Parametrized Sampling for 3D Blood Simulation in Deformable Vessels Using Physics-Informed Neural Networks

Han Zhang[a,b], Lingfeng Li[b], Xue-Cheng Tai[c,*], Raymond Hon-Fu Chan[d,e]

[a] *Department of Mathematics, City University of Hong Kong, Hong Kong, China*
[b] *Hong Kong Centre for Cerebro-Cardiovascular Health Engineering, Hong Kong, China*
[c] *Norwegian Research Centre, Bergen, Norway*
[d] *Department of Operations and Risk Management, Linnan University, Hong Kong, China*
[e] *School of Data Science, Lingnan University, Hong Kong, China*

Abstract

Simulating blood flow in deformable vessels is crucial for advancing the understanding of cardiovascular dynamics. To address this, we employ Physics-Informed Neural Networks (PINNs) to solve the Navier-Stokes equations in full 3D formulations within elastic, deformable vessel geometries. To achieve this, we developed a parameterized sampling strategy that ensures a continuous fluid domain and smooth vessel wall surfaces, facilitating accurate differentiation calculations for various physical interactions across interface surfaces. To enforce periodicity, which poses an even greater challenge to the modeled multi-physics problem, we incorporated periodic feature layers into the neural networks. Additionally, Windkessel boundary conditions were applied at the outlet to replicate physiological behavior through Monte-Carlo integration. Dynamic weighting strategy is introduced to balance multiple loss terms associated with PDE constraints and boundary conditions. Experiments on various geometries are conducted to validate the accuracy of the full 3D model. Comparative studies between the full and reduced-order models highlight the respective advantages and limitations of each approach. Additional analyses of Windkessel parameters further elucidate their impact on flow dynamics.

Keywords: Fluid-Structure Interaction, Physics-Informed Neural Networks, Parameterized Sampling, Periodic Feature, Windkessel Model, Monte-Carlo Integration

1. Introduction

Understanding and accurately measuring blood flow is crucial for effective cardiovascular healthcare. Invasive measurement techniques [1] carry risks for patients and may disrupt original flow dynamics due to the presence of measuring equipment [2]. As a non-invasive

[*] Corresponding author

Email addresses: `hzhang863-c@my.cityu.edu.hk` (Han Zhang), `lfli@hkcoche.org` (Lingfeng Li), `xtai@norceresearch.no` (Xue-Cheng Tai), `raymond.chan@ln.edu.hk` (Raymond Hon-Fu Chan)

alternative, numerical simulations of blood flow offer a promising approach for evaluating cardiovascular dynamics.

Blood flow is commonly modeled as an incompressible Newtonian fluid governed by the incompressible Navier-Stokes equations [3]. Recognizing that vessel deformation influences blood flow, researchers further extended this framework to model blood flow as a Fluid-Structure Interaction (FSI) problem [4, 5]. Over the years, various numerical methods have been developed to address this complex FSI problem. However, solving it numerically remains computationally challenging. Mesh-based methods, such as the Finite Element Method (FEM), require meticulous handling of mesh deformations induced by vessel elasticity. These domain deformations can distort the mesh, leading to convergence issues [6, 7]. Moreover, mesh-based approaches lack continuous solutions across the domain, necessitating interpolation to evaluate results at non-node points, which introduces additional errors. High-resolution configurations — featuring dense grids and small time steps — are typically required for accuracy but come at the cost of significantly increased computational time and memory usage.

Recently, Physics-Informed Neural Networks (PINNs) [8, 9] have achieved lots of success. Some studies [10, 11, 12] have applied PINN models to blood flow simulations. However, those prior works primarily focused on simplified cylindrical or 1D models, which are insufficient for accurately representing realistic cardiovascular vessel systems. This gap highlights the need for developing methods capable of simulating blood flow in full 3D geometries.

However, modeling blood flow in 3D is challenging, as vessels typically exhibit curved and non-uniform geometries. Trivial sampling methods are inadequate because the interaction between fluid and structure requires precise differentiation of variables across their interface. Additionally, the complexity arising from the interplay of multiple physical phenomena makes accurately reproducing the periodicity of the cardiovascular cycle particularly difficult.

To address these challenges, we propose a parameterized sampling strategy specifically designed for the full 3D model. This strategy is crucial for ensuring the smoothness necessary for accurate differentiation calculations across interface surfaces involved in physical interactions. Periodicity is preserved through the use of a periodic feature layer [13], which also eliminates the need for initial boundary conditions [12], often difficult to determine in practice. Additionally, Windkessel boundary conditions are implemented using Monte Carlo integration for the proposed 3D model. A dynamic weighting strategy and an alternative optimization approach are also developed to better handle the optimization of problems involving multiple loss terms.

In the experimental studies, to validate the computational accuracy of the full 3D model, we conduct experiments across various vessel geometries with different degrees of curvature. The results show that our model achieves accuracy comparable to conventional FEM. To assess the impact of dimensionality on solver performance, we systematically compare the full model, which solves the complete 3D formulation of blood flow, with reduced-order models that employ cylindrical or 1D simplifications. While the full model operates in higher dimensions and captures fine details of flow dynamics, the reduced-order models exhibit faster calculations. Further experiments on stenosed curved vessels highlight superior capability of the full model to resolve fine details compared to the simplified cylindrical model. Additionally, we investigate the influence of different Windkessel model parameters, providing deeper insights into how these parameters affect the magnitude and characteristics of blood flow dynamics.

In summary, our contributions are as follows:

1. We solved a Coupled 3D FSI problem Using PINNs for periodic blood flow simulation. under realistic Windkessel Boundary Conditions
2. We propose a parameterized sampling strategy tailored for 3D blood flow simulation in general curved vessel geometries.
3. We apply Periodic Feature Layer to enforce periodicity and Monte Carlo integration for Windkessel model implementation, enabling more realistic blood flow simulation.
4. We provide a systematic comparison between full and reduced-order blood flow models, along with an in-depth investigation of Windkessel parameter effects, offering practical guidance for model selection and insights into their impact on flow dynamics.

2. Related Works

2.1. Blood Simulation

A variety of numerical methods have been performed to tackle blood flow simulations. Blood is commonly modeled as an incompressible Newtonian fluid governed by the incompressible Navier-Stokes equations [3]. Recognizing that vessel deformation influences blood flow, researchers have further extended this model to address blood flow as a Fluid-Structure Interaction (FSI) problem [4, 5]. Formaggia et al. [14] combined 3D and 1D Navier-Stokes equations to reduce computational complexity. Crosetto et al. [15] introduced parallel preconditioners leveraging the block structure of the linear system. Barker and Cai [16] proposed a robust monolithic coupling method for fluid-structure interaction problems, capable of handling large deformations and parameter changes. Bukavc et al. [17] developed a loosely coupled scheme based on modified Lie operator splitting, ensuring unconditional stability by decoupling fluid and structure sub-problems. Chabannesa *et al.* [18] solved the problem using high-order methods. Janela *et al.* [19] model the blood as non-Newtonian, which coupling with 3D compressible elastic vessel wall structures. Ikbal *et al.* [20] analysed blood flow in stenosed artery. Katz *et al.* [21] studied the impact of viscosity modeling and of choosing the turbulence model in generalized Newtonian blood flow simulations in an aortic coarctation.. Other prominent approaches include space-time formulations [22, 23], higher-order discontinuous Galerkin methods [24], immersed boundary methods [25, 26], and coupled momentum methods [27, 28].

2.2. Physics-Informed Neural Networks

Physics-Informed Neural Network (PINN) [8, 9] has achieved significant success in various applications, including thermal energy system modeling [29], sound propagation [30], pulsatile flow [31] source reconstruction [32] and cell research [33]. Mathematicians

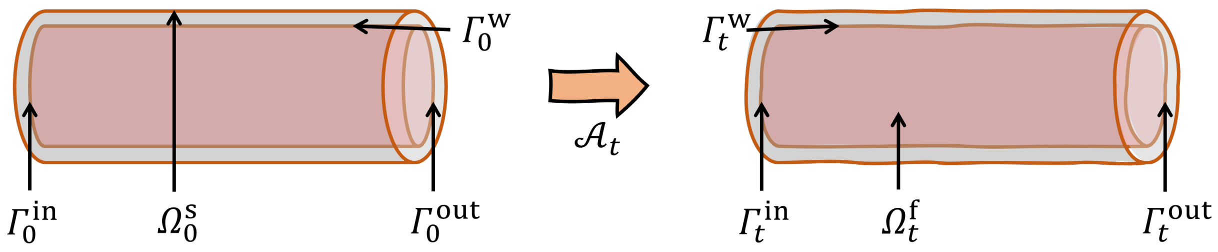


Figure 1: The illustration for the Arbitrary Lagrangian Euler form. Left: reference configuration; Right: current configuration.

also utilize PINN and its variants to solve equations [34, 35]. Further advancements, such as operator learning [36, 37, 38], modeling on irregular domains [39], the Kolmogorov-Arnold Networks (KAN) model [40], and large-scale matrix computation [41], have further expanded the capabilities of PINNs. Griese *et al.* [42] combined the good approximation properties of PINN with the classical FEM to solve for incompressible fluid flows and its related inverse problems.

Beyond those advancements, features like periodicity [43] has gained attention among PINN researchers studying biological and medical applications [44, 45]. For instance, Ahmadi *et al.* [46] utilize sine functions to enforce periodicity or exponential decay in the concerned ODEs. Daryakenari *et al.* [47] propose a feature layer with exponential functions to address the rapid decay to zero in neural networks when modeling minimal changes in drug concentration. Additionally, Huhn *et al.* [48] introduce Fourier features to enhance Radiation Transport simulations in heterogeneous media.

3. Problem Formulation

In this section, we present the mathematical formulation for simulating blood flow in a deformable vessel in full 3D model. In the model, blood flow is treated as an incompressible Newtonian fluid governed by the Navier-Stokes equations in a pressure-velocity formulation. Given the dynamic nature of the computational domain, we employ the Arbitrary Lagrangian-Eulerian (ALE) formulation of the Navier-Stokes equations. The vessel wall undergoes continuous motion under fluid forces, resulting in a moving fluid domain, which is computed using harmonic extension from the structure's displacement.

3.1. Navier-Stokes in Deformable Vessel

To accurately model the movement of the computational domain along with fluid velocity and pressure, we begin with the reference configuration Ω_0^{f}, as shown in Figure 1. This configuration serves as the parameterization for the entire computational domain. The domain is divided into two parts: the fluid domain Ω_0^{f}, where the flow occurs. Γ_0^{w} is the interface boundary between the fluid domains and the solid structure. The inlet and outlet boundaries are labeled as Γ_0^{in} and Γ_0^{out}, respectively. Each point within the reference configuration domain is represented by the coordinates $\boldsymbol{x}_0 = (x_0, y_0, z_0)^\intercal$, with bold symbols indicating matrices or column vectors throughout the paper. As the domain evolves over time, Ω_t^{f} represents the current configuration at time t. The symbols Γ_t^{w}, Γ_t^{in}, and Γ_t^{out} correspond to the interface, the inlet, and the outlet at the current configuration, respectively. Points in the current configuration are also referred to by their physical coordinates $(x_t, y_t, z_t)^{\intercal}$. Following Quarteroni et al. [5, P. 73, Sec. 18], we introduce the ALE mapping defined as:

$$\mathcal{A}_t : \Omega_0 \to \Omega_t, \quad \boldsymbol{x}_0 \to \boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{d}(\boldsymbol{x}_0, t) = \mathcal{A}_t(\boldsymbol{x}_0) \quad \text{in } \Omega_t^{\mathrm{f}}, \tag{1}$$

where the displacement function $\boldsymbol{d}(\cdot)$ characterizes the domain movement defined in the reference configuration, and $\tilde{\boldsymbol{w}} = \frac{\partial \boldsymbol{d}}{\partial t}$ denotes the domain velocity within Ω_0^{f}.

The incompressible unsteady Navier-Stokes equation, describing the movement of a viscous fluid in a moving domain [5, P. 74, Sec. 18], is:

$$\begin{aligned} \rho_{\mathrm{f}} \tfrac{D}{Dt}\boldsymbol{u} + \rho_{\mathrm{f}}[(\boldsymbol{u} - \boldsymbol{w}) \cdot \nabla]\boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma}_{\mathrm{f}} = \boldsymbol{0}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{aligned} \quad \text{in } \Omega_t^{\mathrm{f}},\ t \in [0, T], \tag{2}$$

where $\boldsymbol{w} = \tilde{\boldsymbol{w}} \circ \mathcal{A}_t^{-1}$, ρ_{f} is the fluid density, $\boldsymbol{u}$ is the fluid velocity, $\boldsymbol{\sigma}_{\mathrm{f}} = -P\boldsymbol{I} + 2\mu_{\mathrm{f}} \frac{\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\intercal}}{2}$ is the Cauchy stress tensor, P is fluid pressure, and μ_{f} is the viscosity of the fluid. The term $\frac{D}{Dt}\boldsymbol{u} = \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{w} \cdot \nabla \boldsymbol{u}$ represents the ALE derivative, and ∇ is the gradient with respect to $\boldsymbol{x}_t$ within Ω_t^{f}. Here, $(\nabla \boldsymbol{u})^{\intercal}$ is the transpose of $\nabla \boldsymbol{u}$.

Stress continuity [5, Eq. (16.12), P. 60, Sec. 16] on the interface between the fluid and the vessel wall, as the source of domain movement is written as:

$$\tfrac{\partial^2 \eta}{\partial t^2} + b\eta = K \quad \text{in } \Gamma_0^{\mathrm{w}}, \tag{3}$$

where $b = \frac{E}{\rho_{\mathrm{s}}(1-\xi^2)R_0^2}$, and $K = \frac{1}{\rho_{\mathrm{s}} h_0} \left[\frac{R}{R_0}(P - P_{ext} - g\mu_{\mathrm{f}}\left((\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\intercal}) \cdot \boldsymbol{n}\right) \cdot \boldsymbol{e}_r \right]$. Here, ρ_s represents the vessel wall density, ξ is Poisson's ratio, R_0 is the radius in the reference domain, P_{ext} denotes the external pressure, and h_0 denotes the vessel wall thickness. Note that in this 1D ring model for vessel wall mechanics, all expressions must be converted into cylindrical coordinates. Moreover, as the calculation does not involve the axial position coordinate, the cylindrical coordinate conversion could only be performed over each cross-section.

Fluid continuously enters the vessel through the inlet, following a parabolic velocity profile for the inlet boundary condition:

$$\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{g}(\boldsymbol{x})v(t) \quad \text{on } \Gamma_t^{\mathrm{in}}, t \in [0, T], \tag{4}$$

where $g(\boldsymbol{x}) = 1 - \frac{r^2}{r_0^2}$ characterizes the parabolic profile, r_0 is the radius of inlet cross-section, and r is the distance of $\boldsymbol{x}$ from the center within the cross-section. $v(t)$ is the velocity at the midpoint ($r = 0$) of the inlet (the velocity in Figure 5 is used in this paper). Thus, on the interface boundary, the velocity at the fluid-structure interface matches the wall velocity:

$$\boldsymbol{u} = \tfrac{\partial \boldsymbol{d}}{\partial t} \quad \text{on } \Gamma_t^{\mathrm{w}}, t \in [0, T]. \tag{5}$$

3.2. Windkessel Boundary Condition

To uniquely determine the solution of the governing equations, appropriate initial and boundary conditions must be specified. In traditional numerical simulations, boundary conditions are typically imposed at the inlet and outlet of the computational domain. At the inlet, a Dirichlet boundary condition is commonly used, prescribing either the blood pressure P or the blood flux flow rate Q. At the outlet, boundary conditions play a crucial role in determining the reflection waves traveling backward in the arteries.

A simple choice for the outlet boundary condition is the non-reflecting boundary condition [5], defined as $[\mu_{\mathrm{f}} (\nabla^{\intercal} \boldsymbol{u} + \nabla \boldsymbol{u}) - P\boldsymbol{I}] \cdot \boldsymbol{n} = \boldsymbol{0}$, where μ_{f} is fluid viscosity, $\boldsymbol{n}$ is the outward normal direction of the outlet. Although this condition is easy to implement, it does not accurately capture the reflections that occur due to the interaction between blood flow and vessel walls, particularly in the presence of branching. For more realistic simulations, the three-element Windkessel boundary condition provides a superior alternative [49]. This boundary condition is expressed as:

$$Q(L,t)\left(1+\tfrac{R_1}{R_2}\right)+CR_1\tfrac{\partial Q}{\partial t}(L,t)-\tfrac{P(L,t)}{R_2}-C\tfrac{\partial P}{\partial t}(L,t)=0, \quad t\in[0,T], \tag{6}$$

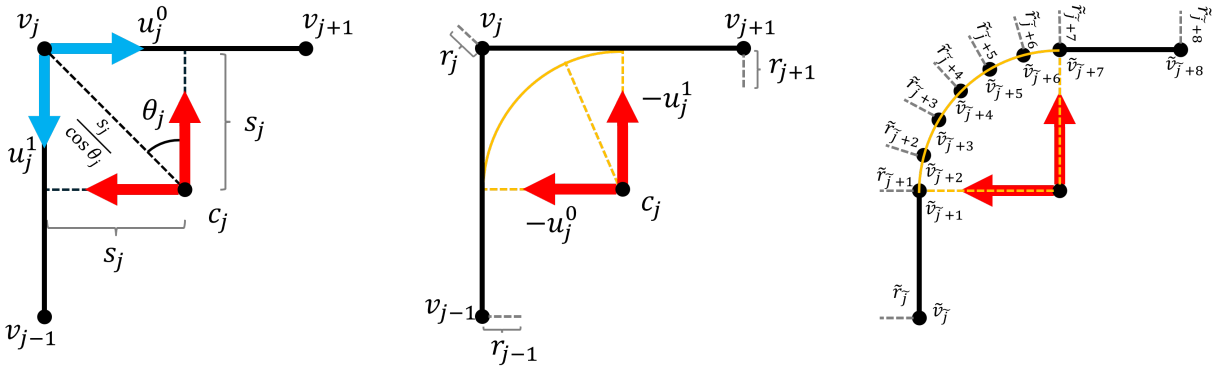
where R_1, R_2, and C are parameters representing proximal resistance, distal resistance, and compliance of the artery, respectively. Importantly, the model relies on the blood flux flow rate Q and pressure P at the outlet. In the 1D model, these quantities can be explicitly solved. However, in the 3D model, only the velocity u and pressure P are solved, without direct access to the flux rate Q. Thus, to acquire the flux Q, an integration of u over the outlet cross-sectional area is needed for Windkessel boundary conditions computation. This is to be done through the usage of Monte-Carlo integration in Section 4.4.

The three-element Windkessel model conceptualizes the hemodynamics of the arterial system in terms of resistance and compliance. The proximal resistance R_1 models the resistance of the arteries close to the outlet, while the distal resistance R_2 accounts for the resistance offered by the microcirculation. The compliance C characterizes the elastic capacity of the artery to store and release blood volume. Together, these elements capture the essential dynamics of blood flow and pressure wave propagation at the outlet, including the generation of reflection waves. Unlike the simple non-reflecting boundary condition, the Windkessel model ensures that reflections are consistent with the physical properties of the vessel and the surrounding circulatory system [50, 51].

Extensions of the Windkessel model include the four-element Windkessel model [49], which incorporates an additional inertial component to account for the dynamic effects of blood acceleration.

4. Method

In this section, we detail the implementation of the PINN method for the full 3D model. We begin by describing the sampling strategy for the computational domain, specifically for 3D vessels. Next, we introduce the loss functions used in the optimization process. Finally, we explain the periodic layers and Monte Carlo integration techniques, which are



v_j
u_j^0
v_{j+1}
θ_j
s_j
$\frac{s_j}{\cos\theta_j}$
u_j^1
c_j
s_j
v_{j-1}
v_j
v_{j+1}
r_j
r_{j+1}
$-u_j^1$
$-u_j^0$
c_j
v_{j-1}
r_{j-1}
$\tilde{r}_{\tilde{j}+8}$
$\tilde{v}_{\tilde{j}+8}$
$\tilde{r}_{\tilde{j}+7}$
$\tilde{v}_{\tilde{j}+7}$
$\tilde{r}_{\tilde{j}+6}$
$\tilde{v}_{\tilde{j}+6}$
$\tilde{r}_{\tilde{j}+5}$
$\tilde{v}_{\tilde{j}+5}$
$\tilde{r}_{\tilde{j}+4}$
$\tilde{v}_{\tilde{j}+4}$
$\tilde{r}_{\tilde{j}+3}$
$\tilde{v}_{\tilde{j}+3}$
$\tilde{r}_{\tilde{j}+2}$
$\tilde{v}_{\tilde{j}+2}$
$\tilde{r}_{\tilde{j}+1}$
$\tilde{v}_{\tilde{j}+1}$
$\tilde{r}_{\tilde{j}}$
$\tilde{v}_{\tilde{j}}$

Figure 2: Illustration of the smoothing process for a parameterized representation. Left: The center c_j for smoothing is first determined using a tangent circle with smoothing radius s_j. Middle: The corresponding circular arc and normal vectors at both endpoints are computed. Right: A set of points, along with their associated normal vectors, is then interpolated between the two endpoints.

employed to model the periodic nature of blood flow and enforce the Windkessel boundary condition.

4.1. Parametrized Sampling for Vessel Objects

A well-designed sampling strategy is crucial for 3D vessel modeling. Both the fluid domain, where blood flows, and the vessel boundary wall, where fluid-structure interactions occur, must be sufficiently smooth to ensure reliable computation of differential quantities and normal vectors. Naive or poorly designed sampling can lead to bad interface representations, where sharp or noisy boundaries result in ill-defined derivatives and inaccurate stress evaluations. To address this, our parameterized sampling strategy is introduced to promote geometric smoothness and improve the fidelity of physical modeling.

Given the vessel's cylindrical-like geometry, a natural approach to parameterization is to extract key points along its centerline [52, 53], which then serve as a basis for structured smooth sampling along the vessel. However, as illustrated in Figure 2, using only a few discrete key points can lead to sharp geometric transitions, compromising smoothness. To address this, smoothing operations are applied to refine the centerline, ensuring a sufficiently smooth representation. Denoting the key centerline points as $\{\boldsymbol{v}_j\}_{j=0}^{M}$, we firstly calculate the curve angles as in Figure 2.(a). To be in detail, for a point v_j where $j = 1, \ldots, M-1$, it's previous and after points are v_{j-1} and v_{j+1}. The angle for point v_j can ba calculated by:

$$\theta_j = \tfrac{1}{2}\pi - \tfrac{1}{2}\arccos(\boldsymbol{u}_j^0 \cdot \boldsymbol{u}_j^1), \tag{7}$$

where $\boldsymbol{u}_j^0 = \frac{(\boldsymbol{v}_{j-1}-\boldsymbol{v}_j)}{|\boldsymbol{v}_{j-1}-\boldsymbol{v}_j|}$ and $\boldsymbol{u}_j^1 = \frac{(\boldsymbol{v}_{j+1}-\boldsymbol{v}_j)}{|\boldsymbol{v}_{j+1}-\boldsymbol{v}_j|}$. Then, based on a given smoothen radius s_j and the calculated curvature angle, we could calculate the center via

$$\boldsymbol{c}_j = \boldsymbol{v}_j + \tfrac{s_j}{\cos(\theta)} \tfrac{\boldsymbol{u}_j^0+\boldsymbol{u}_j^1}{|\boldsymbol{u}_j^0+\boldsymbol{u}_j^1|}. \tag{8}$$

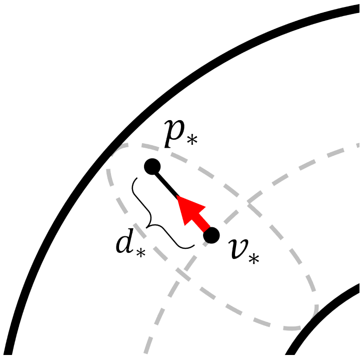


Figure 3: Illustration of the sampling process for a parameterized representation. First, a center point is randomly selected along the centerline. Next, a random direction within the axial cross-section is determined using the local normal and binormal vectors. Finally, the sampled point is obtained by moving a random distance along the chosen direction.

With the solved center, an arc of radius s_j could then be drawn as in Figure 2.(b). A denser sampling will then be performed along the arc to refine the abrupt turn into an approximately circular curve as in Figure 2. (c). The complete algorithm is then given as the following algorithm 1.

Once the vessel has been sufficiently smoothed, we can approximate its centerline as a smooth curve (see Figure 3). To sample points on the vessel, we need to further assign an orientation to each axial cross-section along the centerline. We first compute the normal vectors $\{\tilde{\boldsymbol{n}}_j\}_{j=0}^{M}$ and binormal vectors $\{\tilde{\boldsymbol{b}}_j\}_{j=0}^{M}$ for the centerline curve $\{\tilde{\boldsymbol{v}}_j\}_{j=0}^{M}$ using parallel transport.

Then, we aim to sample new points. Given the smoothened centerline points $\{\tilde{\boldsymbol{v}}_j\}_{j=0}^{M}$ that define the curve, we aim to sample a point along the curve through interpolation. To achieve this, we first parameterize the curve using an arc-length parameter as the cumulative arc length between consecutive points $l_j = \sum_{i=1}^{j} \|\tilde{\boldsymbol{v}}_i - \tilde{\boldsymbol{v}}_{i-1}\|$. We then define a piece-wise interpolation function $\mathcal{F}_l$ that maps the parameter l_j to spatial coordinates $\tilde{\boldsymbol{v}}_j$. We sample new points $\boldsymbol{v}_* = \mathcal{F}(l_*)$, where l_* is uniformly sampled from $[l_0, l_M]$. In the same way, we interpolated for the corresponding radius r_*, the normal vector $\boldsymbol{n}_*$, and the binormal vector $\boldsymbol{b}_*$. Note that the interpolated vectors need to be normalized and may not be the exact normal/binormal vector, but it's still a usable approximation for sampling purposes. Next, we sample an angle θ_* from the interval $[0, 2\pi)$ and use it to determine the direction of the sampled point via the expression $\cos(\theta_*)\boldsymbol{n}_* + \sin(\theta_*)\boldsymbol{b}_*$. This operation effectively rotates $\boldsymbol{n}_*$ around the tangent vector at $\boldsymbol{v}_*$ by θ_*. For sampling points in the fluid domain, an additional random distance d_* is drawn from the interval $[0, r_*]$ to determine the radial displacement from the centerline.

Finally, The sampled point for fluid reference domain Ω_0^{f} is given by:

$$\boldsymbol{x}_*^{\mathrm{f}} = \boldsymbol{v}_* + d_*(\cos(\theta_*)\boldsymbol{n}_* + \sin(\theta_*)\boldsymbol{b}_*), \tag{9}$$

where $\boldsymbol{v}_*$ is the sampled center point along the centerline, and $(\boldsymbol{n}_*, \boldsymbol{b}_*)$ form an orthonormal basis perpendicular to the centerline direction at $\boldsymbol{v}_*$ as discussed. For the vessel wall Γ_0^{w}, where fluid-structure interaction occurs, the sampled points are defined similarly, using the vessel radius r_* instead of d_*:

$$\boldsymbol{x}_*^{\mathrm{w}} = \boldsymbol{v}_* + r_*(\cos(\theta_*)\boldsymbol{n}_* + \sin(\theta_*)\boldsymbol{b}_*), \tag{10}$$

Similarly, for the inlet and outlet boundaries $\tilde{\boldsymbol{v}}_0$ and $\tilde{\boldsymbol{v}}_M$, the sampling centers are fixed at Γ_0^{i} and Γ_0^{o}, respectively. The corresponding sampled points are given by:

$$\boldsymbol{x}_*^{\mathrm{i}} = \tilde{\boldsymbol{v}}_0 + d_*(\cos(\theta_*)\boldsymbol{n}_* + \sin(\theta_*)\boldsymbol{b}_*), \tag{11}$$

$$\boldsymbol{x}_*^{\mathrm{o}} = \tilde{\boldsymbol{v}}_M + d_*(\cos(\theta_*)\boldsymbol{n}_* + \sin(\theta_*)\boldsymbol{b}_*). \tag{12}$$

Algorithm 1 Smoothen vessel system

```
Require: The vessel centerline $\mathcal{V}$, the corresponding axial cross-section radius $\mathcal{R}$, smoothing radius $\mathcal{S}$ and smoothen factor $\epsilon$;
Ensure: The smoothened vessel centerline $\tilde{\mathcal{V}}$, the corresponding axial cross-section radius $\tilde{\mathcal{R}}$, smoothing radius $\tilde{\mathcal{S}}$;
$\tilde{j} = 0$, $\tilde{\boldsymbol{v}}_0 = \boldsymbol{v}_0$, $\tilde{r}_0 = r_0$ and $\tilde{s}_0 = s_0$
for $j = 1 : M - 1$ do
    $\boldsymbol{u}_j^0 = \frac{(\boldsymbol{v}_{j-1} - \boldsymbol{v}_j)}{|\boldsymbol{v}_{j-1} - \boldsymbol{v}_j|}$ and $\boldsymbol{u}_j^1 = \frac{(\boldsymbol{v}_{j+1} - \boldsymbol{v}_j)}{|\boldsymbol{v}_{j+1} - \boldsymbol{v}_j|}$
    $\theta_j = \frac{1}{2}\pi - \frac{1}{2}\arccos(\boldsymbol{u}_j^0 \cdot \boldsymbol{u}_j^1)$
    Center of smooth curve $\boldsymbol{c}_j = \boldsymbol{v}_j + \frac{s_j}{\cos(\theta)} \frac{\boldsymbol{u}_j^0 + \boldsymbol{u}_j^1}{|\boldsymbol{u}_j^0 + \boldsymbol{u}_j^1|}$
    $N_j = ceil(\theta_j / \epsilon)$
    for $l = 0 : N_j$ do
        $\tilde{j} = \tilde{j} + 1$
        $\tilde{\boldsymbol{w}}_{\tilde{j}} = -\cos(l\frac{\theta_j}{N_j})\boldsymbol{u}_j^1 - \sin(2\theta_j - l\frac{\theta_j}{N_j})\boldsymbol{u}_j^0$
        $\tilde{\boldsymbol{v}}_{\tilde{j}} = \boldsymbol{c}_j + s_j \tilde{\boldsymbol{w}}_{\tilde{j}}$, $\tilde{r}_{\tilde{j}} = r_j$ and $\tilde{s}_{\tilde{j}} = s_j$
    end for
end for
$\tilde{j} = \tilde{j} + 1$, $\tilde{M} = \tilde{j}$, $\tilde{\boldsymbol{v}}_{\tilde{M}} = \boldsymbol{v}_M$, $\tilde{r}_{\tilde{M}} = r_M$ and $\tilde{s}_{\tilde{M}} = s_M$
$\tilde{\mathcal{V}} = (\tilde{\boldsymbol{v}}_0, \ldots, \tilde{\boldsymbol{v}}_{\tilde{M}})$,$\tilde{\mathcal{R}} = (\tilde{r}_0, \ldots, \tilde{r}_{\tilde{M}})$,$\tilde{\mathcal{S}} = (\tilde{s}_0, \ldots, \tilde{s}_{\tilde{M}})$
```

4.2. Loss Function from Navier-Stokes Equations

In our research, we move away from traditional numerical techniques like Finite Volume and Finite Element methods. Instead, we explore the fascinating field of neural networks (NNs) to address the intricate problem of coupled fluid-structure interaction. Central to our approach is the use of deep neural networks to approximate solutions for the Navier-Stokes equations. We begin by sampling points in the reference domain Ω_0^{f}. For each point $\boldsymbol{x}_0$ at a given time t, we compute the associated solutions for velocity, pressure, and displacement. To optimize the neural networks for producing accurate solutions, PDE losses with appropriate matching conditions are incorporated as loss functions during training. In this context, we are not fitting to labeled data, but rather seeking a solution

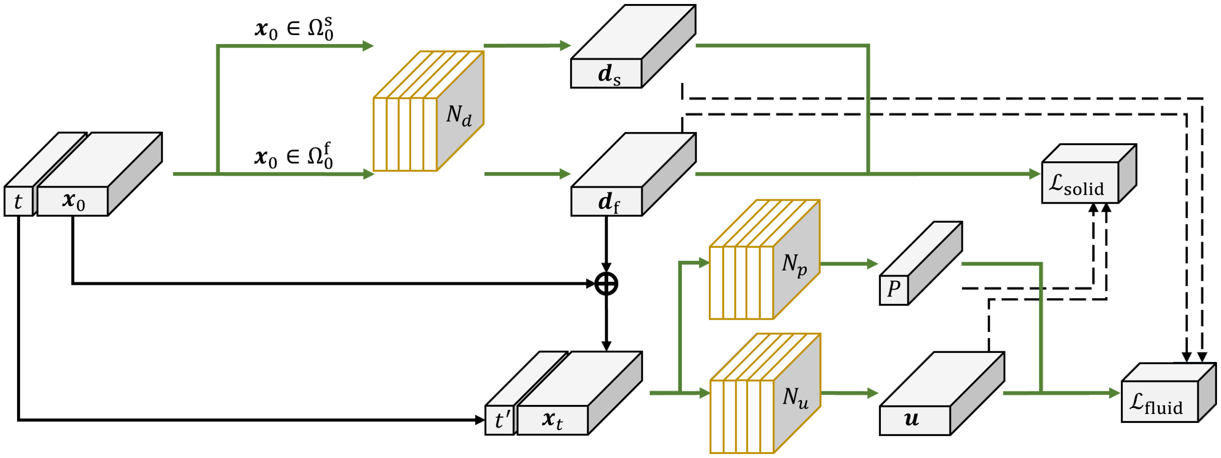


Figure 4: Illustration of Network Architecture for the 3D model: Three networks are employed. The displacement network N_d solves the displacement vector $\boldsymbol{d}$ for both fluid and solid domains. The velocity network N_u computes fluid velocity $\boldsymbol{u}$, while the pressure network N_p handles fluid pressure p. Green lines indicate data flow requiring gradient calculation, while black lines denote no gradient delivery.

that approximately satisfies the PDE system and matching conditions. For a well-trained model, the PDE residuals are close to zero, meaning the physical laws are approximately satisfied throughout the domain.

To be specific, as in Figure 4, we employ a network N_d that takes spatial and temporal coordinates as inputs and produces the displacement at the specified point and time, as given by:

$$\boldsymbol{d}^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) = N_d(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) \quad \text{for } \boldsymbol{x}_0 \in \Omega_t^{\mathrm{f}}; t \in [0, T]. \tag{13}$$

By using the problem coordinates of a point $\boldsymbol{x}_0$ and the computed displacement vector $\boldsymbol{d}^*(\boldsymbol{x}_0, t)$, we can determine the physical coordinates of the point $\boldsymbol{x}_t$ in the current frame at time t as follows:

$$\boldsymbol{x}_t^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) = \boldsymbol{x}_0 + \boldsymbol{d}^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) \quad \text{for } \boldsymbol{x}_0 \in \Omega_t^{\mathrm{f}}; t \in [0, T]. \tag{14}$$

With the acquisition of the coordinates at the current domain Ω_t^{f}, we can compute the fluid velocity and pressure on it (see Figure 4). To do so, we introduce two networks: one for the velocity computation, named N_u, and another for the pressure computation, named N_p. The flow velocity and pressure can be calculated as:

$$\begin{aligned} \boldsymbol{u}^*(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_u, \boldsymbol{\theta}_d) &= N_u(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_u) \\ P^*(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_p, \boldsymbol{\theta}_d) &= N_p(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_p) \end{aligned} \quad \text{for } \boldsymbol{x}_0 \in \Omega_t^{\mathrm{f}}; t \in [0, T]. \tag{15}$$

In this context, we use the symbol t' to represent a separate copy of the value of t. While t and t' hold the same value, they are treated as distinct objects in programming, particularly in our *PyTorch* implementation. This distinction is crucial for differentiating with respect to time in the reference configuration Ω_0^{f} and the current configuration Ω_t^{f}.

Directly applying automatic differentiation to $\boldsymbol{u}^*(\boldsymbol{x}_t^*;t)$ with respect to t would produce a result akin to the material derivative, $\frac{\partial \boldsymbol{u}^*}{\partial t} + \frac{\partial \boldsymbol{d}^*}{\partial t} \cdot \nabla \boldsymbol{u}^*$. By using the distinct variable t', the automatic differentiation of $\boldsymbol{u}^*(\boldsymbol{x}_t^*;t')$ with respect to t' correctly isolates the time dependence.$\frac{\partial \boldsymbol{u}^*}{\partial t'}$.

To enable the training of our neural network, we reformulate these partial differential equations discussed in Section 3 into residual forms. For fluid problem, we rewrite the Navier-Stokes equation and its associated initial/boundary conditions into residual form as:

$$\begin{aligned}\mathcal{L}_{\mathrm{ns}}^{\mathrm{3D}}(\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d) = &||\rho_{\mathrm{f}}\tfrac{\partial}{\partial t'}\boldsymbol{u^*} + \rho_{\mathrm{f}}(\boldsymbol{u}^*\cdot\nabla)\boldsymbol{u}^* - \nabla\cdot\boldsymbol{\sigma}_{\mathrm{f}}^*||^2_{L^2(\Omega_t^{\mathrm{f}};[0,T])}\\ &+ ||\nabla\cdot\boldsymbol{u}^*||^2_{L^2(\Omega_t^{\mathrm{f}};[0,T])},\end{aligned} \tag{16}$$

where $\boldsymbol{\sigma}_{\mathrm{f}}^*$ is computed from $\boldsymbol{u}^*$ and P^* as given in (2), $\boldsymbol{n}$ is the normal vector point outside the outlet boundary.

Regarding the structure problem, we need to rewrite the stress continuity equation (3), its initial and boundary conditions into residual forms. Also, the displacement that extends from the displacement on the interface for the fluid domain is also needed for the calculation of the current domain. Those residual forms are:

$$\mathcal{L}_{\mathrm{sc}}(\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d) = ||\tfrac{\partial^2\eta}{\partial t^2} + b\eta - K||^2_{L^2(\Gamma_0^{\mathrm{w}};[0,T])} + ||\Delta\boldsymbol{d}^*||^2_{L^2(\Omega_0^{\mathrm{f}};[0,T])}, \tag{17}$$

where $\boldsymbol{n}_{\mathrm{s}}$ and $\boldsymbol{n}_{\mathrm{f}}$ are the normal vectors of solid and fluid on their interface. $\boldsymbol{\sigma}_{\mathrm{s}}^*$ is computed from $\boldsymbol{d}^*$ as given in (3).

Integrating the losses from both the flow and deformation problems with proper boundary losses, we obtain:

$$\mathcal{L}_{\mathrm{fluid}}^{\mathrm{3D}}(\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d) = \alpha_{\mathrm{ns}}\mathcal{L}_{\mathrm{ns}}^{\mathrm{3D}} + \text{Boundary losses}, \tag{18}$$

$$\mathcal{L}_{\mathrm{struc}}(\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d) = \alpha_{\mathrm{sc}}\mathcal{L}_{\mathrm{sc}} + \text{Boundary losses}, \tag{19}$$

where α_{ns} is the weight for the Navier-Stokes equations in fluid problem and α_{sc} is the weight for the stress continuity equation in structure problem.

If we minimize the loss $\mathcal{L}_{\mathrm{fluid}}^{\mathrm{3D}}$ to near zero, the physical system described in (2), along with boundary conditions, is approximately satisfied. The networks N_u and N_p then provide valid solutions and are the desired solvers for the modeled problem. Similarly, when $\mathcal{L}_{\mathrm{struc}}$ approaches zero, the stress continuity equation (3), along with the boundary conditions, is effectively satisfied. Our numerical experiments have shown that employing the following multi-objective minimization approach is effective for minimizing both loss functions:

$$\min_{\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d} \mathcal{L}_{\mathrm{struc}}(\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d), \tag{20}$$

and

$$\min_{\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d} \mathcal{L}_{\mathrm{fluid}}^{\mathrm{3D}}(\boldsymbol{\theta}_u,\boldsymbol{\theta}_p,\boldsymbol{\theta}_d). \tag{21}$$

To reduce the loss values, gradient descent methods are employed to update $\boldsymbol{\theta}u$, $\boldsymbol{\theta}p$, and $\boldsymbol{\theta}_d$ with the help of three distinct optimizers by

$$\begin{aligned} \boldsymbol{\theta}_u &\leftarrow \boldsymbol{\theta}_u - \tau_u \nabla_{\boldsymbol{\theta}_u} \mathcal{L}^{\text{3D}}_{\text{fluid}}(\boldsymbol{\theta}_u, \boldsymbol{\theta}_p, \boldsymbol{\theta}_d), \\ \boldsymbol{\theta}_p &\leftarrow \boldsymbol{\theta}_p - \tau_p \nabla_{\boldsymbol{\theta}_p} \mathcal{L}^{\text{3D}}_{\text{fluid}}(\boldsymbol{\theta}_u, \boldsymbol{\theta}_p, \boldsymbol{\theta}_d), \\ \boldsymbol{\theta}_d &\leftarrow \boldsymbol{\theta}_d - \tau_d \nabla_{\boldsymbol{\theta}_d} \mathcal{L}_{\text{struc}}(\boldsymbol{\theta}_u, \boldsymbol{\theta}_p, \boldsymbol{\theta}_d), \end{aligned} \tag{22}$$

where $\nabla_{\boldsymbol{\theta}_u}$, $\nabla_{\boldsymbol{\theta}_p}$ and $\nabla_{\boldsymbol{\theta}_d}$ are the gradients with respect to $\boldsymbol{\theta}_u$, $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_d$. Here, τ_u, τ_p, and τ_d represent adaptive step sizes determined by various methods [54, 55]. In practice, we use three separate *Adam* optimizers for each gradient descent, with step sizes computed according to the approach outlined in [55].

4.3. Periodic Layer

Since cardiovascular dynamics generally exhibit a periodic pattern [45, 44], enforcing periodicity in our model is essential. However, the presence of multiple coupled physical models introduces significant challenges, making it difficult to solve using a standard PINN framework. To address this, incorporating a periodic feature layer [46, 47, 48] is crucial for ensuring the periodicity of the solution and improving the model's accuracy in capturing cyclic physiological behavior. This approach also eliminates the need for initial boundary conditions, which are often challenging to determine in practice.

In physics-informed neural networks (PINNs), a periodic layer [13] is a specialized mechanism designed to incorporate periodic boundary conditions or initial conditions efficiently. To impose the time-periodic constraint $U(z, t + \delta) = U(z, t)$, where $(z, t) \in [0, L] \times [0, T]$, the periodic layer transforms the temporal input t into a representation that inherently satisfies periodicity. This is achieved by expanding the input (z, t) into a set of sine and cosine terms that capture the periodic nature of the system. Specifically, the input transformation takes the form

$$(x, y, z, t) \to \left(x, y, z, \sin\left(\tfrac{2\pi t}{\delta}\right), \cos\left(\tfrac{2\pi t}{\delta}\right), \sin\left(\tfrac{4\pi t}{\delta}\right), \cos\left(\tfrac{4\pi t}{\delta}\right), \ldots, \sin\left(\tfrac{2n\pi t}{\delta}\right), \cos\left(\tfrac{2n\pi t}{\delta}\right)\right), \tag{23}$$

where z represents spatial coordinates or other non-periodic variables, and t is expressed in terms of its fundamental frequency $\frac{2\pi}{\delta}$ and its higher harmonics. The parameter n can be determined based on the specific requirements of the application. This transformation directly encodes periodicity into the input space, allowing the network to model periodic behaviors without explicitly enforcing these conditions during training.

By embedding periodicity into the network's input, the periodic layer provides several benefits. First, it ensures that the network naturally satisfies the time-periodic constraints, reducing the need for additional penalty terms in the loss function. This not only simplifies the training process but also improves convergence by allowing the network.

4.4. Monte-Carlo Integration for Windkessel Model

The implementation of the Windkessel model for boundary conditions in fluid dynamics relies on the flux $Q(l, t)$ at the axial position l. However, the 3D model solve only for the velocity u, and direct computation of $Q(l,t)$ requires integration over the axial cross-sectional area. Given that the PINN approach involves point sampling without connectivity information, Monte Carlo integration emerges as a practical and efficient method to compute this integral. Consequently, we employ this method to convert the resolved velocity field into the flux $Q(l,t)$.

This method relies on randomly sampling points within the axial cross-sectional area $A(l)$ of the vessel. For N random samples $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^N$, the flux $Q^*(l,t)$ is estimated as:

$$Q^*(l,t) \approx \tfrac{A(l)}{N} \sum_{i=1}^{N} u^*(\boldsymbol{x}_i, t_i), \tag{24}$$

where $A(l)$ is the axial cross-sectional area of axial position l. This approach approximates the integral of $u^*(\boldsymbol{x},t)$ over points $\boldsymbol{x}$ lie in the axial cross-section of axial position l by computing the average velocity over the sampled points, scaled by the area. As the number of samples N increases, the accuracy of the approximation improves due to the law of large numbers. In the implementation, the outlet cross-section of the given vessel is fixed without further deformation. Thus, $A(l)$ can be computed and given as a constant value for simplification.

Once the flux $Q(L,t)$ is obtained for the outlet cross-section $l = L$, it can be substituted into the Windkessel model energy:

$$\mathcal{L}_{\text{wks}} = Q^*(L,t)\left(1 + \tfrac{R_1}{R_2}\right) + CR_1 \tfrac{\partial Q^*}{\partial t}(L,t) - \tfrac{P^*(L,t)}{R_2} - C\tfrac{\partial P^*}{\partial t}(L,t). \tag{25}$$

Here, $\frac{\partial Q^*}{\partial t}(L,t)$ and $\frac{\partial P^*}{\partial t}(L,t)$ can be computed using auto gradient methods in *PyTorch* or *TensorFlow*.

4.5. Dynamic Weighting and Training Scheme

The optimization over the modeled fluid-structure interaction problem is challenging due to the coupling of multiple physical systems. To well accomplished this, our approach utilizes three independent networks: N_u, N_p, and N_d. Such configuration makes it even more essential to devise an effective training strategy [56]. In this work, we adopt a dynamic weighting scheme similar to the work of [10]. Initially, we apply zero-initialization to the output layer of N_d, ensuring the solid domain starts in a non-deformed state aligned with the reference domain. With $\alpha_{\text{ns}} = 0$, we alternately optimize N_u and N_p for m_{fluid} epochs, following the scheme outlined in Algorithm 2, to ensure the networks satisfy the initial and boundary conditions.

Once initialized, we train N_u and N_p to solve the Navier-Stokes equations. We start by setting $\alpha_{\text{ns}} = 10^{-7}$ and update it by a factor of 10 after m_{fluid} epochs or once the loss function $\mathcal{L}_{\text{fluid}}$ converges. This process repeats for E iterations (set to 5 in this work). The final training stage couples the fluid and solid systems by alternately solving the fluid and solid problems. We run m_{solid} epochs for the solid and up to m_{fluid} epochs for the fluid, stopping early if convergence is reached. We alternate this process for a maximum of F iterations to ensure stable training and better convergence through the sequential optimization of loss terms.

This sequence of operations that dynamically changes the values of weighting parameters ensures that the networks fulfill the boundary conditions first, which mirrors the finite element method (FEM), where boundary conditions are imposed as hard constraints.

Algorithm 2 Training Scheme

```
Initialize $N_d$ with *zero_initialization*;
$L_{-1} \leftarrow -1$, $L_0 \leftarrow 0$;
$\alpha_{\text{ns}} \leftarrow 0$;
$\boldsymbol{\theta}_u, \boldsymbol{\theta}_p \leftarrow$ Optimize fluid problem by solving (18) for $m_{\text{fluid}}$ epochs;
$\alpha_{\text{ns}} \leftarrow 10^{-7}$;
for i=1:E do
    $\boldsymbol{\theta}_u, \boldsymbol{\theta}_p \leftarrow$ Optimize fluid problem by solving (18) for $m_{\text{fluid}}$ epochs;
    $\alpha_{\text{ns}} \leftarrow 10 \times \alpha_{\text{ns}}$;
end for
for j=1:F and $L_{j-2} < L_{j-1}$ do
    $\boldsymbol{\theta}_d \leftarrow$ Optimize solid problem by solving (19) for $m_{\text{solid}}$ epochs;
    $\boldsymbol{\theta}_u, \boldsymbol{\theta}_p \leftarrow$ Optimize fluid problem by solving (18) for $m_{\text{fluid}}$ epochs;
    $L_j \leftarrow$ The evaluation of $\mathcal{L}^{\text{3D}}_{\text{fluid}}(\boldsymbol{\theta}_u, \boldsymbol{\theta}_p, \boldsymbol{\theta}_d)$ on fixed points;
end for
```

5. Experiment

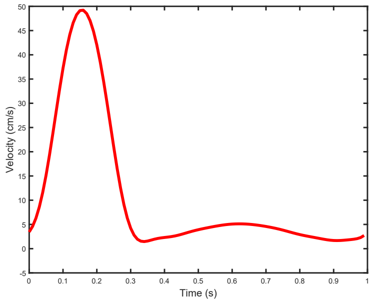


Figure 5: The inlet velocity at the center point of inlet cross-section. The velocity profiles for all models are in parabolic profile expressed in Equation 4.

This section presents numerical results obtained using the proposed PINN solver to simulate blood flow through compliant arteries with varying degrees of bending. The impact of these curved geometries on flow dynamics is evaluated and analyzed. Furthermore, the study explores a straight vessel model, examining the effects of different resistance and compliance parameters in the Windkessel boundary conditions applied at the outlet.

We implemented our methods for all experiments using the *Python* programming language and the *PyTorch* framework with double precision. These experiments were

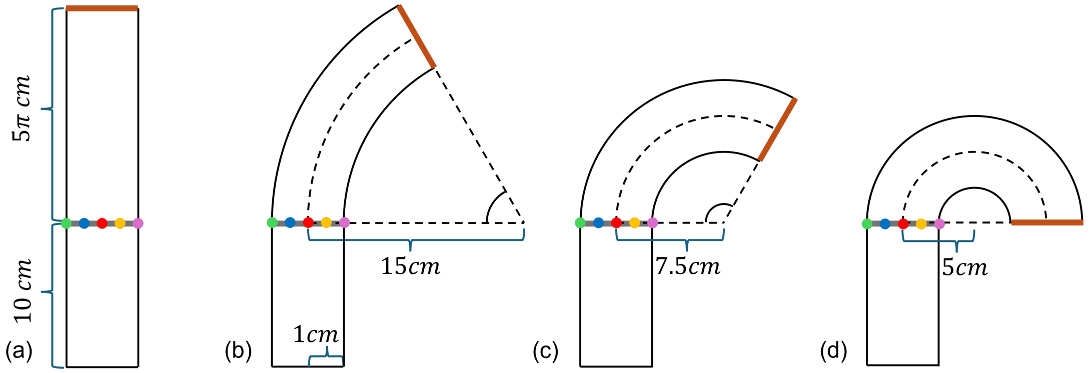


Figure 6: Illustration of the vessel geometry configurations. The blue, red, and yellow markers represent the outer, midpoint, and inner points of each vessel lumen, respectively. The green and purple markers denote the midpoints on the outer and inner walls, respectively. All geometries share the same centerline length.

conducted on a Linux server equipped with a Tesla A6000 Graphics Processing Unit. For the network architecture in the cylindrical and 3D models, the depth of the neural networks for N_u, N_p, and N_d was set to 12. The width of N_u and N_d was configured to 20, while N_p had a width of 10. In the case of the 1D model, the networks for N_Q and N_P were both designed with a depth of 12 and a width of 20. Additionally, a periodic layer was used as the first layer of all the networks, using $n = 3$ as defined in (23). This approach effectively transforms the inputs into periodic features, preparing them for further processing in subsequent layers.

Regarding the point sampling, we independently sample 1000 points for the vessel domain, 1000 for the wall boundary, 1000 for the inlet, and 1000 for the outlet in each training epoch. That is, in every epoch, the points are uniformly and randomly selected as described in Section 4.1. Over the course of training, this approach ensures that the loss is approximately minimized across the entire domain of interest.

5.1. Full Model Validation

To validate the numerical correctness of our method, we conducted a comparative experiment against the Finite Element Method (FEM) applied to the same full 3D problem setup. The material parameters and geometry used in this study are detailed as follows: The fluid density ρ_{f} is specified as 1.06 g/cm^3, and the viscosity μ_{f} is set at 0.04 poise. The vessel has a circular axial cross-section, with a centerline length l as $5\pi + 10$ cm, a radius r_0 as 1 cm, and a flow period of 1 second. The inlet boundary condition is illustrated in Figure 5. We defined the inlet flow based on the example provided in [57, P. 13]. The inflow waveform is first measured and then fitted to a truncated Fourier series, allowing the flow rate to be queried at arbitrary time points [58, 59]. Once the waveform is specified, it is mapped onto the inlet plane using a parabolic profile, as described in Equation (4).

For the outlet, a Windkessel model was applied with the following parameters: proximal resistance R_1 as 6708.98 dyn·s/cm^5, distal resistance R_2 as 17251.66 dyn·s/cm^5, and compliance C as 1.235×10^{-4} cm^5/dyn. These settings mimic the mechanical properties

Relative Error on Velocity Magnitude			
Geometry	60° curved	120° curved	180° curved
Ours	0.0337	0.0261	0.0294
FEM	0.0264	0.0290	0.0331
Relative Error on Pressure Magnitude			
Geometry	60° curved	120° curved	180° curved
Ours	0.0263	0.0293	0.0338
FEM	0.0288	0.0255	0.0285
Computation Time (seconds)			
Geometry	60° curved	120° curved	180° curved
Ours	1819	2034	2166
FEM	1769	1984	2371

Table 1: Comparison between full 3D model and FEM, reporting relative errors for velocity magnitude and pressure. The time step for FEM method is 0.001 s. We compare our method with FEM using a mesh generated with a maximum edge length of 0.30 cm. To compute the relative error, the solution obtained from a much denser mesh with a maximum edge length of 0.10 cm is used as the ground truth.

of the common carotid artery of a young female [59, 57]. The structural parameters of the vessel include a density ρ_s as 1.2 g/cm^3, Young's modulus E_s as 5×10^6 dynes/cm^2, Poisson's ratio ξ_s as 0.5, and wall thickness h as 0.1 cm.

Table 1 presents a quantitative comparison between our proposed method and the Finite Element Method (FEM) for solving problems in vessels with varying degrees of bending (60°, 120° and 180°). The vessel geometries consist of a straight segment of 10cm and a curved segment of 5πcm, as depicted in (b), (c) and (d) of Figure 6. The FEM implementation was conducted using the open-source software *SimVascular* [57] using the Taylor–Hood elements for velocity–pressure discretization. The time step for the FEM method is 0.001 *s*. We compare our method with FEM using a mesh generated with a maximum edge length of 0.30 *cm*. To compute the relative error, the solution obtained from a much denser mesh with a maximum edge length of 0.10 *cm* is used as the ground truth.

The results show that our method achieves error levels comparable to FEM across various bending vessel geometries. In terms of computational efficiency, our approach remains robust even for complex geometries with significant curvature, whereas FEM exhibits a marked increase in computational time as the geometry becomes more intricate with the curvature degree comes to 180°.

5.2. Full and Reduced-Order Models on Curved Vessel

As the curvature of bending vessels significantly influences blood flow dynamics [60, 61], this section begins by examining the effects induced by various degrees of bending vessels under consistent conditions, as in the previous experiment. Also, we compare the solutions obtained from our proposed PINN-based full 3D models with those from

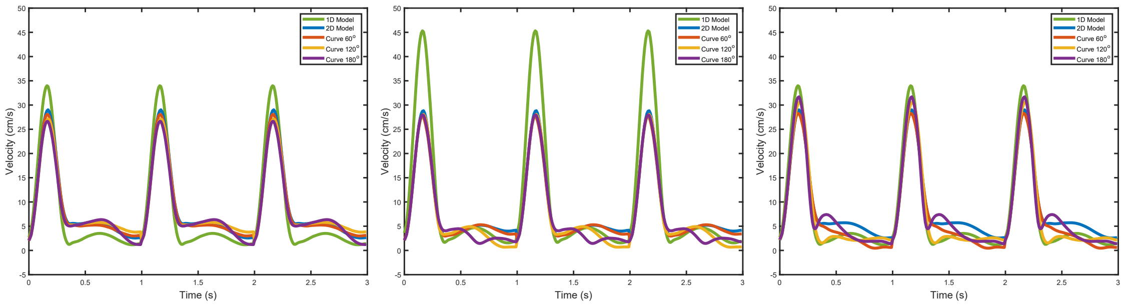


Figure 7: The velocity magnitude profile comparisons. From left to right, they correspond to the outer (blue), middle (red), and inner (yellow) points, as indicated in Figure 6. The discrepancy between the full model and the reduced-order models increases with greater vessel curvature. The 1D model, in particular, exhibits significant differences due to its use of a pre-defined velocity profile.

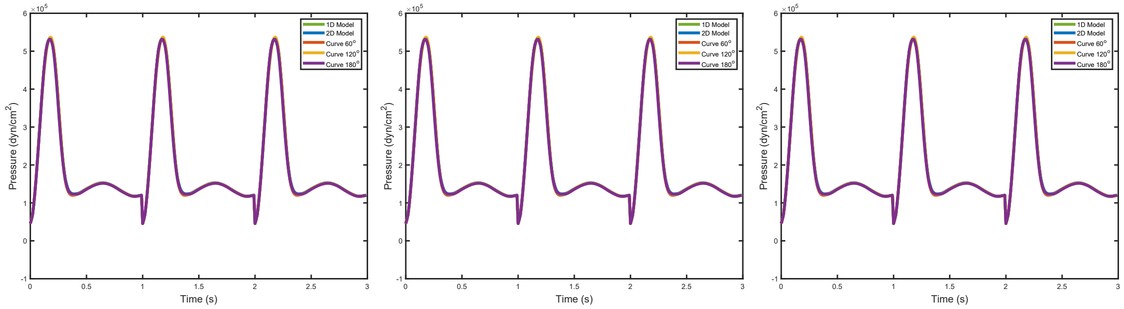


Figure 8: The pressure profile comparisons. From left to right, they correspond to the outer (blue), middle (red), and inner (yellow) points, as indicated in Figure 6. The pressure profiles do not exhibit significant differences across the models, given the specified inlet condition in Figure 5 and the application of the Windkessel outlet boundary condition.

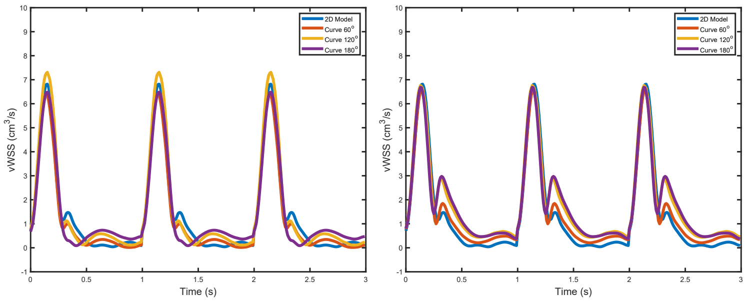


Figure 9: The vertical Wall Shear Stress profile comparisons. From left to right, they correspond to the outer (green) and inner (orange) points, as indicated in Figure 6. The vWSS response varies across different vessel geometries, while the cylindrical model fails to capture these differences. The 1D model does not provide sufficient information for vWSS calculations and is therefore not included in this comparison.

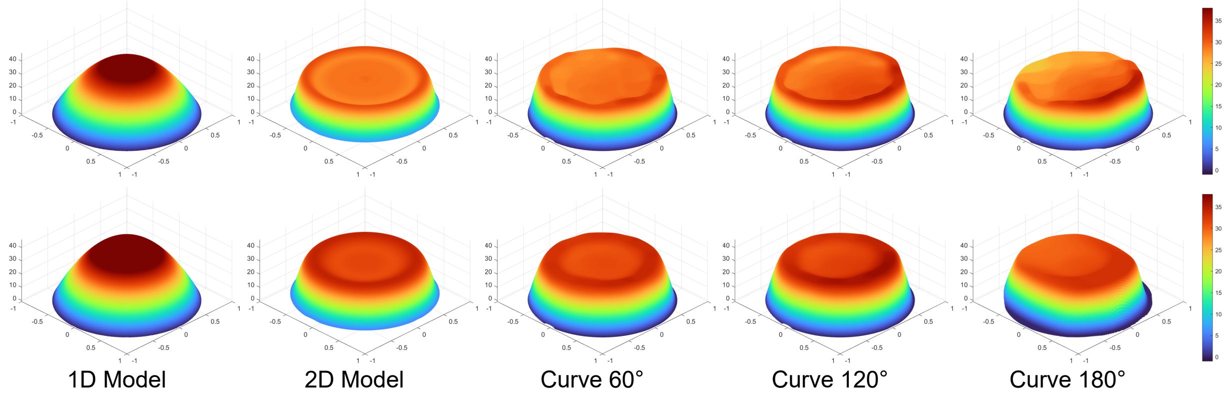


Figure 10: Illustration of velocity magnitude at the midsection and outlet cross-sections, denoted by gray and orange lines in Figure 6. The color indicates the velocity magnitude.

Model Order	1D Model	Cyl. Model	3D Model
Comsumed time /s	312	1326	1723

Table 2: Comparison of computation times for simulating blood flow in the straight cylindrical vessel across models of varying order.

PINN-based reduced-order models (1D and 2D) to assess their geometric expressiveness and computational efficiency.

Figure 6 depicts four geometric configurations for curved vessels, all with identical centerline lengths. The first configuration (a) features a straight centerline. The second configuration (b) includes a $\frac{1}{6}$ circular arc with a radius of 15 cm, preceded by a 10 cm straight segment. The third geometry (c) contains a $\frac{1}{3}$ circular arc with a radius of 7.5 cm following a straight segment, while the fourth configuration (d) features a $\frac{1}{2}$ circular arc with a 5 cm radius preceding the straight segment. Notably, the 1D and cylindrical models are parameterized solely by axial position for the longitudinal sense, which cannot capture the curvature of the vessels. Consequently, these models are effectively computed using the geometry of the straight vessel in the first configuration.

Figures 7 and 8 display the velocity history and pressure profile at the midpoint of the vessel. From left to right, they correspond to the outer (blue), middle (red), and inner (yellow) points, as indicated in Figure 6. In Figure 7, more pronounced differences in velocity profiles emerge with increasing vessel curvature, deviating significantly from the inlet flow profile (Figure 5). The 1D model, which assumes a parabolic velocity profile within each axial cross-section, fails to capture the nuanced effects of vessel curvature. This limitation arises because the 1D model is derived under the assumption that the vessel is a straight tube, inherently neglecting geometric features such as bends or curves that influence the flow dynamics.

Figure 8, organized similarly with outer (blue), middle (red), and inner (yellow) profiles, shows minimal differences across these points. This indicates the robustness of the reduced-order model in capturing pressure distributions, particularly under the application of the Windkessel model as the outlet boundary condition. Figure 9 illustrates the vertical Wall Shear Stress (vWSS) profiles at the outer (green) and inner (purple) vessel walls, as shown in Figure 6. Notably, the 1D model cannot compute such quantities due to the absence of spatially resolved velocity and pressure fields. The cylindrical model, constrained by its inability to account for vessel curvature, produces symmetric wall stress profiles, missing the asymmetry inherent in curved geometries.

Furthermore, Figure 10 provides a visual representation of velocity magnitude at the midsection and outlet cross-sections, denoted by gray and orange lines in Figure 6, respectively. The 1D model, restricted by its parabolic assumption [5, P. 92], shows no discernible pattern variations. Similarly, the cylindrical model lacks the capacity to capture curvature effects. In contrast, the 3D models reveal a clear trend where velocity accumulates more significantly on the inner curvature of the vessel as curvature increases, resulting in higher momentum concentration in these regions.

However, the computational time for reduced-order models is significantly shorter compared to the full model. To demonstrate this, we compare the time required to simulate blood flow in the straight cylindrical vessel depicted in configuration (a) of Figure 6 using 1D, cylindrical, and 3D models. As shown in Table 2, high-dimensional models exhibit substantially higher computational demands and slower convergence rates due to their ability to capture finer details and complex flow dynamics. In contrast, reduced-order models, such as 1D and cylindrical formulations, offer notable advantages in terms of efficiency, making them ideal for scenarios where computational resources are limited or rapid solutions are required. This efficiency does come with trade-offs. While reduced-order models provide a reasonable approximation of the overall flow behavior, they lack the resolution and accuracy of full models, particularly in capturing intricate flow phenomena and localized effects. The full models, by offering a more comprehensive representation, are indispensable for cases where precise detail is critical, such as analyzing blood flow in highly curved or pathological vessels.

5.3. Full and Reduced-Order Models on Curved Stenosis Vessels

The last experiments on blood flow simulation in curved vessels have demonstrated the advantages of the higher-order model in accurately resolving detailed pressure and velocity distributions, while the reduced-order model excels in providing faster computational results. The reduced-order model is particularly useful for scenarios where computational efficiency is the primary concern, such as real-time simulations or large-scale parameter studies. However, when detailed hemodynamic analysis is required—such as in cases involving complex vessel geometries or localized flow disturbances—the full 3D model becomes essential. Here, we aim to further evaluate the capability of the full model to capture intricate details of blood mechanics inside vessels. As the 1D model is obviously not able to capture fine details. We only compare the cylindrical and full 3D models in this section.

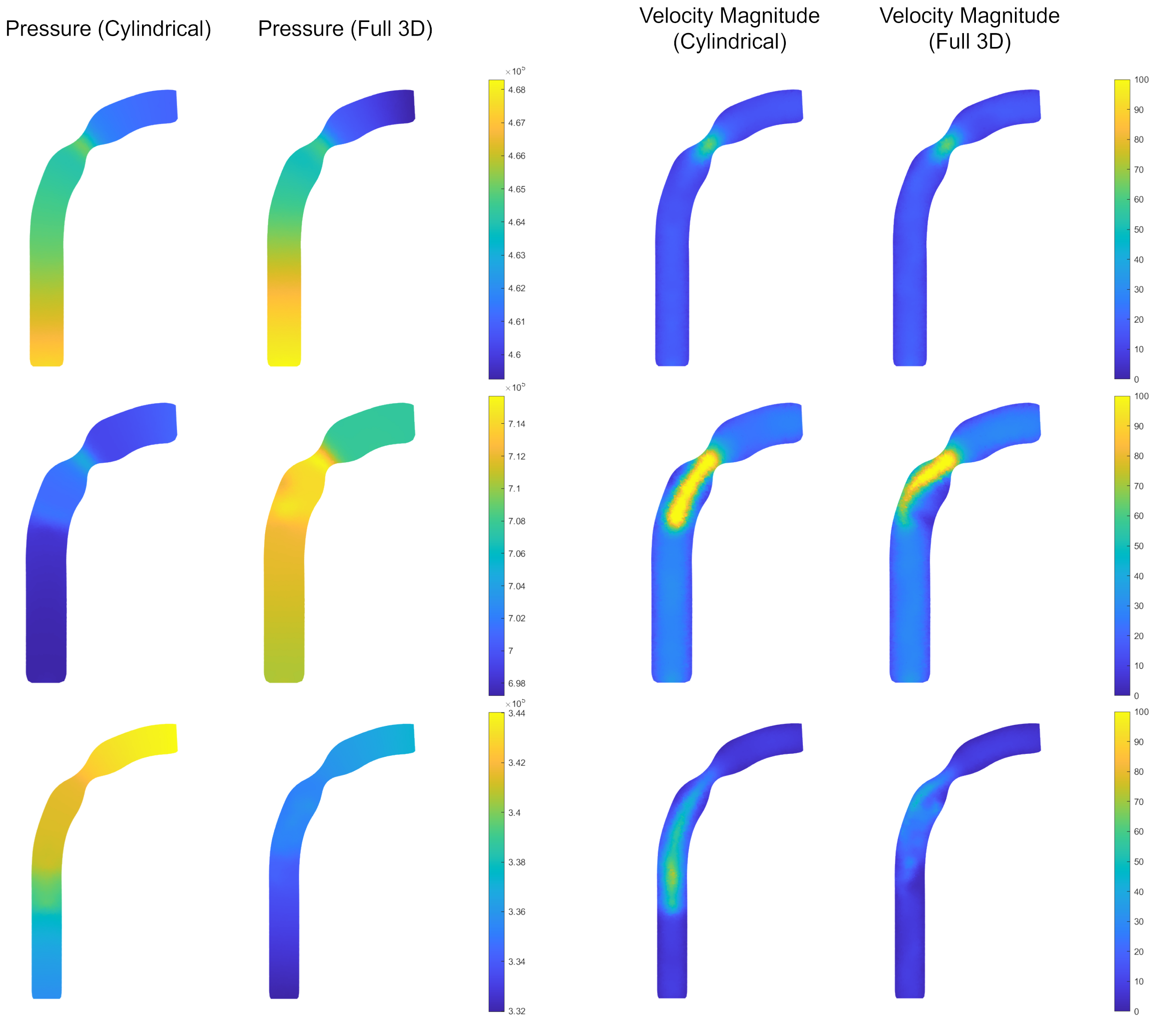


Figure 11: Longitudinal cross-sectional visualization of pressure and velocity magnitude in a 90^o curved vessel. From top to bottom, the results from the cylindrical model (first and third columns) and the full 3D model (second and fourth columns) are shown at 0.1s, 0.2s, and 0.3s within a period.

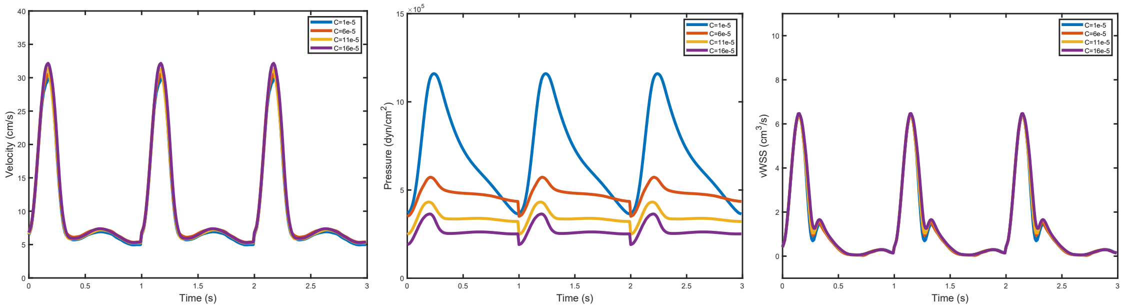


Figure 12: Historical profiles of velocity magnitude, pressure, and vertical wall shear stress (vWSS) for different compliance parameter C, in the Windkessel boundary condition. Velocity and pressure are measured at the red point in Figure 6(a), while vWSS is measured at the green point.

For the numerical experiments, we consider a 90° curved vessel following a straight segment, a common geometry in vascular structures. The fluid properties are set to mimic physiological conditions, with a density of 1.06 g/cm^3 and viscosity of 0.04 poise. The vessel centerline length, radius, and flow period are chosen as $10\pi + 10$ cm, 1 cm, and 1 second, respectively. The inlet boundary condition is identical to that in Figure 5. The vessel wall is modeled with structural properties resembling arterial tissue, with a density of 1.2 g/cm^3, a Young's modulus of 7×10^6 dynes/cm^2, a Poisson's ratio of 0.5, and a thickness of 0.1 cm.

We employ both cylindrical and full 3D models to simulate blood flow. Figure 11 illustrates the pressure and velocity magnitude across the longitudinal cross-section of the vessel at different time points—0.1s, 0.2s, and 0.3s—within a flow period. Observations indicate that the cylindrical model produces results that remain invariant among axial cross-sections orthogonal to the centerline. In contrast, the full 3D model accounts for asymmetric flow characteristics and captures finer details, particularly at 0.2s, and 0.3s, where the accumulated momentum of blood flow leads to significant variations in velocity and pressure, especially in the outer region of the curved segment.

The differences between the cylindrical and full 3D models become more apparent when analyzing the pressure distribution. Over the three time frames, the full 3D model reveals greater pressure accumulation in the stenosed region, which is followed by a more rapid release of accumulated stress compared to the cylindrical model. This behavior is attributed to the complex interplay between convective and inertial effects, which cannot be fully captured using a reduced-order approach. Even though it might bring more computation consumption, these results underscore the necessity of employing a full 3D model when precise hemodynamic details are required, such as in the assessment of vascular diseases, aneurysm formation, or the optimization of medical interventions like stents and bypass grafts.

5.4. Response to Windkessel Parameters

Windkessel models have become integral to blood flow simulations, making the choice of parameters critical to accurately capturing hemodynamic behaviors [49]. In this study,

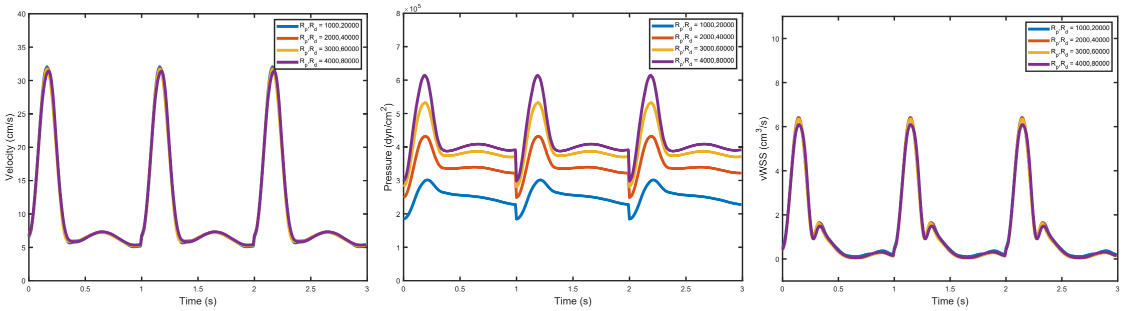


Figure 13: Historical profiles of velocity magnitude, pressure, and vertical wall shear stress (vWSS) for different resistance parameters, R_1 and R_2, in the Windkessel boundary condition. Velocity and pressure are measured at the red point in Figure 6(a), while vWSS is measured at the green point.

we investigate how different Windkessel parameters influence the resulting fluid dynamics. The analysis is conducted using the straight vessel geometry depicted as Figure 6.(a). The fluid density, viscosity, vessel centerline length, vessel radius, and period of the flow are set to be 1.06 g/cm^3, 0.04 poise, $5\pi + 10$ cm, 1 cm and 1 second as the previous. With the same inlet boundary condition as Figure 5, the vessel structure density, Young modulus, Poisson's ratio, and vessel wall thickness are set as 1.2 g/cm^3, 5×10^6 dynes/cm^2, 0.5 and 0.1 cm.

To investigate the impact of compliance, we fix the proximal resistance at $R_1 = 2000$ dyn $\cdot$ s/cm^5 and the distal resistance at $R_2 = 40000$ dyn $\cdot$ s/cm^5. The compliance values of the artery are varied among $C = 1 \times 10^{-5}$, 6×10^{-5}, 11×10^{-5}, and 16×10^{-5} cm^5/dyn. As compliance increases, the artery exhibits enhanced buffering effects, which reduce rapid fluctuations in pressure. This results in smoother pressure profiles and delays in the peaks of pressure waves, as greater compliance allows for more significant energy storage and dissipation. Consequently, the system absorbs and redistributes the energy of pressure waves more effectively, moderating their amplitude and shifting their timing.

To examine the impact of resistance in the Windkessel model, the compliance is fixed at $C = 11 \times 10^{-5}$ cm^5/dyn, while the proximal and distal resistance pairs (R_1, R_2) are varied as $(1000, 20000)$, $(2000, 40000)$, $(3000, 60000)$ and $(4000, 80000)$ dyn $\cdot$ s/cm^5. Figures 13 illustrate, from left to right, the historical profiles for velocity magnitude, pressure at the center of the middle axial cross-section, and vertical wall shear stress (vWSS) on the vessel wall at that section. The velocity magnitude and vWSS show only minor differences across the varying resistance values. However, the pressure profile exhibits significantly more pronounced changes. As the resistances increase, pressure oscillations intensify due to enhanced reflection of pressure waves. Additionally, the overall pressure magnitude rises, as higher resistance impedes the outward flow, creating greater opposition to the fluid dynamics.

Based on these observations, the variations across different parameters are systematically analyzed, offering a framework for approximating how quantities such as velocity, pressure, and wall shear stress are influenced by varying the magnitude of Windkessel model parameters. Conversely, this approach also facilitates approximate estimations of

Windkessel parameters based on observed changes in fluid dynamics.

6. Conclusion

In this work, we proposed a parameterized sampling strategy specifically designed for full 3D blood flow simulations. This strategy ensures the smoothness required for differentiation calculations across interface surfaces where blood interacts with vessel walls. A periodic feature layer is employed to enforce periodicity in the simulation, while Windkessel boundary conditions are implemented using Monte Carlo integration for the 3D model. To effectively optimize problems involving multiple loss terms, we introduce a dynamic weighting strategy combined with an alternative optimization approach.

Through extensive experiments across various vessel geometries with different degrees of curvature, we validated the computational accuracy of the full 3D model, demonstrating performance comparable to conventional FEM methods. We systematically compared the full model with reduced-order models based on cylindrical and 1D simplifications. While the full model excels at capturing intricate flow dynamics, the reduced-order models offer faster computations. Further experiments on stenosed curved vessels highlight the superior capability of the full model in resolving fine details compared to simplified cylindrical models. Additionally, we investigated the impact of different Windkessel model parameters, providing deeper insights into how they influence the magnitude and characteristics of blood flow dynamics.

Future work will focus on extending the framework to accommodate more complex and patient-specific geometries, including geometries with vessel branches. We also plan to explore the development of pre-trained models to accelerate convergence and further improve computational efficiency.

Acknowledgement

The works of Han Zhang and Lingfeng Li are supported by the InnoHK project at Hong Kong Centre for Cerebro-cardiovascular Health Engineering (COCHE). The work of Xue-Cheng Tai is partially supported by HKRGC-NSFC Grant N-CityU214/19, HKRGC CRF. Grant C1013-21GF and NORCE Kompetanseoppbygging program. This work of R. Chan is partially supported by HKRGC Grants No. CityU11301120, C1013-21GF, CityU11309922, ITF Grant No. MHP/054/22, and LU BGR 105824.

References

[1] N. H. Pijls, B. de Bruyne, K. Peels, P. H. van der Voort, H. J. Bonnier, J. Bartunek, J. J. Koolen, Measurement of fractional flow reserve to assess the functional severity of coronary-artery stenoses, New England Journal of Medicine 334 (26) (1996) 1703–1708.

[2] Z. Yan, Z. Yao, W. Guo, D. Shang, R. Chen, J. Liu, X.-C. Cai, J. Ge, Impact of pressure wire on fractional flow reserve and hemodynamics of the coronary arteries: A computational and clinical study, IEEE Transactions on Biomedical Engineering 70 (5) (2022) 1683–1691.

[3] C. S. Peskin, Flow patterns around heart valves: A numerical method, Journal of Computational Physics 10 (2) (1972) 252–271. doi:https://doi.org/10.1016/0021-9991(72)90065-4.

[4] J. Moore, D. Steinman, D. Holdsworth, C. Ethier, Accuracy of computational hemodynamics in complex arterial geometries reconstructed from magnetic resonance imaging, Annals of Biomedical Engineering 27 (1999) 32–41.

[5] A. Quarteroni, L. Formaggia, Mathematical modelling and numerical simulation of the cardiovascular system, Handbook of Numerical Analysis 12 (2004) 3–127.

[6] T. Wick, Fluid-structure interactions using different mesh motion techniques, Computers & Structures 89 (13-14) (2011) 1456–1467.

[7] W.-K. Sun, L.-W. Zhang, K. Liew, A coupled sph-pd model for fluid-structure interaction in an irregular channel flow considering the structural failure, Computer Methods in Applied Mechanics and Engineering 401 (2022) 115573.

[8] 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, Journal of Computational physics 378 (2019) 686–707.

[9] J. D. Toscano, V. Oommen, A. J. Varghese, Z. Zou, N. Ahmadi Daryakenari, C. Wu, G. E. Karniadakis, From pinns to pikans: Recent advances in physics-informed machine learning, Machine Learning for Computational Science and Engineering 1 (1) (2025) 1–43.

[10] H. Zhang, R. H. Chan, X.-C. Tai, A meshless solver for blood flow simulations in elastic vessels using a physics-informed neural network, SIAM Journal on Scientific Computing 46 (4) (2024) C479–C507.

[11] H. Zhang, X.-C. Tai, Full 3d blood flow simulation in curved deformable vessels using physics-informed neural networks, Acta Mathematica Universitatis Comenianae 93 (4) (2024) 235–250.

[12] L. Li, X.-C. Tai, R. H.-F. Chan, A new method to compute the blood flow equations using the physics-informed neural operator, Journal of Computational Physics 519 (2024) 113380.

[13] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, S. G. Johnson, Physics-informed neural networks with hard constraints for inverse design, SIAM Journal on Scientific Computing 43 (6) (2021) B1105–B1132.

[14] L. Formaggia, J.-F. Gerbeau, F. Nobile, A. Quarteroni, On the coupling of 3d and 1d navier–stokes equations for flow problems in compliant vessels, Computer Methods in Applied Mechanics and Engineering 191 (6-7) (2001) 561–582.

[15] P. Crosetto, S. Deparis, G. Fourestey, A. Quarteroni, Parallel algorithms for fluid-structure interaction problems in haemodynamics, SIAM Journal on Scientific Computing 33 (4) (2011) 1598–1622.

[16] A. T. Barker, X.-C. Cai, Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling, Journal of Computational Physics 229 (3) (2010) 642–659.

[17] M. Bukač, S. Čanić, R. Glowinski, J. Tambača, A. Quaini, Fluid-structure interaction in blood flow capturing non-zero longitudinal structure displacement, Journal of Computational Physics 235 (2013) 515–541.

[18] V. Chabannes, G. Pena, C. Prud'Homme, High-order fluid–structure interaction in 2d and 3d application to blood flow in arteries, Journal of Computational and Applied Mathematics 246 (2013) 1–9.

[19] J. Janela, A. Moura, A. Sequeira, A 3d non-newtonian fluid–structure interaction model for blood flow in arteries, Journal of Computational and applied Mathematics 234 (9) (2010) 2783–2791.

[20] M. A. Ikbal, S. Chakravarty, K. K. Wong, J. Mazumdar, P. K. Mandal, Unsteady response of non-newtonian blood flow through a stenosed artery in magnetic field, Journal of Computational and Applied Mathematics 230 (1) (2009) 243–259.

[21] S. Katz, A. Caiazzo, V. John, Impact of viscosity modeling on the simulation of aortic blood flow, Journal of Computational and Applied Mathematics 425 (2023) 115036.

[22] Y. Bazilevs, V. M. Calo, T. J. Hughes, Y. Zhang, Isogeometric fluid-structure interaction: theory, algorithms, and computations, Computational Mechanics 43 (2008) 3–37.

[23] T. E. Tezduyar, S. Sathe, T. Cragin, B. Nanna, B. S. Conklin, J. Pausewang, M. Schwaab, Modelling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics, International Journal for Numerical Methods in Fluids 54 (6-8) (2007) 901–922.

[24] Y. Wang, A. Quaini, S. Čanić, A higher-order discontinuous galerkin/arbitrary lagrangian eulerian partitioned approach to solving fluid–structure interaction problems with incompressible, viscous fluids and elastic structures, Journal of Scientific Computing 76 (2018) 481–520.

[25] H. Wang, J. Chessa, W. K. Liu, T. Belytschko, The immersed/fictitious element method for fluid–structure interaction: volumetric consistency, compressibility and thin members, International Journal for Numerical Methods in Engineering 74 (1) (2008) 32–55.

[26] L. Ge, F. Sotiropoulos, A numerical method for solving the 3d unsteady incompressible navier–stokes equations in curvilinear domains with complex immersed boundaries, Journal of Computational Physics 225 (2) (2007) 1782–1809.

[27] C. A. Figueroa, I. E. Vignon-Clementel, K. E. Jansen, T. J. Hughes, C. A. Taylor, A coupled momentum method for modeling blood flow in three-dimensional deformable arteries, Computer Methods in Applied Mechanics and Engineering 195 (41-43) (2006) 5685–5706.

[28] M. Zhou, O. Sahni, H. J. Kim, C. A. Figueroa, C. A. Taylor, M. S. Shephard, K. E. Jansen, Cardiovascular flow simulation at extreme scale, Computational Mechanics 46 (2010) 71–82.

[29] J. D. Osorio, M. De Florio, R. Hovsapian, C. Chryssostomidis, G. E. Karniadakis, Physics-informed machine learning for solar-thermal power systems, Energy Conversion and Management 327 (2025) 119542.

[30] N. Borrel-Jensen, S. Goswami, A. P. Engsig-Karup, G. E. Karniadakis, C.-H. Jeong, Sound propagation in realistic interactive 3d scenes with parameterized sources using deep neural operators, Proceedings of the National Academy of Sciences 121 (2) (2024) e2312159120.

[31] M. A. Ur Rehman, O. Ekici, M. A. Farooq, K. Butt, M. Ajao-Olarinoye, Z. Wang, H. Liu, Fluid–structure interaction analysis of pulsatile flow in arterial aneurysms with physics-informed neural networks and computational fluid dynamics, Physics of Fluids 37 (3) (2025).

[32] M. Ng, H. M. Yao, Deep learning based source reconstruction method using asymmetric encoder–decoder structure and physics-induced loss, Journal of Computational and Applied Mathematics 438 (2024) 115503.

[33] G. Li, Y. Qiang, H. Li, X. Li, P. A. Buffet, M. Dao, G. E. Karniadakis, A combined computational and experimental investigation of the filtration function of splenic macrophages in sickle cell disease, PLoS Computational Biology 19 (12) (2023) e1011223.

[34] X. Li, J. Wu, Y. Huang, Z. Ding, X. Tai, L. Liu, Y.-G. Wang, Fourier-feature induced physics informed randomized neural network method to solve the biharmonic equation, Journal of Computational and Applied Mathematics 468 (2025) 116635.

[35] M. De Florio, E. Schiassi, F. Calabrò, R. Furfaro, Physics-informed neural networks for 2nd order odes with sharp gradients, Journal of Computational and Applied Mathematics 436 (2024) 115396.

[36] L. Lu, P. Jin, G. E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, arXiv preprint arXiv:1910.03193 (2019).

[37] V. Kumar, S. Goswami, K. Kontolati, M. D. Shields, G. E. Karniadakis, Synergistic learning with multi-task deeponet for efficient pde problem solving, Neural Networks 184 (2025) 107113.

[38] A. Kopaničáková, G. E. Karniadakis, Deeponet based preconditioning strategies for solving parametric linear systems of equations, SIAM Journal on Scientific Computing 47 (1) (2025) C151–C181.

[39] S. Wang, G. E. Karniadakis, Gmc-pinns: A new general monte carlo pinns method for solving fractional partial differential equations on irregular domains, Computer Methods in Applied Mechanics and Engineering 429 (2024) 117189.

[40] Y. Wang, J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, Y. Liu, Kolmogorov–arnold-informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on kolmogorov–arnold networks, Computer Methods in Applied Mechanics and Engineering 433 (2025) 117518.

[41] Y. Gu, M. K. Ng, Deep neural networks for solving large linear systems arising from high-dimensional problems, SIAM Journal on Scientific Computing 45 (5) (2023) A2356–A2381.

[42] F. Griese, F. Hoppe, A. Rüttgers, P. Knechtges, Preconditioned fem-based neural networks for solving incompressible fluid flows and related inverse problems, Journal of Computational and Applied Mathematics (2025) 116663.

[43] A. Yazdani, L. Lu, M. Raissi, G. E. Karniadakis, Systems biology informed deep learning for inferring parameters and hidden dynamics, PLoS computational biology 16 (11) (2020) e1007575.

[44] Y. M. Belousov, V. P. Krainov, S. Revenko, Hydrodynamic model of blood flow in major arteries pulsing in various modes, Medical & Biological Engineering & Computing 59 (9) (2021) 1785–1794.

[45] S. Natarajan, M. Mokhtarzadeh-Dehghan, A numerical and experimental study of periodic flow in a model of a corrugated vessel with application to stented arteries, Medical engineering & physics 22 (8) (2000) 555–566.

[46] N. Ahmadi Daryakenari, M. De Florio, K. Shukla, G. E. Karniadakis, Ai-aristotle: A physics-informed framework for systems biology gray-box identification, PLOS Computational Biology 20 (3) (2024) e1011916.

[47] N. A. Daryakenari, S. Wang, G. Karniadakis, Cminns: Compartment model informed neural networks—unlocking drug dynamics, Computers in Biology and Medicine 184 (2025) 109392.

[48] Q. A. Huhn, M. E. Tano, J. C. Ragusa, Physics-informed neural network with fourier features for radiation transport in heterogeneous media, Nuclear science and engineering 197 (9) (2023) 2484–2497.

[49] N. Westerhof, J.-W. Lankhaar, B. E. Westerhof, The arterial windkessel, Medical & Biological Engineering & Computing 47 (2) (2009) 131–141.

[50] J. Alastruey, A. W. Khir, K. S. Matthys, P. Segers, S. J. Sherwin, P. R. Verdonck, K. H. Parker, J. Peiró, Pulse wave propagation in a model human arterial network: assessment of 1-d visco-elastic simulations against in vitro measurements, Journal of Biomechanics 44 (12) (2011) 2250–2258.

[51] N. Xiao, J. Alastruey, C. Alberto Figueroa, A systematic comparison between 1-d and 3-d hemodynamics in compliant arterial models, International Journal for Numerical Methods in Biomedical Engineering 30 (2) (2014) 204–231.

[52] T.-C. Lee, R. L. Kashyap, C.-N. Chu, Building skeleton models via 3-d medial surface axis thinning algorithms, CVGIP: graphical models and image processing 56 (6) (1994) 462–478.

[53] H. I. Choi, S. W. Choi, H. P. Moon, Mathematical theory of medial axis transform, Pacific Journal of Mathematics 181 (1) (1997) 57–88.

[54] S.-i. Amari, Backpropagation and stochastic gradient gescent method, Neurocomputing 5 (4-5) (1993) 185–196.

[55] D. Kingma, J. Ba, Adam: A method for stochastic optimization, Proc. of IEEE Int. Conf. on Learning Representation 12 (2015).

[56] R. Bischof, M. A. Kraus, Multi-objective loss balancing for physics-informed deep learning, Computer Methods in Applied Mechanics and Engineering 439 (2025) 117914.

[57] N. Freidoonimehr, M. Arjomandi, A. Zander, R. Chin, SimVascular clinical test cases : Aortofemoral normal - 2, https://simvascular.github.io/clinical/aortofemoral2.html (2025).

[58] N. M. Wilson, F. R. Arko, C. A. Taylor, Predicting changes in blood flow in patient-specific operative plans for treating aortoiliac occlusive disease, Computer Aided Surgery 10 (4) (2005) 257–277.

[59] A. Updegrove, N. M. Wilson, J. Merkow, H. Lan, A. L. Marsden, S. C. Shadden, Simvascular: an open source pipeline for cardiovascular simulation, Annals of Biomedical Engineering 45 (2017) 525–541.

[60] K. Govindaraju, G. N. Viswanathan, I. A. Badruddin, S. Kamangar, N. Salman Ahmed, A. A. Al-Rashed, The influence of artery wall curvature on the anatomical assessment of stenosis severity derived from fractional flow reserve: a computational fluid dynamics

study, Computer Methods in Biomechanics and Biomedical Engineering 19 (14) (2016) 1541–1549.

[61] N. Freidoonimehr, M. Arjomandi, A. Zander, R. Chin, Effect of artery curvature on the coronary fractional flow reserve, Physics of Fluids 33 (3) (2021).