40}$$

The global loss function of the vanilla PINN $\mathcal{L}^{v,F}_V$ is constructed as a weighted sum of the individual losses, and has the following form for the forward setting:

$$\mathcal{L}^{v,F}_V = \mathcal{L}^{v,F}_{PDE} + \mathcal{L}_o + \mathcal{L}_{bc} \tag{41}$$

The configuration of a training step of the vanilla RoePINN, for the forward problem, is similar to that in Fig. 1 (left), but now the outputs are a and q.

4.3.2. Inverse vanilla PINN with one single model (IVanPINN)

For the inverse problem, we can define the PDE loss function as:

$$\begin{aligned}\mathcal{L}^{v,I}_{PDE} = {} & l_1 \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(\frac{\partial a}{\partial t}\bigg|_{(x^i_c, t^i_c)} + \frac{\partial q}{\partial x}\bigg|_{(x^i_c, t^i_c)} \right)^2 \\ & + l_2 \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(\frac{\partial q}{\partial t}\bigg|_{(x^i_c, t^i_c)} + \frac{\partial (q/a)^2}{\partial x}\bigg|_{(x^i_c, t^i_c)} + a \frac{\partial p}{\partial x}\bigg|_{(x^i_c, t^i_c)} \right)^2\end{aligned} \tag{42}$$

where $(a, q, p, K_o) = \mathcal{PINN}(x, t; \boldsymbol{w})$, as shown in Fig. 19, where variables q and p are associated with available data (red color), while a and K_o, are discovered variables, with no data available (blue color). Additionally, we define

$$\begin{aligned}\mathcal{L}_{data} = {} & l_7 \frac{1}{n_d} \sum_{i=1}^{n_d} \left(q(x^i_d, t^i_d) - q^i_d(x^i_d, t^i_d)\right)^2 \\ & + l_8 \frac{1}{n_d} \sum_{i=1}^{n_d} \left(p(x^i_d, t^i_d) - p^i_d(x^i_d, t^i_d)\right)^2,\end{aligned} \tag{43}$$

where training data is defined as $\mathcal{D}_q : \{x^i_d, t^i_d, q^i_d\}^{n_d}_{i=1}$ and $\mathcal{D}_p : \{x^i_d, t^i_d, p^i_d\}^{n_d}_{d=1}$. In order to account for the pressure closure model, the following loss function is defined

$$\mathcal{L}_{\mathcal{PM}} = l_9 \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left(p(x^i_c, t^i_c) - p_o(x^i_c) - \frac{K_o(x^i_c, t^i_c)}{\rho U^2} \sigma(a(x^i_c, t^i_c)) \right)^2, \tag{44}$$

where K_o is an output of the model and does not require nondimensionalization because its order of magnitude ranges between 10^2 to 10^3. The insertion of this loss term is not mathematically redundant, as we observe that it helps guide the optimization process to comply with the physics of the problem. Overall, we have the following global loss function for the inverse vanilla PINN:

$$\mathcal{L}^{v,I}_V = \mathcal{L}^{v,I}_{PDE} + \mathcal{L}_o + \mathcal{L}_{bc} + \mathcal{L}_{data} + \mathcal{L}_{\mathcal{PM}}. \tag{45}$$

The configuration of a training step of the IVanPINN, for the inverse problem, is plotted in Fig. 19.

4.4. PINN formulation with RoeFV method: RoePINN

In this section, we show how to build the different loss functions needed for the formulation of the RoePINN in elastic vessels.

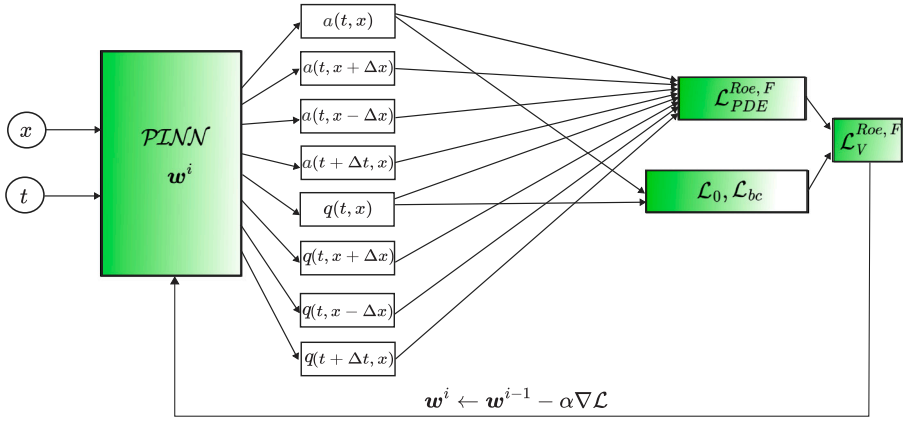


Fig. 20. Scheme of a forward RoePINN for vessels.

4.4.1. Forward RoePINN

The PDE loss $\mathcal{L}_{PDE}^{Roe,F}$ associated with the residual of the numerical scheme RoeFV has the form

$$\mathcal{L}_{PDE}^{Roe,F} = \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left\| \begin{pmatrix} \sqrt{l_1} \\ \sqrt{l_2} \end{pmatrix} \odot \left(\mathbf{U}_{x_c^i}^{t_c^i+\Delta t^i} - \mathbf{U}_{x_c^i}^{t_c^i} + \frac{\Delta t}{\Delta x} \left[(\delta \mathbf{M})^-_{x_c^i+\frac{1}{2}\Delta x} + (\delta \mathbf{M})^+_{x_c^i-\frac{1}{2}\Delta x} \right] \right) \right\|^2. \tag{46}$$

where $\mathbf{U} = (a, q)^T$ and $(a, q) = \mathcal{PINN}(x, t; \boldsymbol{w})$. Therefore, the complete loss function for the forward RoePINN is

$$\mathcal{L}_V^{Roe,F} = \mathcal{L}_{PDE}^{Roe,F} + \mathcal{L}_o + \mathcal{L}_{bc}. \tag{47}$$

Fig. 20 represents the configuration of a training step of a RoePINN for a forward setting.

4.4.2. Inverse RoePINN (IRoePINN)

For the inverse problem, the PDE loss $\mathcal{L}_{PDE}^{Roe,I}$ associated with the residual of the numerical scheme RoeFV has the form

$$\mathcal{L}_{PDE}^{Roe,I} = \frac{1}{n_{coll}} \sum_{i=1}^{n_{coll}} \left\| \begin{pmatrix} \sqrt{l_1} \\ \sqrt{l_2} \end{pmatrix} \odot \left(\mathbf{U}_{x_c^i}^{t_c^i+\Delta t^i} - \mathbf{U}_{x_c^i}^{t_c^i} + \frac{\Delta t}{\Delta x} \left[(\delta \mathbf{M}')^-_{x_c^i+\frac{1}{2}\Delta x} + (\delta \mathbf{M}')^+_{x_c^i-\frac{1}{2}\Delta x} \right] \right) \right\|^2. \tag{48}$$

where $(a, q, p) = \mathcal{PINN}(x, t; \boldsymbol{w})$m as shown in Fig. 21, where variables q and p are associated with available data (red color), while a is the discovered variable with no data available (blue color). Therefore, for the inverse RoePINN the global loss function has the form:

$$\mathcal{L}_V^{Roe,I} = \mathcal{L}_{PDE}^{Roe,I} + \mathcal{L}_o + \mathcal{L}_{bc} + \mathcal{L}_{data}. \tag{49}$$

k_o is computed each training iteration by solving from Eq. (29) as:

$$k_o(a(x_c^i, t_c^i), p(x_c^i, t_c^i)) = \frac{p(x_c^i, t_c^i) - p_o(x_c^i)}{\sigma(a(x_c^i, t_c^i))}, \tag{50}$$

where the quality of the reconstruction of k_o improves as the inference of σ is progressively enhanced. Contrary to the IVanPINN, computing k_o using (50) yields better results than obtaining $k_o(x, t)$ as an output of the RoePINN. Eq. (50) can also be implemented for the vanilla PINN yielding to similar results. Appendix B 8.2 describes the details of the implementation of the IRoePINN algorithm.

5. Numerical results for the 1D model of elastic arteries

The prediction of the RoeFV and that of the different PINNs is compared against a *ground truth* solution that corresponds to the prediction of the RoeFV numerical solver with a finer discretization ($\Delta x = 10^{-4}$), denoted reference solution. The test cases analyzed in this section will be named F4, I2 and I3, where -F- stands for *Forward* problem and -I- for *Inverse* problem.

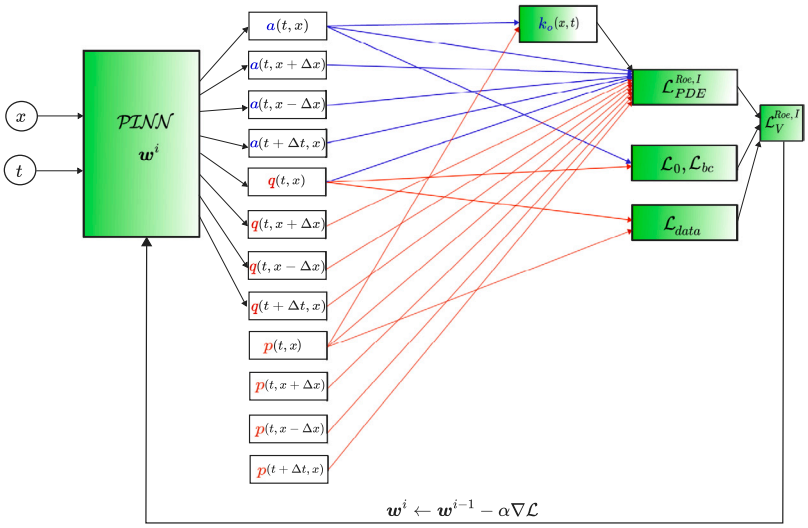


Fig. 21. Scheme of a IRoePINN for vessels, with fitted variables in red color and reconstructed variables in blue color.

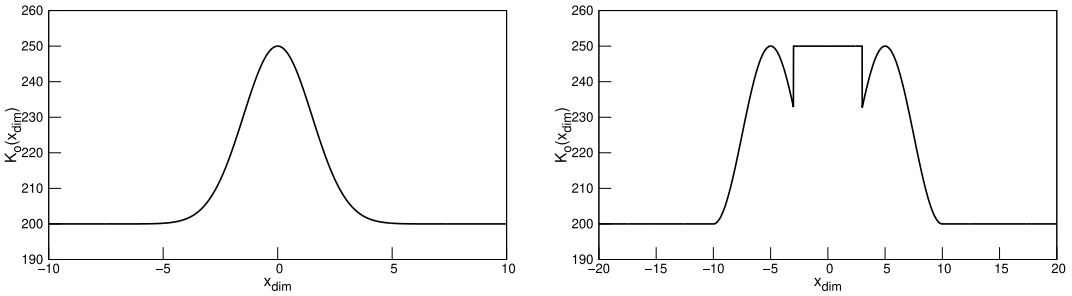


Fig. 22. Stiffness functions employed in Test case I2 (left) and I3 (right).

Test case F4 involves a constant stiffness value. In Test case I2 a smooth gaussian stiffness function is used (Fig. 22, left), with the following form

$$K_o(x_{dim}) = A + B\exp\left(-\frac{\left(x_{dim}-\mu\right)^2}{2\sigma_G^2}\right),$$
$$\mu = \frac{x_o + x_f}{2}, \quad \sigma_G = 0.075(x_o + x_f), \tag{51}$$

with x_{dim} the dimensional coordinate along the vessel centerline and $x_o \leq x_{dim} \leq x_f$. In Test case I3, we use a piecewise function, as shown in Fig. 22) (right):

$$K_o(x_{dim}) = \begin{cases} A & \text{if} \quad |x_o| < |x_{dim} - x_m| < |x_1| \\ A + B\sin^2(x_{dim}) & \text{if} \quad |x_1| < |x_{dim} - x_m| < |x_2| \\ A + B & \text{if} \quad |x_2| < |x_{dim} - x_m| < |x_f| \end{cases}, \tag{52}$$

with $x_m = \left(\frac{x_o+x_f}{2}\right)$ and $x_o \leq x \leq x_f$. A summary of the different parameters involved in (51) and (52) can be found in Appendix D 8.4.

5.1. Forward test case F4. Tourniquet maneuver

In this section we study the solution of the system of equations for 1D blood flow equation with initial velocity condition $u(x, t = 0) = 0\frac{cm}{s}$ and initial radius $r(x, t = 0)$ of value $r_L = 1.5$ cm if $x < 0$ cm and $r_R = 1.0$ cm if $x > 0$ cm, with a constant stiffness K_o. The vessel constricts abruptly on its right half, similar to the application of a *tourniquet* maneuver. Details of the values of the different parameters and hyperparameters in Test case F4 can be found in Appendix D 8.4.

Fig. 23 (upper) shows the flow rates solution at $t = 1.05$ and different values of n_{coll} (from left to right) and Δx. The prediction of the RoeFV method, the RoePINN and the vanilla PINN are compared against the reference solution.

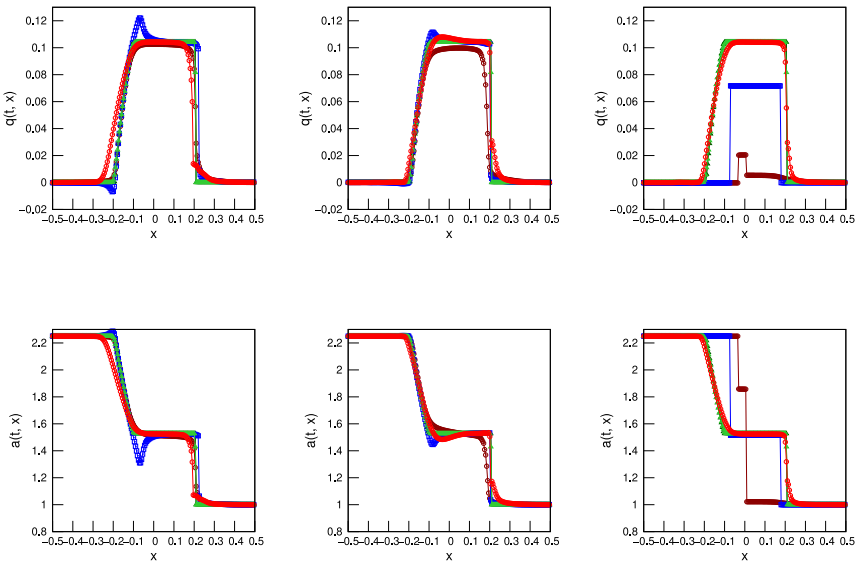


Fig. 23. Test case F4: Flow rate and area solution of the *vanilla* PINN (-□-), RoePINN with $\Delta x = 10^{-3}$ (-o-), RoePINN with $\Delta x = 10^{-2}$ (-o-), RoeFV with $\Delta x = 10^{-3}$ (-▵-), RoeFV with $\Delta x = 10^{-2}$ (-▵-), at time $t = 1.05$. First row corresponds to the flow rate solution with three different values for $n_{coll} = \{10^5, 10^4, 10^3\}$, from left to right, respectively. Second row corresponds to the solution for the area. The plotted discretization is $10 \times \Delta x$.

Both the vanilla PINN and the RoePINN perform worse than the RoeFV, as expected. This is also observed in Section 2 with Burgers' equation for the forward problem. In particular, the RoePINN prediction is diffusive at the discontinuities. Conversely, the vanilla PINN shows a less diffusive behavior at the discontinuities, but features abnormal oscillations at the left discontinuity edge corresponding to the inflection point of the rarefaction wave. Regarding the behavior of both PINNs with hyperparameter n_{coll}, we observe that for a low value of $n_{coll} = 10^3$, the RoePINN stays robust and yields a good approximation of the solution for $\Delta x = 10^{-2}$. However, the vanilla PINN completely fails for this value of n_{coll}. This behavior is analogous to the one in Test cases F1 and F3 with the scalar Burgers' equation, when we observed that increasing hyperparameter Δx significantly improved the accuracy of the RoePINN, while the vanilla PINN simply did not have this hyperparameter.

Fig. 23 (lower) shows the graphical representation of the cross-sectional area solution for Test case F4 at time $t = 0.5$ and different values of n_{coll} (from left to right) and Δx. We observe similar trends as for the prediction of q, but in general a better approximation of the reference solution by both PINNs.

Table 6 (upper) shows the L_2 error in the flow rate q for the different algorithms and combinations of hyperparameters Δx and n_{coll}. The error for the RoeFV method decreases with the reduction of the cell size, as expected. The RoePINN and the vanilla PINN predictions are less accurate than those of the RoeFV for any combination of n_{coll} and Δx. The lowest error among all is achieved by the RoePINN with $n_{coll} = 10^3$ and $\Delta x = 10^{-1}$ (blue), as in previous cases. Also, we observe that for low values of n_{coll}, the RoePINN outperforms the IVanPINN (green), as also depicted in Fig. 23.

Table 6 (lower) shows the L_2 error for the different algorithms and combinations of hyperparameters Δx and n_{coll}, for the cross-sectional area a. Focusing secondly on the prediction of a, the error for the RoeFV method also decreases with the reduction of the cell size, as expected. The RoePINN predictions and the vanilla PINN predictions are less accurate than those of the RoeFV for any combination of n_{coll} and Δx. Error metrics are significantly lower for the prediction of a, as compared with those for the prediction of q. Again, the lowest error among all is achieved by the RoePINN with $n_{coll} = 10^3$ and $\Delta x = 10^{-1}$ (blue), as in previous cases. Also, we observe that for low values of n_{coll}, the RoePINN outperforms the IVanPINN (green), as expected.

Fig. 24 shows the evolution of the RoePINN flow prediction L_2 error as r increases. An optimal value for the error is found at $r = 1.306$, following the same trend as in Test cases F1, F2 and F3 for Burgers' equation. The consistency of this observation when transitioning from Burgers' equation to a more complex system of equations describing blood flow inside a vessel, suggests that the RoePINN can be robustly calibrated across different data sampling regimes, in order to achieve optimal reproduction of sharp non-linearities in blood flow waves.

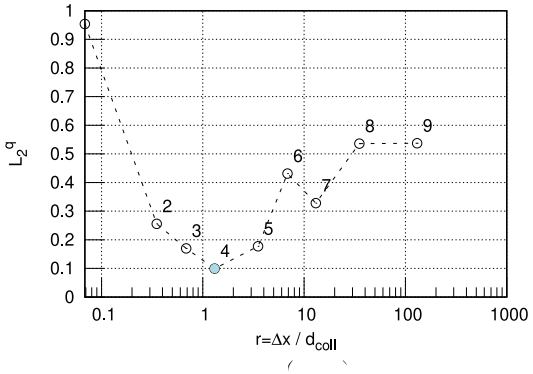
Fig. 25 shows the evolution of the RoePINN area prediction L_2 error as r increases. We observe that for a, the error curve follows the exact same trend and has optimal value of r as the curve of q, but L_2 errors are notably smaller.

Figs. 26 and 27 show the prediction of RoePINN in Test case F4 for different values of the *CFL* hyperparameter, the prediction of the RoeFV and the reference solution. As we observe in Test cases F1, F2 and F3 for the RoePINN solving the scalar Burgers' equation, values of $CFL \ll 1$ such as $CFL = 0.1$, and $CFL \gg 1$, such as $CFL = 5$ give the worst predictions, while predictions for intermediate values achieve much better accuracy.

Table 6
Test case F4: Comparison of L_2^q values for flow (*upper*) and L_2^a values for area (*lower*) for different algorithms and values of Δx and n_{coll}.

		L_2^q	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	RoePINN ($\Delta x = 10^{-3}$)	Vanilla PINN
10^5	–	0.099349	0.253584
10^4	–	0.255708	0.122692
10^3	–	0.953857	0.682354
–	0.014446	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	RoePINN ($\Delta x = 10^{-2}$)	Vanilla PINN
10^5	–	0.328101	0.247617
10^4	–	0.177196	0.127386
10^3	–	0.169894	0.682378
–	0.059321	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	RoePINN ($\Delta x = 10^{-1}$)	Vanilla PINN
10^5	–	0.537162	0.212450
10^4	–	0.536041	0.213234
10^3	–	0.431223	0.681689
–	0.198978	–	–

		L_2^a	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	RoePINN ($\Delta x = 10^{-3}$)	Vanilla PINN
10^5	–	0.013610	0.036873
10^4	–	0.033871	0.017956
10^3	–	0.135771	0.101924
–	0.001768	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	RoePINN ($\Delta x = 10^{-2}$)	Vanilla PINN
10^5	–	0.048980	0.036543
10^4	–	0.024865	0.019043
10^3	–	0.024271	0.101652
–	0.008244	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	RoePINN ($\Delta x = 10^{-1}$)	Vanilla PINN
10^5	–	0.078593	0.037699
10^4	–	0.079066	0.033970
10^3	–	0.061893	0.099029
–	0.028689	–	–

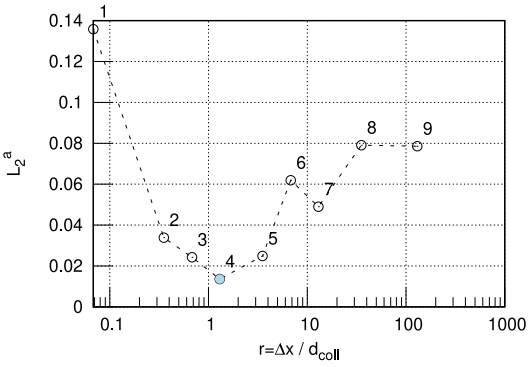


Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-3}	10^3	0.0146	0.068
2	10^{-3}	10^4	0.0028	0.351
4	$\mathbf{10^{-3}}$	10^5	**0.0008**	**1.306**
3	10^{-2}	10^3	0.0146	0.685
5	10^{-2}	10^4	0.0028	3.513
7	10^{-2}	10^5	0.0008	13.063
6	10^{-1}	10^3	0.0146	6.849
8	10^{-1}	10^4	0.0028	35.134
9	10^{-1}	10^5	0.0008	130.628

Fig. 24. Test case F4: (*Left*) Evolution of the RoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$ for the flow rate. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r for the flow rate.

5.2. Inverse test case I2. Unknown Gaussian stiffness

In this Test case we illustrate an inverse problem where, using data of the flow rate and the pressure dynamics along the vessel, we are able to infer the value of the stiffness K_o. This stiffness follows a gaussian distribution along the vessel's centerline. Simultaneously, we reconstruct the dynamics of the cross-sectional area of the vessel. The approach to solving this inverse problem is similar to the one used in Test Case I1 with Burgers' equation, but now more variables are involved.



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-3}	10^3	0.0146	0.068
2	10^{-3}	10^4	0.0028	0.351
4	$\mathbf{10^{-3}}$	10^5	**0.0008**	**1.306**
3	10^{-2}	10^3	0.0146	0.685
5	10^{-2}	10^4	0.0028	3.513
7	10^{-2}	10^5	0.0008	13.063
6	10^{-1}	10^3	0.0146	6.849
8	10^{-1}	10^4	0.0028	35.134
9	10^{-1}	10^5	0.0008	130.628

Fig. 25. Test case F4: (*Left*) Evolution of the RoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$ for the area. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r for the area.

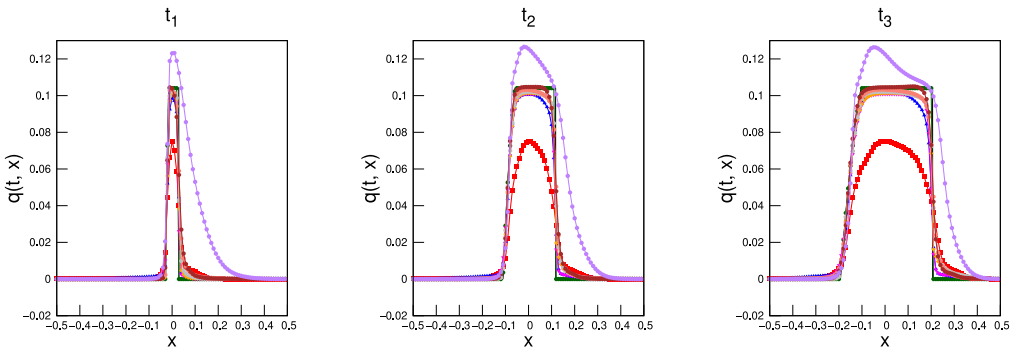


Fig. 26. Test case F4: Flow solution of the RoePINN with $\Delta x = 10^{-3}$ and $n_{coll} = 2 \times 10^5$ for different values of $CFL = \{0.1$ (-■-), 0.3 (-▲-), 0.5 (-▼-), 0.9 (-◆-), 1 (-●-), 1.5 (-●-), 2 (-●-), 5 (-●-)$\}$, RoeFV with $\Delta x = 10^{-3}$ (-●-) and the reference solution (—). Time snapshots are ordered from left to right ($t_1 = 5 \times 10^{-2} < t_2 = 0.275 < t_3 = 0.5$). The plotted discretization is $10 \times \Delta x$.

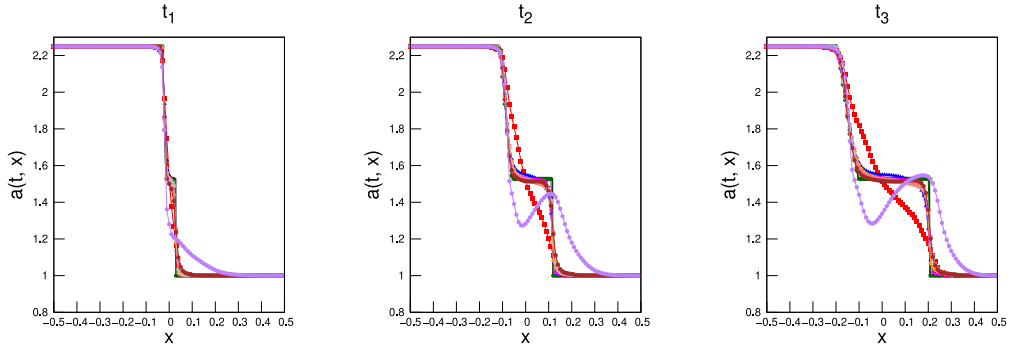


Fig. 27. Test case F4: Cross-sectional area reconstruction of the RoePINN with $\Delta x = 10^{-3}$ and $n_{coll} = 2 \times 10^5$ for different values of $CFL = \{0.1$ (-■-), 0.3 (-▲-), 0.5 (-▼-), 0.9 (-◆-), 1 (-●-), 1.5 (-●-), 2 (-●-), 5 (-●-)$\}$, RoeFV with $\Delta x = 10^{-3}$ (-●-) and the reference solution (—). Time snapshots are ordered from left to right ($t_1 = 5 \times 10^{-2} < t_2 = 0.275 < t_3 = 0.5$). The plotted discretization is $10 \times \Delta x$.

Training data $D_q : \{q_d(x_d, t_d)\}_{d=1}^{n_d}$ and $D_p : \{p_d(x_d, t_d)\}_{d=1}^{n_d}$ is given in the form of snapshots that are equally distributed across the computational domain, while $a(x,t)$ and $K_o(x,t)$ are inferred. Details about the hyperparameters and the number of snapshots are given in Appendix D 8.4.

Fig. 28 (upper) shows the graphical representation of the reconstruction for the area for Test case I2 at time $t = 0.5$ and different number of collocation points n_{coll} (from left to right) and Δx. Fig. 28 (lower) shows the graphical representation of the reconstruction of the stiffness K_o for Test case I2 at time $t = 0.5$ and different number of collocation points n_{coll} (from left to right) and Δx. We observe that both the IVanPINN and the IRoePINN with $\Delta x = 10^{-3}$ are visually as accurate as the RoeFV, but for $\Delta x = 10^{-2}$, the IRoePINN shows less prediction accuracy specially in the multiple peaks. Regarding the change with n_{coll}, the area is identical for the three values of n_{coll}, and the stiffness shows some small differences with n_{coll}.

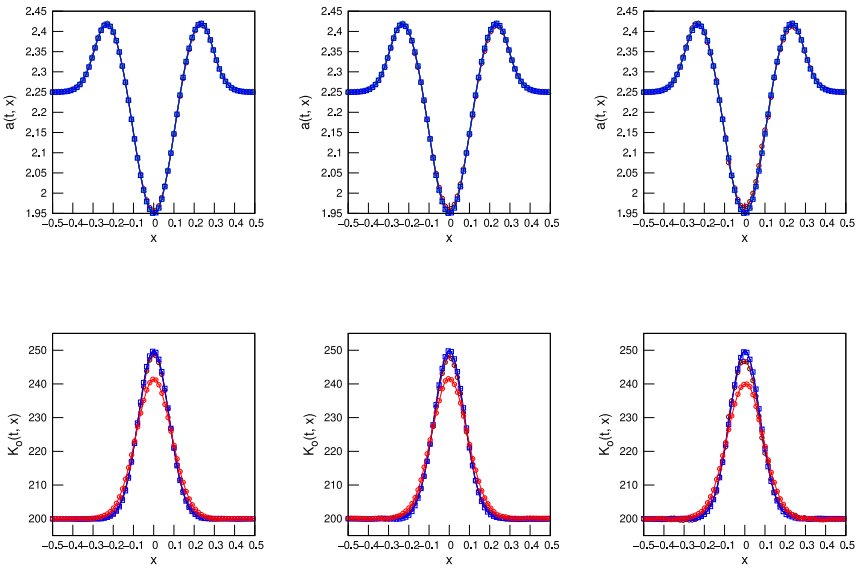


Fig. 28. Test case I2: Reconstructed area and stiffness by IVanPINN (-□-), IRoePINN with $\Delta x = 10^{-3}$ (-o-), IRoePINN with $\Delta x = 10^{-2}$ (-o-), RoeFV with $\Delta x = 10^{-2}$ (-△-), and the reference solution (—), at time $t = 1.05$. First row corresponds to the area solution with three different values for $n_{coll} = \{10^5, 10^4, 10^3\}$, from left to right, respectively. Second row corresponds to the reconstruction of the stiffness, and only the IVanPINN (-□-), IRoePINN with $\Delta x = 10^{-3}$ (-o-), IRoePINN with $\Delta x = 10^{-2}$ (-o-) and the reference solution (—) have been plotted, because the RoeFV did not reconstructed the stiffness (it was only used as a forward solver for comparison). The plotted discretization is $30 \times \Delta x$.).

The RoePINN and vanilla PINN predictions of the flow rate solution are visually identical to the RoeFV and reference solution, because we provide flow rate data, hence these are not show for the sake of brevity. Moreover, since the pressure $p(x,t)$ can be computed using variables a and K_o, plots of the pressure have also been omitted for brevity. In particular, fitting accuracy is important to ensure that the simultaneous reconstruction of $a(x,t)$ and $K_o(x,t)$ is as accurate as possible. Key factors that enhance the accuracy of data fitting include the increase of depth of the neural network (9 layers in our case, which is relatively deep), assigning higher weights l_i in the loss function to prioritize minimizing specific loss terms related to data fitting, and using an exponentially decaying learning rate schedule, which aids in achieving better minimization for each loss term and overall. Detailed information about these parameters can be found in Appendix D 8.4.

Table 7 (upper) shows the L_2 error in the fitting of the flow rate q for the different algorithms and combinations of hyperparameters Δx and n_{coll}. The error for the RoeFV method decreases with the reduction of the cell size, as expected. The IRoePINN predictions and the IVanPINN predictions are approximately as accurate as those of the RoeFV for all combinations of n_{coll} and Δx. No considerable change of the error is observed if varying the value of n_{coll}. With regards to Δx, the error increases one order of magnitude with $\Delta x = 10^{-1}$ for the RoePINN. Comparing the IRoePINN and the IVanPINN for lower values of Δx, the IRoePINN shows lower errors.

Table 7 (lower) shows the L_2 error for the different algorithms and combinations of hyperparameters Δx and n_{coll}, for the reconstruction of the cross-sectional area a. We observe a similar trend as for the fitting of the flow rate q. However, in this case, the IVanPINN shows lower errors than the IRoePINN in general.

Table 8 shows the L_2 error for both PINNs and combinations of hyperparameters Δx and n_{coll}, for the reconstruction of the stiffness K_o. Error increases with Δx, specially for $\Delta x = 10^{-1}$. Comparing the IRoePINN and the IVanPINN, both show similar error metrics, but the IRoePINN yields an error that is an order of magnitude higher for $\Delta x = 10^{-1}$.

Now we analyze the performance of the IRoePINN with the ratio $r = \frac{\Delta x}{d_{coll}}$ of the selected spatial discretization Δx and d_{coll}, for the flow rate q. Fig. 29 shows the evolution of the IRoePINN flow prediction L_2^q error as r increases. An optimal value for the error is found at point 3 with $r = 0.685$. We also observe some low errors for points 5 and 7. The observed trend of the error curve resembles that of previous cases because we find a minimum close to $r = 1$, but the behavior is very oscillating. Again, this is due to the inverse data-driven character of the problem.

Fig. 30 (left) shows the evolution of the IRoePINN area prediction L_2^a error as r increases. We find an optimal value for the error found at point 4 with $r = 0.685$. Fig. 30 (right) shows the evolution of the IRoePINN stiffness reconstruction $L_2^{K_o}$ error as r increases. We observe a similar trend as for L_2^a, with an optimal value for the error found at point 4 with $r = 0.685$.

5.3. Inverse test case I3. Unknown piecewise stiffness

In this Test case we illustrate an inverse problem where, given data on the flow rate and the pressure dynamics along the vessel, we are able to infer the value of the stiffness K_o, which follows a piece-wise distribution along the vessel's centerline, and also

Table 7
Comparison of L_2^q values for flow (*upper*) and L_2^a values for area (*lower*) for different algorithms and values of Δx and n_{coll} in Test case I2.

		L_2^q	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	IRoePINN ($\Delta x = 10^{-3}$)	IVanPINN
10^5	–	0.009097	0.009707
10^4	–	0.010559	0.010389
10^3	–	0.009347	0.009704
–	0.00045	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	IRoePINN ($\Delta x = 10^{-2}$)	IVanPINN
10^5	–	0.004713	0.006177
10^4	–	0.005324	0.006118
10^3	–	0.004447	0.007244
–	0.004943	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	IRoePINN ($\Delta x = 10^{-1}$)	IVanPINN
10^5	–	0.077644	0.077170
10^4	–	0.076759	0.076296
10^3	–	0.076959	0.078262
–	0.049676	–	–

		L_2^a	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	IRoePINN ($\Delta x = 10^{-3}$)	IVanPINN
10^5	–	0.000672	0.000356
10^4	–	0.001188	0.000521
10^3	–	0.001551	0.000419
–	0.000014	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	IRoePINN ($\Delta x = 10^{-2}$)	IVanPINN
10^5	–	0.004105	0.000250
10^4	–	0.004133	0.000262
10^3	–	0.004718	0.000212
–	0.000154	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	IRoePINN ($\Delta x = 10^{-1}$)	IVanPINN
10^5	–	0.023515	0.003738
10^4	–	0.023592	0.003548
10^3	–	0.023610	0.003646
–	0.001535	–	–

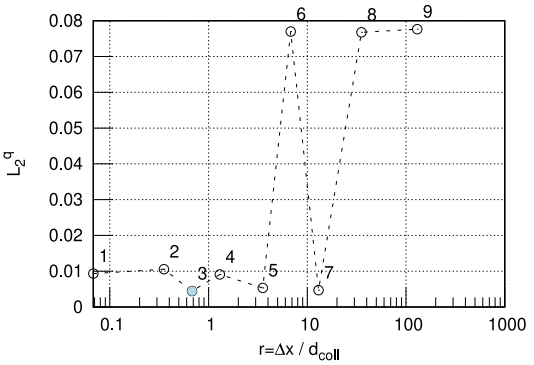
Table 8
Test case I2: Comparison of L_2 errors for the different algorithms and values of Δx and n_{coll} for the stiffness.

	$L_2^{K_o}$ Error	
n_{coll}	IRoePINN ($\Delta x = 10^{-3}$)	IVanPINN
10^5	0.001126	0.006800
10^4	0.002099	0.001366
10^3	0.002787	0.002716
n_{coll}	IRoePINN ($\Delta x = 10^{-2}$)	IVanPINN
10^5	0.007864	0.006784
10^4	0.007765	0.001141
10^3	0.009342	0.002642
n_{coll}	IRoePINN ($\Delta x = 10^{-1}$)	IVanPINN
10^5	0.042216	0.008822
10^4	0.042403	0.005409
10^3	0.042341	0.006060

reconstruct the dynamics of the cross-sectional area of the vessel. In particular, for the sake of consistency, we choose a function for K_o that closely resembles the source term $z(x)$ in Test cases F3 and I1.

Training data $\mathcal{D}_q : \{q_d(x_d, t_d)\}_{d=1}^{n_d}$ and $\mathcal{D}_p : \{p_d(x_d, t_d)\}_{d=1}^{n_d}$ is given in the form of snapshots that are equally distributed across the computational domain, while $a(x,t)$ and $K_o(x,t)$ are inferred. Details about the hyperparameters and the number of snapshots are given in Appendix D 8.4.

Fig. 31 (upper) shows the graphical representation of the reconstruction for the area for the Test case I3 at time $t = 1.05$ and different number of collocation points n_{coll} (from left to right) and Δx. We observe that both the IVanPINN and the IRoePINN with $\Delta x = 10^{-3}$ are less accurate than the RoeFV, but for $\Delta x = 10^{-2}$, the IRoePINN shows less prediction accuracy specially in the sharp



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-3}	10^3	0.0146	0.068
2	10^{-3}	10^4	0.0028	0.351
4	10^{-3}	10^5	0.0008	1.306
3	$\mathbf{10^{-2}}$	10^3	**0.0146**	**0.685**
5	10^{-2}	10^4	0.0028	3.513
7	10^{-2}	10^5	0.0008	13.063
6	10^{-1}	10^3	0.0146	6.849
8	10^{-1}	10^4	0.0028	35.134
9	10^{-1}	10^5	0.0008	130.628

Fig. 29. Test case I2: (*Left*) Evolution of the IRoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$ for the flow rate. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r for the flow rate.

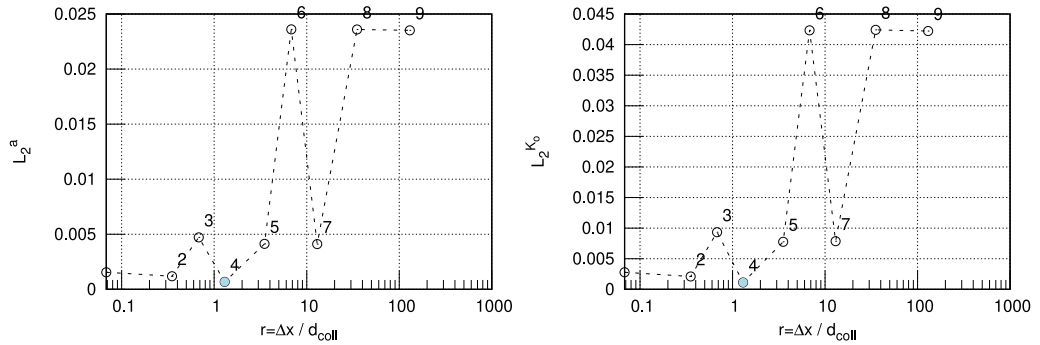


Fig. 30. Test case I2: Evolution of the IRoePINN L_2 error for the area (*Left*) and for the stiffness (*Right*). Blue dot indicates the r with lowest L_2 error.

peaks. For $n_{coll} = 10^3$, we observe an oscillatory behavior of the IVanPINN. As observed in Test cases F1, F3 and F4, this occurs because the IRoePINN is more robust for low values of n_{coll}.

Fig. 31 (lower) shows the graphical representation of the solution for the Test case I3 at time $t = 0.5$ and different number of collocation points n_{coll} (from left to right), for different values of Δx. Again, we depict the reconstruction of stiffness at three different times to emphasize that the solution should remain constant over time, and corroborate that the PINN model inherently learns this independence from t. We observe that both the IVanPINN and the IRoePINN with $\Delta x = 10^{-3}$ are visually as accurate as the RoeFV, but for $\Delta x = 10^{-2}$, the IRoePINN shows less prediction accuracy specially in the multiple peaks. We observe that both the IVanPINN and the IRoePINN with $\Delta x = 10^{-3}$ struggle in the central part of the stiffens curve, where there two discontinuities are. In particular, the IVanPINN shows better accuracy in these regions. As observed in Test case I2, $\Delta x = 10^{-2}$, the IRoePINN shows less prediction accuracy specially in the multiple peaks.

As in Test case I2, the RoePINN and vanilla PINN predictions of the flow rate solution are visually identical to the RoeFV and reference solution, because we provide flow rate data, hence these are not show for the sake of brevity. Moreover, since the pressure $p(x,t)$ can be computed using variables a and K_o, plots of the pressure have also been omitted for brevity.

Table 9 (upper) shows the L_2 error for the different algorithms and combinations of hyperparameters Δx and n_{coll}, for the fitting of the flow rate q. The error for the RoeFV method decreases with the reduction of the cell size, as expected. The IRoePINN predictions and the IVanPINN predictions are approximately as accurate as those of the RoeFV for any combination of n_{coll} and Δx. No considerable change of the error is observed if varying the value of n_{coll}. With regards to Δx, the error increases one order of magnitude with $\Delta x = 10^{-1}$ for the RoePINN. Comparing the IRoePINN and the IVanPINN, the IVanPINN shows lower error in general.

Table 9 (lower) shows the L_2 error for the different algorithms and combinations of hyperparameters Δx and n_{coll}, for the reconstruction of the cross-sectional area a. We observe a similar trend as for the fitting of the flow rate q.

Table 10 shows the L_2 error for both PINNs and combinations of hyperparameters Δx and n_{coll}, for the reconstruction of the stiffness K_o. For the IRoePINN, we do not observe any considerable change of the error n_{coll}, but when looking at the variation of the error with Δx, this increases for $\Delta x = 10^{-1}$. Comparing the IRoePINN and the IVanPINN, both show similar error metrics, but the IRoePINN yields an error that is an order of magnitude higher for $\Delta x = 10^{-1}$.

Now we analyze the performance of the IRoePINN with the ratio $r = \frac{\Delta x}{d_{coll}}$ of the selected spatial discretization Δx and d_{coll}, for the flow rate q. Fig. 32 shows the evolution of the IRoePINN prediction error, L_2^q, as r increases. We observe that the trend is similar to Test case I2. An optimal value for the error is found at point 3 with $r = 0.685$. We also observe some low errors for points 2

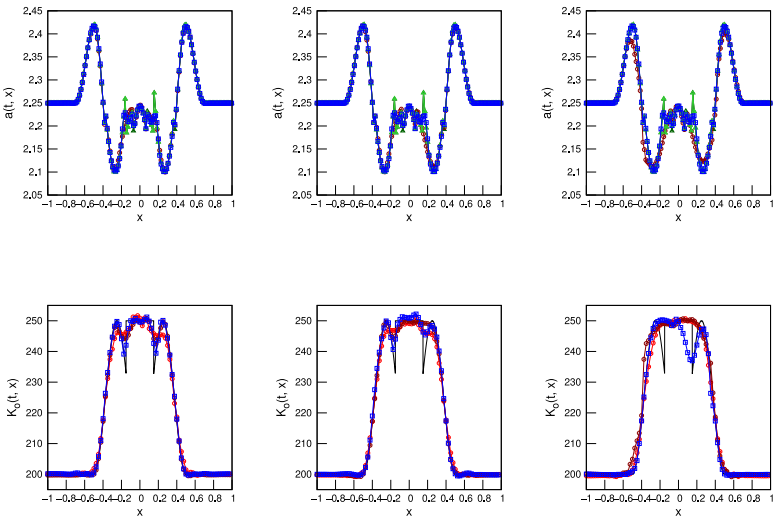
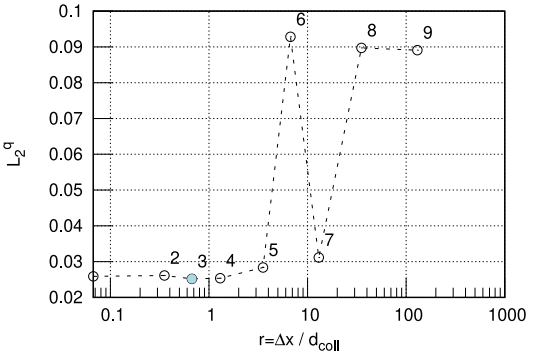


Fig. 31. Test case I3: Reconstructed area and stiffness by IVanPINN (-□-), IRoePINN with $\Delta x = 10^{-3}$ (-○-), IRoePINN with $\Delta x = 10^{-2}$ (-○-), RoeFV with $\Delta x = 10^{-2}$ (-△-), and the reference solution (—), at time $t = 1.05$. First row corresponds to the area reconstruction with three different values for $n_{coll} = \{10^5, 10^4, 10^3\}$, from left to right, respectively. Second row corresponds to the reconstruction of the stiffness, and only the IVanPINN (-□-), IRoePINN with $\Delta x = 10^{-3}$ (-○-), IRoePINN with $\Delta x = 10^{-2}$ (-○-) and the reference solution (—) have been plotted, because the RoeFV did not reconstructed the stiffness (it was only used as a forward solver for comparison). The plotted discretization is $30 \times \Delta x$.



Point	Δx	n_{coll}	d_{coll}	$r = \frac{\Delta x}{d_{coll}}$
1	10^{-3}	10^3	0.0146	0.068
2	10^{-3}	10^4	0.0028	0.351
4	10^{-3}	10^5	0.0008	1.306
3	$\mathbf{10^{-2}}$	10^5	**0.0146**	**0.685**
5	10^{-2}	10^4	0.0028	3.513
7	10^{-2}	10^5	0.0008	13.063
6	10^{-1}	10^3	0.0146	6.849
8	10^{-1}	10^4	0.0028	35.134
9	10^{-1}	10^5	0.0008	130.628

Fig. 32. Test case I3: (*Left*) Evolution of the IRoePINN L_2 error against $r = \frac{\Delta x}{d_{coll}}$ for the flow rate. Blue dot indicates the r with lowest L_2 error. (*Right*) Table of hyperparameters for the computation of r.

and 4. The observed trend of the error curve resembles that of previous cases because we find a minimum close to $r = 1$, but the behavior is very oscillating. Again, this is due to the inverse data-driven character of the problem.

Fig. 33 (left) shows the evolution of the IRoePINN area prediction L_2^a error as r increases, for a. In this case, we observe that for the area prediction, the curve has the same tendency as for q, but the errors has a more pronounced minimum at point 4. Fig. 33 (right) shows the evolution of the IRoePINN prediction $L_2^{K_o}$ error as r increases. In this case, we observe that for the area prediction, the curve has the same tendency as for L_2^q, but, as in the case of L_2^a, the errors has a more pronounced minimum at point 4.

6. Conclusions

In this work, we propose a new type of PINN, the *RoePINN*, where the PDE loss is entirely based on an advanced numerical method, the Roe Finite Volume method (RoeFV), replacing traditional automatic differentiation. Throughout the work, we compare the RoePINN performance against the *vanilla* algorithm as well as the RoeFV numerical scheme, for forward and inverse problems in Burgers' equation with geometrical source terms and in a 1D blood flow model for arteries with space-varying wall stiffness. We show that replacing traditional automatic differentiation with a numerical scheme provides improved predictive accuracy and robustness when tackling source term and solution discontinuities.

Over the past few years, several works in the state-of-the-art have highlighted the challenges faced by PINNs in handling discontinuities [17]. Efforts to mitigate these issues include clustering collocation points around the discontinuity [18], enforcing

Table 9
Test case I3: Comparison of L_2^q values for flow (*upper*) and L_2^a values for area (*lower*) for different algorithms and values of Δx and n_{coll}.

		L_2^q	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	IRoePINN ($\Delta x = 10^{-3}$)	IVanPINN
10^5	–	0.025356	0.027580
10^4	–	0.026150	0.028737
10^3	–	0.025880	0.031678
–	0.001305	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	IRoePINN ($\Delta x = 10^{-2}$)	IVanPINN
10^5	–	0.031148	0.027519
10^4	–	0.016495	0.022139
10^3	–	0.028364	0.030695
–	0.023907	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	IRoePINN ($\Delta x = 10^{-1}$)	IVanPINN
10^5	–	0.089086	0.097440
10^4	–	0.089729	096 525
10^3	–	0.092833	0.098207
–	0.073818	–	–

		L_2^a	
n_{coll}	RoeFV ($\Delta x = 10^{-3}$)	IRoePINN ($\Delta x = 10^{-3}$)	IVanPINN
10^5	–	0.003528	0.002952
10^4	–	0.005291	0.003751
10^3	–	0.007299	0.007529
–	0.000207	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-2}$)	IRoePINN ($\Delta x = 10^{-2}$)	IVanPINN
10^5	–	0.004912	0.002945
10^4	–	0.005135	0.003754
10^3	–	0.007106	0.007534
–	0.000648	–	–
n_{coll}	RoeFV ($\Delta x = 10^{-1}$)	IRoePINN ($\Delta x = 10^{-1}$)	IVanPINN
10^5	–	0.082746	0.071074
10^4	–	0.088110	0.088742
10^3	–	0.123453	0.110607
–	0.039705	–	–

Table 10
Test case I3: Comparison of $L_2^{K_o}$ errors for the different algorithms and values of Δx and n_{coll} for the stiffness.

	$L_2^{K_o}$ Error	
n_{coll}	IRoePINN ($\Delta x = 10^{-3}$)	IVanPINN
10^5	0.005784	0.004707
10^4	0.008668	0.006047
10^3	0.011711	0.013396
n_{coll}	IRoePINN ($\Delta x = 10^{-2}$)	IVanPINN
10^5	0.008187	0.004696
10^4	0.008434	0.006035
10^3	0.011503	0.013422
n_{coll}	IRoePINN ($\Delta x = 10^{-1}$)	IVanPINN
10^5	0.024367	0.006194
10^4	0.024899	0.007252
10^3	0.023937	0.013864

entropy constraints [19], and similar approaches. However, none of these works have proposed a numerical method as an strategy to ensure stability and robustness of the predicted solution. CAN-PINNs represent the most notable attempt to adopt a hybrid formulation of the PDE loss, offering advantages in terms of robustness under low-sampling regimes of collocation points. However, to the best of our knowledge, no existing work to date incorporates both time and space discretization as well as the complete numerical scheme within the PDE loss. While our approach shares certain limitations with vanilla PINNs and more advanced PINNs trained using automatic differentiation, such as the need to fine-tune a notable number of hyperparameters, it also produces, in general, more accurate predictions in the presence of source term discontinues, and an improved robustness under low-sampling of collocation points.

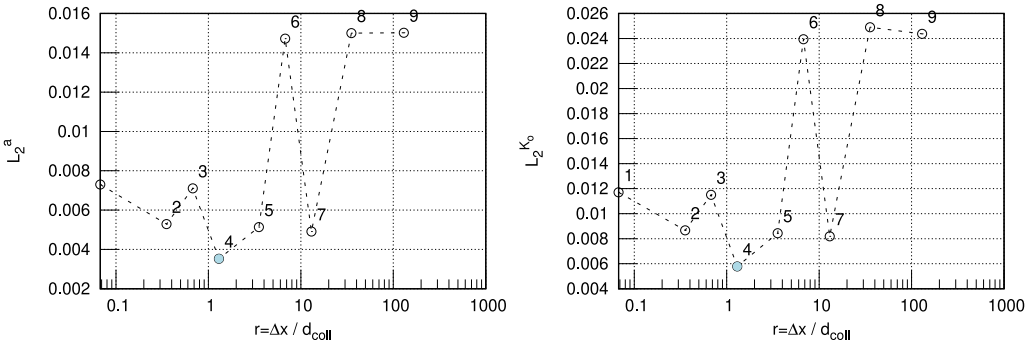


Fig. 33. Test case I3: Evolution of the IRoePINN L_2 error for the area (*Left*) and for the stiffness (*Right*). Blue dot indicates the r with lowest L_2 error.

Methodologically, we observe that the fashion in which the RoeFV method was embedded in the PINN structure can be extrapolated to any other method based on the discretized version of a physical model, not only within the scope of CFD solvers. As a consequence of the dual nature of the RoePINN, specific constraints and features are integrated in the PDE loss, such as the CFL condition and the choice of the space and time discretization. In the realm of Riemann problems, we focus on the ability of the RoePINN to capture sharp discontinuities as compared with vanilla PINNs. We find that, under an appropriate choice of the ratio between the spatial discretization and a characteristic mean distance between collocation points, the RoePINN clearly excels in both forward and inverse scenarios. The trade-off between the number of collocation points and the spatial discretization is a key feature for embedding the advanced numerical solver and proves crucial for calibrating the RoePINN. This balance allows for fine-tuning the model to maintain both its robustness and generalization capacity, even in scenarios with coarse sampling of collocation points (Test cases F1, F2, F3, F4) and missing data (Test cases I1, I2, I3).

The possibility to tune Δx given a predefined value of n_{coll} represents an important advantage of the RoePINN as compared with the vanilla PINN, where Δx does not exist hence one has to blindly trust automatic differentiation. On the contrary, RoePINNs offer the possibility to fine-tune the computation of the derivatives involved in the PDE in order to achieve an optimal trade-off between maximum accuracy and robustness. We study this dependence in a sensitivity analysis.

In this analysis, optimizing r helps ensure that the collocation points are neither too sparse nor too dense relative to the numerical discretization of the problem. We conclude that, in all cases and for forward problems without domain data, the RoePINN error is minimal when r lies in the vicinity of 1. In general, for large values of Δx, it is clear that the RoePINN significantly outperforms the vanilla PINN by at least one order of magnitude, which corroborates the importance of the existence of hyperparameter Δx to fine-tune the RoePINN in scenarios where vanilla PINNs seems to offer no other alternative to significantly improve the accuracy of the predicted solution.

Furthermore, the incorporation of the numerical method in the loss function enables the monitoring of intermediate variables and the enforcement of physical constrains on those intermediate variables, such as the transport velocity λ, that otherwise would completely go unnoticed, and that play a relevant role specially in ill-posed problems such as model discovery.

Moreover, the performance of RoePINN with the CFL hyperparameter exhibits similarities to the RoeFV, but this condition is relaxed in the RoePINN because for values of $CFL \gg 1$, the RoePINN consistently produced stable predictions.

For smooth solutions, both RoePINN and vanilla PINN tend to perform at a comparable level. However, based on the collected evidence, we can conclude that the RoePINN distinctly excels the vanilla PINN when abrupt discontinuities such as jumps, shocks or rarefactions appear.

Throughout the paper, we observe the superiority of numerical solvers over PINNs as forward solvers. Nevertheless, in the last two sections we show that both vanilla PINNs and RoePINNs have great potential for solving challenging inverse problems, particularly in the realm of blood flow modeling and vascular parameter calibration. To illustrate this, a biomedical application of PINNs is showcased. Importantly, vessel stiffness is an indicator of vascular aging and therefore a potential predictor of cardiovascular risk. However, its estimation for simulation and diagnosis purposes poses special difficulties, as this parameter cannot be directly measured using non-surgical procedures, even more so when the mechanical properties of the wall vary across vessel's length. Therefore, we propose RoePINNs as an optimal algorithm to assimilate clinical data into the 1D blood flow model such that, as a result, the stiffness model of the vessel $K_o(x)$ is retrieved by feeding the RoePINN with flow $Q(x,t)$ and pressure $p(x,t)$ domain data, while the cross-sectional area $A(x,t)$ is also reconstructed.

Neural networks are known for their capacity to filter the noise in the data (avoid overfitting) while maintaining robust predictions (avoid underfitting) [53]. In the case of PINNs, the learning process is structured to balance data fitting with the fulfillment of physical equations [54]. One way of dealing with noisy data may be to reduce the Lagrange multipliers associated with data losses and increase those corresponding to PDE losses. Another possible approach may be to increase the density of collocation points, further guiding the optimization process toward the enforcement of the physical constraints rather than overfitting noisy measurements. When comparing RoePINNs and vanilla PINNs, a similar behavior is expected, as both methods involve a comparable loss balancing mechanism, even though the PDE loss computation differs. Future work should evaluate the robustness of our results in the presence of noise.

Other future lines of work will include the reconstruction of the space-dependent vessel stiffness using RoePINNs with real patient's data, the recovery of the viscoelastic properties of the vessels wall, as well as the extension of the RoePINN methodology to junctions of vessels and complex vascular networks.

CRediT authorship contribution statement

J. Orera: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **J. Ramírez:** Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **P. García-Navarro:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **J. Murillo:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by project PID2023-150074NB-I00 funded by the MICIU/AEI/10.13039/501100011033/ and FEDER, UE. JR acknowledges funding from grants PID2021-128972OA-100 and CNS2023-143599, fellowship RYC2021-031413-I from Spanish Ministry of Science and Innovation and project PID2023-148975OB-I00.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cma.2025.117933.

Data availability

Data will be made available on request.

References

[1] J. Murillo, P. García-Navarro, A solution of the junction Riemann problem for 1d hyperbolic balance laws in networks including supersonic flow conditions on elastic collapsible tubes, Symmetry 13 (9) (2021) 1658.
[2] P. García-Navarro, J. Murillo, J. Fernández-Pato, I. Echeverribar, M. Morales-Hernández, The shallow water equations and their application to realistic cases, Environ. Fluid Mech. 19 (2019) 1235–1252.
[3] C. Pozrikidis, Fluid Dynamics: Theory, Computation, and Numerical Simulation, Springer, 2016.
[4] T.J. Chung, Computational Fluid Dynamics, Cambridge University Press, 2002.
[5] J.H. Ferziger, M. Perić, R.L. Street, Computational Methods for Fluid Dynamics, springer, 2019.
[6] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, A reduced computational and geometrical framework for inverse problems in hemodynamics, Int. J. Numer. Methods Biomed. Eng. 29 (7) (2013) 741–776.
[7] G.K. Kenway, C.A. Mader, P. He, J.R. Martins, Effective adjoint approaches for computational fluid dynamics, Prog. Aerosp. Sci. 110 (2019) 100542.
[8] M. Raissi, Z. Wang, M.S. Triantafyllou, G.E. Karniadakis, Deep learning of vortex-induced vibrations, J. Fluid Mech. 861 (2019) 119–137.
[9] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[10] N. McGreivy, A. Hakim, Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations, Nat. Mach. Intell. (2024) 1–14.
[11] J. Orera-Echeverria, J. Ayensa-Jiménez, M. Doblare, Predicting and explaining nonlinear material response using deep physically guided neural networks with internal variables, 2023, arXiv preprint arXiv:2308.03915.
[12] J. Murillo, P. García-Navarro, Weak solutions for partial differential equations with source terms: Application to the shallow water equations, J. Comput. Phys. 229 (11) (2010) 4327–4368.
[13] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018) 1–43.
[14] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review, Acta Mech. Sin. 37 (12) (2021) 1727–1738.
[15] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, Handb. Numer. Anal. 7 (2000) 713–1018.
[16] M. Flaschel, S. Kumar, L. De Lorenzis, Discovering plasticity models without stress data, Npj Comput. Mater. 8 (1) (2022) 91.
[17] X. Zhang, T. Cheng, L. Ju, Implicit form neural network for learning scalar hyperbolic conservation laws, in: Mathematical and Scientific Machine Learning, PMLR, 2022, pp. 1082–1098.
[18] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, Comput. Methods Appl. Mech. Engrg. 360 (2020) 112789.
[19] R.G. Patel, I. Manickam, N.A. Trask, M.A. Wood, M. Lee, I. Tomas, E.C. Cyr, Thermodynamically consistent physics-informed neural networks for hyperbolic systems, J. Comput. Phys. 449 (2022) 110754.
[20] P.-H. Chiu, J.C. Wong, C. Ooi, M.H. Dao, Y.-S. Ong, CAN-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method, Comput. Methods Appl. Mech. Engrg. 395 (2022) 114909.

[21] A.D. Jagtap, G.E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, in: AAAI Spring Symposium: MLPS, vol. 10, 2021.
[22] F.S. Costabal, S. Pezzuto, P. Perdikaris, Δ-PINNs: Physics-informed neural networks on complex geometries, Eng. Appl. Artif. Intell. 127 (2024) 107324.
[23] C. Herrero Martin, A. Oved, R.A. Chowdhury, E. Ullmann, N.S. Peters, A.A. Bharath, M. Varela, EP-PINNs: Cardiac electrophysiology characterisation using physics-informed neural networks, Front. Cardiovasc. Med. 8 (2022) 768419.
[24] G. Pang, L. Lu, G.E. Karniadakis, fPINNs: Fractional physics-informed neural networks, SIAM J. Sci. Comput. 41 (4) (2019) A2603–A2626.
[25] W. Zhang, J. Li, CPINNs: A coupled physics-informed neural networks for the closed-loop geothermal system, Comput. Math. Appl. 132 (2023) 161–179.
[26] E. Kharazmi, Z. Zhang, G.E. Karniadakis, hp-VPINNs: Variational physics-informed neural networks with domain decomposition, Comput. Methods Appl. Mech. Engrg. 374 (2021) 113547.
[27] E. Riccietti, V. Mercier, S. Gratton, P. Boudier, Multilevel physics informed neural networks (MPINNs), 2021.
[28] M. Mahmoudabadbozchelou, G.E. Karniadakis, S. Jamali, nn-PINNs: Non-Newtonian physics-informed neural networks for complex fluid modeling, Soft Matter 18 (1) (2022) 172–185.
[29] K. Tang, X. Wan, C. Yang, DAS-PINNs: A deep adaptive sampling method for solving high-dimensional partial differential equations, J. Comput. Phys. 476 (2023) 111868.
[30] R.J. Gladstone, M.A. Nabian, H. Meidani, FO-PINNs: A first-order formulation for physics informed neural networks, 2022, arXiv preprint arXiv:2210.14320.
[31] R. Mojgani, M. Balajewicz, P. Hassanzadeh, Lagrangian pinns: A causality-conforming solution to failure modes of physics-informed neural networks, 2022, arXiv preprint arXiv:2205.02902.
[32] B. Lütjens, C.H. Crawford, M. Veillette, D. Newman, Pce-pinns: Physics-informed neural networks for uncertainty propagation in ocean modeling, 2021, arXiv preprint arXiv:2105.02939.
[33] Y. Gao, K.C. Cheung, M.K. Ng, SVD-PINNs: Transfer learning of physics-informed neural networks via singular value decomposition, in: 2022 IEEE Symposium Series on Computational Intelligence, SSCI, IEEE, 2022, pp. 1443–1450.
[34] H. Son, S.W. Cho, H.J. Hwang, Al-pinns: Augmented lagrangian relaxation method for physics-informed neural networks, 2022, arXiv preprint arXiv:2205.01059.
[35] K. Teferra, Sc-pinns: Integrating the stochastic collocation method with physics informed neural networks for uncertainty propagation, 2022, Available at SSRN 4339863.
[36] T. Yu, H. Yong, L. Liu, et al., MCMC-PINNs: A modified Markov chain Monte-Carlo method for sampling collocation points of PINNs adaptively, 2023.
[37] H. Barucq, H. Calandra, T. Maron, UW-PINN: Ultra-weak PINNs for the acoustic wave system, 2023.
[38] H. Son, J.W. Jang, W.J. Han, H.J. Hwang, Sobolev training for physics informed neural networks, 2021, arXiv preprint arXiv:2101.08932.
[39] A. Navas-Montilla, J. Murillo, Energy balanced numerical schemes with very high order. The augmented roe flux ADER scheme. Application to the shallow water equations, J. Comput. Phys. 290 (2015) 188–218.
[40] J. Murillo, A. Navas-Montilla, A comprehensive explanation and exercise of the source terms in hyperbolic systems using roe type solutions. Application to the 1D-2D shallow water equations, Adv. Water Resour. (ISSN: 0309-1708) 98 (2016) 70–96, http://dx.doi.org/10.1016/j.advwatres.2016.10.019, URL https://www.sciencedirect.com/science/article/pii/S0309170816305917.
[41] E.F. Toro, Godunov Methods: Theory and Applications, Springer Science & Business Media, 2012.
[42] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction, Springer Science & Business Media, 2013.
[43] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.
[44] S. Wang, S. Sankaran, H. Wang, P. Perdikaris, An expert's guide to training physics-informed neural networks, 2023, arXiv preprint arXiv:2308.08468.
[45] F.M. Rohrhofer, S. Posch, B.C. Geiger, On the pareto front of physics-informed neural networks, 2021, arXiv preprint arXiv:2105.00862.
[46] R. Rojas, R. Rojas, The backpropagation algorithm, in: Neural Networks: A Systematic Introduction, Springer, 1996, pp. 149–182.
[47] T. Beck, H. Fischer, The if-problem in automatic differentiation, J. Comput. Appl. Math. 50 (1–3) (1994) 119–131.
[48] F.F. de la Mata, A. Gijón, M. Molina-Solana, J. Gómez-Romero, Physics-informed neural networks for data-driven simulation: Advantages, limitations, and opportunities, Phys. A 610 (2023) 128415.
[49] J. Murillo, A. Navas-Montilla, P. García-Navarro, Formulation of exactly balanced solvers for blood flow in elastic vessels and their application to collapsed states, Comput. & Fluids 186 (2019) 74–98.
[50] J. Murillo, P. García-Navarro, Numerical coupling of 0D and 1D models in networks of vessels including transonic flow conditions. Application to short-term transient and stationary hemodynamic simulation of postural changes, Int. J. Numer. Methods Biomed. Eng. (2023) e3751.
[51] G. Kissas, Y. Yang, E. Hwuang, W.R. Witschey, J.A. Detre, P. Perdikaris, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 358 (2020) 112623.
[52] E. Pimentel-García, L.O. Müller, E.F. Toro, C. Parés, High-order fully well-balanced numerical methods for one-dimensional blood flow with discontinuous properties, J. Comput. Phys. 475 (2023) 111869.
[53] F. Regazzoni, S. Pagani, A. Cosenza, A. Lombardi, A. Quarteroni, A physics-informed multi-fidelity approach for the estimation of differential equations parameters in low-data or large-noise regimes, Rend. Lincei 32 (3) (2021) 437–470.
[54] L. Yang, X. Meng, G.E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, J. Comput. Phys. 425 (2021) 109913.