A Meshless Solver for Blood Flow Simulations in Elastic Vessels Using Physics-Informed Neural Network [*]

Han Zhang [†], Raymond Chan [‡], and Xue-Cheng Tai [§]

Abstract.
Investigating blood flow in the cardiovascular system is crucial for assessing cardiovascular health. Computational approaches offer some non-invasive alternatives to measure blood flow dynamics. Numerical simulations based on traditional methods such as finite-element and other numerical discretizations have been extensively studied and have yielded excellent results. However, adapting these methods to real-life simulations remains a complex task. In this paper, we propose a method that offers flexibility and can efficiently handle real-life simulations. We suggest utilizing the physics-informed neural network (PINN) to solve the Navier-Stokes equation in a deformable domain, specifically addressing the simulation of blood flow in elastic vessels. Our approach models blood flow using an incompressible, viscous Navier-Stokes equation in an Arbitrary Lagrangian-Eulerian form. The mechanical model for the vessel wall structure is formulated by an equation of Newton's second law of momentum and linear elasticity to the force exerted by the fluid flow. Our method is a mesh-free approach that eliminates the need for discretization and meshing of the computational domain. This makes it highly efficient in solving simulations involving complex geometries. Additionally, with the availability of well-developed open-source machine learning framework packages and parallel modules, our method can easily be accelerated through GPU computing and parallel computing. To evaluate our approach, we conducted experiments on regular cylinder vessels as well as vessels with plaque on their walls. We compared our results to a solution calculated by Finite Element Methods using a dense grid and small time steps, which we considered as the ground truth solution. We report the relative error and the time consumed to solve the problem, highlighting the advantages of our method.

Key words. Fluid-structure interaction, Physics-informed neural network, blood flow simulation, Arbitrary Lagrangian-Eulerian, Computational Fluid Dynamics

1. Introduction.

The measurement of blood velocity and pressure is a fundamental aspect of medical practice, providing crucial insights into the dynamics of the cardiovascular system. Despite the widespread application of established techniques such as Fractional Flow Reserve (FFR) [24], it is important to recognize their limitations, particularly in specific clinical scenarios [32, 10]. For instance, the invasive operation of FFR assessment can increase patient risks, potentially leading to complications or exacerbating existing medical conditions. Inserting an intravascular wire into the vessel can disrupt the original flow dynamics by altering the spatial configuration of the inner lumen [36]. To address these challenges, research has been directed towards using computational approaches, which involves geometry reconstruction from medical images [23, 22] and numerically solving for a properly modeled Fluid-Structure Interaction (FSI) problem [21].

The FSI problem is modeled as a coupled problem of the blood flow and the vessel structures. Specifically, the fluid stress produces the deformation of structures and will reversely change the fluid dynamics. To describe and solve the blood flow simulation problem, the Arbitrary Lagrangian-Eulerian (ALE) formulation is a popular choice. Within this formulation, the displacement of the computational domain is introduced as a variable in the coupled system, ensuring the satisfaction of coupling conditions on the fluid-structure interface. While previous research has used linear elasticity to model the vessel wall [26], others have modeled it as poroelastic media by coupling

[*] Submitted to the editors DATE.

[†] Department of Mathematics, City University of Hong Kong and Hong Kong Centre for Cerebro-Cardiovascular Health EngineeringCity University of Hong Kong, Hong Kong, China. (hzhang863-c@my.cityu.edu.hk)

[‡] Department of Mathematics, City University of Hong Kong and Hong Kong Centre for Cerebro-Cardiovascular Health EngineeringCity University of Hong Kong, Hong Kong, China. (raymond.chan@cityu.edu.hk)

[§] Norwegian Research Centre (NORCE), Nygardsgaten 112, 5008 Bergen, Norway. (xtai@norceresearch.no, xuecheng-tai@gmail.com) the Navier-Stokes equation with the Biot system [3]. Quaini et al. [25] validated the simulated results, which use a semi-implicit, monolithic method, through a mock heart chamber using an elastic aperture.

A variety of methods have been developed to solve these models. Fernandez et al. [11] introduced a semi-implicit coupling method that implicitly couples the pressure stress to the structure while treating the nonlinearity due to convection and geometrical nonlinearities explicitly. To further reduce computational costs, Badia et al. [2] designed two methods based on the inexact factorization of the linearized fluid-structure system. Barker and Cai [4] developed a method that couples the fluid to the structure monolithically, making it robust to changes in physical parameters and large deformation of the fluid domain. A loosely coupled scheme that is unconditionally stable was proposed by Bukavc et al. [7], based on a newly modified Lie operator splitting that decouples the fluid and structure sub-problems. Other approaches include the space-time formulation [6, 31], the higher-order discontinuous Galerkin method [34], the immersed boundary method [33, 13], and the coupled momentum method [12, 37].

However, numerically solving such an FSI problem remains to be challenging. The mesh must be carefully discretized since the deformation of the mesh over time will change the mesh geometry, resulting in the failure of convergence during the solving process [35, 29]. Current autonomous software like Gmsh [14] can produce sub-optimal meshes that can cause methods like the Finite Element Method (FEM) to fail in complex FSI problems. To address the issues caused by large structural displacement, Basting et al. [5] proposed the Extended ALE Method, which relies on a variational mesh optimization technique where mesh alignment with the structure is achieved via a constraint. However, in many cases, manual corrections on the mesh are required to make a result. These corrections may be needed multiple times, with validation required after each correction. Such exhausting operations make solving problems on complex geometry hard and expensive. Besides, solutions by conventional methods are not continuously defined in the domain of our model, leading to further errors if an interpolation is needed to find a solution at a particular point that is not a node in the discretized grid. What's worse, to acquire an accurate solution, the conventional mesh-based methods require a high-resolution configuration of dense grids and small time steps. Such settings would dramatically increase the computation workload and memory consumption, slowing and preventing a fast and satisfying result acquisition.

To address these challenges, we propose a neural solver for the Navier-Stokes equation in deformable vessels using Physics-Informed Neural Networks (PINNs)[27], which has achieved some success on flow problems in non-moving domains [17, 8]. In our methods, we take some networks to produce displacement, pressure and velocity of our problem. To acquire those networks, we first convert the partial differential equations of the formulated physical model into residual forms and use them as loss functions. Next, the automatic differentiation is used to represent all the differential operators. Then, we train the networks to minimize those losses. With those losses getting approximately zero, the networks can be an approximation of the functions for displacement, pressure, and velocity that are used to solve the modeled problem.

This method is a mesh-free approach, eliminating the need for discretization and meshing for the computational domain. This feature makes it highly efficient and straightforward to add other objects into the geometry (e.g., blood plaque) for simulation. To evaluate the performance of our methods, we performed several experiments. The first experiment simulates the blood flow in an elastic cylinder-like vessel. For comparison, we use Finite Element Methods to solve the same problem using different grid resolutions and time steps. The results revealed that our approach can produce comparable results to FEM in high-resolution grids while consuming notably less time. Furthermore, we perform the blood flow simulation on a vessel with a plaque attained further to illustrate the robustness of our methods on complex geometry. In addition, we performed

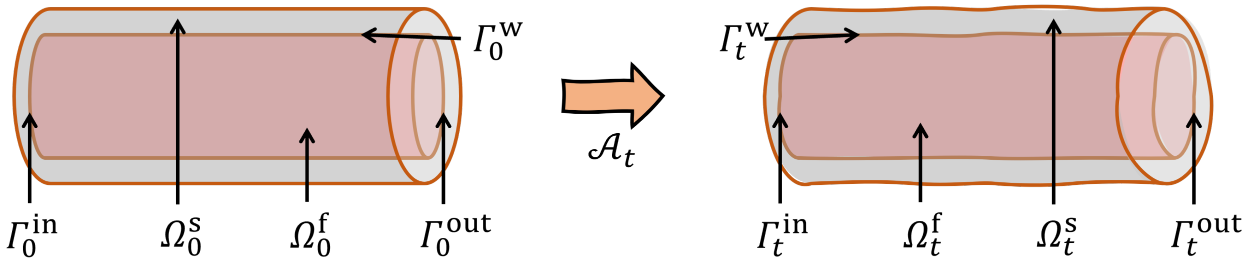

Figure 2.1: The illustration for the Arbitrary Lagrangian Euler form. Left: reference frame; Right: current frame.

self-ablation on networks with different depths and widths to show how we should choose the architecture for the neural networks for our task. Since our methods are easy to parallelize, an evaluation of how our method can be accelerated by using multiple GPUs is performed. Last but not least, the mechanical analysis of how different sizes of plaque might affect the blood flow dynamics is done by using our methods.

Our contributions are summarized as follows:

1. We developed a neural solver based on deep learning techniques for fluid-structure interaction problems, which simulate the blood flow in deformable vessels. To the best of our knowledge, we are the first to use PINN method to solve an FSI problem.
2. Our method is mesh-free, removing the need for careful discretization and meshing of the computation domain. This feature makes it easy and robust to solve the simulation problems in different geometries.
3. The continuous nature of our method ensures that the solutions can be computed for any point in the domain. This advances the conventional grid-based methods, which will introduce extra errors by interpolating for points that are not nodes in the grid.
4. Benefiting from the well-developed deep learning framework packages, our method is easy to deploy on GPU and simple to parallelize. This greatly saves the computation time needed for solving the simulation problems.

2. Problem Formulation.

In this section, we introduce the mathematical model for our problem, which is to simulate the flow of blood in a deformable vessel. In essence, the blood flow, assumed to be incompressible, is described by the Navier-Stokes equations in the pressure-velocity formulation. Since the computational domain is moving, we employ the Arbitrary Lagrangian Euler (ALE) form of the Navier-Stokes equations in this study. The vessel wall, assumed to be linear-elastic, is continuously moving due to the force exerted by the fluid. With the movement of the structures, the computational domain for the fluid is also moving and is computed by the harmonic extension from the displacement of the structure.

To accurately represent the deformation and simulate the flow, it is crucial to properly model the movement of the computational domain, as well as the fluid velocity and pressure. As illustrated in Figure 2.1, we start with the reference frame Ω_0, which serves as the parameterization for the entire domain required for our computations. Dividing this region into two parts, we have the fluid domain Ω_0^{f}, through which the flow occurs, and the structural domain Ω_0^{s}, representing the vessel wall. The interface between the fluid and structural domains is denoted as Γ_0^{w}. It's important to note that most of the interaction takes place at this interface. The inlet boundary is labeled as Γ_0^{in} and the outlet boundary is as Γ_0^{out}. Each point in the domain of the reference frame is denoted as the problem coordinates $\boldsymbol{x}_0 = (x_0, y_0, z_0)^\intercal$, where the bold symbols are used throughout this paper

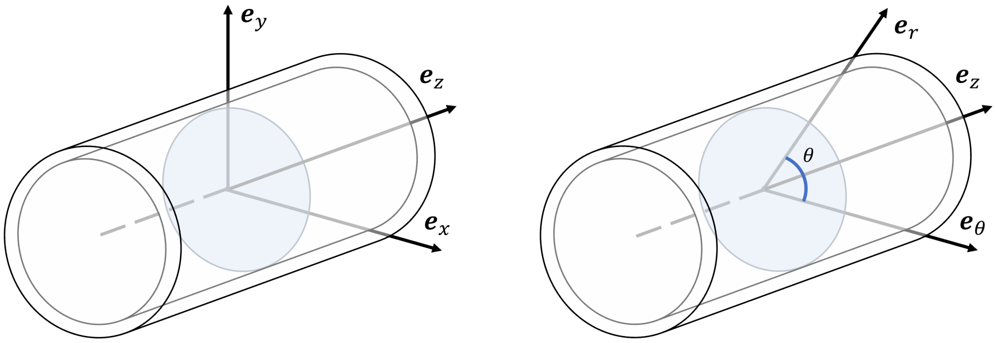

Figure 2.2: The coordinate system illustration. A 3D geometry can be represented using a Cartesian coordinate system as (x, y, z) (left). A cylindrical model of the vessel geometry may be described by using a curvilinear cylindrical coordinate system $r = R(\theta, z; t)$ with the corresponding base unit vectors $\mathbf{e}_r$, $\mathbf{e}_\theta$ and $\mathbf{e}_z$, where $\mathbf{e}_z$ is aligned with the axis of the artery (right).

to signify matrices or column vectors. As the domain continues to move, we use Ω_t to represent the current frame at time t. The symbols Ω_t^{f}, Ω_t^{s}, Γ_0^{w}, Γ_0^{in}, and Γ_0^{out} denote the fluid domain, the structural domain, the interface, the inlet, and the outlet, respectively. The coordinates of points in the current frame are referred to as the physical coordinates $(x_t, y_t, z_t)^\intercal$.

Following Quarteroni et al. [26], we introduce the ALE mapping defined as:

$$\mathcal{A}_t : \Omega_0 \to \Omega_t, \quad \boldsymbol{x}_0 \to \boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{d}(\boldsymbol{x}_0, t) = \mathcal{A}_t(\boldsymbol{x}_0) \quad \text{in } \Omega_t^{\mathrm{f}}, \tag{2.1}$$

where the displacement function $\boldsymbol{d}(\cdot)$ characterizes the domain movement within the reference frame, and $\tilde{\boldsymbol{w}} = \frac{\partial \boldsymbol{d}}{\partial t}$ denotes the domain velocity within Ω_0. In the current frame Ω_t, the domain velocity is defined as $\boldsymbol{w} = \tilde{\boldsymbol{w}} \circ \mathcal{A}_t^{-1}$.

Suppose no external force is applied to the flow, the incompressible unsteady Navier-Stokes equation which discribes the movement of a viscous fluid in a moving domain is:

$$\begin{aligned} \rho_{\mathrm{f}} \frac{D}{Dt}\boldsymbol{u} + \rho_{\mathrm{f}}[(\boldsymbol{u} - \boldsymbol{w}) \cdot \nabla]\boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma}_{\mathrm{f}} = \boldsymbol{0} \\ \nabla \cdot \boldsymbol{u} = 0 \end{aligned} \quad \text{in } \Omega_t^{\mathrm{f}},\ t \in [0, T], \tag{2.2}$$

where ρ_{f} is the fluid density, $\boldsymbol{u}$ is the fluid velocity, $\boldsymbol{\sigma}_{\mathrm{f}} = -p\boldsymbol{I} + 2\mu\frac{\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\intercal}{2}$ is the Cauchy stress tensor, p is fluid pressure, μ is the viscosity of the fluid and is a positive quantity. 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 $\nabla \boldsymbol{u}$.

In modeling the structure of the vessel, we make several simplifying assumptions: (1) The thickness of the vessel wall is small enough to permit a shell-type representation of the vessel geometry; (2) The reference vessel configuration is approximated as a circular cylindrical surface with straight axes, and the displacements occur primarily in the radial direction; (3) The mechanical properties of the vessel wall follow linear elasticity. While these assumptions simplify our problem, it's important to note that extending the model to more complex scenarios is a straightforward possibility.

Assuming the vessel has a cylinder-like geometry at time 0, as illustrated in Figure 2.2, the vessel wall can be rewritten as:

$$\Gamma_0^{\mathrm{w}} = \left\{(r, \theta, z) : r = R_0(\theta, z), \theta \in [0, 2\pi), z \in [0, L]\right\}, \tag{2.3}$$

where $R_0(\theta, z) = r_0$, representing the initial radius of the cylindrical vessel geometry, and L denotes the length of the arterial element under consideration. In our cylindrical coordinate system (r, θ, z), we assume that displacements are mainly in the radial direction and other components are negligible. Thus, we can simplify the displacement $\boldsymbol{d}$ to have only the radial component $\boldsymbol{\eta}$, such that:

$$\boldsymbol{d} = \eta \boldsymbol{e}_r, \tag{2.4}$$

Here, $\boldsymbol{e}_r$ is the unit vector along the r-axis.

Given these settings, we rewrite the current frame Γ_t^w as:

$$\Gamma_t^{\mathrm{w}} = \{(r, \theta, z) : r = R(\theta, z; t), \theta \in [0, 2\pi), z \in [0, L]\}, \tag{2.5}$$

where $R(\theta, z; t) = \eta(\theta, z; t)\boldsymbol{e}_r + R_0(\theta, z)$.

Based on linear elasticity and Newton's second law of momentum, we equate them and obtain the stress continuity equation:

$$\rho_w h_0 R_0 \frac{\partial^2 \eta}{\partial t^2} + \frac{E h_0}{1 - \xi^2} \frac{\eta}{R} = -(2\mu \boldsymbol{D}(\boldsymbol{u}) \cdot \boldsymbol{n}) \cdot \boldsymbol{e}_r g R_0 + pR \quad \text{on } \Gamma_0^{\mathrm{w}}, \tag{2.6}$$

where $g = \frac{R}{R_0}\sqrt{1 + \left(\frac{1}{R}\frac{\partial R}{\partial \theta}\right)^2 + \left(\frac{\partial R}{\partial z}\right)^2}$, ρ_{w} is the vessel wall density, h_0 is the vessel wall thickness, ξ is the Poisson ratio and E is the Young modulus. ??? Refer to a reference for the derivation of this equation. ??? In this equation, the righthand side is the force exerted by the fluid, while the lefthand side is the internal force of the vessel wall structure. The first term on the left is the acceleration times mass by Newton's second law of motion, and the second term is the linear elasticity.

Rewrite (2.6), we obtain the independent ring model for the stress continuity equation:

$$\frac{\partial^2 \eta}{\partial t^2} + b\eta = H \quad \text{in } \Gamma_0^{\mathrm{w}}, \tag{2.7}$$

where $b = \frac{E}{\rho_{\mathrm{s}}(1-\xi^2)R_0^2}$, and $H = \frac{1}{\rho_{\mathrm{s}} h_0}\left[\frac{R}{R_0}P - g\mu\left(\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathsf{T}}\right) \cdot \boldsymbol{n}\right) \cdot \boldsymbol{e}_r\right]$.

The structural domain in the reference frame is then denoted as:

$$\Omega_0^s = \{(r, \theta, z) : r \in [r_0, r_0 + h], \theta \in [0, 2\pi), z \in [0, L]\}\,. \tag{2.8}$$

We assume that the magnitude of the displacement remains constant along the radial direction. Then, the displacement $\boldsymbol{d}$ in the structure domain on the reference frame is expressed as

$$\boldsymbol{d}(r_{\mathrm{s}}, \theta, z, t) = \eta(\theta, z; t)\boldsymbol{e}_r + r_{\mathrm{s}}\boldsymbol{e}_r \quad \text{for } (r_{\mathrm{s}}, \theta, z) \in \Omega_0^s, t \in [0, T]. \tag{2.9}$$

In addition to modeling, we need to define the initial and boundary conditions. Given the variety of specific problem settings, we provide a general representation and define them specifically in the experimental section. For fluid problem, the boundary condition is given by:

$$F_{\mathrm{bdr}}^{\mathrm{f}}(\boldsymbol{u}, P) = 0 \quad \text{on } \Gamma_t, t \in [0, T], \tag{2.10}$$

where $\Gamma_t = \Gamma_t^{\mathrm{in}} \cup \Gamma_t^{\mathrm{out}} \cup \Gamma_t^{\mathrm{w}}$. The initial condition can be written as :

$$F_{\mathrm{init}}^{\mathrm{f}}(\boldsymbol{u}, P) = 0 \quad \text{in } \Omega_0^{\mathrm{f}}, t = 0. \tag{2.11}$$

Similarly, we write the boundary condition and initial condition for the stress continuity equation (2.7) as:

$$F_{\mathrm{bdr}}^{\mathrm{s}}(\boldsymbol{d}) = 0 \quad \text{on } \partial\Gamma_0^{\mathrm{w}}, t \in [0, T], \tag{2.12}$$

and,

$$F^{\mathrm{s}}_{\mathrm{init}}(\boldsymbol{d}) = 0 \quad \text{on } \Gamma^{\mathrm{w}}_0, t = 0. \tag{2.13}$$

With the deformation of the structure domain modeled, we model the deformation of the fluid domain as the harmonic extension from the deformation of the structure, which well-defined the computation domain:

$$\Delta \boldsymbol{d} = \boldsymbol{0} \quad \text{in } \Omega^{\mathrm{f}}_0, t \in [0, T]. \tag{2.14}$$

We may then recognize the coupling between the fluid and the structure models are interactive. The solution to the fluid problem provides stress to the structure, thus the value of H in (2.7). The movement of the vessel wall changes the geometry on which the fluid equations must be solved. In addition, the boundary conditions for the fluid velocity in correspondence to the vessel wall are no longer homogeneous Dirichlet conditions, but they impose equality between the fluid and the structure velocity. They express the fact that the fluid particle in correspondence with the vessel wall should move at the same velocity as the wall [26].

3. Our Proposed Numerical Methods. In this section, we will introduce the fundamental concepts of neural networks (NNs) and elucidate how NNs can be employed to approximate the solution of our proposed physical model.

3.1. Fully connected neural network. The most fundamental and widely utilized type of neural network in applied mathematics is the fully connected neural network (FNN). Let us denote the set of positive integers as $\mathbb{N}^+$. Mathematically, we define a simple nonlinear function, labeled as $h_\ell : \mathbb{R}^{M_{\ell-1}} \to \mathbb{R}^{M_\ell}$, where $M_\ell \in \mathbb{N}^+$ for $\ell = 0, 1, \dots, L$. The expression for h_ℓ can be represented as

$$h_\ell\left(\boldsymbol{x}_\ell\right) := \phi\left(\boldsymbol{W}_\ell \boldsymbol{x}_\ell + \boldsymbol{b}_\ell\right), \tag{3.1}$$

where $\boldsymbol{W}_\ell \in \mathbb{R}^{M_\ell \times M_{\ell-1}}$ is the weight parameter; $\boldsymbol{b}_\ell \in \mathbb{R}^{M_\ell}$ is the bias parameter; the function $\phi(\boldsymbol{y})$ represents a specified acitivation function applied element-wise to a vector $\boldsymbol{y}$ and output another vector of the same size. Commonly used activation functions include the rectified linear unit $\phi_{\mathrm{r}} = \max(0, y)$ and the sigmoidal function $\phi_{\mathrm{s}} = \left(1 + e^{-y}\right)^{-1}$.

Set the input dimension $M_0 = d_{\mathrm{in}}$, the output dimension $M_L = d_{\mathrm{out}}$. Then an FNN $N : \mathbb{R}^{d_{\mathrm{in}}} \to \mathbb{R}^{d_{\mathrm{out}}}$ is formulated as the composition of these L simple nonlinear functions. Then, for an input $\boldsymbol{x}_0 = (x_1, x_2, \cdots, x_{d_{\mathrm{in}}-1})^\intercal$ and temporal coordinate t into:

$$\hat{\boldsymbol{x}} = (x_1, x_2, \cdots, x_{d-1}, t)^\intercal, \tag{3.2}$$

we have

$$N(\boldsymbol{x}, t; \boldsymbol{\theta}) = h_L \circ h_{L-1} \circ h_{L-2} \circ \cdots \circ h_1(\hat{\boldsymbol{x}}) \quad \text{for } \hat{\boldsymbol{x}} \in \mathbb{R}^d, \tag{3.3}$$

where $\boldsymbol{\theta} := \{\boldsymbol{W}_\ell, \boldsymbol{b}_\ell : 1 \le \ell \le L\}$, denotes the set of all weighting parameters that defined the neural network. Here M_ℓ is called the width of the ℓ-th layer, and L is called the depth. These parameters collectively define the architecture of a fully connected neural network (FNN). By specifying the activation function ϕ, the depth L, and the sequence of widths $\{M_\ell\}_{\ell=1}^L$, the architecture of the FNN is determined. However, the parameters within $\boldsymbol{\theta}$ remain undetermined and need to be initialized and optimized. For simplicity, we will focus on an architecture where the width is consistent across all layers, specifically $M_\ell = M$ for ℓ ranging from 1 to $L-1$. We use $\mathcal{F}_{L,M,\sigma}$ to denote the set of all FNNs characterized by a depth of L, a uniform width of M, and an activation function σ [15]. In

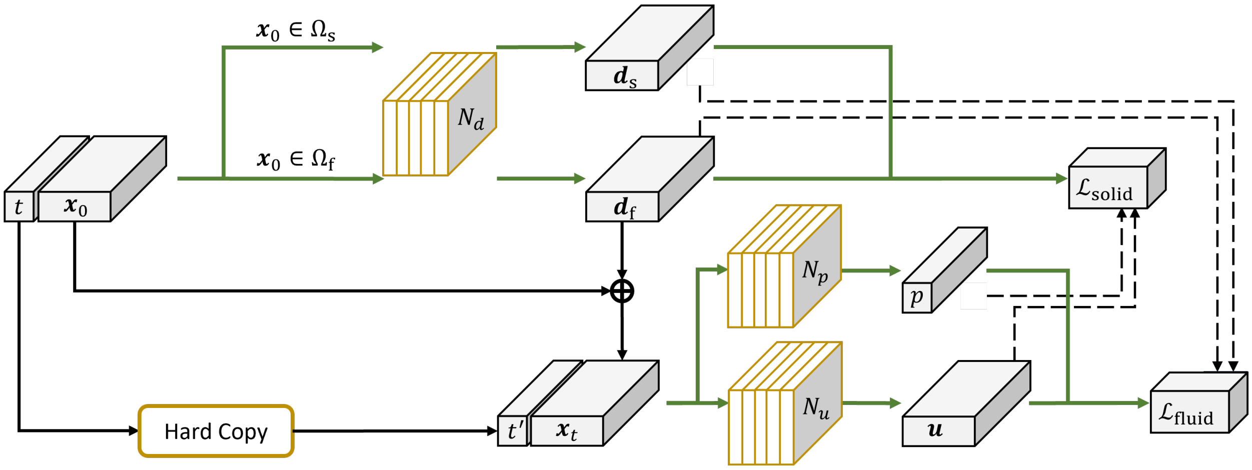

Figure 3.1: The illustration for the Network Architecture. Three networks are used. Displacement network N_d is for solving the displacement vector $\boldsymbol{d}$ for both the fluid and the solid domain. Velocity network N_u is to compute the fluid velocity $\boldsymbol{u}$ and pressure network N_p is for the fluid pressure p. Note that the green line indicates data flow where gradient calculation needs to be done, while the black line denotes that no gradient needs to be delivered. The dotted black line means that the computed results need to be given in the loss function calculation. But it will not enable a back-propogation which might change the network producing it. For example, when solving for the fluid problem in the alternative optimization, the loss function (3.10) matters with $\boldsymbol{d}$. However, the gradient will not be computed with respect to $\boldsymbol{d}$ and the parameters of N_d, which is explained in more detail in Section.3.4.

this sense, any two functions within $\mathcal{F}_{L,M,\sigma}$ only differ in the values of their respective weighting parameters $\boldsymbol{\theta}$.

To ensure the smoothness of the solutions approximated by neural networks, the inclusion of the *Sigmoid* activation function in some of our networks is essential. However, based on our experiments, we observed that utilizing only the *Sigmoid* activation function in the network can result in slow optimization and convergence. This issue may stem from the problem of gradient explosion associated with the *Sigmoid* [28, 16].

In response to this challenge, we have thoughtfully incorporated a combination of alternating activation functions into our network architecture. Specifically, after the first layer, which is activated by the *Sigmoid* function, in this work, we set the activation function for odd-numbered layers as *Sigmoid* while setting that in the even-numbered layer as *ReLU*. Besides, it's important to note that the layer before the final output layer does not incorporate any activation function so as to ensure the correct scaling. This deliberate choice of activation functions strikes a balance between ensuring sufficiently continuous derivatives, which are crucial for precise fluid dynamics modeling, and the fast convergence offered by the *Relu* activation.

3.2. Loss Function. In our study, we depart from conventional numerical approaches such as Finite Volume and Finite Element methods. Instead, we venture into the intriguing domain of neural networks (NNs) to tackle the complex issue of coupled fluid-structure interaction. At the heart of our methodology lies the utilization of deep neural networks for the approximation of solutions to the Navier-Stokes equations.

Our process initiates with the sampling of points in the reference frame. For each of these points, denoted as $\boldsymbol{x}_0$ at a specific time t, we calculate the corresponding solutions for velocity, pressure, and displacement. To achieve this, we introduce a network N_d that takes spatial and temporal coordinates as inputs and outputs the displacement on the corresponding point and time, which is expressed as:

$$\eta^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) = N_d(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) \quad \text{for } \boldsymbol{x}_0 \in \Omega_t; t \in [0, T]. \tag{3.4}$$

and by (2.4)

$$\boldsymbol{d}^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) = \eta^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d)\boldsymbol{e}_r \quad \text{for } \boldsymbol{x}_0 \in \Omega_t; t \in [0, T]. \tag{3.5}$$

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

$$\boldsymbol{x}_t^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) = \boldsymbol{x}_0 + \boldsymbol{d}^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d) \quad \text{for } \boldsymbol{x}_0 \in \Omega_t; t \in [0, T]. \tag{3.6}$$

With the acquirement of the physical coordinates, we can get sets of points in the current frame Ω_t. When the problem coordinate of $\boldsymbol{x}_t^*$ is within the fluid domain, we can compute the fluid velocity and pressure on it. To achieve this, 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:

$$\begin{matrix} \boldsymbol{u}^*(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_u, \boldsymbol{\theta}_d) = N_u(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_u) = N_u(\boldsymbol{x}_0 + \boldsymbol{d}^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d), t'; \boldsymbol{\theta}_u) \\ P^*(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_p, \boldsymbol{\theta}_d) = N_p(\boldsymbol{x}_t^*, t'; \boldsymbol{\theta}_p) = N_p(\boldsymbol{x}_0 + \boldsymbol{d}^*(\boldsymbol{x}_0, t; \boldsymbol{\theta}_d), t'; \boldsymbol{\theta}_p) \end{matrix} \quad \text{for } \boldsymbol{x}_0 \in \Omega_t^{\mathrm{f}}; t \in [0, T]. \tag{3.7}$$

where for convenience of discussion, we denote $\boldsymbol{\theta}_f = \{\boldsymbol{\theta}_u, \boldsymbol{\theta}_p\}$.

Please note that in this context, we are using the symbol t', which is distinct from the t, to represent a separate hard copy of the value of t. Although they have the same value, they are considered different objects in the programming. This distinction is necessary for our implementation using *PyTorch*, as we need to distinguish the differentiation with respect to time in the reference configuration Ω_0 and the current configuration Ω_t.

To clarify, if we were to directly apply automatic differentiation to $\boldsymbol{u}^*(\boldsymbol{x}_t^*; t)$ with respect to t, it would yield a result similar to the material derivative as $\frac{\partial \boldsymbol{u}^*}{\partial t} + \frac{\partial \boldsymbol{d}^*}{\partial t} \cdot \nabla \boldsymbol{u}^*$. However, by using the separate variable t', which is a distinct copy of t, the automatic differentiation to $\boldsymbol{u}^*(\boldsymbol{x}_t^*; t')$ with respect to t' is $\frac{\partial \boldsymbol{u}^*}{\partial t'}$.

To enable the training of our neural network, we reformulate these partial differential equation (PDE) systems discussed in Section.2 into residual forms. In the case of the fluid problem, we rewrite the Navier-Stokes equation and its associated initial/boundary conditions into residual form as:

$$\begin{aligned} \mathcal{L}_{\mathrm{ns}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) &= ||\frac{\partial}{\partial t'}\boldsymbol{u}^* + (\boldsymbol{u}^* \cdot \nabla)\boldsymbol{u}^* + \nabla p^* - 2\nabla \cdot (\mu \boldsymbol{D}(\boldsymbol{u}^*))||^2_{L^2(\Omega_t^{\mathrm{f}};[0,T])}, \\ &+ ||\nabla \cdot \boldsymbol{u}^*||^2_{L^2(\Omega_t^{\mathrm{f}};[0,T])} \\ \mathcal{L}^{\mathrm{f}}_{\mathrm{bdr}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) &= ||F^{\mathrm{f}}_{\mathrm{bdr}}(\boldsymbol{u}^*, P^*)||^2_{L^2(\Gamma_t;[0,T])}, \\ \mathcal{L}^{\mathrm{f}}_{\mathrm{init}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) &= ||F^{\mathrm{f}}_{\mathrm{init}}(\boldsymbol{u}^*, P^*)||^2_{L^2(\Omega_0^{\mathrm{f}};t=0)}. \end{aligned} \tag{3.8}$$

Note that the L^2 norm is employed here and will be redefined by a discrete approximation in the next subsection.

Regarding the solid problem, we need to rewrite the stress continuity equation (2.7), which is the source of the deformation. The initial and boundary conditions associated are also necessary.

The displacement that extends from the displacement on the interface for the fluid domain is also converted for the calculation of the current domain. Those residual forms are:

$$\begin{aligned}
\mathcal{L}_{\text{sc}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) &= ||\frac{\partial^2 \eta^*}{\partial t^2} + b\eta^* - H||^2_{L^2(\Gamma_0^{\text{w}};[0,T])}, \\
\mathcal{L}^{\text{s}}_{\text{bdr}}(\boldsymbol{\theta}_d) &= ||F^{\text{s}}_{\text{bdr}}(\boldsymbol{d}^*)||^2_{L^2(\partial\Gamma_0^{\text{w}};[0,T])}, \\
\mathcal{L}^{\text{s}}_{\text{init}}(\boldsymbol{\theta}_d) &= ||F^{\text{s}}_{\text{init}}(\boldsymbol{d}^*)||^2_{L^2(\Gamma_0^{\text{w}},t=0)}, \\
\mathcal{L}_{\text{he}}(\boldsymbol{\theta}_d) &= ||\Delta \boldsymbol{d}^*||^2_{L^2(\Omega_0^{\text{f}};[0,T])};
\end{aligned} \tag{3.9}$$

where $\eta^*(\boldsymbol{x}_0)$ is the projection of $\boldsymbol{d}^*(\boldsymbol{x}_0)$ onto $\boldsymbol{e}_r$ for $\boldsymbol{x}_0 \in \Gamma_0^{\text{w}}$ as discussed in Sec.2.

Combining all the losses for the flow problem and deformation problem, we get

$$\mathcal{L}_{\text{fluid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) = \alpha_{\text{ns}}\mathcal{L}_{\text{ns}} + \alpha^{\text{f}}_{\text{bdr}}\mathcal{L}^{\text{f}}_{\text{bdr}} + \alpha^{\text{f}}_{\text{init}}\mathcal{L}^{\text{f}}_{\text{init}}, \tag{3.10}$$

$$\mathcal{L}_{\text{solid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d) = \alpha_{\text{sc}}\mathcal{L}_{\text{sc}} + \alpha_{\text{he}}\mathcal{L}_{\text{he}} + \alpha^{\text{s}}_{\text{bdr}}\mathcal{L}^{\text{s}}_{\text{bdr}} + \alpha^{\text{s}}_{\text{init}}\mathcal{L}^{\text{s}}_{\text{init}}. \tag{3.11}$$

where α_{ns}, $\alpha^{\text{f}}_{\text{bdr}}$ and $\alpha^{\text{f}}_{\text{init}}$, are corresponding weight parameters for the Navier-Stokes equations, boundary conditions, and initial conditions. In the context of the deformation problem, we utilize α_{sc}, α_{he}, $\alpha^{\text{s}}_{\text{bdr}}$, and $\alpha^{\text{s}}_{\text{init}}$ to assign weights to the loss components associated with the stress continuity, harmonic extension, initial condition, and boundary conditions.

Then, if we can minimize the loss $\mathcal{L}_{\text{fluid}}$ to be approximately zero, the physical systems described in Eq.(2.2) with boundary condition Eq.(2.10) and initial condition Eq.(2.11) are approximately satisfied. The solutions provided by networks N_u and N_p have successfully solved the model. Similarly, for $\mathcal{L}_{\text{solid}}$, when it converges sufficiently close to zero, the stress continuity equation in Eq.(2.7), along with the appropriate initial and boundary conditions (Eq.(2.13) and Eq.(2.12)), as well as the harmonic extension into the fluid domain, are effectively solved. From our numerical experiments, we found that it is efficient to use the following mutli-objective minmization approach to minimize the two loss functions:

$$\min_{\boldsymbol{\theta}_d, \boldsymbol{\theta}_f} \mathcal{L}_{\text{solid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d), \tag{3.12}$$

and

$$\min_{\boldsymbol{\theta}_f, \boldsymbol{\theta}_d} \mathcal{L}_{\text{fluid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d). \tag{3.13}$$

Our method to solve the above FSI problem is to solve the fluid problem and the solid problem iteratively. In particular, the solution of the fluid problem is obtained by imposing the displacement obtained from solving the solid problem at the interface as boundary conditions. The result for the solid problem is solved by imposing H in the stress continuity (2.7) on the interface. Correspondingly, in our method, which needs to minimize the two losses, we take an alternative optimization strategy to decouple the FSI problem into two sub-problems. The optimization scheme will be given more specifically in 3.4.

3.3. Mini-batch gradient descent algorithm. Our FSI problem contains the fluid problem (3.10) and the solid problem (3.11). Each problem contains multiple losses defined as the norm in individual domains. To obtain those norms, we follow the principle of *PINN* to sample a sufficient number of points in the domain and treat them as an approximate representation of it. For example, we sample a set of M points in domain Ω_0^{f} to obtain $\mathcal{S}(\Omega_0^{\text{f}})$, which is an approximate representation of it. We additionally sample M temporal points $\mathcal{T}([0,T])$ in $[0,T]$ as an approximate representation of this time interval. Then, we can have

$$\mathcal{S}(\Omega_0^{\mathrm{f}}; t=0) = \{(\boldsymbol{x}_0^{(i)}, 0) \quad | \quad \text{where } \boldsymbol{x}_0^{(i)} \in \mathcal{S}(\Omega_0^{\mathrm{f}})\} \tag{3.14}$$

which can be an approximate representation of $(\Omega_0^{\mathrm{f}}, t=0)$, and

$$\begin{aligned}\mathcal{S}(\Omega_t^{\mathrm{f}}; [0,T]) = \{(\boldsymbol{x}_{t^{(i)}}^{(i)}, t^{(i)}) \quad | \quad & \boldsymbol{x}_{t^{(i)}}^{(i)} := \boldsymbol{x}_0^{(i)} + \boldsymbol{d}^*(\boldsymbol{x}_0^{(i)}, t^{(i)}; \boldsymbol{\theta}_d) \\ & \text{where } \boldsymbol{x}_0^{(i)} \in \mathcal{S}(\Omega_0^{\mathrm{f}}) \text{ and } t^{(i)} \in \mathcal{T}([0,T])\}\end{aligned} \tag{3.15}$$

which can be an approximate representation of $(\Omega_t^{\mathrm{f}}, [0,T])$.

Naturally, we define the norm with $\mathcal{S}(\Omega_t^{\mathrm{f}}; [0,T])$ as

$$||f||^2_{\mathcal{S}(\Omega_t^{\mathrm{f}};[0,T])} := \frac{1}{M} \sum_{(\boldsymbol{x},t)\in\mathcal{S}(\Omega_t^{\mathrm{f}};[0,T])} |f(\boldsymbol{x},t)|^2 \tag{3.16}$$

Similarly, we sample N points for Γ_0^{w} and K points for $\Gamma_0^{\mathrm{in}} \cup \Gamma_0^{\mathrm{out}}$. Then $\mathcal{S}(\Gamma_t; [0,T])$, $\mathcal{S}(\Gamma_0^{\mathrm{w}}; [0,T])$, $\mathcal{S}(\Omega_0^{\mathrm{f}}; [0,T])$, $\mathcal{S}(\partial\Gamma_0^{\mathrm{w}}; [0,T])$ can be obtained in the same way as (3.15), while $\mathcal{S}(\partial\Gamma_0^{\mathrm{w}}; t=0)$ can refer to (3.14). Then, we can compute for the losses in (3.8) and (3.9) by redefining those norms as

$$||\cdot||^2_{L^2(\cdot;\cdot)} := ||\cdot||^2_{\mathcal{S}(\cdot;\cdot)} \tag{3.17}$$

To decrease the value of the losses, gradient descent methods are applied to update $\boldsymbol{\theta}_f$ in and $\boldsymbol{\theta}_f$ by

$$\boldsymbol{\theta}_f \leftarrow \boldsymbol{\theta}_f - \tau_f \nabla_{\boldsymbol{\theta}_f} \mathcal{L}_{\mathrm{fluid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d), \tag{3.18}$$

$$\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d - \tau_d \nabla_{\boldsymbol{\theta}_d} \mathcal{L}_{\mathrm{solid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d), \tag{3.19}$$

where $\nabla_{\boldsymbol{\theta}_f} \mathcal{L}_{\mathrm{fluid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d)$ is the gradient of $\mathcal{L}_{\mathrm{fluid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d)$ with respect to $\boldsymbol{\theta}_f$, and $\nabla_{\boldsymbol{\theta}_d} \mathcal{L}_{\mathrm{solid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d)$ is the gradient of $\mathcal{L}_{\mathrm{solid}}(\boldsymbol{\theta}_f, \boldsymbol{\theta}_d)$ with respect to $\boldsymbol{\theta}_d$. Here, τ_f, τ_d are adaptive step lengths calculated by some methods [1, 18]. In this paper, since we utilize *Adam* optimizer in the implementation, the step lengths are all calculated using the method in [18].

3.4. Training Scheme. The fluid-structure interaction problem is particularly challenging due to the coupling of multiple physical phenomena. Adding to this complexity, our approach employs three independent networks: N_u, N_p, and N_d. As a result, devising an effective training strategy for these networks is crucial. Based on the consideration to solve the fluid problem and solid problem alternatively, we present our training scheme in Algorithm 3.1.

Before stepping into the training scheme, we define a training unit for the fluid problem to be a collection of m training epochs to make our discussion convenient as given in Algorithm 3.2. Within each unit for the fluid problem, using the loss function (3.10), N_u is trained for the first $m_1 = 80\% \times m$ epochs, and N_p is trained for the remaining $m_2 = 20\% \times m$ epochs. The training unit for the solid problem is defined as a group of m training epochs for N_p using (3.11).

To initialize, we perform zero-initialization on the parameters for the output layer of N_d so that its output is all zeros, aligning with the boundary and initial conditions for solving displacements in solid problems for most cases. Then, with $\alpha_{\mathrm{ns}} = 0$, we optimize N_u and N_p for a maximum of m_{fluid} units, which can be stopped when it converges before that. The unit-by-unit training results in an alternative training for N_u and N_p until the optimization converges (the improvement on loss is less than a control parameter ϵ). This sequence of operations ensures that the networks fulfill the initial and boundary conditions.

After the initialization phase, we further train the networks N_u and N_p to solve the Navier-Stokes equations in the domain of the reference frame. During this stage, we initially set $\alpha_{\text{ns}} = 10^{-7}$. Then, we update it, according to Section.4.1.2, by a factor of 10 when the loss function $\mathcal{L}_{\text{fluid}}$ converges or the training units have run for m_{fluid} units. The update is performed for E times for one simulation problem.

The final stage of training involves coupling the fluid and solid systems. In this phase, we alternately solve the fluid and solid problems. Specifically, we run m_{solid} training units to solve the solid problem for the displacement computation, followed by a maximum of m_{fluid} training units for the fluid problem. Either training can be stopped in advance in one alternation if it converges. We alternate between these two problems until they converge or run for a maximum of F alternations.

Algorithm 3.1 Fluid-Structure Interaction Training Scheme

```
Initialize $N_d$ with *zero_initialization*;
Set $\alpha_{\text{ns}} \leftarrow 0$;
Optimize the fluid problem for $m_{\text{fluid}}$ training units at most;
Take $\alpha_{\text{ns}} \leftarrow 10^{-8}$;
**for** i=1:5 **do**
    $\alpha_{\text{ns}} \leftarrow 10 \times \alpha_{\text{ns}}$;
    Optimize the fluid problem for $m_{\text{fluid}}$ training units at most;
**end for**
**for** i=1:F **and** $\mathcal{L}_{\text{fluid}}$ doesn't converge **do**
    Optimize the solid problem for $m_{\text{solid}}$ training units at most;
    Optimize the fluid problem for $m_{\text{fluid}}$ training units at most;
**end for**
```

Algorithm 3.2 Training Scheme for Fluid Problem

```
$k = 1$;
**while** $|L^{(k)} - L^{(k-1)}| < \epsilon$ or $k > m_{\text{fluid}}$ **do**
    Train $N_u$ for $m_1$ epochs, and then $N_p$ for $m_2$ epochs using $\mathcal{L}_{\text{fluid}}$;
    $L^{(k)} \leftarrow$ the minimal value for $\mathcal{L}_{\text{fluid}}$ within this training unit;
    $k = k + 1$;
**end while**
```

3.5. Parallel Training. Neural network training can be parallelized in various ways to improve efficiency and reduce training time. We focus on data parallelism, a popular technique for parallelizing the training of neural networks, especially in distributed computing environments.

Taking the solid problem as an example, the network model N_a is first replicated on P processing units (CPU or GPU), where each replica is denoted as $N_a^{(i)}$ with $i = 1, \ldots, P$ indexing the processing unit for P units. Each replica processes a different batch of data independently. Specifically, the sampled point $\mathcal{N}$ is divided into P groups $\mathcal{N}^{(i)}$, where $i = 1, \ldots, P$. The forward pass, involving the computation of predictions, is performed independently on each replica as $N_a^{(i)}(\mathcal{N}^{(i)})$. The loss is computed for each batch as $\mathcal{L}_{\text{solid}}(\boldsymbol{\theta}_d{}^{(i)};\mathcal{N}^{(i)})$, where $\boldsymbol{\theta}_d{}^{(i)}$ is the weight parameters for replica $N_a^{(i)}$. Backpropagation is performed independently on each model replica to calculate the gradients of the loss with respect to the model parameters to acquire the corresponding gradients $\boldsymbol{\theta}_d{}^{(i)}$. Gradients computed on each replica are then averaged and used to update the shared model parameters:

$$\nabla_{\boldsymbol{\theta}_d}\mathcal{L}_{\text{solid}}(\boldsymbol{\theta}_{\text{d}};\mathcal{N}) = \frac{1}{P}\sum_{i=1}^{P}\nabla_{\boldsymbol{\theta}_{\text{d}}^{(i)}}\mathcal{L}_{\text{solid}}(\boldsymbol{\theta}_{\text{d}}^{(i)};\mathcal{N}^{(i)}) \tag{3.20}$$

4. Experiment.

This section embarks on a comprehensive evaluation of the efficacy of our proposed methodology through a series of experiments. The primary aim is to demonstrate the robustness and reliability of our approach in a variety of settings.

Firstly, we undertake a comprehensive assessment of the simulation quality for blood flow within a segment of a cylinder-like vessel. To do a quantitative evaluation, we refer to the calculations by Finite Element Method on a highly dense grid with a small time step as the ground truth. Then, we compare our methods with Finite Element Methods on different grid resolutions and time step sizes according to the relative error. The second facet of our experiments focuses on a self-ablation test to determine the optimal network architecture for fluid problems. We employ networks with varying depths and widths to discern the architecture that best suits our specific setting by simulating the same scenario as the previous blood flow in cylinder vessel simulation. Evaluation of the time for using different numbers of GPUs for parallel training is taken. Also, we expand our evaluation on vessel segments afflicted with a plaque for the third set of experiments. Comparison with Finite Element Method calculations are also performed to provide a holistic view. In the final experiment, we evaluate different sizes of plaque to see how it will influence the stress from the fluid as well as how the blood flow after the stenosis will be affected.

4.1. Implementation Details.

4.1.1. Environment and Softwares. Our methods are implemented across all experiments using the *Python* programming language coupled with the *Pytorch* framework. The Finite Element Method is implemented using the open-source package *FEniCS*. It's noteworthy that the computational meshes are generated using *Gmsh*; however, certain meshes, generated automatically, necessitate manual modifications to make them computable with *FEniCS*. A Linux server equipped with four Tesla A600 Graphics Processing Units is employed to execute these experiments.

4.1.2. Hyperparameters. The most important consideration for the weights on the loss functions is ensuring the satisfaction of the boundary and initial conditions. To achieve this, we first train the network to minimize the residuals with $\alpha_{\text{ns}} = 0$, $\alpha^{\text{f}}_{\text{bdr}} = 1$, and $\alpha^{\text{f}}_{\text{init}} = 10^{-1}$ for the fluid problem at the beginning. This setting is motivated by the treatment of initial and boundary conditions in conventional finite element methods, where they are enforced strictly. When the network N_u and N_p are sufficiently trained, we set $\alpha_{\text{ns}} = 10^{-7}$ and update it by $\alpha_{\text{ns}} = 10 \times \alpha_{\text{ns}}$ until it converges for the current α_{ns}. This dynamic update helps the neural solver to sufficiently satisfy the initial and boundary conditions before optimizing the network to produce accurate solutions for velocity and pressure. Regarding the solid problem, based on empirical experiments, we set $\alpha_{\text{sc}} = 1$, $\alpha_{\text{he}} = 10$, $\alpha^{\text{s}}_{\text{bdr}} = 10^{-2}$, and $\alpha^{\text{s}}_{\text{init}} = 10^{-2}$.

We set $M = N = 1000$ points with temporal and spatial coordinates sampled on each domain and boundary, as discussed in Section 3.3. Adam optimizer with a learning rate of 1×10^{-3} is employed to initialize and optimize the learnable variables of neural network. Regarding the network architecture, unless specified otherwise, the network depth is set to 12 for N_u, N_p, and N_d. The width is set to 20 for N_u and N_d, and 10 for N_p.

For parameters in Algorithm 3.1, a training unit for the fluid problem contains $m = 100$ training epochs. Within each unit, N_u is trained for the first $m_1 = 80\% \times m = 80$ epochs, while N_p is trained for the remaining $m_2 = 20\% \times m = 20$ epochs. The training unit for the solid problem also contains $m = 100$ training epochs.

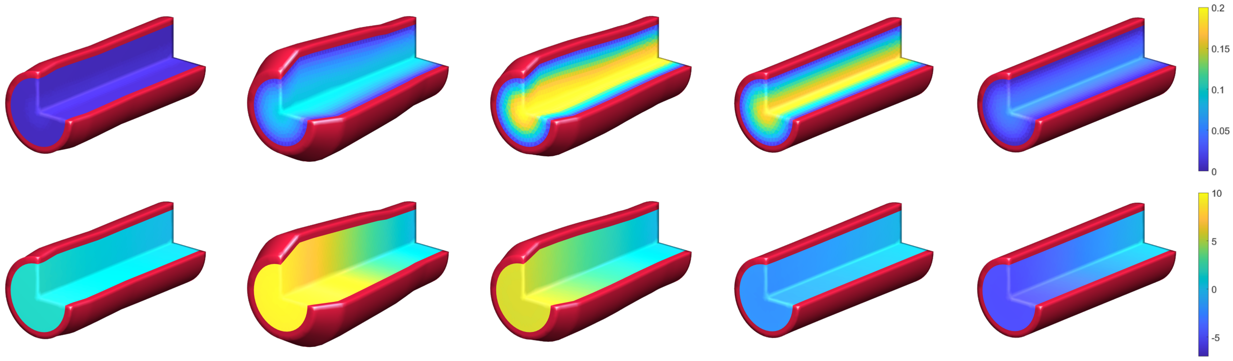

Figure 4.1: The illustration for the velocity magnitude (up row) and pressure (bottom row) variation at time points 0.01, 0.21, 0.41, 0.61, and 0.81. The red outside indicates the vessel wall. The color inside the lumen indicates the quantity magnitude. For a more convenient observation, the displacement vectors are enlarged by a factor of 10.

For parameters in Algorithm 3.1, when $\alpha_{\text{ns}} = 0$, N_u and N_p are optimized for a maximum of $m_{\text{fluid}} = 20$ units which can be stopped when it converges before that. The convergence parameter is $\epsilon = 10^{-2}$. After the initialization stage, we set $\alpha_{\text{ns}} = 10^{-7}$, which is then multiplied by 10 for $E = 5$ times. In the phase of iteratively solving the two sub-problems, we run $m_{\text{solid}} = 5$ training units to solve the solid problem, followed by a maximum of $m_{\text{fluid}} = 20$ training units for the fluid problem. There are a maximum of $F = 6$ alternations for solving the FSI problem.

4.1.3. Error Metric. To evaluate the accuracy and speed of our methods, the relative error and computation time are reported in the sections below. The relative error for velocity magnitude is defined as:

$$E(\boldsymbol{u}^*, \boldsymbol{u}^{\mathrm{GT}}) = t_l \sum_{\boldsymbol{t}\in\mathcal{T}} \frac{\sum_{\mathcal{P}\in\mathcal{F}} A(\mathcal{P}) \frac{\sum_{\boldsymbol{p}\in\mathcal{P}} |\boldsymbol{u}^*(\boldsymbol{p})-\boldsymbol{u}^{\mathrm{GT}}(\boldsymbol{p})|^2}{|\mathcal{P}|}}{\sum_{\mathcal{P}\in\mathcal{F}} A(\mathcal{P}) \sum_{\boldsymbol{p}\in\mathcal{P}} \frac{|\boldsymbol{u}^{\mathrm{GT}}(\boldsymbol{p})|^2}{|\mathcal{P}|}}, \tag{4.1}$$

where $\boldsymbol{u}^*$ represents the solution obtained using our method, while $\boldsymbol{u}^{\mathrm{GT}}$ denotes the ground truth solution, which we utilize the solution obtained using FEM on a high-resolution grid and small step size. The collection of sampled times is denoted by $\mathcal{T}$ with the time step length t_l. The set $\mathcal{F}$ represents the collection of elements, while $\mathcal{P}$ denotes one element containing $|\mathcal{P}|$ nodes, and each node is written as $\boldsymbol{p}$. Here $A(\mathcal{P})$ calculates the volume of the element $\mathcal{P}$. For the evaluation of the relative error in pressure, a similar definition is employed.

4.2. Blood Vessel with Elastic Wall. In this section, we delve into the details of our simulation process, focusing on the incompressible viscous flow within a linearly elastic vessel. Let us begin by outlining the material parameters and geometry for this study.

The fluid density, denoted by ρ, is set to 1.025 g/cm^3, and the viscosity, denoted as μ, is 0.035 poise. The vessel has a cylinder-like shape, with a length l of 2 cm and a radius r_0 of 0.25 cm. The vessel wall thickness is $h = 0.05$ cm. The entire simulation runs for a duration T of 2 seconds.

Because of the specific geometry, we can simplify the problem by transforming it from a 3-D coordinate system (x, y, z) to a 2.5-D cylindrical coordinate system (r, θ, z). This simplification allows us to fix the longitudinal section while varying the radial angle θ, effectively reducing the problem to a 2-D setting. This transformation streamlines the computation and conserves computational resources. It also needs to be mentioned again that the longitudinal direction is designated as the z-axis, which is the same as the original coordinate system illustrated in 2.2.

To elucidate the initial and boundary conditions, let us begin with the initial state of the fluid. At the starting point in time $t = 0$, the fluid is at rest, and its velocity $\boldsymbol{u}$ is zero within the fluid domain Ω_0^{f}, as Equation (4.2).

$$\boldsymbol{u} = 0 \quad \text{in } \Omega_0^{\mathrm{f}}, t = 0. \tag{4.2}$$

Fluid continuously enters the vessel through the inlet, following a parabolic velocity profile $g(\boldsymbol{x}) = 1 - \frac{r^2}{r_0^2}$. Equation (4.3) provides the mathematical representation for this inlet velocity profile. The maximum value of velocity oscillates sinusoidally, achieving a maximum of 20 cm/s and a minimum of 0 cm/s, fitting the nature of the cardiac cycle, approximately 60 times per minute.

$$\boldsymbol{u}(\boldsymbol{x}, t) = g(\boldsymbol{x})(0.1 - 0.1\cos(2\pi t)) \quad \text{on } \Gamma_t^{\mathrm{in}}, t \in [0, T]. \tag{4.3}$$

On the outlet of the vessel segment, no stress is imposed. This condition is represented as a neutral boundary condition, written as

$$[\mu(\nabla^{\intercal}\mathbf{u} + \nabla\mathbf{u}) - p\mathbf{I}] \cdot \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_t^{\mathrm{out}}, t \in [0, T]. \tag{4.4}$$

For the interface between the fluid domain and the solid domain, the velocity of the fluid is matched with the speed of the structural movement. Thus, the boundary condition here is as:

$$\mathbf{u} = \frac{\partial \eta}{\partial t}\mathbf{e}_r \quad \text{on } \Gamma_t^{\mathrm{w}}, t \in [0, T]. \tag{4.5}$$

Thus, in the fluid problem, the functional $F_{\mathrm{init}}^{\mathrm{f}}$ that indicates the initial condition and $F_{\mathrm{bdr}}^{\mathrm{f}}$ that indicates the boundary condition can be expressed as

$$F_{\mathrm{init}}^{\mathrm{f}} = \boldsymbol{u} \qquad \text{in } \Omega_0^{\mathrm{f}}, t = 0. \tag{4.6}$$

$$F_{\mathrm{bdr}}^{\mathrm{f}} = \begin{cases} \boldsymbol{u} - g(0.1 - 0.1\cos(2\pi t)) & \text{in } \Gamma_t^{\mathrm{in}}, t \in [0, T], \\ [\mu(\nabla^{\intercal}\mathbf{u} + \nabla\mathbf{u}) - p\mathbf{I}] \cdot \mathbf{n} & \text{on } \Gamma_t^{\mathrm{out}}, t \in [0, T], \\ \mathbf{u} - \frac{\partial \eta}{\partial t}\mathbf{e}_r & \text{on } \Gamma_t^{\mathrm{w}},\ t \in [0, T]. \end{cases} \tag{4.7}$$

Turning to the parameters for the solid problem, the vessel structure density ρ_{s} is set at 1.2 g/cm^3, and its mechanical properties include a Young modulus of $E_{\mathrm{s}} = 0.5 \times 10^6$ dynes/cm^2 and a Poisson's ratio of $\xi_{\mathrm{s}} = 0.5$ [26, 30]. Given these mechanical parameters and the governing equation for deformation (Equation (2.7)), we need suitable initial and boundary conditions to complete the system.

As we discussed in the previous section, the vessel initially maintains a strictly cylindrical shape, ensuring no deformation occurs. This state can be precisely expressed as:

$$\mathbf{d} = \mathbf{0} \quad \text{in } \Omega_0, t = 0. \tag{4.8}$$

The boundary conditions are established by fixing the starting and ending portions of the vessel segment in their respective positions results in the boundary condition:

$$\mathbf{d} = \mathbf{0} \quad \text{on } \partial\Gamma_0^{\mathrm{w}}, t = 0. \tag{4.9}$$

Therefore, the functionals $F_{\mathrm{init}}^{\mathrm{s}}$ and $F_{\mathrm{bdr}}^{\mathrm{s}}$, indicating the initial and boundary conditions in the solid problem, can be expressed as:

$$F_{\mathrm{init}}^{\mathrm{s}} = \boldsymbol{d} \quad \text{in } \Omega_0, t = 0, \tag{4.10}$$

$$F_{\mathrm{bdr}}^{\mathrm{s}} = \mathbf{d} \quad \text{on } \partial\Gamma_0^{\mathrm{w}}, t = 0. \tag{4.11}$$

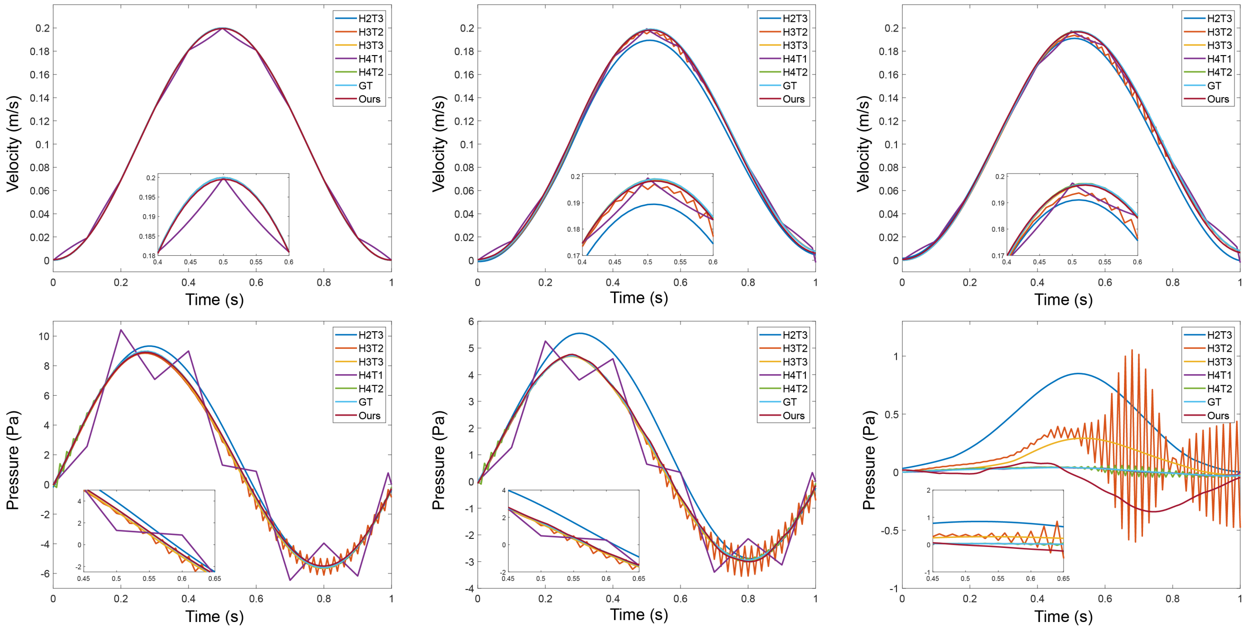

Figure 4.2: The temporal profiles of velocity magnitude and pressure at different points. From left to right, the upper three plots illustrate velocity magnitude on the centerline's inlet, middle, and outlet. The lower three represent pressure magnitude for the same corresponding points. The notations H1, H2, H3, and H4 in the legend denote the grids whose minimum length h are set to be $\frac{1}{2} \times 10^{-3}$, $\frac{1}{4} \times 10^{-3}$, $\frac{1}{8} \times 10^{-3}$, and $\frac{1}{16} \times 10^{-3}$, respectively. T1, T2, and T3 denote time step lengths 1×10^{-1}, 1×10^{-2}, and 1×10^{-3}.

With these critical conditions and material properties defined, we are now fully equipped to proceed with solving the simulation problem for blood flow in elastic vessels, as outlined in Section 3. The results of our computations are presented in Table 4.1 where the relative errors for velocity magnitude and pressure are reported. In this comparison, we utilize the solution computed by the Finite Element Method (FEM) on a grid with a minimum length of $\frac{1}{16} \times 10^{-3}$ meters and a time step of 10^{-3} seconds as the ground truth. In the table, h^{-1} signifies the inverse of the minimum length h, which is used to discretize the geometry. The results for simulation using FEM also relates to time step size, which is set to be three different lengths in our experiments. Notably, our method is mesh-free, so we don't need to configure for time step and discretization length. Instead, we test with a different number of sample points, denoted as N, M, and K as discussed in Section 3.

The results indicate that our method performs impressively with N, M, and K set to 1000, achieving high accuracy. Remarkably, the accuracy achieved with these settings is comparable to traditional grid-based methods with a minimum length of 1×10^{-4} meters and a time step of 10^{-2} seconds. However, our approach significantly reduces the computational time.

Figure 4.1 visually displays the variation in velocity magnitude and pressure at specific time points (0.01, 0.21, 0.41, 0.61, and 0.81). Notably, velocity magnitude is more pronounced near the vessel's centerline, and pressure is generally higher near the inlet, particularly during the systole phase of a cardiac cycle. In contrast, during the diastole phase of a cardiac cycle, the pressure in the front part appears to be negative and lower than at the outlet.

These observations are further supported by the plots in Figure 4.2, which depict the temporal profiles of velocity magnitude and pressure at different points. The upper three plots illustrate

Relative Error on Velocity Magnitude						
h^{-1}	2×10^3	4×10^3	8×10^3	16×10^3	Ours	Points
Time Step 10^{-1}	0.2342	0.1089	0.0798	0.0413	0.0363	10^2
Time Step 10^{-2}	0.1142	0.0567	0.0112	0.0008	0.0128	10^3
Time Step 10^{-3}	0.0730	0.0385	0.0003	0.0000	0.0098	10^4
Relative Error on Pressure Magnitude						
h^{-1}	2×10^3	4×10^3	8×10^3	16×10^3	Ours	Points
Time Step 10^{-1}	1546.5336	4.9901	0.0933	0.1275	0.0415	10^2
Time Step 10^{-2}	17.1869	1.4504	0.0212	0.0017	0.0235	10^3
Time Step 10^{-3}	1.9128	0.0897	0.0083	0.0000	0.0187	10^4
Computation Time (seconds)						
h^{-1}	2×10^3	4×10^3	8×10^3	16×10^3	Ours	Points
Time Step 10^{-1}	18	51	103	877	732	10^2
Time Step 10^{-2}	87	214	1053	9155	863	10^3
Time Step 10^{-3}	283	825	7168	72539	2417	10^4

Table 4.1: The comparison between our methods and FEM. The relative errors for velocity magnitude and pressure are reported. The FEM result using a high-resolution grid with a small time step is taken as the ground truth. In the table, h^{-1} signifies the inverse of the minimum length h, used to discretize the geometry. Since our method is mesh-free, we test with different numbers of sample points in the column of the results by our method, which is of different time step lengths for FEM.

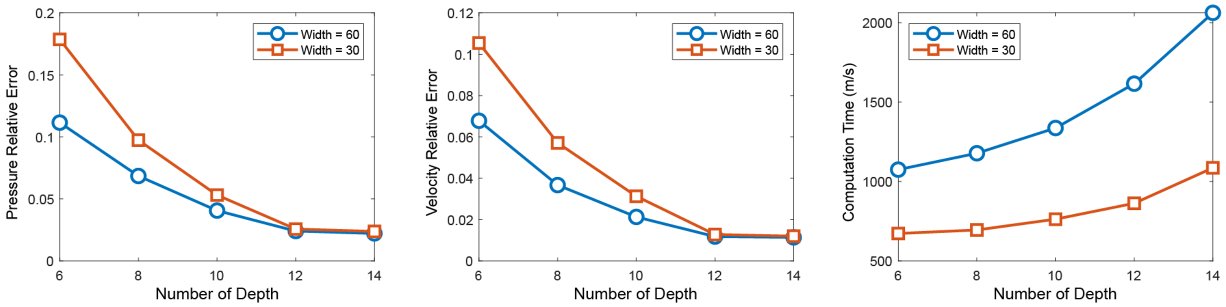

Figure 4.3: The plots of the solution quality and computational time for networks with different depths and widths. From left to right are pressure relative error, velocity magnitude relative error, and time consumed for reaching a convergence.

velocity magnitude, while the lower three represent pressure. To understand the plots, from left to right, they correspond to the points on the centerline's inlet, middle, and outlet. The symbols H1, H2, H3, and H4 in the legend denote the grids whose minimum length h are set to be $\frac{1}{2} \times 10^{-3}$, $\frac{1}{4} \times 10^{-3}$, $\frac{1}{8} \times 10^{-3}$, and $\frac{1}{16} \times 10^{-3}$, respectively. Similarly, the time step lengths are denoted as T1, T2, and T3. For instance. As an example, the H4T3 corresponds to a grid with a minimum length of $\frac{1}{16} \times 10^{-3}$ meters and a time step of 1×10^{-3} seconds.

Upon closer inspection, we have noticed that the time steps of 1×10^{-1} and 1×10^{-2} can introduce oscillations that lead to more significant errors over time while our approach consistently produces a smooth solution, which is more accurate and usable in practice.

	6×60	8×60	10×60	12×60	14×60
No. of Parameters	8583	12703	16823	20943	25063
Velocity Relative Error	0.0678	0.0367	0.0213	0.0118	0.0114
Pressure Relative Error	0.1115	0.0685	0.0406	0.0241	0.0222
Computational Time	1075	1178	1337	1616	2062
	6×30	8×30	10×30	12×30	14×30
No. of Parameters	2293	3353	4413	5473	6533
Velocity Relative Error	0.1054	0.0571	0.0313	0.0128	0.0120
Pressure Relative Error	0.1786	0.0974	0.0531	0.0235	0.0238
Computational Time	673	695	763	863	1086

Table 4.2: The velocity magnitude relative error, pressure relative error, and computational time for our methods using different depths and widths.

4.3. Self Ablation. Next, we delve into the influence of different neural network architectures on the model performance. In this section, we maintain the same geometric setup, initial conditions, boundary conditions, and other parameters outlined in the previous simulation problem. The primary variation lies in the neural network architecture, specifically in its depth and the total width for N_u and N_p. In this study, we employ two different width settings. The first setting allocates a width of 20 to N_u and 10 to N_p, resulting in a total width of 30. For the second setting, with a total width of 60, we maintain the same proportion, assigning 40 to N_u and the remaining 20 to N_p. All tests are conducted under identical optimization settings as detailed in Section 4.1.

The data presented in Table 4.2 illustrates that deeper and wider neural networks deliver more precise solutions for velocity and pressure. However, it is worth highlighting that these architectures also impose greater computational consumptions, primarily due to the increased number of trainable parameters. As the networks become deeper and wider, the incremental performance improvement becomes less pronounced, demonstrating a pattern of convergence. Figure 4.3 visually encapsulates these findings. Notably, when the network width is set to be fixed (either 60 or 30), the errors exhibit a discernible convergence pattern. Expanding the number of layers beyond this point does not substantially enhance the relative errors for velocity magnitude and pressure. However, it does significantly extend the duration of the network optimization process.

Considering the trade-off between simulation quality and computational efficiency, we identify the architecture with a depth of 12 and a total width of 30 are the optimal configuration. This architecture is selected for use in the problem of simulating blood flow in a regular cylinder-like vessel segment, as well as the simulation of a vessel with hard plaque within the lumen.

Another important feature of the proposed methods is its parallelizability. Benefiting from the well-developed *PyTorch* package, using multiple GPUs to optimize neural networks in a short time is straightforward and simple. Given in Table.4.3, we reported the time needed when using different numbers of GPUs for training a network with a depth of 12 and width of 30.

No. of GPUs	1	2	3	4
Time (s)	863	571	481	440

Table 4.3: The computational time for our methods using parallel training on different number of GPUs

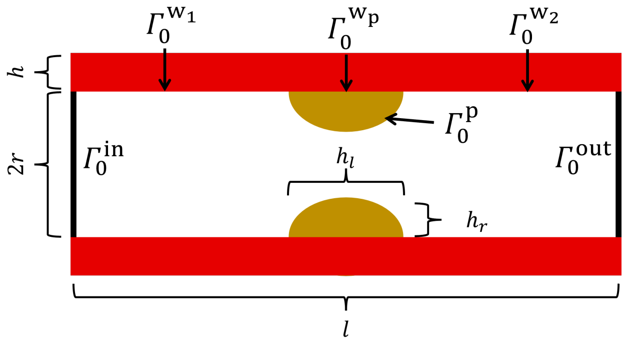

Figure 4.4: The illustration for the configuration of the cylindrical vessel containing plaque in longitudinal cross-section.

4.4. Blood Vessel with Plaque. Here, we are simulating the flow of incompressible viscous fluid within a linearly elastic vessel containing a plaque. The parameters and settings closely resemble those detailed in the first problem discussed in Section 4.2. The parameters of the fluid remain the same, with a density of $\rho_{\mathrm{f}} = 1.025$ g/cm^3 and a viscosity μ of 0.035 poise.

The vessel retains its cylinder-like shape, as illustrated in Figure 4.4, with a length of $l = 2$ cm, a radius of $r_0 = 0.25$ cm, and a vessel wall thickness of $h = 0.05$ cm. In the view of the longitudinal section, the plaque inside the vessel takes the form of a half ellipse centered at the middle point of the vessel wall, as illustrated in Figure 4.4. It has a long radius in the longitudinal direction ($h_l = 0.15$ cm) and a short radius in the radial direction ($h_r = 0.1$ cm). Similar to the previous problem, we simplify this model to a 2-D setting according to its radial symmetry configuration.

The initial conditions for the fluid remain the same as in the first problem. The fluid starts at rest at the beginning, as described in Equation (4.2). The functional $F^{\mathrm{f}}_{\mathrm{init}}$ from Equation (4.7) remains applicable in this context. The same inlet and outlet boundary conditions are applied as outlined in Equations (4.3) and (4.4). However, the interface is changed into $\Gamma^{\mathrm{w}_1} \cup \Gamma^{\mathrm{w}_2} \cup \Gamma^{\mathrm{p}}$. The continuity of the velocity between the fluid and solid is maintained at the boundaries $\Gamma_t^{\mathrm{w}_1} \cup \Gamma_t^{\mathrm{w}_2} \cup \Gamma_t^{\mathrm{p}}$ by setting the fluid velocity equal to the structure movement speed:

$$\mathbf{u} = \frac{\partial \eta}{\partial t}\mathbf{e}_r \quad \text{on } \Gamma_t^{\mathrm{w}_1} \cup \Gamma_t^{\mathrm{w}_2} \cup \Gamma_t^{\mathrm{p}}, t \in [0, T]. \tag{4.12}$$

The equation for the functional $F^{\mathrm{f}}_{\mathrm{bdr}}$, indicating the boundary condition in the fluid problem, is described as follows:

$$F^{\mathrm{f}}_{\mathrm{bdr}} = \begin{cases} \boldsymbol{u} - g(0.1 - 0.1\cos(2\pi t)) & \text{in } \Gamma_t^{\mathrm{in}}, t \in [0, T], \\ [\mu(\nabla^\intercal \mathbf{u} + \nabla \mathbf{u}) - p\mathbf{I}] \cdot \mathbf{n} & \text{on } \Gamma_t^{\mathrm{out}}, t \in [0, T], \\ \mathbf{u} - \frac{\partial \eta}{\partial t}\mathbf{e}_r & \text{on } \Gamma_t^{\mathrm{w}_1} \cup \Gamma_t^{\mathrm{w}_2} \cup \Gamma_t^{\mathrm{p}}, t \in [0, T]. \end{cases} \tag{4.13}$$

The key difference in this simulation lies in the geometry and mechanical properties introduced by the plaque. Here, we assume the plaque is a hard plaque characterized by the presence of calcium deposits within it[9]. This type of plaque is rigid and relatively inflexible compared to its softer counterpart. Unlike soft plaques, which can rupture suddenly, hard plaques tend to narrow the arteries, reducing blood flow to the heart muscle and may lead to conditions like coronary artery disease or atherosclerosis.

We retain the parameters for the vessel wall from the previous problem. The vessel structure density is set as $\rho_{\mathrm{s}} = 1.2$ g/cm^3, with Young modulus as $E_{\mathrm{s}} = 0.5 \times 10^6$ dynes/cm^2, and a Poisson's

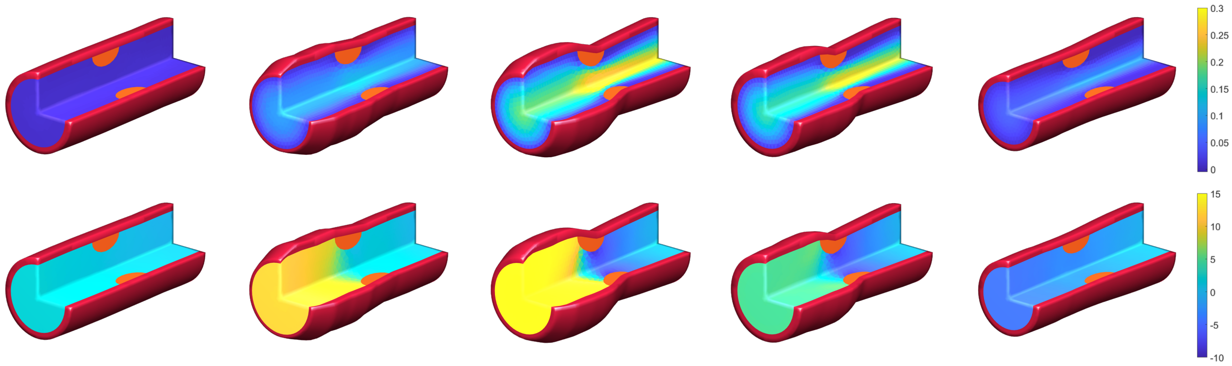

Figure 4.5: The illustration for the velocity magnitude (up row) and pressure (bottom row) variation at time points 0.01, 0.21, 0.41, 0.61, and 0.81. The red outside indicates the vessel wall. The color inside the lumen indicates the quantity magnitude. For a more convenient observation, the displacement vectors are enlarged by a factor of 10.

ratio of $\xi_\mathrm{s} = 0.5$. For the plaque, we set its properties as $\rho_\mathrm{p} = 1.1$ g/cm^3 for density, $E_\mathrm{p} = 1 \times 10^6$ dynes/cm^2 for Young modulus, and a Poisson's ratio of $\xi_\mathrm{p} = 0.5$.

The stress continuity equation (2.7) is modified to account for these different properties:

$$\begin{aligned} &\frac{\partial^2 \eta}{\partial t^2} + b_1 \eta = H_1 \qquad \text{on } \Gamma_0^{\mathrm{w}_1} \cup \Gamma_0^{\mathrm{w}_2}, \\ &\frac{\partial^2 \eta}{\partial t^2} + b_2 \eta = H_2 \qquad \text{on } \Gamma_0^{\mathrm{p}}. \end{aligned} \tag{4.14}$$

where

$$\begin{aligned} b_1 &= \frac{E_1}{\rho_{\mathrm{s}_1}(1-\xi^2)R_0^2}, \quad H = \frac{1}{\rho_{\mathrm{s}_1} h_0}\left[\frac{R}{R_0}P - g\mu\left((\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\intercal)\cdot \boldsymbol{n}\right)\cdot \boldsymbol{e}_r\right], \\ b_2 &= \frac{E_2}{\rho_{\mathrm{s}_2}(1-\xi^2)R_0^2}, \quad H = \frac{1}{\rho_{\mathrm{s}_1} h_0}\left[\frac{R}{R_0}P - g\mu\left((\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\intercal)\cdot \boldsymbol{n}\right)\cdot \boldsymbol{e}_r\right]. \end{aligned} \tag{4.15}$$

Please note that R_0 in (2.7) has been replaced by R_p, a real-value function represented as:

$$R_p(\theta, z) = \begin{cases} R_0 & \text{on } \Gamma_0^{\mathrm{w}_1} \cup \Gamma_0^{\mathrm{w}_2}, \\ R_0 - \sqrt{h_r^2 - \frac{h_r^2}{h_l^2}(z - z_p)^2} & \text{on } \Gamma_0^{\mathrm{p}}. \end{cases} \tag{4.16}$$

For the initial conditions, the fluid remains at rest initially, consistent with (4.8) in the first problem. Consequently, the functional $F_{\text{init}}^{\mathrm{s}}$ in (4.10) is directly applicable in this context. Regarding boundary conditions, as the start and end points of the interface boundary $\Gamma_0^{\mathrm{w}_1} \cup \Gamma_0^{\mathrm{p}} \cup \Gamma_0^{\mathrm{w}_2}$ remain the same as those for Γ_0^{w}, their descriptions do not differ from what is outlined in (4.9). Thus, we can directly apply the functional (4.11).

We follow the same approach as in the first problem, using a solution obtained from a highly dense grid and a sufficiently small time step as our ground truth. Table 4.4 presents the relative errors for velocity magnitude and pressure. As a mesh-free method, we provide results corresponding to different numbers of sample points instead of different time steps and discretization length parameters.

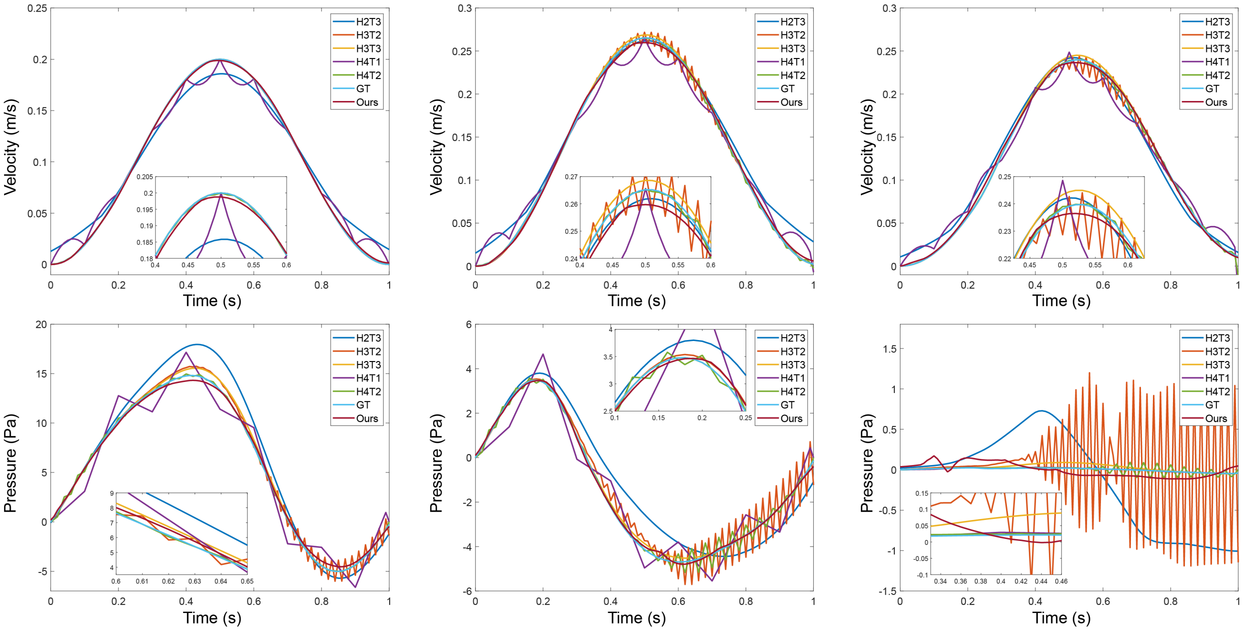

Figure 4.6: The temporal profiles of the velocity magnitude and pressure at different points. From left to right, the upper three plots illustrate the velocity magnitude on the centerline's inlet, middle, and outlet. The lower three represent the pressure magnitude for the same corresponding points. The symbols H1, H2, H3, and H4 in the legend denote the grids whose minimum length h are set to be $\frac{1}{2} \times 10^{-3}$, $\frac{1}{4} \times 10^{-3}$, $\frac{1}{8} \times 10^{-3}$, and $\frac{1}{16} \times 10^{-3}$, respectively. Here T1, T2, and T3 denote time step lengths 1×10^{-1}, 1×10^{-2}, and 1×10^{-3}.

The table shows that our method can deliver accurate results, particularly when using N, M, and K values greater than 1000. The accuracy of our method surpasses that achieved by using a grid with a minimum length of $\frac{1}{8} \times 10^{-3}$ m and a time step of 1×10^{-2}. This demonstrates the capability of our method for simulating blood flow in complex geometries.

Following the evaluation in the cylinder-like vessel simulation, we give in Figure 4.5 the evolution of velocity magnitude and pressure at time points: 0.01, 0.21, 0.41, 0.61, and 0.81. The observation reveals a significant increase in velocity magnitude after the stenosis caused by the plaque. This increase results in greater momentum concentration towards the center right between the plaque. This can also be revealed by the plots in Figure.4.6, which depict the time histories of the velocity and pressure at the points on the centerline's inlet, middle, and outlet, respectively.

4.5. Analysis on Plaque with Different Sizes. Using computational fluid dynamics (CFD) for hemodynamic analysis is an intriguing and highly relevant topic in cardiovascular medicine[19]. In this section, we apply our proposed method to investigate the impact of varying plaque sizes. Specifically, we examine four different levels of plaque severity. The configuration for the *No Plaque* case remains the same as the initial problem, featuring a regular, cylinder-like vessel. The other scenarios all involve plaques of different sizes but share similarities with the description provided in Section 4.4. To elaborate further, when we focus on the longitudinal cross-section at $\theta = \hat{\theta}$ (as illustrated in Figure 4.4), we set the plaques within the vessel as a half-ellipse centered at the midpoint of the vessel wall. These plaques share a common major radius $h_l = 0.15$ cm in the longitudinal direction and possess varying minor radii h_r in the radial direction, specifically 0.05 cm for *Mild*, 0.1 cm for *Moderate*, and 0.15 cm for *Severe*.

Relative Error on Velocity Magnitude						
h^{-1}	2×10^3	4×10^3	8×10^3	16×10^3	Ours	Points
Time Step 10^{-1}	2.2252	1.7873	0.2809	0.0515	0.0153	10^1
Time Step 10^{-2}	0.2104	0.0776	0.0136	0.0035	0.0094	10^2
Time Step 10^{-3}	0.0885	0.0390	0.0023	0	0.0092	10^3
Relative Error on Pressure Magnitude						
h^{-1}	2×10^3	4×10^3	8×10^3	16×10^3	Ours	Points
Time Step 10^{-1}	68.7149	41.6116	0.9274	0.1580	0.0378	10^1
Time Step 10^{-2}	22.8401	7.3519	0.0772	0.0153	0.0188	10^2
Time Step 10^{-3}	0.8589	0.2443	0.0073	0	0.0173	10^3

Table 4.4: The comparison between our methods and FEM. The relative errors for velocity magnitude and pressure are reported. The FEM result using a high-resolution grid with a small time step is taken as the ground truth. In the table, h^{-1} signifies the inverse of the minimum length h, used to discretize the geometry. Since our method is mesh-free, we test with different numbers of sample points in the column of the results by our method, which is of different time step lengths for FEM.

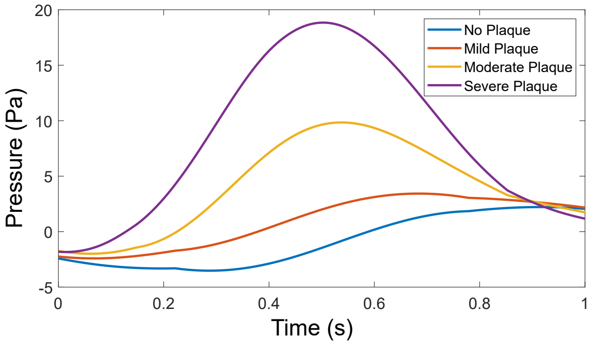

Figure 4.7: The temporal profiles of the stress on the peak point of the plaque (the endpoint of the minor axis of the ellipse) for different plaque sizes.

Firstly, we investigate the effects of force on the vessel for different plaque sizes. In our problem setup, the force on the plaque sourced from the stress exerted from the fluid flow, which is the norm of the traction vector as:

$$F = ||\boldsymbol{\sigma}_{\mathrm{f}} \cdot \boldsymbol{n}||_2 \tag{4.17}$$

where $\boldsymbol{\sigma}_{\mathrm{f}}$ is the Cauchy stress tensor as in (2.6), $\boldsymbol{n}$ is the normal vector to the interface between the plaque and fluid. In Figure 4.7, we present a plot illustrating the force applied by the fluid flow to the highest point of the elliptical plaque. The plot reveals that a healthy vessel without any plaque experiences the least stress in the corresponding region. In contrast, the largest plaque, which significantly alters the vessel geometry, encounters the highest stress from the blood flow. As the minor radius h_r of the elliptical plaque increases linearly, the maximum stress within one cardiac cycle rises exponentially. This demonstrates that an increase in the amount of deposited materials within the plaque leads to a much more rapid and significant escalation in the level of risk.

Next, we explore the impact of plaque on blood flow, specifically on how different plaque sizes affect the flow rate. As shown in Figure 4.8, a healthy vessel without plaque allows ample blood flow

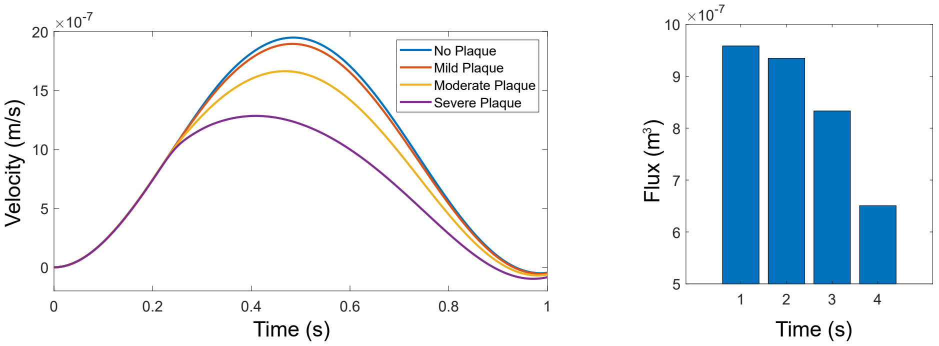

Figure 4.8: Left: the temporal profiles of the total flux on the outlet section of the vessel for different plaque sizes; Right: total flux within one cardiac cycle measured on the outlet section of the vessel for different plaque sizes.

to pass through the outlet boundary of the vessel segment. However, vessels with plaques impede the flow, with larger plaques causing more pronounced restrictions. Similarly, the increase in the short radius of the elliptical plaque results in a notably more dramatic reduction in blood flow.

5. Conclusion. In this paper, we proposed a mesh-free approach to simulate the blood flow in a deformable vessel. The blood flow is formulated as the Navier-Stokes equation associated with proper initial and boundary conditions. The mechanical model for the vessel wall structure is modeled by equating Newton's second law of motion and linear elasticity to the force exerted by the fluid flow, which we referred to as the stress continuity equation (2.6). Our method utilizes the physics-informed neural network (PINN) to solve the Navier-Stokes equation coupled with stress continuity on the interface of vessel structure and fluid flow. As a mesh-free approach, our method does not necessitate discretization and meshing of the computational domain and has been proven highly efficient in solving simulations for complex geometries. With the well-developed neural network packages and parallel modules, our method can be easily accelerated through GPU computing and parallel computing. Furthermore, our model can easily be adapted to arteries in different geometries without requiring additional discretization. Therefore, our method is significantly more efficient and user-friendly than employing the conventional Finite Element Method for clinical purposes. To evaluate our methods, we performed experiments on vessels with and without plaque inside. The results indicate the advantages of our methods. Self-ablation is also performed to determine the optimal setting for network architecture. The time saved by parallel training is evaluated. Additionally, we analyze the influence of different plaque sizes using our method. In future work, we plan to use PINNs to inversely design[20] an optimal shape of stents in patient-specific arteries to help with clinical treatment for patients who encounter arterial stenosis.

REFERENCES

[1] S.-i. Amari, *Backpropagation and stochastic gradient descent method*, Neurocomputing, 5 (1993), pp. 185–196.

[2] S. Badia, A. Quaini, and A. Quarteroni, *Splitting methods based on algebraic factorization for fluid-structure interaction*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1778–1805.

[3] S. Badia, A. Quaini, and A. Quarteroni, *Coupling biot and navier–stokes equations for modelling fluid–poroelastic media interaction*, Journal of Computational Physics, 228 (2009), pp. 7986–8014.

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

[5] S. Basting, A. Quaini, S. Čanić, and R. Glowinski, *Extended ale method for fluid–structure interaction*

problems with large structural displacements, Journal of Computational Physics, 331 (2017), pp. 312–336.

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

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

[8] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, *Physics-informed neural networks (pinns) for fluid mechanics: A review*, Acta Mechanica Sinica, 37 (2021), pp. 1727–1738.

[9] M. H. Criqui, J. O. Denenberg, J. H. Ix, R. L. McClelland, C. L. Wassel, D. E. Rifkin, J. J. Carr, M. J. Budoff, and M. A. Allison, *Calcium density of coronary artery plaque and risk of incident cardiovascular events*, Jama, 311 (2014), pp. 271–278.

[10] P. B. Dattilo, A. Prasad, E. Honeycutt, T. Y. Wang, and J. C. Messenger, *Contemporary patterns of fractional flow reserve and intravascular ultrasound use among patients undergoing percutaneous coronary intervention in the united states: insights from the national cardiovascular data registry*, Journal of the American College of Cardiology, 60 (2012), pp. 2337–2339.

[11] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont, *A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid*, International Journal for Numerical Methods in Engineering, 69 (2007), pp. 794–821.

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

[13] L. Ge and 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 (2007), pp. 1782–1809.

[14] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities*, International journal for numerical methods in engineering, 79 (2009), pp. 1309–1331.

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

[16] B. Hanin, *Which neural net architectures give rise to exploding and vanishing gradients?*, Proc. of Int. Conf. on Neural Information Processing Systems, 31 (2018).

[17] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, *Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations*, Journal of Computational Physics, 426 (2021), p. 109951.

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

[19] S. Li, C. Chin, V. Thondapu, E. K. Poon, J. P. Monty, Y. Li, A. S. Ooi, S. Tu, and P. Barlis, *Numerical and experimental investigations of the flow–pressure relation in multiple sequential stenoses coronary artery*, The International Journal of Cardiovascular Imaging, 33 (2017), pp. 1083–1088.

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

[21] J. Moore, D. Steinman, D. Holdsworth, and C. Ethier, *Accuracy of computational hemodynamics in complex arterial geometries reconstructed from magnetic resonance imaging*, Annals of biomedical engineering, 27 (1999), pp. 32–41.

[22] B. L. Nørgaard, J. Leipsic, S. Gaur, S. Seneviratne, B. S. Ko, H. Ito, J. M. Jensen, L. Mauri, B. De Bruyne, H. Bezerra, et al., *Diagnostic performance of noninvasive fractional flow reserve derived from coronary computed tomography angiography in suspected coronary artery disease: the nxt trial (analysis of coronary blood flow using ct angiography: Next steps)*, Journal of the American College of Cardiology, 63 (2014), pp. 1145–1155.

[23] M. I. Papafaklis, T. Muramatsu, Y. Ishibashi, L. S. Lakkas, S. Nakatani, C. V. Bourantas, J. Ligthart, Y. Onuma, M. Echavarria-Pinto, G. Tsirka, et al., *Fast virtual functional assessment of intermediate coronary lesions using routine angiographic data and blood flow simulation in humans: comparison with pressure wire-fractional flow reserve.*, EuroIntervention: journal of EuroPCR in collaboration with the Working Group on Interventional Cardiology of the European Society of Cardiology, 10 (2014), pp. 574–583.

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

[25] A. Quaini, S. Canic, R. Glowinski, S. Igo, C. Hartley, W. Zoghbi, and S. Little, *Validation of a 3d computational fluid–structure interaction model simulating flow through an elastic aperture*, Journal of biomechanics, 45 (2012), pp. 310–318.

[26] A. Quarteroni and L. Formaggia, *Mathematical modelling and numerical simulation of the cardiovascular system*, Handbook of numerical analysis, 12 (2004), pp. 3–127.

[27] M. Raissi, P. Perdikaris, and 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), pp. 686–707.

[28] M. Roodschild, J. Gotay Sardiñas, and A. Will, *A new approach for the vanishing gradient problem on sigmoid activation*, Progress in Artificial Intelligence, 9 (2020), pp. 351–360.

[29] W.-K. Sun, L.-W. Zhang, and 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), p. 115573.

[30] K. Takashima, T. Kitou, K. Mori, and K. Ikeuchi, *Simulation and experimental observation of contact conditions between stents and artery models*, Medical engineering & physics, 29 (2007), pp. 326–335.

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

[32] G. G. Toth, B. Toth, N. P. Johnson, F. De Vroey, L. Di Serafino, S. Pyxaras, D. Rusinaru, G. Di Gioia, M. Pellicano, E. Barbato, et al., *Revascularization decisions in patients with stable angina and intermediate lesions: results of the international survey on interventional strategy*, Circulation: Cardiovascular Interventions, 7 (2014), pp. 751–759.

[33] H. Wang, J. Chessa, W. K. Liu, and 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 (2008), pp. 32–55.

[34] Y. Wang, A. Quaini, and 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), pp. 481–520.

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

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

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