

Physical-Informed Neural Networks (PINNs) for Solving Shape Optimization Problems

Huanyu Li¹, Xiaoyan Li², Fangying Song^{1*}

¹School of Mathematics and Statistics, Fuzhou University, Fuzhou, China

²College of Computer and Data Science, Fuzhou University, Fuzhou, China

Email: *fangying_song@fzu.edu.cn

How to cite this paper: Li, H.Y., Li, X.Y. and Song, F.Y. (2024) Physical-Informed Neural Networks (PINNs) for Solving Shape Optimization Problems. *Journal of Applied Mathematics and Physics*, **12**, 3626-3637. <https://doi.org/10.4236/jamp.2024.1210217>

Received: October 12, 2024

Accepted: October 28, 2024

Published: October 31, 2024

Abstract

In this paper, we use Physics-Informed Neural Networks (PINNs) to solve shape optimization problems. These problems are based on incompressible Navier-Stokes equations and phase-field equations. The phase-field function is used to describe the state of the fluids, and the optimal shape optimization is obtained by using the shape sensitivity analysis based on the phase-field function. The sharp interface is also presented by a continuous function between zero and one with a large gradient. To avoid the numerical solutions falling into the trivial solution, the hard boundary condition is implemented for our PINNs' training. Finally, numerical results are given to prove the feasibility and effectiveness of the proposed numerical method.

Keywords

PINNs, Phase-Field, Shape Optimization, Incompressible Navier-Stokes Equations

1. Introduction

Shape optimization for the fluid flows is widely interested in the optimum designs [1]-[4]. To obtain minimum drag wing profiles [5] or minimum pressure drop diffusers [6], when striving for a decreased drag or pressure drop, the feasible design modifications have concerned adjusting selected parts of the boundary to the fluid region. Borrvall and Petersson [7] optimized the dissipated power of the stokes flow. Allaire *et al.* [8] minimized the compliance of elastic structure.

Let u and p be the velocity and pressure of the incompressible flow which is governed by the incompressible Navier-Stokes equations in a domain Ω . To obtain optimal shape in Ω , the purpose is to find an optimal shape Ω^* in certain admissible shape set $\Omega_{set} \in \mathbb{R}^*$, $d = 2, 3$, satisfying

$$J(\Omega^*) = \min_{\Omega_{set}} J(\Omega), \quad (1)$$

where $J(\Omega) = \int_{\Omega} j(\mathbf{u}, p) \, d\mathbf{x}$ is constantly referred to as objective functional or cost functional. A typical choice of $J(\Omega)$ used in this context is the total potential energy, which is given by $j(\mathbf{u}, p) = 0.5\mu\nabla\mathbf{u} \cdot \nabla\mathbf{u}$ [4], which is used to represent the shear stress and influences the damage of fluid flows on the optimized shape surface [9]. In the framework of the model, the incompressible Navier-Stokes equations for the absence of body fluid forces is

$$\begin{cases} -\mu\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = 0 & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \end{cases} \quad (2)$$

where the parameter μ is the viscosity of the fluid. The inflow boundary Γ_{in} and outflow boundary Γ_{out} are fixed during the optimization, where inflow velocity \mathbf{u}_in is specified. The no-slip or free-slip boundary ξ is the concerned part for an optimized shape.

There are many numerical methods for solving the shape optimization problem Equations (1), (2). Among those methods, the phase-field method is an outstanding approach for various free boundary problems [10] [11], topology optimization for elastic structure [12] [13], and photonic devices [14] [15]. The phase field method stands out in its treatment of the interface as a physically diffuse thin layer. The interface is sharp conceptually but regularized numerically by a continuous function between zero and one. Due to the generation of the interface, mixing energy is introduced to govern the dynamics of the diffusive interface. Wallin [13] and Garcke [4] use the Ginzburg-Landau energy as the mixing energy over the interfacial layer, but the mixing energy can be dated back to Van Der Waals energy \mathcal{W} .

Neural networks are mathematical approximations that have been effectively applied in various fields, including speech recognition and classification tasks [16]-[18]. Neural networks have been applied to multiple scientific problems, such as turbulence model design from data [19] and for simulating heat transfer problems [20]. Among the methods that use deep learning for scientific computing, Physics-Informed Neural Networks (PINNs) [21] is a popular approach for simulating physical problems. Leverage all the information available about the problem, such as the underlying PDEs, boundary conditions, physical properties, and known measurement data, to train a neural network that best approximates the model. These networks transform boundary-value problems into optimization problems, providing flexibility to include any constraint in the problem. PINNs have been applied to a variety of forward, inverse, and parameterized problems [20] [22] [23], including fluid dynamics problems.

The remainder of this paper is organized as follows. In the next section, we introduce the phase-field model and its shape sensitivity analysis. The architecture of the PINNs and the loss functions used in training are presented in Section 3. We present the experimental results in Section 4. Finally, we provide the conclusions in Section 5.

2. Phase-Field Model for Fluid Shape Optimization

The fundamental idea of the phase-field model is to assume that the interface is a diffusive thin layer, as shown in the figure of **Figure 1**. We define the phase-field function ϕ over the whole design domain D to regularize numerically the sharp conceptually, which consists of three different parts of **Figure 1**: the material domain Ω with $\phi=1$, the void domain $D \setminus \Omega$ with $\phi=0$, and a fictitious diffused interface ξ with $0 < \phi < 1$.

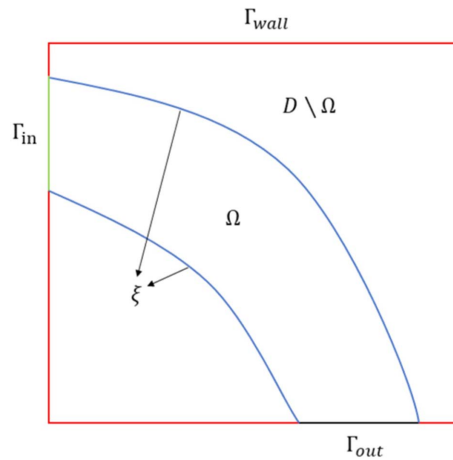


Figure 1. The shape optimization domain D .

In the framework of the phase-field method, Equation (2) should be extended to the design domain D . The basic idea of a porous medium approach [4] [7] [9] is to add the term $\phi(1-\phi)\mathbf{u}$ to Equation (2), then Equation (2) ‘interpolate’ between the steady-state Navier–Stokes equations where $\phi(\mathbf{x})=1$ and a virtual Darcy flows through a porous medium where $\phi(\mathbf{x})=0$. Thus we get

$$\begin{cases} -\mu\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p + \alpha(1-\phi) = 0 & \text{in } D, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } D, \end{cases} \quad (3)$$

where the parameter α penalizes the phase field. The boundary of D is also extended into three parts: $\Gamma_D = \Gamma_{in} + \Gamma_{out} + \Gamma_{wall}$, where homogeneous boundary conditions are supposed for the fluid flow. The inflow boundary Γ_{in} and the out-flow boundary Γ_{out} are fixed during the optimization.

To ensure that the velocity vanishes in the void domain $D \setminus \Omega$, an additional penalization term [4] is added to the objective function (1). Thus, the new objective function is rewritten as

$$J(\phi, \mathbf{u}) = \int_D j(\phi, \mathbf{u}) \, dx = \frac{1}{2} \int_D \mu \nabla \mathbf{u} : \nabla \mathbf{u} \, dx + \frac{\alpha}{2} \int_D (1-\phi) |\mathbf{u}|^2 \, dx. \quad (4)$$

The phase-field model for fluid shape optimization is to find a ϕ^* to minimize the Van Der Waals free energy, given by

$$\mathcal{W} = \int_D \frac{k}{2} |\Delta\phi|^2 + f(\phi, \mathbf{u}) \, dx, \quad (5)$$

with the constraints of state Equation (3). the k characters are the width of the diffusive thin layer and $f(\phi, \mathbf{u})$ is responsible for the moving direction of interfaces. Following the suggestion of Takezawa in [12], it is

$$f(\phi, \mathbf{u}) = \frac{1}{4}w(\phi) + 30\frac{\partial J(\phi, \mathbf{u})}{\partial \phi}g(\phi), \tag{6}$$

where $w(\phi) = \phi^2(1-\phi)^2$ and $g(\phi) = \phi^3(6\phi^2 - 15\phi + 10)$ are two sets of basis functions. To fulfill the movement of the interface with a diffusive thin layer in the direction indicated by $\frac{\partial J(\phi, \mathbf{u})}{\partial \phi}$, $w(0) = w(1) = w'(0) = w'(1) = 0$ and $g(0) = 0, g(1) = 1$ are the necessary conditions.

Authors presented a new objective function (4) in reference [24], Then the adjoint equation of (3) is rewritten as follows

$$\begin{cases} -\mu(\Delta \mathbf{w} + \Delta \mathbf{u}) + (\mathbf{w} \cdot \nabla) \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{w} - \nabla q + \alpha(1-\phi)(\mathbf{w} + |\mathbf{u}|) = 0 & \text{in } D, \\ \nabla \cdot \mathbf{w} = 0 & \text{in } D, \\ \mathbf{w} = 0 & \text{on } \Gamma_{in} \cup \Gamma_{wall}, \\ \mu \nabla \mathbf{w} \cdot \mathbf{n} + (\mathbf{u} \cdot \mathbf{n}) \mathbf{w} + q \mathbf{n} = 0 & \text{on } \Gamma_{out}, \end{cases} \tag{7}$$

where \mathbf{w} and q are the adjoint functions. \mathbf{n} is the unit outward normal vector. The shape sensitivity of the objective function is given as follows

$$\frac{\partial J(\phi, \mathbf{u})}{\partial \phi} = -\frac{\alpha}{2} \int_D |\mathbf{u}|^2 \, dx + \alpha \int_D \mathbf{u} \cdot \mathbf{w} \, dx. \tag{8}$$

The gradient flow method $\frac{\partial \phi}{\partial t} = -\frac{\partial \mathcal{W}}{\partial \phi}$ is used to find the minimum of the Equations (4) and (5). From this, it can be obtained the phase-field equation

$$\frac{\partial \phi}{\partial t} = k\Delta \phi + \phi(\phi-1) \left\{ \phi - \frac{1}{2} - 900 \frac{\partial J(\phi, \mathbf{u})}{\partial \phi} (1-\phi)\phi \right\}. \tag{9}$$

Moreover, we can simplify the model in the steady state. In this paper, we consider the problem with steady state, i.e. $\frac{\partial \phi}{\partial t} = 0$. So we can get the phase-field

model for fluid shape optimization as follows: Find $[\mathbf{u}, p, \mathbf{w}, q, \phi]$ satisfies

$$\begin{cases} -\mu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p + \alpha(1-\phi) = 0 & \text{in } D, \\ -\mu(\Delta \mathbf{w} + \Delta \mathbf{u}) + (\mathbf{w} \cdot \nabla) \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{w} - \nabla q + \alpha(1-\phi)(\mathbf{w} + |\mathbf{u}|) = 0 & \text{in } D, \\ k\Delta \phi + \phi(\phi-1) \left\{ \phi - \frac{1}{2} - 900 \frac{\partial J(\phi, \mathbf{u})}{\partial \phi} (1-\phi)\phi \right\} = 0 & \text{in } D, \\ \frac{\partial J(\phi, \mathbf{u})}{\partial \phi} = -\frac{\alpha}{2} \int_D |\mathbf{u}|^2 \, dx + \alpha \int_D \mathbf{u} \cdot \mathbf{w} \, dx & \text{in } D, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } D, \\ \nabla \cdot \mathbf{w} = 0 & \text{in } D, \end{cases} \tag{10}$$

and boundary conditions

$$\begin{cases} \mu \nabla \mathbf{w} \cdot \mathbf{n} + (\mathbf{u} \cdot \mathbf{n}) \mathbf{w} + q \mathbf{n} = 0 & \text{on } \Gamma_{out}, \\ \mathbf{w} = 0 & \text{on } \Gamma_{in} \cup \Gamma_{wall}. \end{cases} \tag{11}$$

3. Numerical Methods

In this section, we present the numerical method for solving the shape optimization problem (10), (11). To gain a better understanding of neural networks, we consider a simple example of a one-layer neural network. If \mathbf{x} is the input vector and \mathbf{y}_1 is the output vector to have dimensions n and m , respectively, the output can be expressed as

$$\mathbf{y}_1 = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{12}$$

where \mathbf{W} is a weight matrix of dimension $m \times n$ and \mathbf{b} is a bias vector of dimension m . The activation function $\sigma(z)$ is user-defined and can take various forms, such as a cosine or a hyperbolic tangent. This function is applied to each component of a vector \mathbf{z} . By stacking multiple layers, we can establish a feedforward neural network as illustrated in **Figure 2**. The output of the network is then expressed as

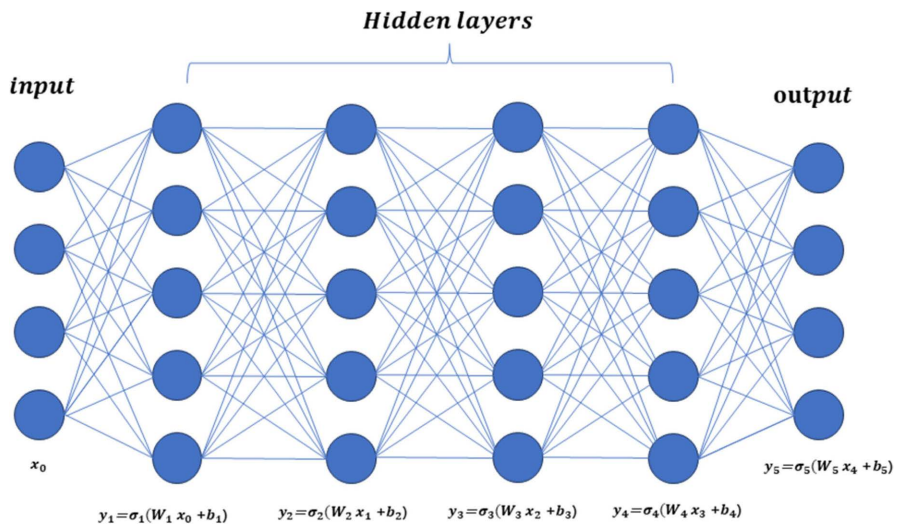


Figure 2. Feedforward neural network.

$$\mathcal{N}_\theta(\mathbf{x}) = \sigma_l(\mathbf{W}_l \sigma_{l-1}(\mathbf{W}_{l-1} \sigma_{l-2}(\dots(\sigma_1(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1))\dots))), \tag{13}$$

where l and θ represent the number of layers and all the parameters of the neural network, respectively. One can build a more complex neural network by changing the connections between the layers.

Denoted the differential operator about PDEs in regions D and ∂D by F and B , the phase-field model can be expressed as

$$\begin{cases} F[U(\mathbf{x})] = \mathbf{0} & x \in D, \\ B[U(\mathbf{x})] = b(\mathbf{x}) & x \in \partial D. \end{cases} \tag{14}$$

Once the problem has been described, the next step core is using a formulation of the physics-informed loss function to determine the optimal values for the parameter θ . The loss function $\mathcal{L}(\mathbf{x}, \theta)$, constructed by evaluating the residuals

about the governing PDEs ($\mathcal{L}_f(\mathbf{x}, \theta)$) and boundary conditions ($\mathcal{L}_b(\mathbf{x}, \theta)$), quantifies the network's deviation from the physical laws encapsulated by the governing differential equations and are defined as follows

$$\begin{cases} \mathcal{L}_f(\mathbf{x}; \theta) = F[\mathcal{N}_\theta(\mathbf{x})] - \mathbf{0} & \mathbf{x} \in D, \\ \mathcal{L}_b(\mathbf{x}; \theta) = B[\mathcal{N}_\theta(\mathbf{x})] - b(\mathbf{x}) & \mathbf{x} \in \partial D, \end{cases} \quad (15)$$

The aggregated loss function for the PINNs, combining these residuals, is given by

$$L(\mathbf{x}; \theta) = \frac{1}{N_D} \sum_{i=1}^{N_D} \|\mathcal{L}_f(x_i; \theta)\|^2 + \frac{1}{N_\Gamma} \sum_{j=1}^{N_\Gamma} \|\mathcal{L}_b(x_j; \theta)\|^2, \quad (16)$$

where N_Ω, N_Γ denote the number of collocation points in D and on ∂D . And $\|\cdot\|$ is the discrete L_2 norm. With the PINNs, we approximate the solution of the shape optimization problem using a feedforward neural network.

While various optimization methods could address this minimization problem, in the context of this work, we will only consider gradient-based optimization methods. Gradient-based methods leverage the property that the gradient of a function points in the direction of the function's steepest. To minimize the function, one would generally move in the direction opposite to the gradient, with a step size proportional to a user-defined learning rate. Popular gradient-based methods include the Stochastic Gradient Descent (SGD), the Limited-memory Broyden-Fletcher-Goldfarb-Shanno method(L-BFGS) [25], and the Adaptive Moment Estimation method (Adam) [26]. The gradient is calculated using the back-propagation algorithm, which employs the chain rule and automatic differentiation to compute the partial derivatives of the output concerning the weights, the biases, and the network's input.

Since the problem includes integral terms, we incorporate Gaussian integration in the training of PINNs to approximate the calculation of shape sensitivity (8). During the training process, we implement hard boundary conditions in the PINNs' boundary conditions. This can prevent excessive void domain where $\phi = 0$ from causing trivial numerical solutions. The training flow chart of PINNs is shown in the following **Figure 3**.

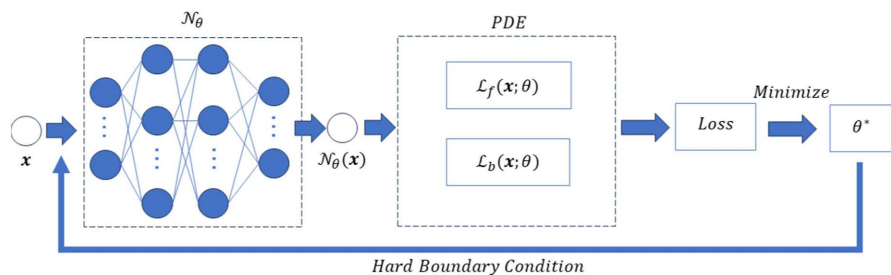


Figure 3. PINN's training flowchart.

4. Numerical Results

In this section, we first present an accuracy test of the algorithm by comparing the

numerical results obtained from the PINNs with the exact solution. Our numerical algorithm exhibits excellent convergence properties for models with known exact solutions. Next, we conduct numerical simulations for practical problems, presenting the optimized forms of the pipeline under different boundary conditions.

4.1. Accuracy Test

In the two-dimensional shape optimization domain $D = [0,1]^2$, we take the following exact solution

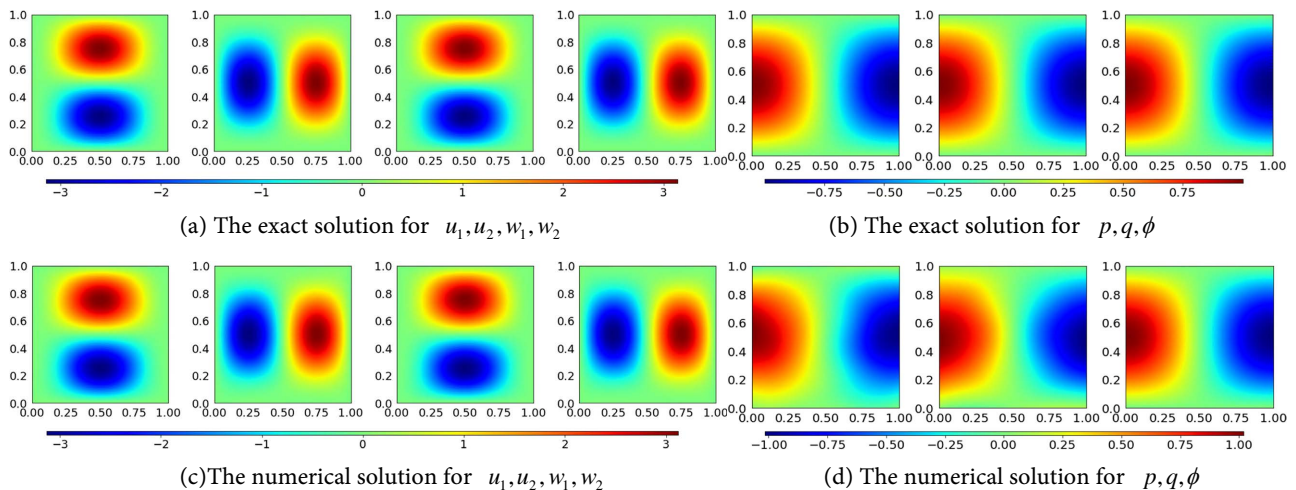
$$\begin{cases} \mathbf{u}_{exa}(x) = (\pi \sin(2\pi y) \sin^2(\pi x), -\pi \sin(2\pi x) \sin^2(\pi y)), \\ \mathbf{w}_{exa}(x) = (\pi \sin(2\pi y) \sin^2(\pi x), -\pi \sin(2\pi x) \sin^2(\pi y)), \\ p_{exa}(x) = q_{exa}(x) = \phi_{exa}(x) = \cos(\pi x) \sin(\pi y). \end{cases} \quad (17)$$

This will be used to evaluate the accuracy and convergence behavior of the shape optimization problem. To enhance the training effectiveness, we will randomly select 81 points within D to construct the loss functions for p and q , which will be incorporated into the experimental model:

$$\mathcal{L}_{pq}(\mathbf{x}; \theta) = \sum_{i=1}^{81} \|p_{exa}(x_i) - p_{N_\theta}(x_i)\|^2 + \sum_{i=1}^{81} \|q_{exa}(x_i) - q_{N_\theta}(x_i)\|^2. \quad (18)$$

Substituting the numerical solution (17) into the model, we obtain the residual of the right-hand side term of the corresponding equation and set the parameters: $\alpha = 10$, $k = 0.0006$, $\mu = 0.025$, $u_{in}(0, y) = (0, 0)$. We selected 2601 training points and 10,000 Gaussian points in D and solved the experimental results using PINNs with four layers, each containing fifty neurons. The Adam and L-BFGS optimizers will be used for 10,000 and 700,000 iterations, respectively, with a learning rate of 0.5×10^{-3} .

Figure 4(a) and Figure 4(b) are exact solution images, and Figure 4(c) and Figure 4(d) are numerical solution images. By observing the error images in Figure 4(e), Figure 4(f), and the loss value images in Figure 5, we can see that the numerical solutions converge to the exact solution with high accuracy.



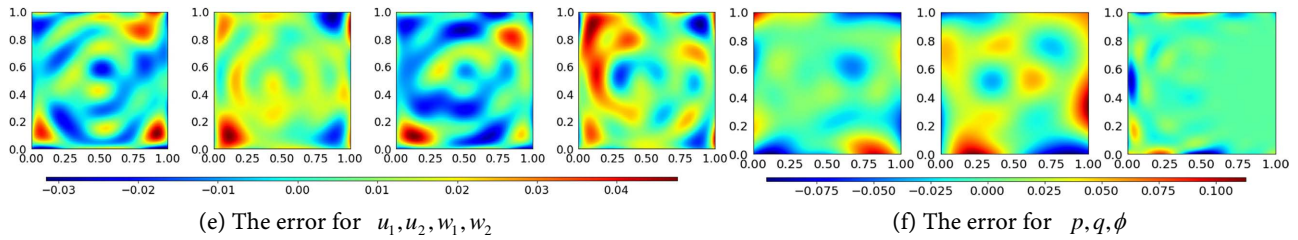


Figure 4. The result of the accuracy test.

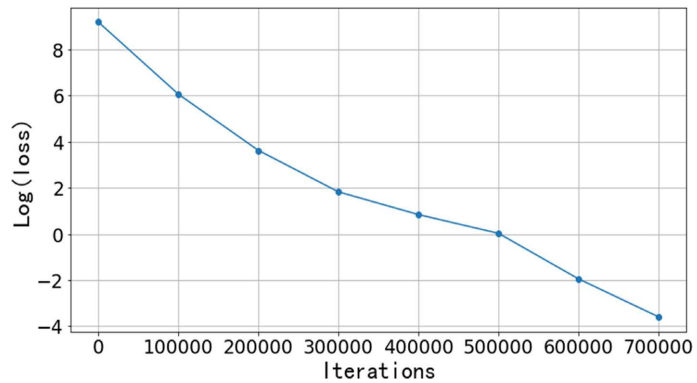
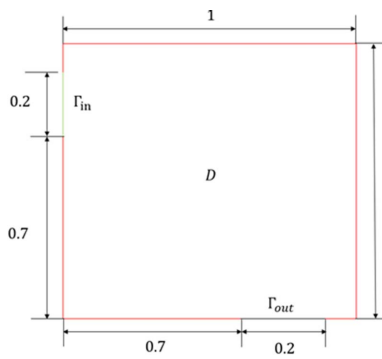


Figure 5. The loss value images of the PINNs.

4.2. The Shape Optimization Problem for Pipes

Now, we consider the design of a bend, and the purpose of designing work is to find an optimized connection between the inlet and outlet [27]. The domain is shown in **Figure 6(a)**. Set the parameters: $\alpha = 100$, $k = 0.0006$, $\mu = 0.025$, $u_{in} = (100(0.9 - y)(y - 0.7), 0)$. We selected 2601 training points and 10,000 Gaussian points in D and solved the experimental results using a PINN with four layers, each containing fifty neurons. The Adam and L-BFGS optimizers will be used for 10,000 and 200,000 iterations, respectively, with a learning rate of 0.5×10^{-3} . To avoid converging to a trivial solution, boundary conditions are handled as hard boundaries.

We find that the shape and loss in **Figure 6(b)**, **Figure 6(c)** are in the same direction as those reported in the literature (see [27] and references therein) and exhibit excellent convergence performance.



(a) Pipe domain

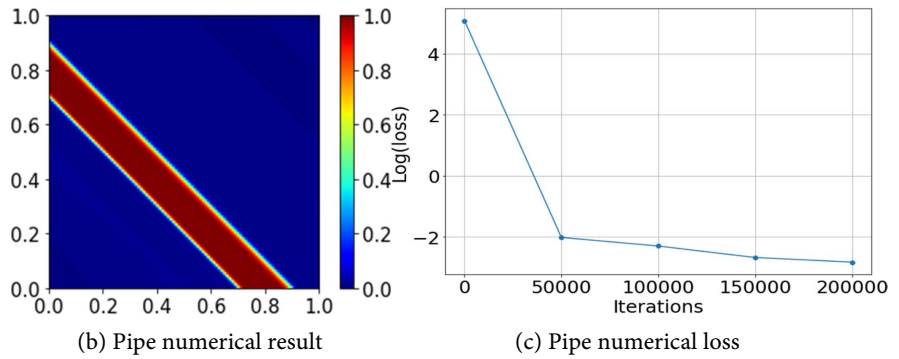


Figure 6. The pipe experiment.

As shown in **Figure 7(a)**, we set the inlet to be twice that of the pipe experiment and conducted another expansion experiment for prediction. Set the parameters: $\alpha = 10$, $k = 0.0006$, $\mu = 0.025$, $u_{in} = (100(0.9 - y)(y - 0.5), 0)$. We select 2601 training points and 10,000 Gaussian points and use PINNs with four layers, each containing fifty neurons. The Adam and L-BFGS optimizers will be used for 10,000 and 200,000 iterations, respectively, with a learning rate of 0.5×10^{-3} . To avoid converging to a trivial solution, boundary conditions are handled as hard boundaries. **Figure 7(b)** and **Figure 7(c)** present our results, showing that the model exhibits excellent convergence.

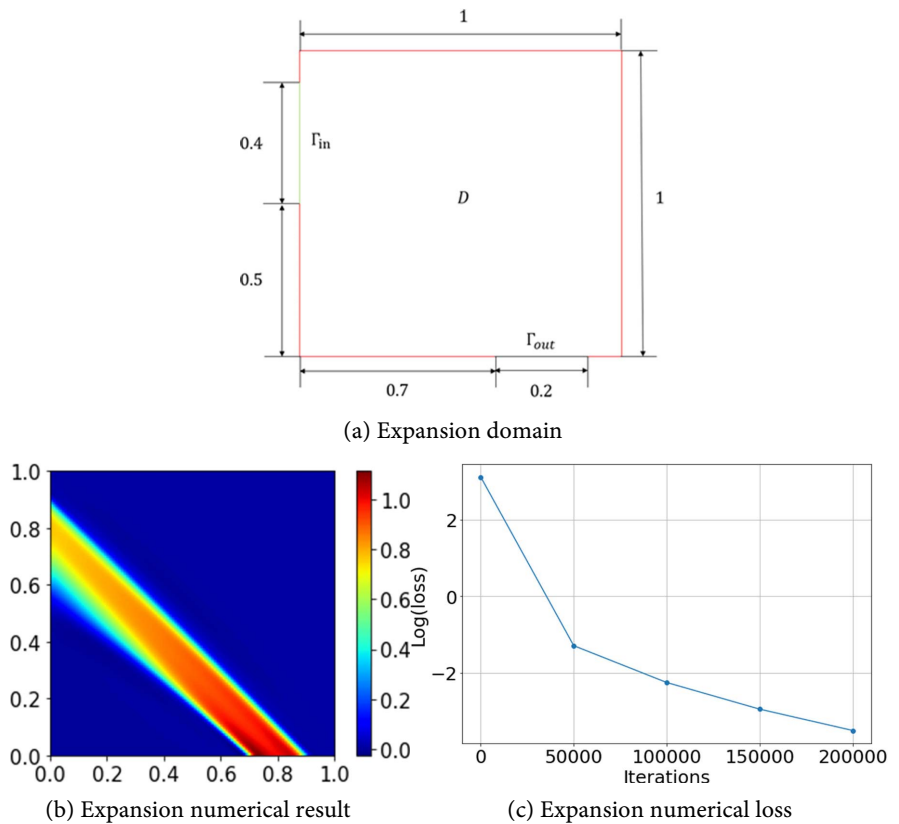


Figure 7. The expansion experiment.

5. Conclusion

In this paper, we use PINNs with hard boundary conditions to solve the phase-field model for fluid shape optimization. The accuracy test and the pipe experiment show that the established phase-field model aligns well with results from the literature, and the convergence achieved is also accurate. Moreover, we provide the expansion experiment result, demonstrating that using PINNs with hard boundaries to solve the phase-field model is effective and feasible for fluid shape optimization problems.

Funding

This work was supported by the NSF of China (Grant No. 11901100) and the Natural Science Foundation of Fujian Province of China (Grant No. 2020J05111).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Pironneau, O. (1982) Optimal Shape Design for Elliptic Systems. In: Drenick, R.F. and Kozin, F., Eds., *System Modeling and Optimization*, Springer, 42-66. <https://doi.org/10.1007/bfb0006123>
- [2] Mohammadi, B. and Pironneau, O. (2009) *Applied Shape Optimization for Fluids*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780199546909.001.0001>
- [3] Guest, J.K. and Prévost, J.H. (2006) Topology Optimization of Creeping Fluid Flows Using a Darcy-Stokes Finite Element. *International Journal for Numerical Methods in Engineering*, **66**, 461-484. <https://doi.org/10.1002/nme.1560>
- [4] Garcke, H., Hecht, C., Hinze, M. and Kahle, C. (2015) Numerical Approximation of Phase Field Based Shape and Topology Optimization for Fluids. *SIAM Journal on Scientific Computing*, **37**, A1846-A1871. <https://doi.org/10.1137/140969269>
- [5] Pironneau, O. (1974) On Optimum Design in Fluid Mechanics. *Journal of Fluid Mechanics*, **64**, 97-110. <https://doi.org/10.1017/s0022112074002023>
- [6] Çlabuk, H. and Modi, V. (1992) Optimum Plane Diffusers in Laminar Flow. *Journal of Fluid Mechanics*, **237**, 373-393. <https://doi.org/10.1017/s0022112092003458>
- [7] Borrvall, T. and Petersson, J. (2002) Topology Optimization of Fluids in Stokes Flow. *International Journal for Numerical Methods in Fluids*, **41**, 77-107. <https://doi.org/10.1002/flid.426>
- [8] Allaire, G., Jouve, F. and Toader, A. (2004) Structural Optimization Using Sensitivity Analysis and a Level-Set Method. *Journal of Computational Physics*, **194**, 363-393. <https://doi.org/10.1016/j.jcp.2003.09.032>
- [9] Zhang, B. and Liu, X. (2014) Topology Optimization Study of Arterial Bypass Configurations Using the Level Set Method. *Structural and Multidisciplinary Optimization*, **51**, 773-798. <https://doi.org/10.1007/s00158-014-1175-y>
- [10] Shen, J. and Yang, X. (2009) An Efficient Moving Mesh Spectral Method for the Phase-Field Model of Two-Phase Flows. *Journal of Computational Physics*, **228**, 2978-2992. <https://doi.org/10.1016/j.jcp.2009.01.009>

- [11] Sun, Y. and Beckermann, C. (2007) Sharp Interface Tracking Using the Phase-Field Equation. *Journal of Computational Physics*, **220**, 626-653. <https://doi.org/10.1016/j.jcp.2006.05.025>
- [12] Takezawa, A., Nishiwaki, S. and Kitamura, M. (2010) Shape and Topology Optimization Based on the Phase Field Method and Sensitivity Analysis. *Journal of Computational Physics*, **229**, 2697-2718. <https://doi.org/10.1016/j.jcp.2009.12.017>
- [13] Wallin, M., Ristinmaa, M. and Askfelt, H. (2011) Optimal Topologies Derived from a Phase-Field Method. *Structural and Multidisciplinary Optimization*, **45**, 171-183. <https://doi.org/10.1007/s00158-011-0688-x>
- [14] Takezawa, A. and Kitamura, M. (2014) Phase Field Method to Optimize Dielectric Devices for Electromagnetic Wave Propagation. *Journal of Computational Physics*, **257**, 216-240. <https://doi.org/10.1016/j.jcp.2013.09.051>
- [15] Wu, S., Hu, X. and Zhu, S. (2018) A Multi-Mesh Finite Element Method for Phase-Field Based Photonic Band Structure Optimization. *Journal of Computational Physics*, **357**, 324-337. <https://doi.org/10.1016/j.jcp.2017.12.031>
- [16] Alam, M., Samad, M.D., Vidyaratne, L., Glandon, A. and Iftekharuddin, K.M. (2020) Survey on Deep Neural Networks in Speech and Vision Systems. *Neurocomputing*, **417**, 302-321. <https://doi.org/10.1016/j.neucom.2020.07.053>
- [17] Chai, J., Zeng, H., Li, A. and Ngai, E.W.T. (2021) Deep Learning in Computer Vision: A Critical Review of Emerging Techniques and Application Scenarios. *Machine Learning with Applications*, **6**, Article ID: 100134. <https://doi.org/10.1016/j.mlwa.2021.100134>
- [18] Nassif, A.B., Shahin, I., Attili, I., Azzeh, M. and Shaalan, K. (2019) Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access*, **7**, 19143-19165. <https://doi.org/10.1109/access.2019.2896880>
- [19] Beck, A., Flad, D. and Munz, C. (2019) Deep Neural Networks for Data-Driven LES Closure Models. *Journal of Computational Physics*, **398**, Article ID: 108910. <https://doi.org/10.1016/j.jcp.2019.108910>
- [20] Cai, S., Wang, Z., Wang, S., Perdikaris, P. and Karniadakis, G.E. (2021) Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*, **143**, Article ID: 060801. <https://doi.org/10.1115/1.4050542>
- [21] Raissi, M. (2018) Deep Hidden Physics Models: Deep Learning of Non-Linear Partial Differential Equations. *Journal of Machine Learning Research*, **19**, 1-24.
- [22] Jin, X., Cai, S., Li, H. and Karniadakis, G.E. (2021) NSFnets (Navier-Stokes Flow Nets): Physics-Informed Neural Networks for the Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, **426**, Article ID: 109951. <https://doi.org/10.1016/j.jcp.2020.109951>
- [23] Lim, J. and Psaltis, D. (2022) MaxwellNet: Physics-Driven Deep Neural Network Training Based on Maxwell's Equations. *APL Photonics*, **7**, Article ID: 011301. <https://doi.org/10.1063/5.0071616>
- [24] Li, F. and Hu, X. (2019) A Phase-Field Method for Shape Optimization of Incompressible Flows. *Computers & Mathematics with Applications*, **77**, 1029-1041. <https://doi.org/10.1016/j.camwa.2018.10.029>
- [25] Bollapragada, R., Nocedal, J., Mudigere, D., Shi, H.J. and Tang, P.T.P. (2018) A Progressive Batching L-BFGS Method for Machine Learning. *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 620-629.
- [26] Kingma, D. and Ba, J. (2014) Adam: A Method for Stochastic Optimization.

<https://api.semanticscholar.org/CorpusID:6628106>

- [27] Duan, X., Ma, Y. and Zhang, R. (2008) Shape-Topology Optimization for Navier-Stokes Problem Using Variational Level Set Method. *Journal of Computational and Applied Mathematics*, **222**, 487-499. <https://doi.org/10.1016/j.cam.2007.11.016>