

Fluid-Structure Interaction Analysis of Pulsatile Flow in Arterial Aneurysms with Physics-Informed Neural Networks and Computational Fluid Dynamics

Michael Ajao-Olarinoye¹

¹Center for computational science and Mathematical modelling

olarinoyem@coventry.ac.uk

1 Physics-Informed Neural Networks

High-fidelity Computational Fluid Dynamics (CFD) simulations, as outlined in Section 2.4, offer precise insights into arterial hemodynamics and are central to modeling fluid dynamics in complex vascular geometries. However, these simulations can be both computationally expensive and time-consuming, especially when dealing with patient-specific anatomies or conducting numerous parameter sweeps. Such challenges are further compounded by the need for significant domain expertise to refine meshes, tune solvers, and interpret results.

Physics-Informed Neural Networks (PINNs), first introduced by [1], have emerged as a powerful alternative that combines the strengths of traditional deep learning with established physics-based modeling. Rather than relying exclusively on data-driven learning, PINNs integrate the governing partial differential equations (PDEs) in this case, the continuity and momentum conservation equations (Section 2.2) directly into the network's loss function. This approach ensures that the neural network is "informed" by the physical laws of fluid motion, guiding it toward realistic flow solutions even when labeled data are sparse or imperfectly measured.

By integrating PDE residuals and boundary conditions into the training process, PINNs inherently enforce the fundamental laws of fluid dynamics, yielding physically consistent solutions even with limited labeled data. This physics-based integration minimizes overfitting and eliminates the need for conventional mesh discretization, enabling the network to be evaluated at arbitrary spatial and temporal resolutions, as a result, they can effectively complement or even partially replace traditional CFD

simulations. Moreover, the outputs of PINNs can be used to generate high-resolution flow fields predictions that exceed the original CFD. Consequently, once trained, PINNs provide rapid inference and are highly adaptable to new boundary conditions or geometrical changes with minimal retraining, making them an efficient, mesh-free alternative to traditional CFD simulations [?].

Furthermore, the mesh-free framework of PINNs eliminates the need for domain discretization, thereby simplifying simulation workflows and reducing computational overhead. This attribute is particularly beneficial for complex geometries like arterial aneurysms, where conventional CFD methods often face challenges during mesh generation [? ?]. Techniques like adaptive weighted loss functions further improve prediction accuracy and generalization capabilities, making PINNs a versatile tool for a wide range of fluid dynamics applications [?].

Although PINNs provide numerous benefits, ensuring robust and accurate simulations across various physiological conditions remains a challenge. Incorporating clinical data and further refining the neural network architectures could significantly broaden the utility of PINNs in real-world cardiovascular diagnostics and treatment planning.

In this study, we combine PINNs with traditional CFD data to estimate core hemodynamic variables such as pressure, velocity components, and wall shear stress (WSS) across healthy and Marfan Syndrome aortic geometries presented in (Section 2.1). This unified approach enables direct comparisons of velocity patterns, shear forces, and other critical flow indicators between non-pathological and aneurysmal models, providing valuable information for understanding the progression of vascular disease. The following sections detail the formulation, architecture, and training strategy of the PINNs used in this study.

1.1 Formulation of Physics-Informed Neural Networks

Modeling pulsatile flow in arterial segments necessitates the enforcement of both the continuity and momentum conservation equations to capture the essential hemodynamic effects. In this work, blood flow is modeled using the Reynolds-averaged Navier–Stokes (RANS) equations (see Sections 2.2 and 2.3), which describe the flow behavior in the aortic geometries. Let $\mathbf{u} = (u, v, w)$ denote the velocity field, p the

pressure, ρ the fluid density, and μ the dynamic viscosity. The continuity and momentum conservation equations are combined to form the Navier–Stokes system, which governs the evolution of the velocity field and pressure in response to external forces and viscous effects.

The boundary conditions employed in the simulations include a no-slip condition on the arterial walls ($\mathbf{u} = \mathbf{0}$ on $\partial\Omega_{\text{wall}}$), zero relative pressure at the outlet, and a pulsatile inlet velocity prescribed on the inlet boundary ($\partial\Omega_{\text{inlet}}$) as detailed in Equation (2.1) (Section 2.3). These boundary and inlet constraints are incorporated into the PINNs through penalty terms in the loss function, ensuring that the network’s predictions adhere to the specified physical conditions. In addition, the temporal dependence is explicitly modeled to reflect the transient nature of blood flow.

Within the PINN framework, the neural network approximates the functions

$$u(x, y, z, t), \quad v(x, y, z, t), \quad w(x, y, z, t), \quad p(x, y, z, t),$$

as well as the corresponding wall shear stress (WSS) components

$$\tau_x(x, y, z, t), \quad \tau_y(x, y, z, t), \quad \tau_z(x, y, z, t).$$

Collectively, these functions capture the spatiotemporal evolution of blood flow and the shear forces acting on the vessel walls.

Let $F(\cdot)$ denote the non-linear operator corresponding to the Navier–Stokes system of equations as defined in Equation (3.1). The physics residual loss is then defined as

$$L_{\text{physics}} = \|F(\mathbf{u}, p)\|_2, \quad (1)$$

where $\|\cdot\|_2$ represents the Euclidean norm. Minimization of this loss drives the PINN to approximate velocity and pressure fields that satisfy the Navier–Stokes equations, thereby capturing the fundamental flow dynamics even in regions with sparse or missing data.

Furthermore, the loss function includes additional terms to enforce the boundary conditions. Specifically, the penalty for the no-slip condition and the prescribed inlet

velocity are given by

$$L_{\text{boundary}} = \left\| \mathbf{u}|_{\partial\Omega_{\text{wall}}} - \mathbf{0} \right\|_2, \quad L_{\text{inlet}} = \left\| \mathbf{u}|_{\partial\Omega_{\text{inlet}}} - \mathbf{u}_{\text{inlet}}(t) \right\|_2, \quad (2)$$

where $\mathbf{u}_{\text{inlet}}(t)$ is the time-varying inlet velocity profile. When available, CFD reference data for pressure, velocity, and WSS are incorporated into the training through a data loss term. Let \mathbf{u}_{NN} and p_{NN} denote the predictions of the PINN, while \mathbf{u}_{CFD} and p_{CFD} denote the corresponding CFD reference data. The data loss term is defined as

$$L_{\text{data}} = \left\| \mathbf{u}_{\text{NN}} - \mathbf{u}_{\text{CFD}} \right\|_2 + \left\| p_{\text{NN}} - p_{\text{CFD}} \right\|_2 + \left\| \boldsymbol{\tau}_{\text{NN}} - \boldsymbol{\tau}_{\text{CFD}} \right\|_2. \quad (3)$$

The total loss function is then formulated as

$$L_{\text{total}} = \lambda_{\text{physics}} L_{\text{physics}} + \lambda_{\text{boundary}} L_{\text{boundary}} + \lambda_{\text{inlet}} L_{\text{inlet}} + \lambda_{\text{data}} L_{\text{data}}, \quad (4)$$

where λ_{physics} , $\lambda_{\text{boundary}}$, λ_{inlet} , λ_{data} are self-adaptive weighting coefficients [?]. These coefficients are optimized during training to automatically balance the importance of each loss component, ensuring that the PINN faithfully respects the fundamental fluid dynamics as well as the available CFD data. The training workflow for the PINNs, which encompasses the forward pass, loss computation, and weight updates via backpropagation, is illustrated in Figure ??.

1.2 Neural Network Architecture

The neural network architecture plays a crucial role in the performance and generalisation capabilities of PINNs. In this study, We train separate PINNs for pressure p , velocity components (u, v, w) , and the three WSS components (τ_x, τ_y, τ_z) , all time dependant (t) . Each network is a feed-forward, fully connected architecture with $L = 10$ hidden layers and $N = 64$ neurons per layer. The Swish activation function [?] is employed to promote smooth gradients. batch normalization is applied to each layer to stabilize training, and all weights are initialized using Kaiming normal [?] to ensure stable gradient flow.

The Swish activation function [?] is defined as:

$$\text{Swish}(x) = x \sigma(\beta x), \quad (5)$$

where β is a learnable parameter and $\sigma(\cdot)$ is the sigmoid function. This choice often yields faster convergence compared to ReLU or other activations. The network architecture is visualized in Figure ??, illustrating the flow of spatial-temporal coordinates (x, y, z, t) through the hidden layers to predict the pressure, velocity, and WSS components.

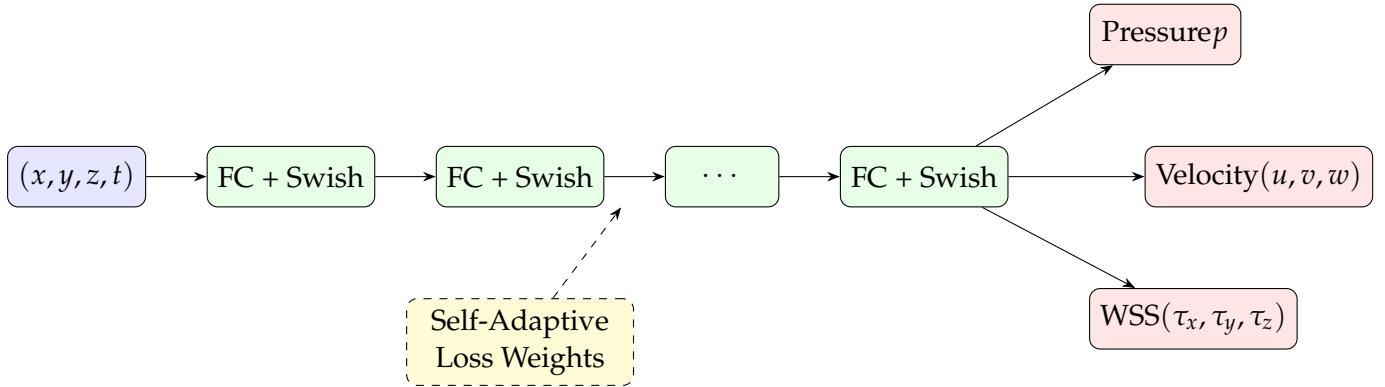


Figure 1: Architecture for predicting pressure, velocity, and WSS from spatiotemporal inputs (x, y, z, t) via fully connected layers with Swish activations. The network branches into three outputs and includes self-adaptive loss weights.

1.2.1 Self-Adaptive Loss Weighting Scheme

The self-adaptive loss weighting scheme is a key component of the PINN training process, enabling the network to dynamically adjust the relative importance of different loss components based on their contributions to the total loss. This mechanism prevents any single loss term from dominating, thereby ensuring that the network effectively balances the physics residuals, boundary conditions, inlet constraints, and supervised data.

Balancing multiple loss components is particularly challenging in multi-physics scenarios, where various physical phenomena and boundary conditions must be simultaneously satisfied [?]. In our study, the primary loss components include the physics residuals derived from the governing PDEs, the boundary conditions, the inlet conditions, and the supervised data from CFD simulations. Improper weighting of

these components can lead to suboptimal training outcomes, such as overfitting or failure to capture the underlying physical laws.

To address this challenge, we implement a self-adaptive loss weighting scheme inspired by [?]. Specifically, we introduce learnable parameters $\log \lambda_i$ for each loss component $i \in \{\text{phys, bound, inlet, data}\}$. These parameters are optimized jointly with the network weights during training, and the corresponding weights are defined as

$$\lambda_i = \exp(\log \lambda_i), \quad (6)$$

which ensures that each λ_i remains positive throughout training. By automatically adjusting $\log \lambda_i$, the network dynamically balances the contributions of each loss component based on the training dynamics, eliminating the need for manual tuning of hyperparameters and facilitating a more stable and efficient convergence [?]. The self-adaptive weighting scheme is implemented as part of the total loss function in Equation (??), where the loss components are weighted by the corresponding λ_i coefficients. The self-adaptive weighting mechanism is summarized in Algorithm ??.

Algorithm 1 Self-Adaptive Loss Weighting in PINNs

Require: Initial network weights θ , initial $\log \lambda_i$, learning rates for θ and $\log \lambda_i$

Ensure: Trained network weights θ^* and optimized λ_i^*

- 1: **while** not converged **do**
 - 2: Sample mini-batch of training points
 - 3: Forward pass to compute predictions $\mathbf{u}_{\text{NN}}, p_{\text{NN}}, \boldsymbol{\tau}_{\text{NN}}$
 - 4: Compute individual loss components:
 - 5: $\mathcal{L}_{\text{phys}} = \|F(\mathbf{u}_{\text{NN}}, p_{\text{NN}})\|_2^2$
 - 6: $\mathcal{L}_{\text{bound}} = \|\mathbf{u}_{\text{NN}}|_{\partial\Omega_{\text{wall}}} - \mathbf{0}\|_2^2$
 - 7: $\mathcal{L}_{\text{inlet}} = \|\mathbf{u}_{\text{NN}}|_{\Gamma_{\text{inlet}}} - \mathbf{u}_{\text{inlet}}(t)\|_2^2$
 - 8: $\mathcal{L}_{\text{data}} = \|\mathbf{u}_{\text{NN}} - \mathbf{u}_{\text{CFD}}\|_2^2 + \|p_{\text{NN}} - p_{\text{CFD}}\|_2^2 + \|\boldsymbol{\tau}_{\text{NN}} - \boldsymbol{\tau}_{\text{CFD}}\|_2^2$
 - 9: $\mathcal{L}_{\text{total}} = \lambda_{\text{phys}} \mathcal{L}_{\text{phys}} + \lambda_{\text{bound}} \mathcal{L}_{\text{bound}} + \lambda_{\text{inlet}} \mathcal{L}_{\text{inlet}} + \lambda_{\text{data}} \mathcal{L}_{\text{data}}$
 - 10: Backpropagate the total loss to compute gradients for θ and $\log \lambda_i$
 - 11: Update network weights $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}$
 - 12: Update log-weights $\log \lambda_i \leftarrow \log \lambda_i - \eta_\lambda \nabla_{\log \lambda_i} \mathcal{L}$
 - 13: StepLR scheduler updates learning rates η_θ and η_λ every N epochs
 - 14: **end while**
 - 15: **Return** θ^*, λ_i^*
-

The self-adaptive loss weighting algorithm is summarized in Algorithm ???. During training, the network weights θ and the log-weights $\log \lambda_i$ are updated simultaneously using stochastic gradient descent. The loss components are computed for each mini-

batch, and the total loss is evaluated based on the current predictions and the CFD reference data. The gradients of the total loss with respect to the network weights and log-weights are then backpropagated to update the parameters. A StepLR scheduler is employed to adjust the learning rates for the network weights and log-weights every N epochs, ensuring stable convergence and preventing oscillations during training. The self-adaptive weighting mechanism enables the network to effectively balance the physics residuals, boundary conditions, inlet constraints, and supervised data, leading to robust and accurate predictions of arterial flow dynamics.

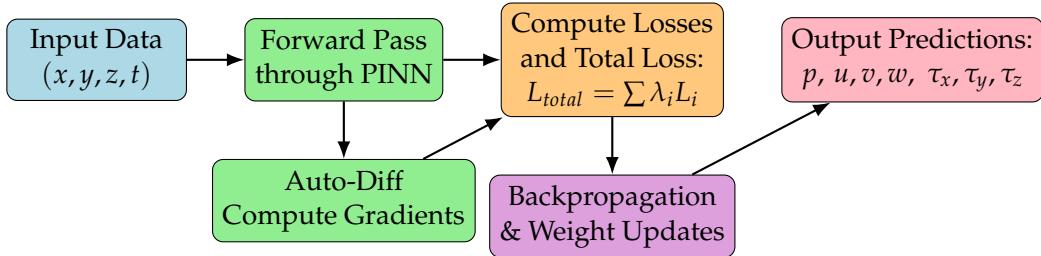


Figure 2: PINN training workflow: input (x, y, z, t) undergoes forward propagation and back-propagation to update the network and yield pressure, velocity, and WSS predictions.

1.3 Data Processing and Experimental Setup

The experimental data were derived from CFD simulations of both healthy and aneurysmal aortic geometries, with measurements acquired during the systolic and diastolic phases. Each dataset comprises spatial coordinates (X, Y, Z), time points, pressure values, velocity components (u, v, w), and wall shear stress (WSS) components. Prior to model training, the raw data underwent a comprehensive preprocessing pipeline. Initially, the data were cleaned by removing incomplete records and verifying the physical consistency of the measurements, resulting in, for example, the 0021 diastolic aneurysm dataset containing approximately 11,005 spatial points per time step. Subsequently, all input features and target variables were normalized via Min-Max scaling; spatial coordinates were scaled to the range $[-1, 1]$, time points were normalized over the cardiac cycle, and each of the physical variables (pressure, velocity, and WSS components) was scaled independently to ensure robust training. To enforce the no-slip condition at the vessel walls, points were identified using a velocity magnitude threshold ($|u|, |v|, |w| < 10^{-5}$). The processed data were then organized by case—healthy cases (0024 and 0142) and aneurysmal cases (0021, 0022, 0023, and 0025) with uniform

temporal sampling across the cardiac cycle. Finally, rigorous quality control measures were applied, including consistency checks on velocity and pressure fields, validation of WSS components against analytical solutions, verification of mass conservation at the inlet and outlet boundaries, and confirmation of temporal continuity across cardiac cycles.

1.4 Training of the PINNs for Pulsatile Flow Simulations

The PINN framework was implemented in Python using the PyTorch library [?], which leverages automatic differentiation for efficient computation of PDE residuals. All codes and numerical experiments for the PINN-based aneurysm flow analyses were developed using PyTorch, with the self-adaptive weighting and physics-based PDE residuals integrated into the backpropagation routine. The detailed hyperparameters and training protocol, described below, ensure reproducibility and transparency.

Input spatiotemporal data points (x_j, y_j, z_j, t_j) were randomly sampled from the aortic geometries over the cardiac cycle and normalized via Min–Max scaling. The PINNs were trained using the AdamW optimizer [?] with an initial learning rate of 1×10^{-4} , momentum parameters $\beta = (0.9, 0.999)$, and a weight decay of 1×10^{-4} . A StepLR scheduler reduced the learning rate by a factor of $\gamma = 0.9$ every 200 epochs to facilitate finer convergence.

Training was performed using mixed-precision arithmetic via `torch.cuda.amp` [?] to enhance computational efficiency and reduce memory consumption without compromising accuracy. The training process was carried out for up to 1000 epochs, with early stopping applied if the validation loss did not improve for 5 consecutive epochs. Gradient clipping, with a maximum norm of 1.0, was employed to prevent exploding gradients. The entire training was executed on a high-performance computing cluster equipped with NVIDIA RTX 8000 GPUs, which enabled efficient parallel processing and rapid convergence. The overall training workflow, as depicted in Figure ??, encompasses the forward pass, loss computation (which combines physics, boundary, inlet, and data losses as $L_{total} = \sum \lambda_i L_i$), and the subsequent backpropagation and weight update steps.

1.5 Relevance to Aneurysm Studies and Clinical Applications

The use of PINNs presents a transformative approach in the study of aneurysms, particularly when compared with conventional CFD simulations. Traditional CFD methods require the complete re-simulation of flow fields each time boundary conditions or geometrical configurations are modified, a process that is computationally intensive due to the demands of mesh generation, solver tuning, and the resolution of large-scale systems. In contrast, PINNs embed the governing physical laws directly into the neural network's loss function. This mesh-free approach enables the network to learn robust representations of the flow dynamics, thereby significantly reducing the computational overhead and facilitating rapid, high-resolution evaluations.

Once trained, PINNs serve as efficient surrogates that can be queried at arbitrary spatial and temporal resolutions, allowing for iterative parameter sweeps, sensitivity analyses, and even real-time clinical decision-making. For instance, in the case of Marfan Syndrome aortic aneurysms, PINNs enable the rapid exploration of a wide range of inflow waveforms, wall properties, and geometric variations—scenarios that would otherwise require multiple, resource-intensive CFD simulations. Furthermore, the ability of PINNs to incorporate physical constraints directly into the training process minimizes overfitting and ensures that the solutions remain physically realistic, even in regions where direct measurements are sparse or unavailable.

An additional advantage of PINNs lies in their adaptability. Through transfer learning, a PINN trained on a specific patient's CFD data can be fine-tuned to accommodate new cases or altered conditions with minimal extra data, thus supporting personalized treatment strategies. This synergy between classical CFD and PINNs not only preserves the accuracy of detailed fluid physics but also enhances computational scaling, making it a promising tool for accelerating patient-specific hemodynamic evaluations and ultimately improving clinical outcomes.

1.6 Results

The PINN-based simulations were validated against benchmark CFD data for healthy and Marfan Syndrome aortic geometries, demonstrating excellent agreement in velocity profiles, pressure distributions, and wall shear stress (WSS) patterns. The PINNs

accurately captured the complex flow dynamics within the aneurysm sac, including vortex formation, flow separation, and recirculation zones. The self-adaptive loss weighting mechanism effectively balanced the physics residuals, boundary conditions, and data-fitting objectives, ensuring that the network learned realistic flow solutions while adhering to the governing Navier–Stokes equations. The figures below illustrate the comparison between PINN predictions and CFD reference data for pressure, and wall shear stress in the aortic aneurysms.

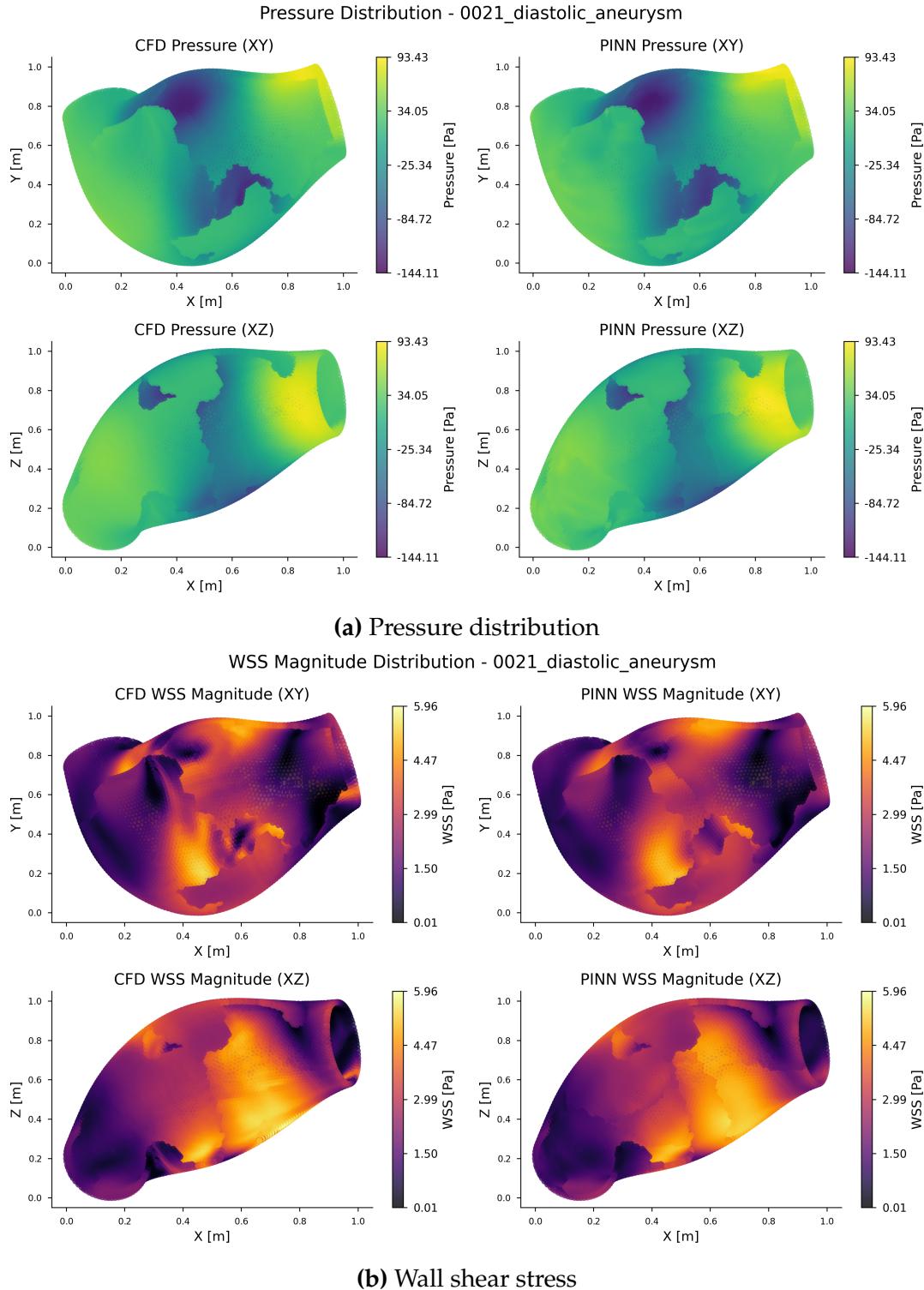


Figure 3: Von Mises stress (Pa) and wall shear stress contours for diseased cases. Figures a–d in each show the xy and xz planes for case 0021 diastolic aneurysm comparison between CFD and PINN results.

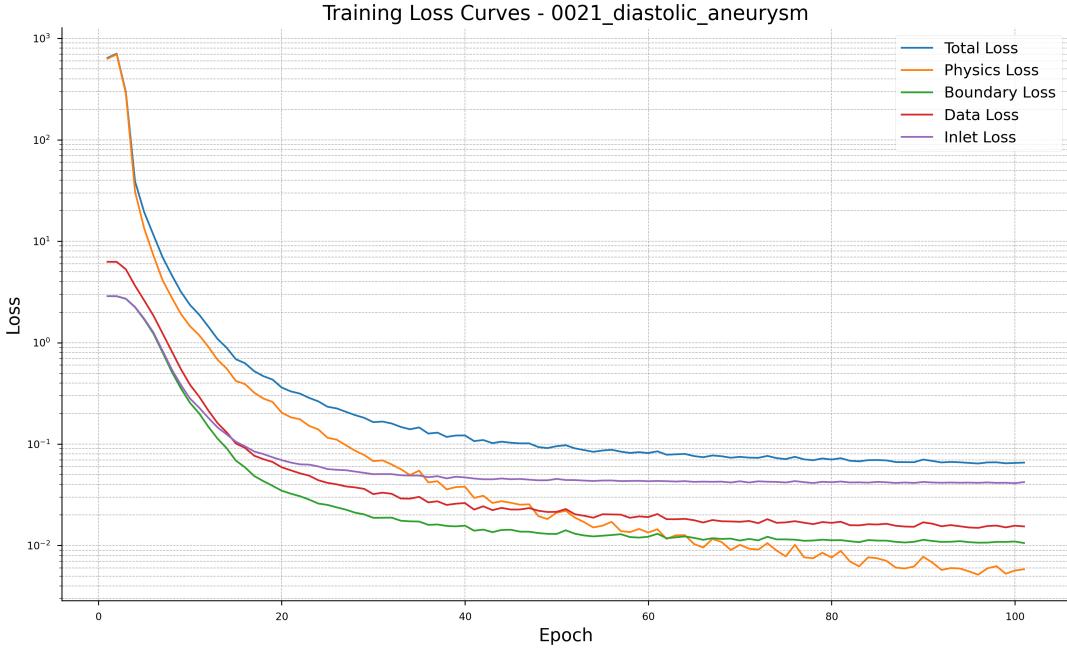


Figure 4: Loss curves for the PINN training process, showing the evolution of the physics, boundary, inlet, and data-fitting losses over 1000 epochs. The self-adaptive loss weighting mechanism effectively balances the contributions of each loss component, ensuring that the network learns realistic flow solutions while adhering to the governing Navier–Stokes equations.

Figure ?? illustrates the comparison between PINN predictions and CFD reference data for both pressure and WSS. The subfigures demonstrate the spatial distribution of pressure and WSS on both the xy and xz planes for a representative aneurysm case (0021 diastolic aneurysm). These visual comparisons confirm that the PINN accurately reproduces the key flow features observed in the CFD simulations. Furthermore, Figure ?? presents the evolution of the loss curves over 1000 training epochs, highlighting the convergence behavior of the physics, boundary, inlet, and data losses. The steady decline of these loss components reflects the efficacy of the self-adaptive weighting mechanism in guiding the network toward a physically consistent solution.

The mesh-free nature of PINNs [?] enables the rapid evaluation of flow fields at arbitrary spatial and temporal resolutions, facilitating detailed hemodynamic analyses and personalized treatment planning. By leveraging the self-adaptive loss weighting scheme, the PINNs effectively balance the physics residuals, boundary conditions, and data-fitting objectives, ensuring that the network learns realistic flow solutions while adhering to the governing Navier–Stokes equations. The results presented here demonstrate the potential of PINNs as a powerful tool for simulating pulsatile flow in arterial geometries, with applications ranging from disease diagnosis to treatment

planning and optimization.