Error estimates and physics informed augmentation of neural networks for thermally coupled incompressible Navier Stokes equations

Shoaib Goraya · Nahil Sobh · Arif Masud

Received: 28 October 2022 / Accepted: 27 March 2023 / Published online: 14 June 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract
Physics Informed Neural Networks (PINNs) are shown to be a promising method for the approximation of partial differential equations (PDEs). PINNs approximate the PDE solution by minimizing physics-based loss functions over a given domain. Despite substantial progress in the application of PINNs to a range of problem classes, investigation of error estimation and convergence properties of PINNs, which is important for establishing the rationale behind their good empirical performance, has been lacking. This paper presents convergence analysis and error estimates of PINNs for a multi-physics problem of thermally coupled incompressible Navier–Stokes equations. Through a model problem of Beltrami flow it is shown that a small training error implies a small generalization error. Posteriori convergence rates of total error with respect to the training residual and collocation points are presented. This is of practical significance in determining appropriate number of training parameters and training residual thresholds to get good PINNs prediction of thermally coupled steady state laminar flows. These convergence rates are then generalized to different spatial geometries as well as to different flow parameters that lie in the laminar regime. A pressure stabilization term in the form of pressure Poisson equation is added to the PDE residuals for PINNs. This physics informed augmentation is shown to improve accuracy of the pressure field by an order of magnitude as compared to the case without augmentation. Results from PINNs are compared to the ones obtained from stabilized finite element method and good properties of PINNs are highlighted.

Keywords
Machine learning · PINNs · Thermally coupled flows · Physics informed augmentation · Navier–Stokes equations · Neural networks · Error estimates · Convergence

1 Introduction
Thermally coupled Navier–Stokes equations serve as a model for a range of flows in engineering and natural sciences. The complexity of flow fields invariably requires high fidelity computational methods that have their associated cost of computation. Traditional numerical methods come with the challenges of elaborate computer codes for involved mathematical formulations, prohibitive computational cost for resolving all the physical scales, and uncertainty in the numerical values of the parameters in a problem under consideration. In these situations, physics-based data driven techniques can play an important role in filling the knowledge gap between the physical phenomenon and the modeled physics.

Weighted Residual Methods (WRM), when introduced a century ago, provided approximate semi-analytical solution to Partial Differential Equations (PDEs) [1, 2]. Their popularity stemmed from the fact that approximate solutions (trial functions) were generated by forming a linear combination of known functions and unknown coefficients which could vary over the domain of interest. The number of components in a linear combination were dictated by the size of the resulting system of equations of the unknown coefficients. Since the introduction of WRM, exploring with various function spaces and weights has led to what is known today as the finite difference and the finite element methods as well as
meshless and mesh free methods [3]. Neural networks with
unknown functions (activations) and unknown coefficients
(weights and biases) have also been proposed to solve linear
and nonlinear partial differential equations via the WRM.
WRM-based neural network trial function approximations
of PDE solution first appeared in 1990s in the works of [4,
5]. The use of Physics Informed Neural Networks (PINNs)
to approximate partial differential equations was popularized
by the work of [6, 7] who leveraged PDE based loss functions
along with TensorFlow’s deep learning library to automate
the differentiation and subsequently solve for the unknown
weights and functions.

In Deep Neural Networks (DNN), it is assumed that an
underlying function maps inputs to outputs in the target
domain. The functional map is determined by minimizing
the loss function via optimizing network’s parameters during
the process called training. For instance, given a loss function
and a class of neural networks \( N^m \) with \( m \) parameters,
the goal is to find a mapping \( \hat{N}_S(\hat{W}, \hat{b}) : \mathbf{X} \rightarrow \mathbf{Y} \) between inputs
\( \mathbf{X} = \bigcup_{i=1}^S \{ x_i, y_i \} \) and outputs \( \mathbf{Y} = \bigcup_{i=1}^S \{ u_i, p_i \} \) such that
it minimizes the loss function over a training dataset of \( S \)
collocation points. The mapping \( \hat{N}_S(\hat{W}, \hat{b}) \in N^m \) rep-
resents trained DNN approximation function with optimized
weights \( \hat{W} \) and biases \( \hat{b} \). \( \hat{N}_S \) can also be understood as a
global approximation function used in the spectral methods.
The structure of such DNN consists of an input-layer fol-
lowed by one or more internal layers ending with an output
layer where each layer consists of neurons. For the input
layer the neurons are typically the spatio-temporal locations
of the domain points. For example, the input layer for a two-
dimensional spatial domain \((x, y)\) will consist of two inputs
or neurons \( \mathbf{X} = \{ x, y \} \). The internal layers (also known as
hidden layers) can have a variable number of neurons. These
neurons represent the unknown functions used in building
the WRM trial approximation while the output layer neurons
contain the trial function values. The values of any layer \( l \)
in the network can be represented in an algebraic form as
\( \mathbf{z}_l = \sigma(\mathbf{W}_l\mathbf{z}_{l-1} + \mathbf{b}_l) \) for \( 1 \leq l \leq L \). The values in the out-
put layer are computed as \( \mathbf{Y} = \sigma(\mathbf{W}_{L+1}\mathbf{z}_L + \mathbf{b}_{L+1}) \) where
\( L \) is the total number of hidden layers, \( \mathbf{W} \) is the tensor
of weights, \( \mathbf{b} \) is the bias vector, \( \mathbf{Y} \) is the output vector, and \( \sigma \)
is the activation function that acts as a transfer function to
propagate data in the network.

The suitability of the DNNs for approximating PDEs
arises from their universal approximation properties as well
as automatic differentiation [8]. The universal approxima-
tion theorem states that any compactly supported continuous
function can be approximated by a neural network generated
group to an arbitrary precision [9]. It was shown in [10] that
a single layer neural network of sufficiently large width can
uniformly approximate a function and its derivative. Furth-
more, the availability of open-source deep learning libraries
such as TensorFlow [11] and PyTorch [12] makes the imple-
mentation straightforward. In the context of solving general
PDEs using DNN, let us consider a second order PDE in
an open-bounded domain \( \Omega \in \mathbb{R}^{n_d} \), where \( n_d \) is the
number of spatial dimensions, with non-homogeneous Dirichlet
boundary conditions applied at its boundary \( \Gamma = \partial \Omega \).

\[
\begin{align*}
\mathcal{L}u(x) &= f \quad \text{in } \Omega \\
\ u(x) &= g \quad \text{on } \Gamma
\end{align*}
\]

where \( \mathcal{L} \) is the differential operator, \( u \) is the unknown solution
field, and \( f \) is the source term. The loss function is comprised
of the governing equation which is based on some physical
balance laws.

\[
\text{Loss} = \left\| \mathcal{R}^D \right\|_\Omega^2 + \left\| \mathcal{R}^B \right\|_\Gamma^2 = \left\| \mathcal{L}\hat{u}(x; \mathbf{W}, \mathbf{b}) - f \right\|_\Omega^2 + \left\| \hat{u}(x; \mathbf{W}, \mathbf{b}) - g \right\|_\Gamma^2 \quad (3)
\]

where \( \hat{u}(x; \mathbf{W}, \mathbf{b}) \) is the DNN approximation to the exact
solution \( \tilde{u}(x) \), \( \mathcal{R}^D \) is the domain residual, and \( \mathcal{R}^B \) is the
boundary residual. Equation 3 yields a constrained optimization
problem in which the residual or the loss is minimized
using optimization strategies such as the stochastic gradient
descent methods during the training process:

\[
\left\{ \hat{W}, \hat{b} \right\} = \arg\min_{\mathbf{W}, \mathbf{b}} (\mathcal{R})
\]

where \( \hat{W} \) and \( \hat{b} \) are optimized weights and biases, re-
spectively, and are used by the DNN to approximate the PDE
solution.

The error in supervised learning through neural networks
can be decomposed into three components: (i) optimization
error \( e_O \), (ii) approximation error \( e_A \), and (iii) training/ esti-
mation error \( e_T \). Figure 1 shows a schematic diagram of
total error decomposition. Optimization error arises from the
minimization problem 2 and is not well understood because
the objective function 1 is highly non-convex with multiple
local minima. Although several gradient descent algorithms
such as Adam and BFGS have been successfully employed
to obtain solutions with enough accuracy, establishing the
convergence of these algorithms to a unique global mini-
mum remains an open problem [13]. Approximation error
describes the discrepancy between the exact solution and the
neural network mapping function on a given network archi-
tecture. It is relatively well understood in the light of universal
approximation theorem [9] as discussed earlier. Training or
estimation error \( e_T \) arises when the network is trained on a
finite dataset \( S \) to get a mapping \( \hat{N}_S \) on the target domain.
The generalization error \( e_G \) is the combination of approxima-
tion and training error and defines the accuracy of the neural
network predicted solution. In PDE problems, the general-
ization error is the distance between a global minimizer of

\( e_G \).
The first objective in this paper is to carry out a numerical investigation on the convergence of PINNs for thermally coupled incompressible Navier–Stokes Equations and provide a rationale for their empirical performance. This involves establishing that for a small generalization error $\epsilon$ the total error is also small i.e., $e < \delta(\epsilon)$ for some $\delta(\epsilon) < O(\epsilon)$. Since PINNs minimize training error instead of generalization error, the above assumption holds under the premise that the generalization error decreases proportionately with the training error.

Another issue PINNs face for incompressible Navier–Stokes equations is the poor pressure prediction for velocity-pressure formulations. In fact the pressure prediction of the Beltrami flow problem in [15] has an $L_2$ error of 13% which is an order of magnitude higher than the velocity field. In the context of multi-physics problems, a less accurate pressure field in turn affects the accuracy of the velocity field. The second objective of this paper is to improve the pressure prediction by introducing a physics-based augmentation in the loss function of the coupled system of PDEs.

In this paper, we have presented convergence analysis of PINNs applied to 2D steady-state flow of incompressible Navier–Stokes PDEs coupled with a scalar energy PDE and subjected to Dirichlet boundary conditions. Through a model problem of Beltrami flows for which an analytical solution exists, effect of (i) network size, (ii) collocation points, and (iii) training error on total error of the predicted solution is investigated and posteriori error estimates are obtained. The systemic study establishes the convergence and consistency of PINNs for the multiphysics problems. Moreover, we introduce a physics-informed augmentation to PINNs in the form of a pressure Poisson equation and its effect on the prediction of the pressure field is shown.

The first contribution in this paper is the convergence analysis and error estimates of PINNs for a multiphysics problem of thermally coupled incompressible Navier–Stokes equations. We show that with enough number of network parameters, collocation points, and a tighter training convergence criterion that leads to a small training error, the PINNs predicted solution converges to the exact solution in various Sobolev norms. Under these conditions, a small training residual implies a small total error. We present posteriori convergence rates of total error w.r.t training residual and collocation points. This is of practical importance in determining adequate number of training parameters and residual thresholds to get a good PINNs prediction of coupled thermofluidics for steady state laminar flows. We also show that these convergence rates can be generalized for different geometries and higher Reynolds number flows in the laminar regime.

The second major contribution in the paper deals with physics-informed augmentation of PINNs to get better prediction of the pressure field in the coupled system. Borrowing ideas from the stabilized methods [31], we introduce a pres-
sure stabilization type term in the form of pressure Poisson equation into the PDE residuals for PINNs. This term is shown to improve the accuracy of the pressure field by an order of magnitude as compared to the case without augmentation. More sophisticated approaches to stabilized methods for Navier–Stokes equations are presented in [32–34], and to turbulent flows in [35]. A class of variationally derived stabilized methods for thermofluidic systems is presented in [35, 36]. In a recent development, the Variationally Embedded Measured Data (VEMD) method has been proposed [37] that results in forward simulations which are not only driven by boundary and initial conditions but are also augmented by real measurements taken at only a small number of observation points. In the present work we have opted to employ a simple approach to keep the focus of the paper on the ML aspects. In the present work we have opted to employ a simple approach to keep the focus of the paper on the ML aspects. This renders the proposed loss function free of any user-defined penalty parameters, and it is shown to perform robustly in finding a minimizer for the coupled system of PDEs.

An outline of the paper is as follows; Sect. 2 presents a physics informed neural network in the context of thermally coupled steady state incompressible Navier–Stokes equations. We propose physics-informed augmentation to the standard PINNs method in Sect. 3. In Sect. 4, the solution to a subclass of Beltrami flows with a known analytical solution is predicted using the developed framework, and an error analysis is carried out. The effect of physics-informed augmentation, network architecture, global training loss, and collocation points on the prediction error is analyzed. The error estimates are shown to have been generalizad at different geometries and Reynolds’s numbers in Sect. 5. Moreover, the PINNs predicted solution is compared with the solution from a stabilized finite element method [36] in Sect. 6. Finally, concluding remarks are presented in Sect. 7.

2 PINNs for thermally coupled steady state Navier Stokes equations

In this section, we develop PINNs model for solving thermally coupled steady state incompressible Navier–Stokes equations. In thermally coupled flows, the spatially varying active temperature field \( T(x) \) results in a variable density field \( \rho(x) \) in the domain. In the context of steady-state flow of incompressible Newtonian fluids, the effect of density variation is accounted for by introducing a buoyancy force in the momentum balance equation, while the continuity equation appears as an incompressibility condition. In addition, a background thermal gradient is applied via thermal boundary condition, and it serves as the driving mechanism for the thermal phase. The governing system of equations for density-stratified flows known as the Boussinesq equations [38] defined in an open bounded domain \( \Omega \subset \mathbb{R}^{n+1} \), can be written as follows.

\[
\begin{align*}
\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot (2\nu \nabla^2 \mathbf{u}) + g \beta \theta &= \mathbf{f}_b \\
\nabla \cdot \mathbf{u} &= 0 \\
\mathbf{u} \cdot \nabla \theta - \nabla \cdot (\alpha \nabla \theta) &= f
\end{align*}
\]  

where \( \mathbf{u} = [u, v] \) and \( p \) are the velocity and kinematic pressure fields in two dimensions, respectively; \( \theta = T - T_0 \) is the relative temperature (\( T \) is the absolute temperature and \( T_0 \) is the reference temperature); \( \nu \) is the kinematic viscosity of the fluid; \( \beta \) is the thermal expansion coefficient; \( \mathbf{g} = [g_x, g_y] \) is the gravity acceleration vector, \( \alpha \) is the thermal diffusivity; \( \mathbf{f}_b = [f_{bx}, f_{by}] \) is the non-gravitational body force, and \( f \) is the heat source/sink. \( \nabla^2 = (\nabla + \nabla^T)/2 \) is the symmetric gradient operator. Equation (8) is the momentum balance equation with a buoyancy term accounting for the thermal effects, Eq. (9) is the continuity equation to enforce the incompressibility condition, and Eq. (10) is the energy conservation in the form of the convection-diffusion of the relative temperature field. The boundary conditions on the domain boundary \( \Gamma \) are:

\[
\begin{align*}
\mathbf{u}(x) &= \mathbf{g}_M \quad \text{on} \quad \Gamma^M_g \\
\theta(x) &= \mathbf{g}_E \quad \text{on} \quad \Gamma^E_g \\
\sigma \cdot \mathbf{n} &= (2\nu \nabla^2 \mathbf{u} - \rho \mathbf{I}) \cdot \mathbf{n} = \mathbf{h}_M \quad \text{on} \quad \Gamma^M_h \\
\phi \cdot \mathbf{n} &= \alpha \nabla \theta \cdot \mathbf{n} = \mathbf{h}_E \quad \text{on} \quad \Gamma^E_h
\end{align*}
\]  

where \( \mathbf{g}_M \) and \( \mathbf{g}_E \) are the Dirichlet boundary conditions for the velocity field and the relative temperature field, while \( \mathbf{h}_M \) and \( \mathbf{h}_E \) are the Neumann boundary conditions for the total stress \( \sigma \) and heat flux \( \phi \), respectively. \( \mathbf{n} \) is the unit outward normal vector at the boundary. Moreover, these boundaries satisfy the following conditions: \( \Gamma^M_g \cap \Gamma^M_h = \emptyset \), \( \Gamma^M_g \cup \Gamma^M_h = \Gamma \), \( \Gamma^E_g \cap \Gamma^E_h = \emptyset \) and \( \Gamma^E_g \cup \Gamma^E_h = \Gamma \).

For PINNs in two-dimensions, the inputs to the network are the spatial coordinates \( \mathbf{X} = \{x, y\} \) and the outputs are the four solution fields \( \mathbf{Y} = \{u, v, p, \theta\} \). As opposed to other proposed approaches [39] where each solution field is predicted using a separate neural network, we solve the four unknown fields in a coupled fashion using a single network. This approach has advantage in the way that the coupling effect between the mechanical and thermal fields is preserved. The neural network architecture is illustrated in Fig. 2.

In this model, only Dirichlet boundary conditions are taken into account. Since PINNs essentially solve an optimization problem, typically the boundary conditions are satisfied by penalizing the residual/loss function at the
boundaries through a Lagrange multiplier [7, 15, 40, 41], wherein a suitable value of the Lagrange multiplier is chosen through a tedious trial and error procedure. We have not employed any penalty parameter for the boundary residuals, and it is shown through numerical experiments that although the prediction error is mostly accumulated at the boundaries, the boundary residual is less than that of the domain residual during training.

As mentioned in the previous section, a neural network approximates the solution to a PDE by optimized parameters that are obtained by minimizing an objective or loss function. We develop our residual or loss functions using the strong form of the governing PDEs presented in Eqs. (8)–(10). In this case, the loss function or the residual is computed using Mean Squared Error (MSE) of the neural network predicted values on the left hand side and the exact values on the right hand side of Eqs. (8)–(10). The left hand side of the equations require spatial derivatives and is evaluated using the automatic differentiation [8]. The domain residual for our PINNs model is given as,

$$R_{\text{domain}} = R_D^u + R_D^v + R_D^{div} + R_D^\theta$$

where

$$R_D^u = \left\| \nabla \cdot u + \frac{\partial p}{\partial x} - \nu \Delta u + g_x \beta \theta \right\|^2 - (f_{bx})$$

$$R_D^v = \left\| \nabla \cdot v + \frac{\partial p}{\partial y} - \nu \Delta v + g_y \beta \theta \right\|^2 - (f_{by})$$

$$R_D^{div} = \| \nabla \cdot u \|^2$$

$$R_D^\theta = \| (u \nabla \theta - \alpha \Delta \theta) - (f) \|^2$$

whereas boundary residuals are computed by comparing exact values with the

$$R_{\text{boundary}} = R_B^u + R_B^v + R_B^\theta$$

where,

$$R_B^u = \| u - g_Mx \|^2$$

$$R_B^v = \| v - g_My \|^2$$

$$R_B^\theta = \| \theta - g_E \|^2$$

The total loss or residual is comprised of the residual from both the domain and the boundaries:

$$R_{\text{total}} = R_{\text{domain}} + R_{\text{boundary}}$$

### 3 Physics informed augmentation to PINNs

For PINNs applied to incompressible Navier–Stokes equations, satisfying incompressibility constraint is an important issue. Since pressure acts as a Lagrange multiplier that enforces the incompressibility constraint, any error arising due to the weak enforcement of this constraint leads to less accurate prediction of the pressure field [15]. In order to get
a better prediction of the pressure field, strategies involving training of neural networks via an augmented dataset, enforcement of penalty on the pressure field during the training process, and input/output perturbation are employed \([15, 42, 43]\). We propose a physics-informed augmentation in which the residuals are constructed based on a pressure Poisson equation and spatial derivatives of the incompressibility constraint. The augmentation is intended to help improve the pressure prediction by satisfying additional physics-based constraints within the domain which in turn reduces the number of local minima and helps in constraining the search space of the stochastic gradient descent optimization. The governing equations of the augmentation in the domain \( \Omega \in \mathbb{R}^{n_d} \) are given as follows,

\[
\Delta p = \nabla \cdot (\mathbf{f}_b - \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{g} \beta \theta) \tag{18}
\]

\[
(\nabla \cdot \mathbf{u})_x = 0 \tag{19}
\]

\[
(\nabla \cdot \mathbf{u})_y = 0 \tag{20}
\]

This gives rise to an augmented residual or loss function which is given as,

\[
R_{augm} = R^D_p + R^D_{div,x} + R^D_{div,y} \tag{21}
\]

where,

\[
R^D_p = \| (\Delta p) - (\nabla \cdot (\mathbf{f}_b - \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{g} \beta \theta)) \|^2
\]

\[
R^D_{div,x} = \| (\nabla \cdot \mathbf{u})_x \|^2
\]

\[
R^D_{div,y} = \| (\nabla \cdot \mathbf{u})_y \|^2
\]

The proposed augmentation sits in the total residual given in Eq. (17), and in the next section it is shown to significantly improve the pressure field prediction.

\[
\hat{R}_{total} = R_{domain} + R_{boundary} + R_{augm} \tag{22}
\]

**Remark 1** From the mean value theorem, if the derivative of a continuous function is zero then the function is constant over the domain. Combined with Eq. (9), this ensures stricter enforcement of the incompressibility constraint. Consequently, the derivatives of the velocity divergence field were added to the augmented residual.

---

### 4 Numerical experiments: Beltrami flows

We employ the PINNs model developed in the previous sections to predict solution of a subclass of Beltrami flows \([44]\). The problem has an analytical solution against which the trained neural networks are tested. This class of problems is used to study buoyancy-induced convection and heat transfer phenomenon and has wide-ranging applications in engineering systems, such as solar collectors, electronic cooling, heat exchangers, and thermal insulating systems etc. \([45]\). The analytical expressions for the velocity, pressure and relative temperature fields are as follows,

\[
\mathbf{u} (x, y) = \begin{bmatrix} -\cos (\pi x) \sin (\pi y) \\ \sin (\pi x) \cos (\pi y) \end{bmatrix} \tag{23}
\]

\[
p (x, y) = -\frac{1}{4} (\cos (2\pi x) + \cos (2\pi y)) \tag{24}
\]

\[
\theta (x, y) = \cos (\pi x) \cos (\pi y) \tag{25}
\]

and the body force and heat source driving the flow are as follows,

\[
f_b (x, y) = \begin{bmatrix} -2\pi^2 v \cos (\pi x) \sin (\pi y) \\ + \frac{\pi^2}{\nu} \cos (\pi x) \cos (\pi y) \end{bmatrix} \tag{26}
\]

\[
f (x, y) = 2\pi^2 \alpha \cos (\pi x) \cos (\pi y) \tag{27}
\]

The computational domain is a bi-unit square \( \Omega = [-1, 1] \times [-1, 1] \) that covers one period of the sinusoidal velocity and pressure solution fields. Dirichlet boundary conditions \( \mathbf{g} (x) = [\mathbf{u}, p, \theta] \big|_{x \in \Gamma} \) for the velocity, pressure, and the relative temperature fields are applied at the domain boundaries \( \Gamma = \partial \Omega \). The non-dimensional material parameters used in our problem are all unity i.e., \( \alpha = \beta = \nu = 1.0 \). The exact solution and boundary conditions are shown in Fig. 3.

For training, random collocation points are generated in the domain as well as on the boundaries using Latin hyper-cube sampling strategy \([46]\). Total collocation points are chosen such that the domain has twice the number of points than on all the four boundaries combined. For convergence analysis, collocation points are hierarchically increased such that the next set of points also contains the previous points. The details of each training dataset is given in Table 1. The trained network is tested on a 100 × 100 uniform grid of collocation points that covers the problem domain and boundaries.

A schematic diagram of training and test points is shown in Fig. 4.

#### 4.1 Convergence analysis and error estimates

In order to answer the questions put forth in Sect. 1, we systematically evaluate the impact of the choice of network architecture, physics-informed augmentation, collocation points, and training error on the total error. This methodology enables us to numerically estimate *a-posteriori* convergence rates of the total error that are of practical importance in
selecting appropriate PINNs model for thermally coupled Navier-Stokes equations.

For this purpose, the solution \( \hat{u} \) from a trained network is predicted on the 100 \( \times \) 100 grid of testing points and the total error is computed w.r.t analytical solution \( \tilde{u} \) in Sobolev norms as defined below:

\[
\| \tilde{u} - \hat{u} \|_{W^{k,\infty}(\Omega)} = \max_{0 \leq m \leq k} \| D^\alpha (\tilde{u} - \hat{u}) \|_{L^\infty(\Omega)}
\]

for \( m = 0, \ldots, k \). \( D^\alpha \) is the operator for spatial derivative of order \( \alpha \).

The total error on test points ensures that the error estimates are also applicable to the data not used in the training and without any prior knowledge of the exact solution.

\[
\| \tilde{u} - \hat{u} \|_{W^{m,\infty}(\Omega)} = \max_{|\alpha| = m} \| D^\alpha (\tilde{u} - \hat{u}) \|_{L^\infty(\Omega)}
\]
4.1.1 Effect of network architecture on approximation and training errors

Since generalization error comprises of both the approximation and training errors, it is important to have a neural network that is a consistent estimator, and is capable of approximating the given loss function with a mapping that converges to a specified threshold value for the training error. This section investigates the effect of network architecture and trainable parameters on approximation and convergence capabilities of PINNs. Six different network architectures are employed with single and double hidden layers and varying number of neurons in each layer (32, 64, and 128). As we increase the number of hidden layers or neurons, the number of trainable parameters of network also increases as shown in Fig. 5. The stopping criterion for training of each network is either the training error of $10^{-4}$ or the maximum epochs of 350,000. With this in place, each network is trained on a range of collocation points. The numbers within each cell indicate the epochs it took for a network to reach the specified training error while N.C. indicates that the network has failed to converge to the training threshold and instead exceeded the maximum number of epochs.

From Fig. 5 it is observed that as we increase the number of collocation points, shallower networks with lesser trainable parameters fail to converge. As we go to deeper and wider networks, network parameters as well as the approximation capability increases, and the network converges faster to the training threshold [47]. This is consistent with the universal approximation theorem. Based on the study, we chose 2-128-128-4 architecture for subsequent error analysis.

Remark 2 One key take away from this study is that the convergence properties of a given neural network depends on the size of training dataset as well as on the training error. For instance, networks of fixed size have irreducible approximation error irrespective of the size of training dataset (beyond a specific threshold), i.e., increasing the number of collocation points for the augmented and the non-augmented cases (training error = $10^{-4}$, collocation points = 1536).
points beyond this threshold would not reduce total error and networks are no longer consistent estimators.

### 4.1.2 Effect of physics informed augmentation on pressure prediction

In this section, we investigate the effect of proposed augmentation on PINNs predicted pressure field. Figure 6 shows the convergence of total error as a function of collocation points for several training error thresholds. Physics informed augmentation significantly reduces error in the pressure field at every training threshold. This shows that the additional terms in the loss function are significant for a better pressure prediction in the coupled system of PDEs. The observation is further confirmed by the qualitative error plots for pressure field in Fig. 7. Without augmentation, the error w.r.t analytical solution is an order of magnitude higher than the case with augmentation.

We also investigate the computational cost of the physics informed augmentation. There is no significant increase in the computational cost with the added augmentation for larger training loss thresholds ($10^{-1}$, $10^{-2}$, and $10^{-3}$) as shown in Fig. 8. For the training error of $10^{-4}$, computational cost of training PINNs with physics informed augmentation is 2–7 times than that of no augmentation. The computational complexity w.r.t the total collocation points for the augmented and the non-augmented case is $O\left(N^{1.9}\right)$ and $O\left(N^{1.7}\right)$, respectively, where $N$ is the number of collocation points.

### 4.1.3 Convergence analysis of total error

As discussed earlier, the underlying idea in PINNs is to minimize the training error over some network parameters. To justify their empirical performance, we have to establish that for a given small training error and in turn the generalization error, the corresponding total error is also small. This section investigates the relation between the training and total error and we estimate the convergence rates that are important in establishing the convergence of PINNs. The network architecture of 2-128-128-4 with augmentation is selected and the total training threshold is varied as $10^{-1}$, $10^{-2}$, $10^{-3}$, and $10^{-4}$. For a specified number of collocation points, the network is trained to a given training error and the total error between the exact solution and the PINNs predicted solution is computed over $W^{0,\infty}$, $W^{1,\infty}$, and $W^{2,\infty}$ Sobolev norms. The number of collocation points are hierarchically increased to determine enough number of points for optimal convergence.

![Fig. 9 Convergence of $W^{0,\infty}$ norm of error w.r.t training error](image-url)
Fig. 10 Convergence of $W^{1,\infty}$ norm of error w.r.t training error

Fig. 11 Convergence of $W^{2,\infty}$ norm of error w.r.t training error
It is observed that the number of collocation points are important in getting optimal convergence of total error. All the plots in Figs. 9, 10 and 11 show that after 200 points, the convergence rates becomes optimal. For lower number of collocation points, there is no significant convergence. We estimate the convergence rates for the solution fields, first derivatives, and second derivatives in Figs. 9, 10 and 11, respectively. In general the convergence rates improve as we go to higher order derivatives. For 1536 collocation points, the approximate convergence rates of total error for velocity, pressure, and temperature fields are \( O\left(\frac{e_T}{3}\right) \), \( O\left(\frac{e_T}{2}\right) \), and \( O\left(\frac{e_T}{2}\right) \), respectively.

We also analyze the training error in the domain as well as on the boundaries for a single and double layer network in Fig. 12. The threshold value is set at \( 10^{-4} \). It is seen that the boundary error is an order of magnitude less than the domain error which indicates that the total training error is not limited by the weak enforcement of the boundary conditions in this case. Moreover, it is observed that deeper network shows greater oscillations during the training. Since deeper networks have larger number of parameters, these oscillations might be due to redundancy in the parameterisation of the function space in a neighborhood of a local minimum [48].

Recently, theoretical error estimates for incompressible Navier–Stokes equations under periodic boundary conditions were presented in [30]. Although the PDE system and the boundary conditions do not exactly match the present case, the error bounds give scaling between total error and training error as follows.

\[
\int_\Omega \| \tilde{u} - \hat{u} \|_2 \, dx \lesssim O\left(\sqrt{e_T}\right) \tag{29}
\]

Figure 13 shows that total error scales approximately as the square root of the training error for all the solution fields which validates our numerical findings.

### 4.1.4 Consistency of PINNs

A consistent neural network converges to the exact solution if the sample size grows to infinity [49, 50]. This section explores the consistency of PINNs within a sample limit, which is equivalent to the convergence of the generalization error. As seen in the previous sections, number of collocation points affects the accuracy of PINNs predicted solution. Typically the number is selected using a trial-and-error procedure, however, a systematic approach is needed to sample enough number of points that can predict correct physics of the problem while minimizing the total error. For this
purpose, an error convergence study w.r.t the training collocation points is carried out. The points are hierarchically increased such that the current set of points are fully represented in the subsequent larger training dataset, thereby leading to a consistent convergence analysis. Figures 14, 15 and 16 show convergence plots for each of the velocity, pressure, and temperature fields and their gradients at various training residual thresholds. A good estimate for the number of collocation points is reached when the error convergence rate drops and there is not much reduction in error with the subsequent increase in collocation points.

The convergence analysis shows that the rate of convergence increases for lower training error thresholds. The drop in the convergence rates indicate that the network has reached its approximation capacity with the given number of collocation points and the training error threshold. For the training error of $10^{-4}$, the convergence rates in the $W^{0,\infty}$ norm of error for all the solution fields are super-linear i.e., between $O\left(N^{3/2}\right)$ and $O\left(N^{2}\right)$. The convergence rates drop to $O\left(N\right)$ for $W^{1,\infty}$ and $W^{2,\infty}$ norms for error that also includes first and second order derivatives of the solutions fields, respectively.

The PINNs predicted solution fields for various number of collocation points are shown in Fig. 17. Although the network converges to the given training error at lower number of collocation points (first row in Fig. 17), it fails to produce an accurate solution because it converges to wrong minima. Moreover, even though the qualitative solution fields in the second and third row look quite similar, the error plots in Fig. 18 indicate that there is an order of magnitude difference in the error between the two solutions. This highlights the importance of having a network that is consistent, and an error convergence study for selecting the appropriate number of collocation points for a particular problem. In Fig. 18, the error bar on the right indicates the magnitude of point-wise error in the predicted velocity, pressure, and temperature fields corresponding to that row.

**Remark 3** Number of collocation points affects the accuracy of PINNs predicted solution and these points should be chosen depending on the length scale of the physics of the problem.

### 5 Generalization of error estimates for different problem domain and higher Reynolds number flow using transfer learning

In general, PINNs need to be trained again if flow parameters or problem domain is changed. Since the computational cost of training PINNs is very high, re-training from scratch is not cost effective for its generalization. However, the training of PINNs can be accelerated by using transfer learning. In transfer learning, instead of training from scratch, the parameters of the already trained network (e.g., at a lower Reynolds

![Fig. 14](image_url) Convergence of $W^{0,\infty}$ norm of error w.r.t training collocation points

![Diagram](image_url)
Fig. 15 Convergence of $W^{1,\infty}$ norm of error w.r.t training collocation points

Fig. 16 Convergence of $W^{2,\infty}$ norm of error w.r.t training collocation points
Fig. 17 Qualitative plots of the solution fields for PINNs trained at different number of collocation points (first row = 12 points, second row = 96 points, third row = 1536 points)

Fig. 18 Qualitative plots of the pointwise error w.r.t the exact solution for PINNs trained at different number of collocation points (first row = 12 points, second row = 96 points, third row = 1536 points)
number) are transferred to a network for different flow conditions (e.g., a higher Reynolds number flow). In this way, only a fraction of the original epochs are needed to obtain an accurate solution. We have used BFGS optimizer in transfer learning which is faster when the network parameters are already trained and they only need some fine-tuning.

To establish the generalization of the error estimates presented in Sect. 4, we employ transfer learning to train networks for two test cases. The first test case involves a different geometric description of the problem domain which is half of the original domain in the x-direction i.e., $\Omega = [0, 1] \times [-1, 1]$. The sinusoidal solution is no longer axisymmetric for the new geometric layout. In the second test case, we retain the spatial domain as that of the original problem, while we increase the Reynolds number from 1 to 10 and modify the value of gravitational acceleration from $-1$ to $-9.8$. In both the cases, the $L_2$-norm of the total error scales as the square root of the training error in accordance with Eq. 29 as shown in Fig. 19. Moreover, the convergence rates of total error w.r.t the training error and the number of collocation points are in the same ballpark as that for the original problem. These convergence rates are presented in A and B. These results show the practical significance of the numerical error estimates obtained in this work as they are consistent and can be generalized to different spatial domains or flow parameters in the coupled thermal flow problems.

### 6 Comparison between FEM and PINNs

In this section, we compare PINNs and a stabilized finite element method (FEM) [36] for the solution of Beltrami flow. The FEM solution is computed on a $100 \times 100$ structured mesh of linear quadrilateral and triangular elements with 39, 204 degrees of freedom. For PINNs, 2-128-128-4 network is employed with 1536 total collocation points and a training error of $10^{-4}$. The trained network is then used to predict solution over a $100 \times 100$ grid of testing points as shown in Fig. 4 and point-wise error w.r.t analytical solution is computed.

The solution fields as well as error w.r.t analytical solution for both the methods are presented in Figs. 20 and 21, respectively. Qualitatively there is no significant difference between the solution fields from both the methods. However, PINNs show better error performance in the domain interior as the FEM solution has a larger magnitude of error in there as compared to PINNs. Particularly for the pressure field, PINNs predicted solution looks much better in the domain interior as compared to FEM which is due to the pressure augmentation term we have introduced in our model. On the boundaries, error from the FEM solution is lesser as compared to the error in PINNs. This is because the boundary conditions are strongly enforced in FEM while they are weakly enforced in PINNs. We also notice the non-symmetry of the error fields for the PINNs predicted solution which is due to the randomly chosen collocation points that are not symmetric. Selecting a structured grid of collocation points for training would result in symmetric error plots.

Computationally PINNs take much more computational time and resources to train and predict the solution as compared to FEM. FEM takes about 25 s to generate the solution on a single CPU (excluding the time taken for meshing) while PINNs take almost 20 h to train on a single CPU for the largest training dataset (1536 points) and a training residual of $10^{-4}$. Once the network is trained, it predicts the solution instantaneously on any given set of points. Moreover, for a lower training error threshold and lesser number of collocation points (e.g., $10^{-3}$ and 768), PINNs take only about an hour to train and the solution is good enough for all practical design purposes. The training time for PINNs can be further reduced by using GPUs, faster optimization algorithm such as BFGS, and transfer learning. Although PINNs are computationally less efficient for forward problems with structured meshes, they can be efficient for problems where meshing takes considerable time.
7 Conclusions

We have presented convergence analysis for PINNs, applied to the solution of 2D steady incompressible Navier–Stokes PDEs coupled to the scalar energy PDE and subjected to Dirichlet boundary conditions. A pressure stabilization term in the form of pressure Poisson equation is added to the PDE residuals for PINNs. This physics informed augmentation of thermally coupled Navier–Stokes equations improves accuracy of the pressure field by an order of magnitude as compared to the case without augmentation. Via a model problem of Beltrami flows for which an analytical solution exists, we investigated the effects of (i) network size, (ii) collocation points, (iii) training error, and (iv) physics-informed augmentation of the model, on the reduction in the total error in the predicted solution. Through a carefully designed set of numerical test cases, posteriori error estimates were obtained. The network convergence study showed better approximation capability and faster convergence of deeper networks that have larger trainable parameters, an outcome which is in accordance with the universal approximation theorem. The convergence of PINNs was established by showing that with enough training parameters and collocation points, a small training error leads to a small total error. The conver-
Convergence rate of total error w.r.t the training error was observed to be sub-linear for different Sobolev norms, while the total error was observed to scale as the square root of training error in the $L_2$ norm. The consistency of PINNs was shown through convergence study of total error w.r.t the training dataset. The rates of convergence of total error w.r.t the number of collocation points were super-linear for all the solution fields ($W^{0,\infty}$ norm) and were linear for their first and second derivatives ($W^{1,\infty}$ and $W^{2,\infty}$ norms). Faster convergence rates were observed as the training error threshold was reduced. Moreover, we showed that all these convergence rates can be generalized for problems with different spatial domain, and/or different flow parameters such as a larger body force and a higher Reynolds number within the laminar range. A comparison of the PINNs predicted solution with the finite element based solution also highlighted the various mathematical attributes of PINNs for application to coupled multiphysics problems.

Acknowledgements Computing resources for the numerical test cases were provided by the Teragrid/XSEDE Program under NSF Grant TG-DMS100004. This research was partially supported by NIH-USA Grant No. R01GN135921.

Declarations

Conflicts of interest Authors have no competing interests.

Appendix A . Error estimates for a different problem domain

In this appendix, error estimates for a problem with a different geometric description than that of the problem in Sect. 4 are presented. The new domain is half of the original domain in the x-direction i.e., $\Omega = [0, 1] \times [-1, 1]$ and the sinusoidal solution is no longer axisymmetric. Convergence rate of total error w.r.t the training error is shown in Fig. 22 and the convergence of $W^{0,\infty}$, $W^{1,\infty}$, and $W^{2,\infty}$ norms of total error w.r.t total number of collocation points is shown in Figs. 23, 24 and 25. The convergence rates are in the same ballpark as in Sect. 4.
Fig. 23  Convergence of $W^{0,\infty}$ norm of error w.r.t the number of training collocation points

Fig. 24  Convergence of $W^{1,\infty}$ norm of error w.r.t the number of training collocation points
Appendix B . Error estimates for higher Reynolds number

Here error estimates for a problem with a different flow parameters than that of the problem in Sect. 4 are presented. Reynolds number is increased from 1 to 10 and the value of gravitational acceleration is changed from -1 to -9.8. Convergence rate of total error w.r.t the training error is shown in Fig. 26 and the convergence of $W^{2,\infty}$, $W^{1,\infty}$, and $W^{0,\infty}$ norms of total error w.r.t total number of collocation points is shown in Figs. 27, 28 and 29. The convergence rates are similar to the ones shown in Sect. 4.
Fig. 26 Convergence of total error w.r.t training error under different Sobolev norms

Fig. 27 Convergence of $W^{0,\infty}$ norm of error w.r.t the number of training collocation points
Fig. 28 Convergence of $W^{1,\infty}$ norm of error w.r.t the number of training collocation points
Fig. 29 Convergence of $W^{2,\infty}$ norm of error w.r.t the number of training collocation points

References

1. Ames WF (1967) Nonlinear partial differential equations, vol 1. Elsevier, Amsterdam
2. Finlayson BA (2013) The method of weighted residuals and variational principles. SIAM, New Delhi
3. Fletcher CA (1988) Computational techniques for fluid dynamics. Volume 1-fundamental and general techniques. Volume 2-specific techniques for different flow categories. Springer-Verlag, Berlin and New York
4. Dissanayake M, Phan-Thien N (1994) Neural-network-based approximations for solving partial differential equations. Commun Numer Methods Eng 10(3):195–201
5. Lagaris IE, Likas A, Fotsiadis DI (1998) Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans Neural Netw 9(5):987–1000
6. Raissi M, Wang Z, Karniadakis GE (2019) Deep learning of vortex-induced vibrations. J Fluid Mech 861:119–137
7. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
8. Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2018) Automatic differentiation in machine learning: a survey. J Mach Learn Res 18:1–43
9. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
10. Pinkus A (1999) Approximation theory of the mlp model in neural networks. Acta Numer 8:143–195
11. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al (2016) Tensorflow: a system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI’16), pp 265–283
12. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in PyTorch
13. Shin Y, Darbon J, Karniadakis GE (2020) On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. arXiv:2004.01806
14. Gao H, Sun L, Wang J-X (2021) Phylgeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. J Comput Phys 428:110079
15. Jin X, Cai S, Li H, Karniadakis GE (2021) Nsfnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations. J Comput Phys 426:109951
16. Mao Z, Jagtap AD, Karniadakis GE (2020) Physics-informed neural networks for high-speed flows. Comput Methods Appl Mech Eng 360:112789
17. Rao C, Sun H, Liu Y (2020) Physics-informed deep learning for incompressible laminar flows. Theor Appl Mech Lett 10(3):207–212
18. Sun L, Wang J-X (2020) Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data. Theor Appl Mech Lett 10(3):161–169
19. He Q, Barajas-Solano D, Tartakovsky G, Tartakovsky AM (2020) Physics-informed neural networks for multiphysics data assimila-
tion with application to subsurface transport. Adv Water Resour 141:103610
20. Tripathy RK, Bilionis I (2018) Deep uq: learning deep neural network surrogate models for high dimensional uncertainty quantification. J Comput Phys 375:565–588
21. Zhu Y, Zabaras N (2018) Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. J Comput Phys 366:415–447
22. Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE (2021) Physically informed neural networks for heat transfer problems. J Heat Transf 143(6):060801
23. Kissas G, Yang Y, Hwuang E, Witschey WR, Detre JA, Perdikaris P (2020) Machine learning in cardiovascular flows modeling: predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. Comput Methods Appl Mech Eng 358:112623
24. Sahil Costabal F, Yang Y, Perdikaris P, Hurtado DE, Kuhl E (2020) Physics-informed neural networks for cardiac activation mapping. Front Phys 8:42
25. Zhang X, Garikipati K (2020) Machine learning materials physics: multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures. Comput Methods Appl Mech Eng 372:113362
26. Karumuri S, Tripathy R, Bilionis I, Panchal J (2020) Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. J Comput Phys 404:109120
27. Patel DV, Ray D, Oberai AA (2022) Solution of physics-based Bayesian inverse problems with deep generative priors. Comput Methods Appl Mech Eng 400:115428
28. Mishra S, Molinaro R (2022) Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for pdes. IMA J Numer Anal 42(2):981–1022
29. Guo M, Haghighat E (2022) Energy-based error bound of physics-informed neural network solutions in elasticity. J Eng Mech 148(8):04022038
30. De Ryck T, Jagtap AD, Mishra S (2022) Error estimates for physics informed neural networks approximating the Navier–Stokes equations. arXiv:2203.09346
31. Masud A, Khurram R (2004) A multiscale/stabilized finite element method for the advection-diffusion equation. Comput Methods Appl Mech Eng 193(21–22):1997–2018
32. Masud A, Khurram R (2006) A multiscale finite element method for the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 195(13–16):1750–1777
33. Masud A, Calderer R (2009) A variational multiscale stabilized formulation for the incompressible Navier–Stokes equations. Comput Mech 44(2):145–160
34. Masud A, Al-Naseem AA (2018) Variationally derived discontinuity capturing methods: fine scale models with embedded weak and strong discontinuities. Comput Methods Appl Mech Eng 340:1102–1134
35. Zhu L, Masud A (2021) Residual-based closure model for density-stratified incompressible turbulent flows. Comput Methods Appl Mech Eng 386:113931
36. Zhu L, Goraya SA, Masud A (2023) A variational multiscale method for natural convection of nanofluids. Mech Res Commun 127:103960
37. Masud A, Goraya SA (2022) Variational embedding of measured data in physics-constrained data-driven modeling. J Comput Mech 89(11):11001
38. Doering CR, Gibbon JD (1995) Applied analysis of the Navier-Stokes equations, vol 12. Cambridge University Press, Cambridge
39. Haghighat E, Raissi M, Moure A, Gómez H, Juanes R (2021) A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput Methods Appl Mech Eng 379:113741
40. Sirignano J, Spiliopoulos K (2018) Dgm: a deep learning algorithm for solving partial differential equations. J Comput Phys 375:1339–1364
41. Rao C, Sun H, Liu Y (2021) Physics-informed deep learning for computational elastodynamics without labeled data. J Eng Mech 147(8):04021043
42. Simard PY, Steinkraus D, Platt JC, et al (2003) Best practices for convolutional neural networks applied to visual document analysis. In: Icdar, Vol. 3
43. Yaeger L, Lyon R, Webb B (1996) Effective training of a neural network character classifier for word recognition, Advances in neural information processing systems, Vol. 9, MIT Press, Cambridge MA
44. Wong JC-F (2007) Numerical simulation of two-dimensional laminar mixed-convection in a lid-driven cavity using the mixed finite element consistent splitting scheme. Int J Numer Methods Heat Fluid Flow 17:46–93
45. Garnier C, Currie J, Muneer ET (2009) Integrated collector storage solar water heater: temperature stratification. Appl Energy 86(9):1465–1469
46. McKay MD, Beckman RJ, Conover WJ (2000) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42(1):55–61
47. Raghu M, Poole B, Kleinberg J, Ganguli S, Sohl-Dickstein J (2017) On the expressive power of deep neural networks. In: International conference on machine learning. PMLR, pp 2847–2854
48. Saarinen S, Bramley R, Cybenko G (1993) Ill-conditioning in neural network training problems. SIAM J Sci Comput 14(2):693–714
49. Strang G (2019) Linear algebra and learning from data, vol 4. Wellesley-Cambridge Press, Cambridge
50. Kutz JN (2013) Data-driven modeling & scientific computation: methods for complex systems & big data. Oxford University Press, Oxford

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.