

## Article

# Extrapolation of Hydrodynamic Pressure in Lubricated Contacts: A Novel Multi-Case Physics-Informed Neural Network Framework

Faras Brumand-Poor <sup>\*</sup>, Niklas Bauer , Nils Plückerhahn , Matteo Thebelt , Silas Woyda  and Katharina Schmitz 

Institute for Fluid Power Drives and Systems (ifas), RWTH Aachen University, 52074 Aachen, Germany

\* Correspondence: faras.brumand@ifas.rwth-aachen.de

**Abstract:** In many technical applications, understanding the behavior of tribological contacts is pivotal for enhancing efficiency and lifetime. Traditional experimental investigations into tribology are often both costly and time-consuming. A more profound insight can be achieved through elastohydrodynamic lubrication (EHL) simulation models, such as the ifas-DDS, which determines precise friction calculations in reciprocating pneumatic seals. Similar to other distributed parameter simulations, EHL simulations require a labor-intensive resolution process. Physics-informed neural networks (PINNs) offer an innovative method to expedite the computation of such complex simulations by incorporating the underlying physical equations into the neural network's parameter optimization process. A hydrodynamic PINN framework has been developed and validated for a variant of the Reynolds equation. This paper elucidates the framework's capacity to handle multi-case scenarios—utilizing one PINN for various simulations—and its ability to extrapolate solutions beyond a limited training domain. The outcomes demonstrate that PINNs can overcome the typical limitation of neural networks in extrapolating the solution space, showcasing a significant advancement in computational efficiency and model adaptability.

**Keywords:** elastohydrodynamic simulation; hydrodynamic pressure extrapolation; physics-informed machine learning; physics-informed neural networks; pneumatic sealing; tribology



**Citation:** Brumand-Poor, F.; Bauer, N.; Plückerhahn, N.; Thebelt, M.; Woyda, S.; Schmitz, K. Extrapolation of Hydrodynamic Pressure in Lubricated Contacts: A Novel Multi-Case Physics-Informed Neural Network Framework. *Lubricants* **2024**, *12*, 122. <https://doi.org/10.3390/lubricants12040122>

Received: 14 February 2024

Revised: 25 March 2024

Accepted: 4 April 2024

Published: 5 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Lubricated tribological contacts, such as sealing contacts, are critical parts of many technical systems since they have a decisive influence on the performance, efficiency, and durability of the corresponding technical system. The complexity of lubricated contacts requires a comprehensive understanding of all occurring phenomena through a precise model. The dynamic friction, influenced significantly by fluid dynamics, plays an essential role in the accurate modeling of lubricated contacts. Achieving an analytical solution for the behavior of lubricated contacts necessitates numerous simplifications, making it impractical in many cases. Additionally, experimental characterization is often unfeasible due to its time-consuming and costly nature. As a result, the dynamic friction behavior is modeled using elastohydrodynamic lubrication (EHL) simulations, which account for the phenomena through the Reynolds equation, delineating the pressure distribution within the sealing gap and a calculation for the deformation of the contact partners. At ifas, an EHL simulation model for the simulation of reciprocating seals, the ifas-DDS, was implemented. The ifas-DDS calculates the friction of reciprocating seals by analyzing the hydrodynamics within the sealing contact according to the Reynolds equation, the contact mechanics, and the seal's deformation. In prior work, a thorough investigation of the agreement of the EHL simulation model with a test rig across various operating conditions, offering validation through experimental measurements has been conducted [1]. However, a notable limitation is the extensive computation time required for simulating short-term physical processes, attributed to the complexity of the underlying partial differential equations and

their computationally intensive numerical solutions. Enhancing computational power presents a straightforward method to reduce computation time, but this may not always be feasible, especially as simulation complexity increases (e.g., system with multiple contacts or even multiple components) or for applications with the necessity to perform the calculations in near real time (e.g., control applications or digital twins).

A possible alternative to the time-expensive EHL simulations is the use of machine learning, e.g., with neural networks. However, traditional neural networks are constrained by their inherent lack of understanding of the underlying physical principles governing the datasets they are trained on. This limitation significantly restricts their capacity for predictive extrapolation beyond the scope of the input data. Typically, the primary objective of a neural network in regression tasks is to minimize the discrepancy between its predicted outputs and the actual observed data. Although the parameters of the network can be optimized to achieve this objective, the optimality of these parameters is often confined to the range of data provided, potentially leading to overfitting. In recent years, a novel method for encapsulating physical laws within computational models has been developed, termed physics-informed neural networks (PINNs). This drawback underscores the necessity for embedding physical rules directly into the model, enabling PINNs to overcome the limitations of conventional neural networks by enhancing their predictive accuracy and generalizability across unobserved data regimes.

PINNs integrate physical rules directly into the network's architecture, representing a significant advancement in physics-based machine learning. These innovative methods are adept at addressing challenges characterized by partial differential equations (PDEs). In contrast to conventional neural networks, which rely on a data-driven loss function, the loss function of a PINN encompasses not only initial and boundary conditions but also the PDEs' residual. Cuomo et al. have described PINNs as a mesh-free approach that transforms the complexities of solving PDEs by reformulating the direct solution of these equations into an optimization problem focused on minimizing the loss function [2]. Most research on PINNs applied to EHL simulations has focused on the hydrodynamic aspect of the computation, thus neglecting the deformation, contact mechanics, and friction and purely focusing on calculating the pressure. As a pioneer in this field, Almqvist investigated the interpolation with PINNs for the determination of hydrodynamic pressure, described by a simplified variant of the Reynolds equation [3]. In subsequent work, PINNs were applied to 2D problems [4–6]. The newest achievements consider the computation of pressure and cavitation in tribological systems [7–9]. The conducted research displays the potential of PINNs to combine the strengths of distributed simulation models with the computational efficiency of classical neural networks. This approach ensures that computations are not only based on physical rules, offering robustness and accuracy, but also benefit from the significantly reduced computation times associated with parameterized neural networks.

This contribution assesses the capabilities of PINNs in performing extrapolation and handling multi-case scenarios. For this analysis, a hydrodynamic PINN framework is employed, which has previously been implemented and validated for single-case and interpolation scenarios [10]. Here, a single-case scenario refers to a specific set of inputs, including pressure boundary conditions, geometry, and viscosity, while a multi-case scenario involves a collection of inputs that vary across different configurations. Building on this groundwork, this study investigates the performance of PINNs concerning not only single-case, but also multi-case tasks involving variations in pressure boundaries and geometry and extrapolation tasks across parameters such as pressure boundary, position, and geometry. The difficulty rises for multi-case and extrapolation tasks compared to the single-case and interpolation tasks investigated in the prior publication. Consistent with the studies referenced earlier, this research focuses on the hydrodynamic aspect of EHL. Consequently, it concentrates on calculating the pressure distribution while disregarding factors such as friction, deformation, and contact mechanics. The subsequent section introduces hydrodynamic lubrication, followed by a detailed description of PINNs. A detailed overview of the latest research on the application of PINNs in hydrodynamic lubrication

scenarios is also provided. Section 2.2.1 presents the investigated loss function, a variant of the Reynolds equation. The comprehensive framework and the specific tasks examined are delineated in Sections 2.3 and 2.4. Subsequently, the results are presented and validated against a modified version of the ifas-DDS, herein referred to as the rigid ifas-DDS. This variant similarly omits consideration of all previously mentioned aspects of the complete EHL model. The paper concludes with a discussion and summary of the findings.

## 2. Materials and Methods

### 2.1. Hydrodynamic Lubrication

EHL simulations are an essential tool for the detailed analysis of friction, leakage, and wear within lubricated mechanical interfaces. These simulations evaluate the dynamic relationship between lubricants and the surfaces in contact, focusing on the computational modeling of surface deformations and the resulting development of hydrodynamic pressure within the contact zone. Such simulations are vital for designing and optimizing tribological contacts in various industrial applications.

The ifas-DDS model is an advanced simulation framework, validated with experimental measurements for different operation points [1], specifically designed to elucidate the complex interactions between a seal and its mating counterface. Central to this model is the consideration of a lubricating film that separates the seal and counterface. This film plays a pivotal factor in determining the behavior of the seal. The model employs the finite element software Abaqus to accurately simulate the seal's deformation under operational conditions. The hydrodynamic phenomena are computed through the Reynolds equation, which is seamlessly integrated into Abaqus via custom user subroutines.

Within the scope of this research, the primary emphasis is placed on solving the Reynolds equation, while simplifying the model by excluding the deformation of the contacting surfaces, the contact mechanics, and the friction. This approach focuses on investigating the lubrication aspects without the added complexity of accommodating deformations.

To validate the PINN approach presented in this study, a variant of the ifas-DDS model, the rigid ifas-DDS, is utilized. This validation allows for a direct comparison of both solvers, PINN and rigid ifas-DDS, as both are implemented to solve the same underlying equations. The rigid ifas-DDS omits the deformational, friction, and contact mechanics aspects to concentrate on the lubrication dynamics. The original ifas-DDS model incorporates an extended form of the Reynolds equation, augmented with flow factors  $\Phi^\tau$  and  $\Phi^p$ , as described by Patir and Cheng [11]. This enhancement allows for the accounting of surface topography effects on lubrication, a critical consideration in accurately modeling real-world scenarios. Additionally, the model integrates the Jakobsson–Floberg–Olsson cavitation model, which describes the formation of a gaseous phase, e.g., due to vaporization or dissolution of solved air in the fluid due to localized pressure drops by introducing the cavity fraction  $\theta$  [12]. Here,  $\theta$  describes the local volume fraction of the gaseous phase ranging from 0 at no cavitation to 1 for full cavitation.

The comprehensive Reynolds equation, as incorporated within the ifas-DDS model, expands upon the original equation originally formulated by Osborne Reynolds in 1886. This extended form is detailed as follows [13]:

$$\underbrace{\frac{v}{2} \frac{\partial}{\partial x} \left( (1 - \theta)(\rho h + \rho R_q \Phi^\tau) \right)}_{\text{Couette flow}} - \underbrace{\frac{1}{12\eta} \frac{\partial}{\partial x} \left( \Phi^p \rho h^3 (1 - \theta) \frac{\partial p}{\partial x} \right)}_{\text{Poiseuille flow}} + \underbrace{\frac{\partial}{\partial t} \left( (1 - \theta) \rho h \right)}_{\text{Transient term}} = 0 \quad (1)$$

The Reynolds equation describes the hydrodynamic pressure  $p$ , in a lubricated contact. This pressure depends on several parameters: the fluid's density  $\rho$ , viscosity  $\eta$ , the relative velocity  $v$  between the contact surfaces, the gap height  $h$ , the time  $t$ , the axial coordinate  $x$ , and derivatives of pressure with respect to time and position,  $\frac{\partial}{\partial t}$  and  $\frac{\partial}{\partial x}$ . The parameters  $\Phi^\tau$  and  $\Phi^p$  signify the shear and pressure flow factors, respectively, which adjust the equation to include the effects of surface roughness on the hydrodynamics. The parameter  $R_q$  represents the root mean square roughness of the contact surfaces and includes the

effect of surface textures on the distribution of hydrodynamic pressure. The cavity friction  $\theta$  measures the extent of cavitation within the lubricated contact.

## 2.2. Physics-Informed Neural Networks

The Reynolds equation, as previously introduced, serves as an essential mathematical tool for modeling pressure distribution in lubricated contacts. Given the absence of an analytical solution, numerical methods such as finite volume, element, or difference approaches are employed for solving tribological problems. However, these methods often require extensive computational efforts due to their numerical characteristics.

In recent years, machine learning has emerged as a promising methodology in tribology, demonstrating significant advancements in the field [14,15]. A notable application is the development of deep neural networks for detecting faults in tribological systems, including ball bearings, journal bearings, and slipper bearings [16–18]. For instance, Hess and Shang have adeptly utilized a convolutional neural network to calculate the elasto-hydrodynamic pressure distribution in journal bearings [19]. Usually, machine learning models are data-driven, often referred to as black-box models, and benefit due to their flexibility and straightforward implementation. However, since their emergence, hybrid models, which combine data-driven approaches with physics-based insights, have gained significance. These models profit from the frequent unavailability of sufficient measurement data and a comprehensive mathematical system description, rendering both purely data-driven and exclusively physics-based (white-box) modeling approaches impractical [20]. Over time, various configurations of hybrid models, including sequential, parallel, and structured forms, have been explored [21–23].

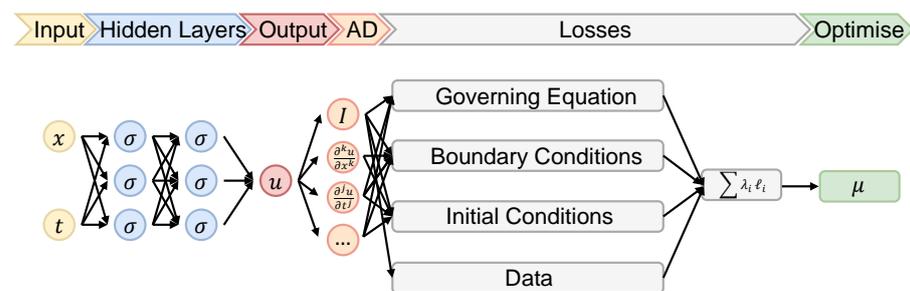
A promising advancement in the field of tribology is physics-informed machine learning (PIML), which enriches machine learning models with physics-based rules, thereby enhancing their capability to accurately represent phenomena such as friction, wear, and lubrication [24]. Applications of PIML in tribology range from assessing lubrication conditions in hydrodynamic interfaces to predicting wear or damage. Unlike conventional machine learning methods, which rely exclusively on data-driven strategies (black-box models), PIML, particularly through PINNs, incorporates physical principles to guide the learning process. Consequently, these models often yield outputs that are not only more accurate but also more reliable than those determined by data-driven approaches. Compared to the prior mentioned hybrid models, a PINN can be viewed as a hybrid model, with the neural network as the prediction model, but is fed with information from governing physical laws during training by adding residual terms in the loss function [25].

The foundational work in the domain of physics-based regularization of neural networks was conducted by Hyuk [26] and Lagaris [27], following the proofs by Cybenko [28] and Hornik [29]. Cybenko demonstrated that feed-forward neural networks with at least one hidden layer are capable of approximating any continuous function to a desired degree of accuracy. Hornik expanded this to include Borel measurable functions.

Although Hyuk and Lagaris did not explicitly use the term “physics-informed” in their research, the objectives of their work bear a significant resemblance to what is now understood as the principles underlying PINNs. In Hyuk’s approach, the loss function of the neural network was extended to embed the governing differential equation, laying the groundwork for the later-called field of PINNs. This exploration into guiding neural network training with physical laws has significantly influenced subsequent developments in the field of PIML, bridging the gap between traditional machine learning and domain-specific knowledge. The concept of integrating physical laws with neural network training initially encountered rarely attention, primarily due to the lack of computational resources and the underdevelopment of computational algebra techniques at the time. However, this idea has experienced a resurgence, led by advancements in efficient gradient calculation methods, such as automatic differentiation, and significant improvements in hardware capabilities.

The reappearance of PIML began with Owhadi in 2014, who incorporated prior knowledge into the solution process. He proposed formulating the solution of PDEs as Bayesian inference tasks, thereby introducing the idea of enriching algorithms with pre-existing knowledge [30]. Building on this foundation, Raissi and colleagues implemented a probabilistic machine learning algorithm for solving general linear equations via the Gaussian process, adapting it specifically for the integro-differential or partial differential equations [31,32]. This computation was further extended to tackle nonlinear partial differential equations [33,34]. A further notable advance was the invention of PINNs, which can be described as mesh-free models reformulating the solution of PDEs into an optimization task of a loss function [2]. Raissi introduced PINNs as a new class of hybrid solvers, which can determine the solution to several forward and inverse problems described by PDEs with high accuracy [35–37]. In addition to solving PDEs, Antonello et al. advanced the PINN concept to control tasks by adding the control inputs to the network, resulting in an algorithm capable of solving control applications [38]. The research on PINNs' extrapolation capabilities includes work by Kim et al., who developed an algorithm to adjust the gradient for the residual and boundary condition loss, aiming to extrapolate the solution of Burger's equation over time [39]. Fesser et al. explored transfer learning on equations like Burger and Allen-Cahn to enhance time extrapolation [40].

Figure 1 illustrates an exemplary PINN. The depicted loss is hybrid since it is physics-informed and data-based.



**Figure 1.** Schematic illustration of a physics-informed neural network [10].

The architecture of a PINN is analogous to that of a traditional neural network. It processes an input array, such as position  $x$  and time  $t$ , through a deep network to determine the network's output. The physics-informed loss is achieved by efficient gradient computation using automatic differentiation. This computer algebra technique leverages the principle that computer calculations consist of sequences of elementary operations and functions. By recurrently applying the chain rule, automatic differentiation enables the precise and rapid calculation of partial derivatives of any order [41]. These derivatives are used to compute the residual loss, which resembles the governing equation and is defined as an unsupervised loss [42].

In addition to residual loss, PINNs also integrate two other physics-informed losses: the losses associated with boundary conditions and initial conditions. These are considered supervised losses because the target values are known and explicitly included in the loss computation. Figure 1 illustrates an extra loss component in a hybrid PINN, which is the classical data-driven loss. The computation of these losses typically involves the calculation of the L2-norm (mean squared error, MSE) across collocation points sampled within the computational domain [43]. The losses are summed and eventually used to find the optimal parameters, weights, and biases, of the hidden layers by an optimizer.

In the subsequent subsection, the physics-informed loss function investigated in this study is detailed. Before this, an overview of existing research on the application of PINNs to hydrodynamic lubrication is provided.

### 2.2.1. Physical-Informed Loss

The initial publication on the application of PINNs to solve a simplified variant of the Reynolds equation was by Almqvist in 2021 [3]. This pioneering work was expanded upon by researchers such as Zhao et al., Li et al., and Yadav et al., who developed more advanced algorithms to address the 2D Reynolds equation in contexts ranging from linear sliders to gas bearings and journal bearings, respectively, [4–6]. A significant advancement was achieved by Rom, who became the first to implement PINNs for the stationary Reynolds equation, incorporating the Jakobsson–Floberg–Olsson (JFO) cavitation model. Moreover, Rom extended the PINN’s input to include relative eccentricity, thereby enabling the PINN to handle several geometrical configurations [7].

Furthering these advancements, Cheng et al. successfully implemented a PINN capable of solving the Reynolds equation with either the JFO or Swift–Stieber (SS) cavitation models. Their approach utilized three different multi-task learning strategies to effectively balance the loss components associated with the models [8]. Most recently, Xi et al. investigated the stationary Reynolds equation with cavitation, introducing both soft and hard constraints within the loss function to enhance the precision of the solution [9]. Additionally, Rimón et al. explored the feasibility of applying PINNs to EHL simulations, employing a simplified Reynolds equation and describing the seal’s deformation through the Lamé equation [44].

It is important to emphasize that the primary emphasis of these contributions has largely been on refining the PINN itself rather than on the development of a comprehensive framework for deploying PINNs in hydrodynamic lubrication tasks. Consequently, a significant degree of manual work remains necessary. Moreover, the research has focused on interpolation capabilities, thereby limiting the PINNs’ ability to predict pressure distributions for parameter combinations outside their training domain.

In this work, an automated framework was utilized to undertake the complex and time-consuming task of tuning the essential parameters of the whole training process, reducing the required manual work. Additionally, this study presents the capability of PINNs to solve several multi-case and extrapolation tasks. These applications, particularly in solving the Reynolds equation, have previously been either minimally investigated or completely overlooked. The focus lies in a variant of the Reynolds equation, but can be extended to the complete mathematical equation in the future. The assumptions made in this work are as follows:

- $\theta = 0$  (no cavitation)
- $\Phi^\tau = 0, \Phi^p = 1$  (ideally smooth surface)
- $\rho = \text{constant}$  (incompressible)
- Stationary: Partial derivatives with respect to time are irrelevant

These assumptions result in the following Reynolds equation:

$$\frac{v}{2} \frac{\partial}{\partial x} h - \frac{1}{12\eta} \frac{\partial}{\partial x} \left( h^3 \frac{\partial p}{\partial x} \right) = 0 \quad (2)$$

Figure 2 shows the hydrodynamic (HD)-PINN with two hidden layers as an example. The pressure  $p$  is computed by the network, which obtains the position  $x$ , the fluid density  $\rho$ , the dynamic viscosity  $\eta$ , the velocity of the counter surface  $v$ , and the four parameters  $h_1, h_2, h_3$ , and  $x_b$ , which describe the investigated height profile.

The height profile between the seal and the counter surface is described with the four coefficients over the position interval  $x \in [x_l, x_r]$  as follows:

$$h(x) = \begin{cases} \frac{h_1 - h_2}{x_b - x_l} \cdot \text{ReLU}(x_b - x) + h_2 + h_3 \cdot \left( x - \frac{x_l + x_r}{2} \right)^2 - h_3 \cdot \frac{x_l + x_r}{2} & x_b \neq x_l \\ h_1 + (h_2 - h_1) \cdot (x - x_l) + h_3 \cdot \left( x - \frac{x_l + x_r}{2} \right)^2 - h_3 \cdot \frac{x_l + x_r}{2} & x_b = x_l \end{cases} \quad (3)$$

The parameters  $h_1$  and  $h_2$  describe the height at the left and right end, respectively. The investigation of curvature is achieved by setting the coefficient  $h_3$  and the  $\text{ReLU}$

function allows the creation of bends at the specific position determined by  $x_b$ . In this work, rigid surfaces are investigated; therefore, the geometry, described by the parameters, is fixed and provided to the PINN as a part of its input. After the pressure is computed, the loss is computed as shown in Figure 2. The physics-informed Reynolds loss, residual loss, is determined by applying automatic differentiation and described by the following equation:

$$l_{Rey} = MSE\left(\frac{v}{2} \frac{\partial}{\partial x} \rho h - \frac{1}{12\eta} \frac{\partial}{\partial x} \left(\rho h^3 \frac{\partial p}{\partial x}\right), 0\right) \tag{4}$$

The MSE of the Reynolds equation is compared to zero and mirrors the residual term of common EHL simulations. The boundary condition is considered by the boundary loss, as follows:

$$l_{BC} = MSE(p_{l,r}, p_{b,l,r}) \tag{5}$$

The total loss is calculated as the sum of the residual loss and the boundary loss, both of which are incorporated into the HD-PINN framework described herein.

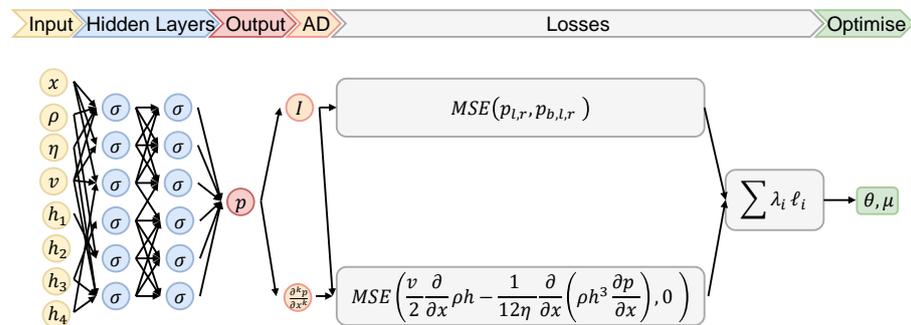


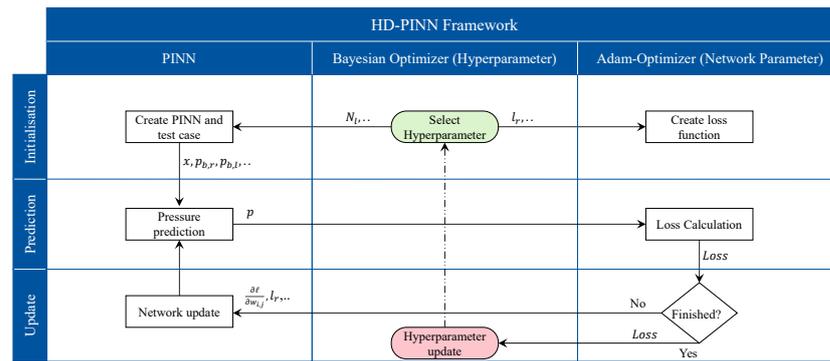
Figure 2. Illustration of the HD-PINN.

### 2.3. HD-PINN Framework

In previous research, an HD-PINN framework was developed and validated for single-case tasks. This framework, depicted in Figure 3, comprises three primary components. The core of the framework is the PINN itself, which computes the pressure distribution for each collocation point. For the actual network parameter optimization, the Adaptive Moment Estimation (Adam) algorithm, a gradient-based, first-order optimization method designed for stochastic objective functions, is embedded [45]. Recognized as a state-of-the-art optimizer for classical neural network [46,47], Adam’s application to PINNs has shown success, attributed to its minimal memory requirements and effectiveness in handling large-scale problems.

A significant challenge in PINN development is the need to balance various loss terms effectively; while conventional neural networks typically focus on a single loss function, PINNs engage in multi-objective optimization, necessitating loss balancing strategies. To address this, the Relative Loss Balancing with Random Lookback (ReLoBRaLo) method proposed by Bischof et al. was integrated into the HD-PINN framework. ReLoBRaLo leverages four algorithms: boundary scaling [48], tracking learning progress by considering losses from current and previous iterations, implementing exponential decay of the learning rate over iterations [49], and employing a random lookback to account for the entirety of the learning process [43]. The balancing algorithms achieved a relevant performance enhancement during the training.

Unlike traditional neural networks, PINNs possess a greater number of hyperparameters, particularly due to their loss balancing algorithm. To achieve efficient and automated tuning of these hyperparameters, the Bayesian optimizer was integrated into the HD-PINN framework. This optimizer utilizes a probabilistic surrogate model to estimate the loss function, thereby facilitating the tuning of many hyperparameters [50]. Its effectiveness in application to PINNs has been demonstrated and validated [51].



**Figure 3.** The HD-PINN framework and its training process.

Figure 3 provides an overview of the training process of the framework. Initially, the Bayesian optimizer determines the hyperparameters for initializing both the PINN and Adam. Subsequently, the PINN determines the pressure distribution, and the corresponding loss is calculated. If the training is not finished, due to a low loss or a high number of iterations, the loss is used to update the PINN's weights and biases, resulting in a new computed pressure distribution. After this cycle is finished, the hyperparameters are reevaluated by the Bayesian optimizer, and the whole framework starts over.

#### 2.4. Test Cases

The focus and novelty of this research lies on the capability of PINNs to solve multi-case and extrapolation tasks. In prior research, the ability to solve single-case scenarios has been proven. Single-case represents a scenario with a fixed set of inputs, while a multi-case scenario refers to a collection of inputs, including the pressure boundary conditions and the geometry in this study. The focus on extrapolation pertains to these two parameters, as well as to the position. Here, the PINN was trained within a constrained range of the actual sealing and housing geometries and was then assessed across the entirety of the positional space. The following five scenarios were investigated:

- The height multi-case analysis involved a linear convergent profile, where the parameter  $h_1$  varied within the range from 0.3 to 1.0, holding  $h_2$  constant at 0.2, and the pressure boundaries  $p_{b,l}$  and  $p_{b,r}$  were set to 0.5 and 1.0, respectively.
- For the pressure boundary multi-case scenarios, a linear convergent profile was considered with both  $p_{b,l}$  and  $p_{b,r}$  varying across the range from 0 to 1.
- Height extrapolation tasks were extended beyond the original multi-case domain for  $h_1$ , exploring values from 0.2 down to 0.1 and up from 1.1 to 1.3.
- Pressure boundary extrapolation tested the PINN with scenarios outside the multi-case training domain, specifically for pressure boundary combinations of  $(p_{b,l}, p_{b,r}) = (0.5, 1.2)$ ,  $(p_{b,l}, p_{b,r}) = (1.1, 1.2)$ , and  $(p_{b,l}, p_{b,r}) = (1.5, 2.0)$ .
- Position extrapolation was investigated for a linear convergent height profile with fixed pressure boundaries  $(p_{b,l}, p_{b,r}) = (0.5, 1.0)$ , where the right boundary position  $x_r$  varied from 0.9 down to 0.1.

Each scenario contained the two losses described in Equations (4) and (5), with the velocity of the counter surface  $v$  set to 1 and the fluid viscosity  $\eta$  also equal to 1. As previously mentioned, the focus was on calculating the hydrodynamic pressure distribution; thus, deformation, contact mechanics, and friction were neglected. The validation data was derived from the rigid ifas-DDS, which similarly disregards all aforementioned aspects and numerically computes the pressure distribution. In the next section, the results of the training are presented. After obtaining a feasible set of hyperparameters by the Bayesian optimizer the PINNs were trained and compared with the rigid ifas-DDS.

### 3. Results

#### 3.1. Height Multi-Case

The initial analysis focused on the height multi-case scenario, examining the PINN’s capability to predict the pressure distribution across different geometries. Figure 4 presents the outcomes for  $h_1 = 0.3$  and  $0.4$ , with  $0.3$  representing the lower limit of the training domain. It is observed that the accuracy of the PINN improves in the second scenario. This enhancement in PINN performance, with an increase in the parameter  $h_1$ , is further observed in the subsequent Figures 5–7. The HD-PINN framework demonstrates the capability to address the height multi-case task effectively across a considerable segment of the examined geometric range, exhibiting improved performance at higher values of  $h_1$ .

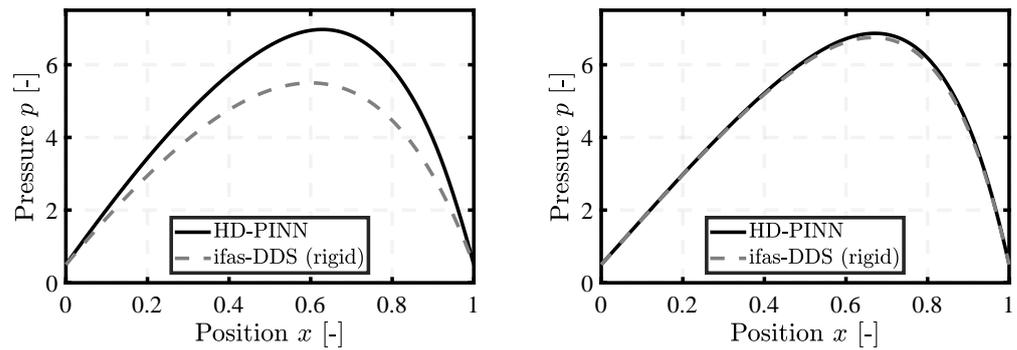


Figure 4. Pressure distribution for the multi-case PINN for  $h_1 = 0.3$  and  $0.4$ .

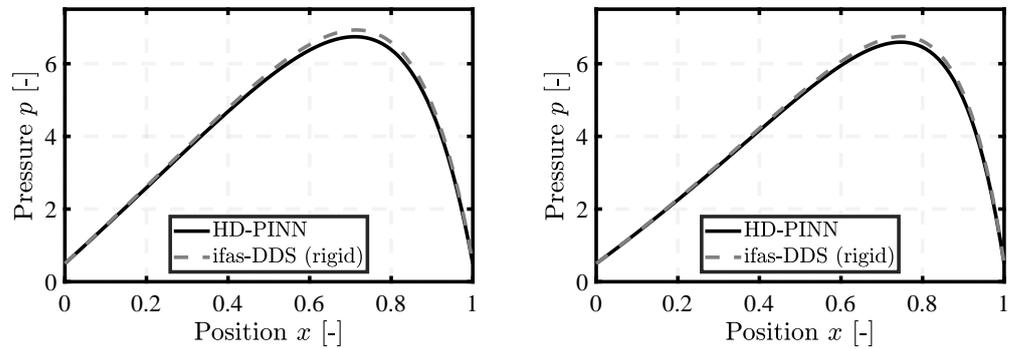


Figure 5. Pressure distribution for the multi-case PINN for  $h_1 = 0.5$  and  $0.6$ .

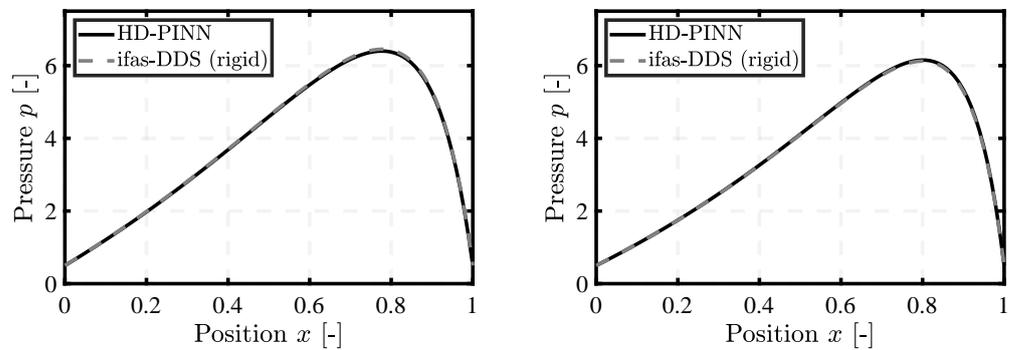
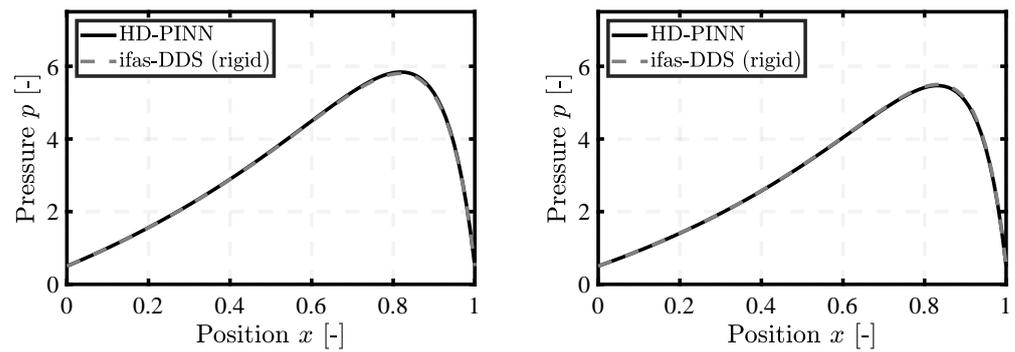


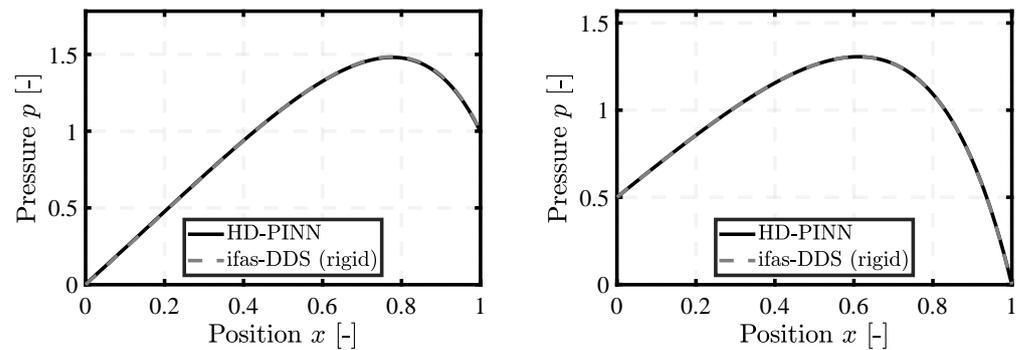
Figure 6. Pressure distribution for the multi-case PINN for  $h_1 = 0.7$  and  $0.8$ .



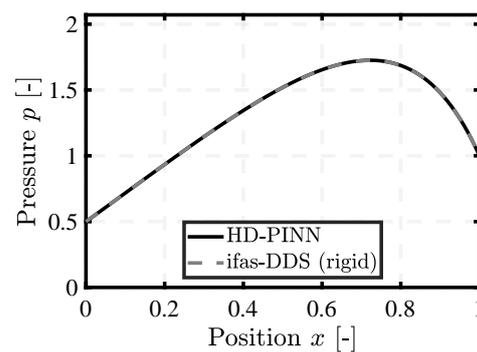
**Figure 7.** Pressure distribution for the multi-case PINN for  $h_1 = 0.9$  and  $1.0$ .

### 3.2. Pressure Boundary Multi-Case

The framework was further evaluated in a multi-case scenario, where the pressure boundaries varied. The results are illustrated in Figures 8 and 9, showcasing the PINN's performance in accurately predicting the pressure distribution across various boundaries within the designated training range.



**Figure 8.** Pressure distribution for the multi-case PINN for  $(p_{b,l}, p_{b,r}) = (0.0, 1.0)$  and  $(p_{b,l}, p_{b,r}) = (0.5, 0.0)$ .

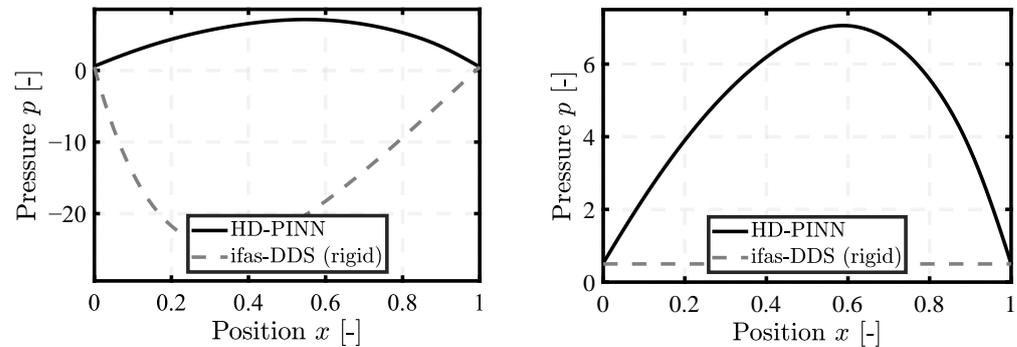


**Figure 9.** Pressure distribution for the multi-case PINN for  $p_l = 0.5$  and  $p_r = 1.0$ .

### 3.3. Height Extrapolation

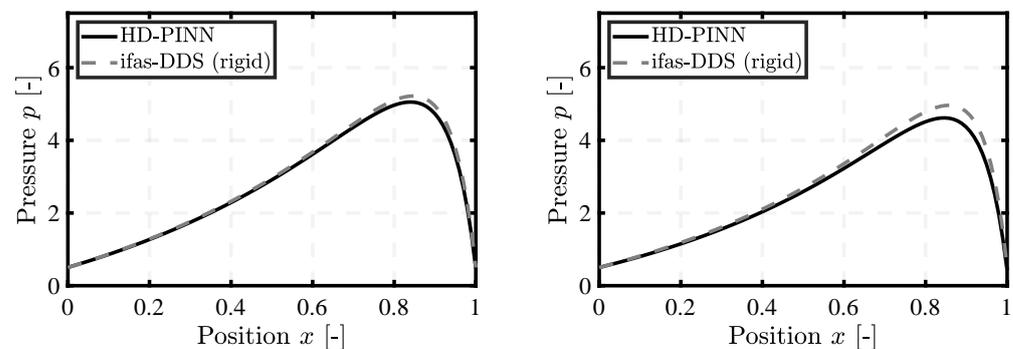
After evaluating the interpolation capabilities of the height multi-case PINN, its performance in extrapolation tasks was investigated. Figure 10 depicts the results when the parameter  $h_1$  is set to  $0.2$  and  $0.1$ , leading to a divergent height profile. Under these conditions, the multi-case PINN fails to predict even the correct sign of the pressure distribution. This outcome suggests that the PINN has not adequately captured the concepts underlying convergent and divergent height profiles. Notably, for divergent profiles, the pressure distribution becomes negative, which, in practical scenarios, would typically

signify the onset of cavitation. However, prior research has shown the PINN's ability to accurately predict negative pressure distributions for divergent gaps [10], indicating that the parameter  $h_1$ 's representation might not be sufficient to facilitate optimal PINN training. Compared to the work of Rom [7], the parameter  $h_1$  is more nested than the relative eccentricity provided to the multi-case PINN, therefore the influence of the  $h_1$  on the geometry is more complicated than the linear operation in the case of Rom.

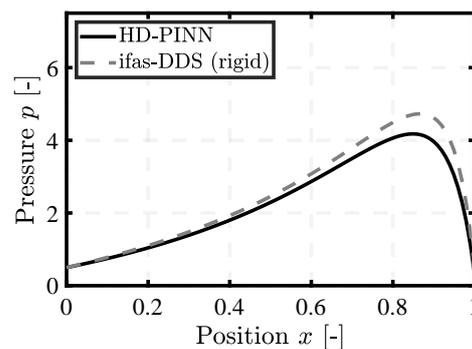


**Figure 10.** Pressure distribution for the multi-case PINN for  $h_1 = 0.1$  and  $0.2$ .

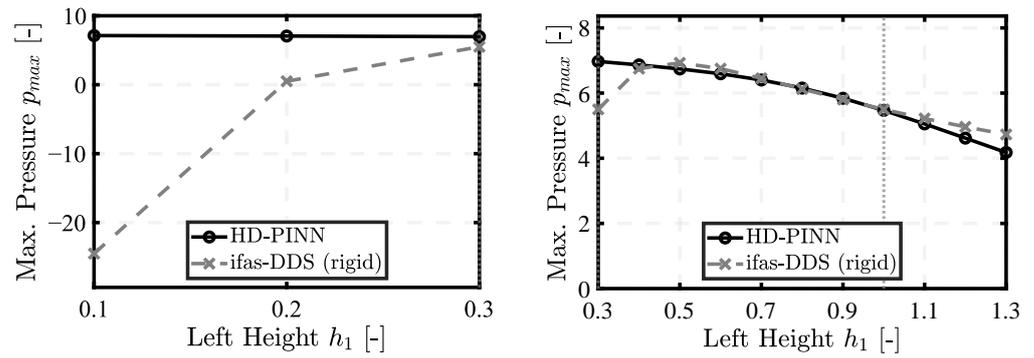
Figures 11 and 12 display the results for the extrapolation of  $h_1$  up to 1.3. The height profile remains convergent and the prediction accuracy is maintained within an acceptable range. However, it is observed that the performance diminishes as the value of the height parameter  $h_1$  is further increased. Figure 13 displays the maximum pressure determined by the ifas-DDS and the HD-PINN for the different values of  $h_1$ . The difference increases significantly for the lower range of  $h_1$ . For higher values, starting from 0.4, good agreement is observed in the upper range of the multi-case training domain. Increasing the value of  $h_1$  beyond that results in an increased deviation of ifas-DDS and HD-PINN.



**Figure 11.** Pressure distribution for the multi-case PINN for  $h_1 = 1.1$  and  $1.2$ .



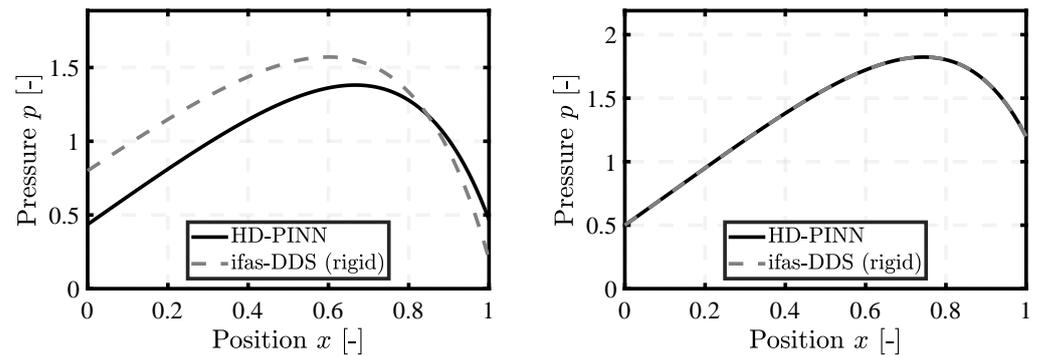
**Figure 12.** Pressure distribution for the multi-case PINN for  $h_1 = 1.3$ .



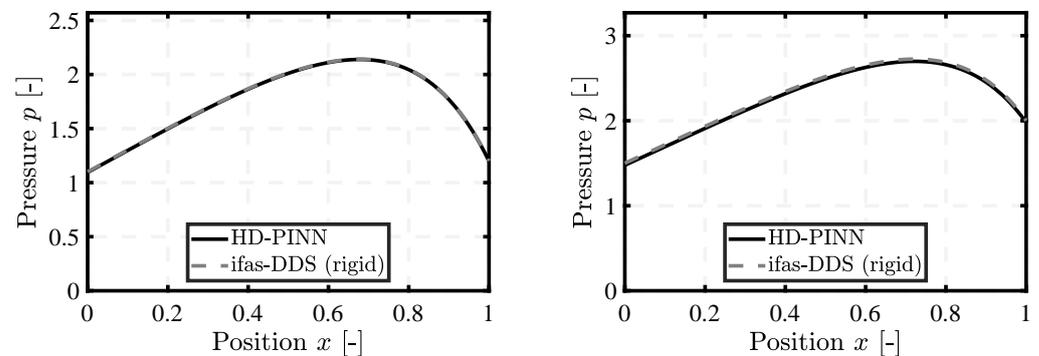
**Figure 13.** Maximum pressure of ifas-DDS and HD-PINN for  $h_1$  between 0.1 and 0.3 (left), and 0.3 and 1.3 (right). The vertical line represents the upper limit of the trained  $h_1$ .

### 3.4. Pressure Boundary Extrapolation

Similar to height extrapolation tasks, extrapolations for pressure boundaries were also investigated. Figure 14 shows the extrapolation results for two PINNs with identical hyperparameters but trained under different conditions: one for a single-case scenario (with  $p_l = 0.5$  and  $p_r = 0.2$ ) and the other for a multi-case scenario. It can be observed that the extrapolation performance benefits from multi-case training. This improvement is further validated in Figure 15, where the pressure boundaries are successfully extrapolated up to 2.0.



**Figure 14.** Pressure distribution for the single-case PINN for  $(p_{b,l}, p_{b,r}) = (0.8, 0.2)$  (trained on  $(p_{b,l}, p_{b,r}) = (0.5, 0.2)$ ) (left) and multi-case PINN (right) for  $(p_{b,l}, p_{b,r}) = (0.5, 1.2)$ .

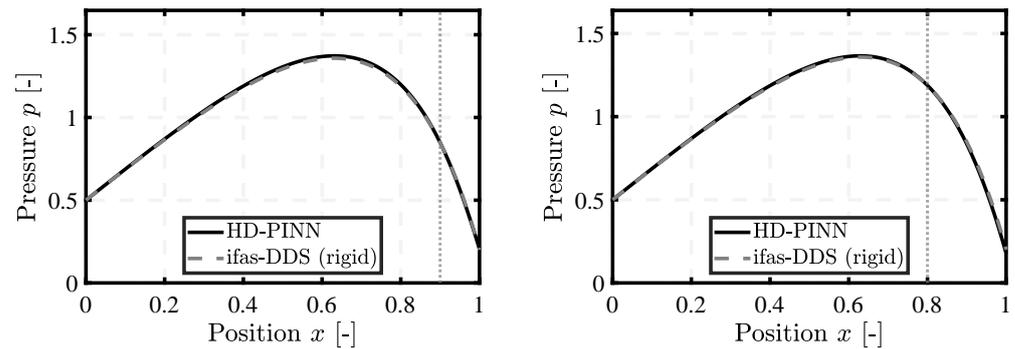


**Figure 15.** Pressure distribution for the multi-case PINN for  $(p_{b,l}, p_{b,r}) = (1.1, 1.2)$  (left) and  $(p_{b,l}, p_{b,r}) = (1.5, 2.0)$  (right).

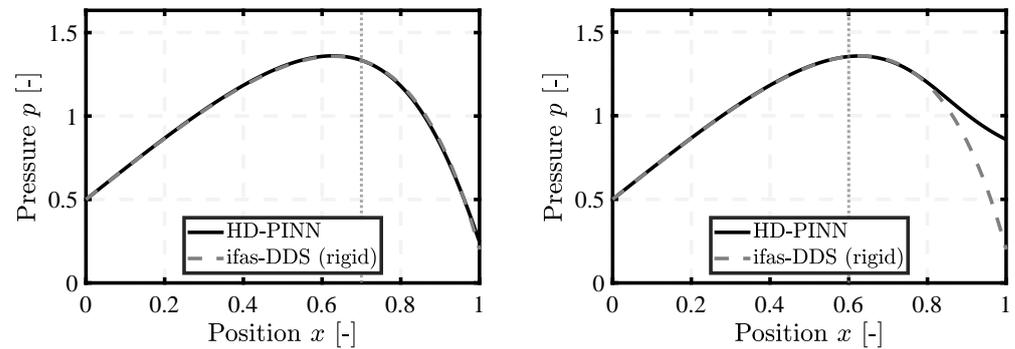
### 3.5. Position Extrapolation

The final scenario investigated the extrapolation of position, thereby restricting the PINN's training domain in terms of position data. The right pressure boundary was as-

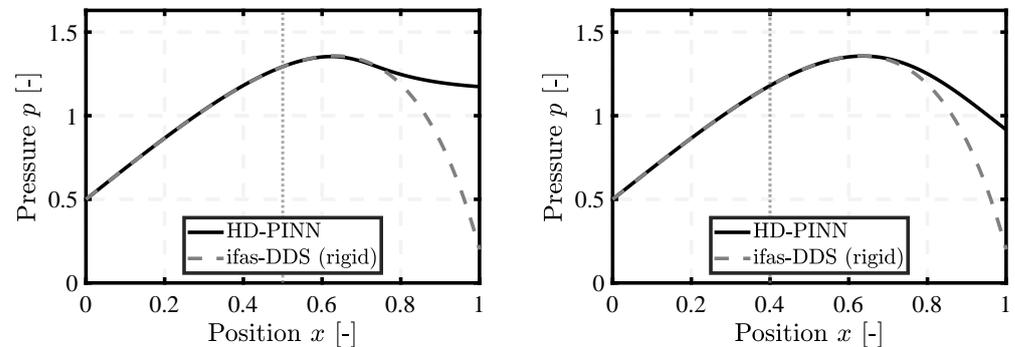
signed a specific value, obtained from the rigid ifas-DDS model. The range of extrapolation spanned from 0.9 to 0.1. The PINN calculated the pressure distribution across the entire position domain, from 0.0 to 1.0. As depicted in Figures 16–20, with a vertical line representing the training boundary, there is a decline in accuracy as the position limit becomes more constrained. However, no significant deviation from the rigid ifas-DDS model is detected up to a positional limit of  $x_r = 0.7$ . Beyond this point, starting from  $x_r = 0.6$ , the predicted pressure distributions begin to deviate from those of the complete extrapolation region according to the rigid ifas-DDS. It is worth mentioning that the extrapolation maintains good performance over a considerable range. For instance, Figure 20 demonstrates extrapolation starting from  $x_r = 0.1$ , where a good agreement with the rigid ifas-DDS model is observed up to  $x = 0.5$ .



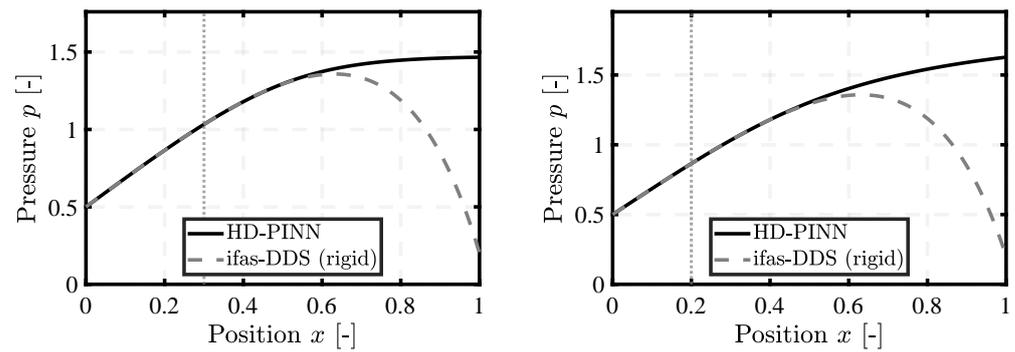
**Figure 16.** Pressure distribution for the extrapolation task for  $x_r = 0.9$  (left) and  $0.8$  (right).



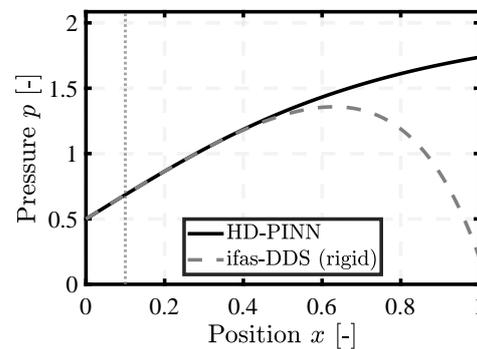
**Figure 17.** Pressure distribution for the extrapolation task for  $x_r = 0.7$  (left) and  $0.6$  (right).



**Figure 18.** Pressure distribution for the extrapolation task for  $x_r = 0.5$  (left) and  $0.4$  (right).



**Figure 19.** Pressure distribution for the extrapolation task for  $x_r = 0.3$  (left) and  $0.2$  (right).



**Figure 20.** Pressure distribution for the extrapolation task for  $x_r = 0.1$ .

#### 4. Discussion

The investigation of PINNs in solving multi-case scenarios and performing extrapolation tasks revealed their potential in the context of accelerating hydrodynamic lubrication computation. In multi-case scenarios and extrapolation of pressure boundaries, PINNs exhibited commendable accuracy. Their ability to precisely predict pressure distributions across a variety of conditions underscores their adaptability and robustness. Position extrapolation tasks also demonstrated the efficiency of PINNs. Even with the omission of data to train on, PINNs maintained an acceptable level of agreement with the rigid ifas-DDS. These results indicate a promising possibility for applications where datasets are not completely attainable, underlining the PINNs' capability to infer missing information to a reasonable degree of precision.

However, challenges emerged in height multi-case scenarios, especially as the parameters approached the boundaries of the trained height range. Within the confines of the designated range, PINNs performed adeptly, accurately capturing the pressure distribution. Yet, as the parameter values approached the extremities of this range, a noticeable decline in performance was observed. The extrapolation tasks, in particular, suffered significantly. This issue can be attributed to the complexity of the height parameter, which, unlike the simplified forms used in previous studies such as those used by Rom [7], presents a more complex challenge for the network to learn and generalize. Rom extended the PINN's input by the relative eccentricity, which was not nested in the formula of the geometry. This discrepancy in performance highlights the critical importance of the formulation and representation of input parameters in training PINNs. The complex nature of parameters like height requires careful consideration and potentially more sophisticated preprocessing or reformulation of the height formula.

## 5. Conclusions

This contribution demonstrates the capacity of PINNs to undertake multi-case and extrapolation tasks in determining the pressure distribution within sealing contacts in a housing, as governed by the Reynolds equation. It began with an introduction to PINNs and surveyed current research on their application in hydrodynamic lubrication, followed by a detailed presentation of the physics-informed loss and a description of the whole HD-PINN framework.

In terms of position extrapolation, the PINN aligned well with the rigid ifas-DDS, despite the constraints of limited information. The first notable deviation occurred for right boundaries smaller than 0.6. This deviation did not occur immediately beyond the boundary. For instance, in the case of  $x_r = 0.1$ , a noticeable deviation became evident at around  $x > 0.5$ . This observation underscores the capabilities of PINNs in performing extrapolation tasks to a certain extent. The good results, regarding the pressure boundary extrapolation tasks, further support this statement. For extrapolation pressure boundary tasks, the single-case PINN did not compute the pressure distribution accurately. In contrast, the multi-case PINN demonstrated proficiency in both extrapolation and interpolation tasks related to pressure boundaries. Nonetheless, the PINN encountered challenges when addressing multi-case and height-related extrapolation tasks. Specifically, for values of  $h_1$  approaching or falling below the lower range, resulting in a divergent gap, the model's accuracy declined. This may be attributed to how the height parameter's information was presented to the PINN. However, for higher  $h_1$ , even in the extrapolation regime, the PINN showed satisfying results compared to the rigid ifas-DDS.

The findings of this study represent a notable advancement in the domain of lubricated contact simulations, presenting a novel approach to accelerate hydrodynamic pressure computation. This is due to the application of PINNs, which directly compute pressure, compared to distributed parameter simulation models, which depend on numerical methods. Additionally, the demonstrated capability of a single PINN to solve multi-case tasks highlights the potential of PINNs to be applied to several scenarios. This could allow for accelerated calculations of hydrodynamic pressure for varying parameters following one initial training. Furthermore, the ability to extrapolate the pressure distribution beyond the confines of the trained regime is a promising advancement in the investigation of tribological simulations, providing the potential to acquire insights into system areas where direct measurements are unfeasible. Traditional simulation models, in comparison, typically do not offer the capability for such extrapolation, thereby necessitating the numerical computation across the entirety of the seal's geometry.

Future research will focus on integrating transient and cavitation terms of the Reynolds equation into the framework. Particularly, exploring extrapolation across the time domain emerges as a promising avenue for advancing the field of tribology. Furthermore, the height parameter inputs will be investigated so that the PINN can solve multi-case and extrapolation tasks for different geometries.

**Author Contributions:** Conceptualization, F.B.-P., N.B. and K.S.; methodology, F.B.-P. and N.B.; software, F.B.-P., N.P., M.T. and S.W.; validation, F.B.-P. and N.P.; formal analysis, F.B.-P., N.B., N.P., M.T. and S.W.; investigation, F.B.-P., N.B., N.P., M.T. and S.W.; resources, F.B.-P., N.B., N.P., M.T., S.W. and K.S.; data curation, F.B.-P. and N.P.; writing—original draft preparation, F.B.-P.; writing—review and editing, F.B.-P., N.B. and K.S.; visualization, F.B.-P. and N.P.; supervision, F.B.-P., N.B. and K.S.; project administration, F.B.-P.; funding acquisition, F.B.-P., N.B. and K.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors thank the Research Association for Fluid Power of the German Engineering Federation VDMA (Verband Deutscher Maschinen- und Anlagenbau, grant number 7058400) for its financial support. Special gratitude is expressed to the participating companies and their representatives in the accompanying industrial committee for their advisory and technical support.

**Data Availability Statement:** The datasets presented in this article are not readily available because the data are part of an ongoing study.

**Conflicts of Interest:** The authors declare no conflicts of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

Adam	Adaptive moment estimation
EHL	Elastohydrodynamic lubrication
HD	Hydrodynamic
JFO	Jakobsson–Floberg–Olsson
MSE	Mean squared error
PDE	Partial differential equation
PINN	Physics-informed neural network
PIML	Physics-informed machine learning
ReLU	Rectified linear unit
ReLoBRaLo	Relative Loss Balancing with Random Lookback
SS	Swift–Stieber

### Nomenclature

Symbol	Definition	Unit
$h$	Gap height	[-]
$h_1$	Height at left end	[-]
$h_2$	Height at right end	[-]
$h_3$	Curvature of sealing	[-]
$h_4$	Position for sealing bend	[-]
$p$	Hydrodynamic pressure	[-]
$p_{b,l,r}$	Pressure boundary condition for left and right boundary	[-]
$p_{l,r}$	Pressure at the left and right boundary	[-]
$R_q$	Root mean squared contact surface roughness	[-]
$t$	Time	[-]
$v$	Velocity of counter surface	[-]
$x$	Axial coordinate	[-]
$x_b$	Position of sealing bend	[-]
$x_l$	Left end of geometry	[-]
$x_r$	Right end of geometry	[-]
$\eta$	Fluid viscosity	[-]
$\theta$	Cavity friction	[-]
$\rho$	Fluid density	[-]
$\Phi^p$	Pressure flow factors	[-]
$\Phi^\tau$	Shear flow factors	[-]
$\frac{\partial p}{\partial x, t}$	Partial derivative of pressure with regards to time and position	[-]

### References

- Bauer, N.; Sumbat, B.; Feldmeth, S.; Bauer, F.; Schmitz, K. Experimental determination and EHL simulation of transient friction of pneumatic seals in spool valves. In Proceedings of the Sealing Technology—Old School and Cutting Edge: International Sealing Conference: 21st ISC, Stuttgart, Germany, 7–8 October 2022; pp. 503–522.
- Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What is Next. *J. Sci. Comput.* **2022**, *92*, 88. [\[CrossRef\]](#)
- Almqvist, A. Fundamentals of Physics-Informed Neural Networks Applied to Solve the Reynolds Boundary Value Problem. *Lubricants* **2021**, *9*, 82. [\[CrossRef\]](#)
- Zhao, Y.; Guo, L.; Wong, P.P.L. Application of physics-informed neural network in the analysis of hydrodynamic lubrication. *Friction* **2023**, *11*, 1253–1264. [\[CrossRef\]](#)
- Li, L.; Li, Y.; Du, Q.; Liu, T.; Xie, Y. ReF-nets: Physics-informed neural network for Reynolds equation of gas bearing. *Comput. Methods Appl. Mech. Eng.* **2022**, *391*, 114524. [\[CrossRef\]](#)
- Yadav, S.K.; Thakre, G. Solution of Lubrication Problems with Deep Neural Network. In *Advances in Manufacturing Engineering; Dikshit, M.K., Soni, A., Davim, J.P., Eds.; Lecture Notes in Mechanical Engineering; Springer: Singapore, 2023; pp. 471–477. [CrossRef]*
- Rom, M. Physics-informed neural networks for the Reynolds equation with cavitation modeling. *Tribol. Int.* **2023**, *179*, 108141. [\[CrossRef\]](#)

8. Cheng, Y.; He, Q.; Huang, W.; Liu, Y.; Li, Y.; Li, D. HL-nets: Physics-informed neural networks for hydrodynamic lubrication with cavitation. *Tribol. Int.* **2023**, *188*, 108871. [[CrossRef](#)]
9. Xi, Y.; Deng, J.; Li, Y. A new method to solve the Reynolds equation including mass-conserving cavitation by physics informed neural networks (PINNs) with both soft and hard constraints. *Friction* **2024**, *12*, 1165–1175. [[CrossRef](#)]
10. Brumand-Poor, F.; Bauer, N.; Plückerhahn, N.; Schmitz, K. Fast Computation of Lubricated Contacts: A Physics-Informed Deep Learning Approach. In Proceedings of the 14th International Fluid Power Conference: Fluid Power—Sustainable Productivity, Dresden, Germany, 19–21 March 2024.
11. Bauer, N.; Baumann, M.; Feldmeth, S.; Bauer, F.; Schmitz, K. Elastohydrodynamic Simulation of Pneumatic Sealing Friction Considering 3D Surface Topography. *Chem. Eng. Technol.* **2023**, *46*, 167–174. [[CrossRef](#)]
12. Bauer, N.; Rambaks, A.; Müller, C.; Murrenhoff, H.; Schmitz, K. Strategies for Implementing the Jakobsson-Floberg-Olsson Cavitation Model in EHL Simulations of Translational Seals. *Int. J. Fluid Power* **2021**, *22*, 199–232. [[CrossRef](#)]
13. Angerhausen, J.; Woyciniuk, M.; Murrenhoff, H.; Schmitz, K. Simulation and experimental validation of translational hydraulic seal wear. *Tribol. Int.* **2019**, *134*, 296–307. [[CrossRef](#)]
14. Marian, M.; Tremmel, S. Current Trends and Applications of Machine Learning in Tribology—A Review. *Lubricants* **2021**, *9*, 86. [[CrossRef](#)]
15. Paturi, U.M.R.; Palakurthy, S.T.; Reddy, N.S. The Role of Machine Learning in Tribology: A Systematic Review. *Arch. Comput. Methods Eng.* **2023**, *30*, 1345–1397. [[CrossRef](#)]
16. Kanai, R.A.; Desavale, R.G.; Chavan, S.P. Experimental-Based Fault Diagnosis of Rolling Bearings Using Artificial Neural Network. *J. Tribol.* **2016**, *138*, 4032525. [[CrossRef](#)]
17. Sadık Ünlü, B.; Durmuş, H.; Meriç, C. Determination of tribological properties at CuSn10 alloy journal bearings by experimental and means of artificial neural networks method. *Ind. Lubr. Tribol.* **2012**, *64*, 258–264. [[CrossRef](#)]
18. Canbulut, F.; Yildirim, Ş.; Sinanoğlu, C. Design of an Artificial Neural Network for Analysis of Frictional Power Loss of Hydrostatic Slipper Bearings. *Tribol. Lett.* **2004**, *17*, 887–899. [[CrossRef](#)]
19. Hess, N.; Shang, L. Development of a Machine Learning Model for Elastohydrodynamic Pressure Prediction in Journal Bearings. *J. Tribol.* **2022**, *144*, 4053815. [[CrossRef](#)]
20. Velioglu, M.; Mitsos, A.; Dahmen, M. Physics-Informed Neural Networks (PINNs) for Modeling Dynamic Processes Based on Limited Physical Knowledge and Data. In Proceedings of the 2023 AIChE Annual Meeting, Orlando, FL, USA, 5–10 November 2023.
21. Psychogios, D.C.; Ungar, L.H. A hybrid neural network—first principles approach to process modeling. *AIChE J.* **1992**, *38*, 1499–1511. [[CrossRef](#)]
22. Su, H.T.; Bhat, N.; Minderman, P.A.; McAvoy, T.J. Integrating Neural Networks with First Principles Models for Dynamic Modeling. *IFAC Proc. Vol.* **1992**, *25*, 327–332. [[CrossRef](#)]
23. Kahrs, O.; Marquardt, W. The validity domain of hybrid models and its application in process optimization. *Chem. Eng. Process. Process Intensif.* **2007**, *46*, 1054–1066. [[CrossRef](#)]
24. Marian, M.; Tremmel, S. Physics-Informed Machine Learning—An Emerging Trend in Tribology. *Lubricants* **2023**, *11*, 463. [[CrossRef](#)]
25. Nabian, M.A.; Meidani, H. Physics-Driven Regularization of Deep Neural Networks for Enhanced Engineering Design and Analysis. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 436. [[CrossRef](#)]
26. Lee, H.; Kang, I.S. Neural algorithm for solving differential equations. *J. Comput. Phys.* **1990**, *91*, 110–131. [[CrossRef](#)]
27. Lagaris, I.E.; Likas, A.; Fotiadis, D.I. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Trans. Neural Netw.* **1998**, *9*, 987–1000. [[CrossRef](#)] [[PubMed](#)]
28. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [[CrossRef](#)]
29. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
30. Owhadi, H. Bayesian Numerical Homogenization. *Multiscale Model. Simul.* **2015**, *13*, 812–828. [[CrossRef](#)]
31. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Inferring solutions of differential equations using noisy multi-fidelity data. *J. Comput. Phys.* **2017**, *335*, 736–746. [[CrossRef](#)]
32. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Machine learning of linear differential equations using Gaussian processes. *J. Comput. Phys.* **2017**, *348*, 683–693. [[CrossRef](#)]
33. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Numerical Gaussian Processes for Time-dependent and Non-linear Partial Differential Equations. *Siam J. Sci. Comput.* **2018**, *40*, 17M1120762. [[CrossRef](#)]
34. Raissi, M.; Karniadakis, G.E. Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **2018**, *357*, 125–141. [[CrossRef](#)]
35. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
36. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. *arXiv* **2017**, arXiv:1711.10566v1.
37. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]

38. Antonelo, E.A.; Camponogara, E.; Seman, L.O.; Souza, E.R.d.; Jordanou, J.P.; Hubner, J.F. Physics-Informed Neural Nets for Control of Dynamical Systems. *Neurocomputing* **2024**, *579*, 127419. [[CrossRef](#)]
39. Kim, J.; Lee, K.; Lee, D.; Jin, S.Y.; Park, N. DPM: A Novel Training Method for Physics-Informed Neural Networks in Extrapolation. *arXiv* **2021**, arXiv:2012.02681v1.
40. Fesser, L.; D'Amico-Wong, L.; Qiu, R. Understanding and Mitigating Extrapolation Failures in Physics-Informed Neural Networks. *arXiv* **2023**, arXiv:2306.09478v2.
41. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *Atılım Gunes Baydin*. **2017**, *18*, 5595–5637.
42. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [[CrossRef](#)]
43. Bischof, R.; Kraus, M. Multi-Objective Loss Balancing for Physics-Informed Deep Learning. *arXiv* **2021**, arXiv:2110.09813v2.
44. Rimon, M.T.I.; Hassan, M.F.; Lyathakula, K.R.; Cesmeçi, S.; Xu, H.; Tang, J. A Design Study of an Elasto-Hydrodynamic Seal for sCO<sub>2</sub> Power Cycle by Using Physics Informed Neural Network. In Proceedings of the ASME Power Applied R&D 2023, American Society of Mechanical Engineers, Long Beach, CA, USA, 6–8 August 2023. [[CrossRef](#)]
45. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Schmidt, R.M.; Schneider, F.; Hennig, P. Descending through a Crowded Valley—Benchmarking Deep Learning Optimizers. *arXiv* **2020**, arXiv:2007.01547v6.
47. Reyad, M.; Sarhan, A.M.; Arafa, M. A modified Adam algorithm for deep neural network optimization. *Neural Comput. Appl.* **2023**, *35*, 17095–17112. [[CrossRef](#)]
48. Heydari, A.A.; Thompson, C.A.; Mehmood, A. SoftAdapt: Techniques for Adaptive Loss Weighting of Neural Networks with Multi-Part Loss Functions. *arXiv* **2019**, arXiv:1912.12355.
49. Wang, S.; Teng, Y.; Perdikaris, P. Understanding and mitigating gradient pathologies in physics-informed neural networks. *arXiv* **2020**, arXiv:2001.04536.
50. Močkus, J. On bayesian methods for seeking the extremum. In Proceedings of the Optimization Techniques IFIP Technical Conference Novosibirsk, Novosibirsk, Russia, 1–7 July 1974; Goos, G., Hartmanis, J., Brinch Hansen, P., Gries, D., Moler, C., Seegmüller, G., Wirth, N., Marchuk, G.I., Eds.; Springer: Berlin/Heidelberg, Germany, 1975; Volume 27, pp. 400–404. [[CrossRef](#)]
51. Escapil-Inchauspé, P.; Ruz, G.A. Hyper-parameter tuning of physics-informed neural networks: Application to Helmholtz problems. *Neurocomputing* **2023**, *561*, 126826. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.