Galerkin Neural Network Approximation of Singularly-Perturbed Elliptic Systems

Mark Ainsworth*, Justin Dong†

June 17, 2022

Abstract

We consider the neural network approximation of systems of partial differential equations exhibiting multiscale features such as the Reissner-Mindlin plate model which poses significant challenges due to the presence of boundary layers and numerical phenomena such as locking. This work builds on the basic Galerkin Neural Network approach established in [1] for symmetric, positive-definite problems. The key contributions of this work are (1) the analysis and comparison of several new least squares-type variational formulations for the Reissner-Mindlin plate, and (2) their numerical approximation using the Galerkin Neural Network approach. Numerical examples are presented which demonstrate the ability of the approach to resolve multiscale phenomena such for the Reissner-Mindlin plate model for which we develop a new family of benchmark solutions which exhibit boundary layers.

1 Introduction

Neural networks offer an interesting alternative to traditional numerical methods for partial differential equations (PDEs) such as finite elements, finite differences, and finite volumes, and have been used to approximate various linear and nonlinear elliptic, parabolic, and hyperbolic PDEs [9, 23, 29, 30, 33, 39]. Generally, such approaches seek to approximate the true solution by the realization of a neural network which is trained by minimizing the ℓ^2-norm of the strong residual of the PDE. The fact that the PDE will not have strong solutions in general has prompted the development of neural network frameworks based on variational principles [25,26,38,40]. In [1], we proposed an adaptive neural network framework (Galerkin Neural Networks) for approximating symmetric, positive-definite variational equations which also incorporated error control and showed it to be capable of achieving a high level of accuracy on a range of standard test problems.

Of course, all of this begs the question of what advantages (if any) are offered by neural networks over traditional numerical methods? Traditional methods have the benefit of being refined over many decades and are often capable of delivering high-fidelity approximations

*Brown University, Division of Applied Mathematics

†Brown University, Division of Applied Mathematics

efficiently while the potential benefits of a neural network-based approach for such problems are unclear. It is nevertheless the case that there remain many classes of problem that pose difficulties for traditional numerical methods. Examples include parameter-dependent problems that arise in linear elasticity and plate theory which exhibit locking [17]. Locking means that the numerical approximation deteriorates for small parameter values even though the solution itself may not be sensitive to this parameter. Additionally, in some models such as the Reissner-Mindlin plate model, a characteristic feature of solutions is the presence of boundary layers. In conjunction with the possibility of locking, the robust and accurate approximation of such problems still poses a serious challenge, which has led to the development of a whole gamut of sophisticated techniques in an attempt to obtain schemes capable of delivering robust and high-resolution approximations [3, 6, 10, 13, 16, 20, 22, 34, 35].

In this work, we consider the question of whether a neural network approach is capable of approximating Reissner-Mindlin plates uniformly in the plate thickness while also resolving multiscale features such as boundary layers. The universal approximation property of neural networks should, in theory, mean that neural networks have the capability to resolve multiscale features. While results exist quantifying how the accuracy of a neural network approximation varies with respect to the width and depth of the network [**?**, **?**, 18, 27, 28, 32] the capability of neural networks to uniformly approximate parameter-dependent functions exhibiting boundary layers remains open. The relationship between ReLU networks and piecewise linear approximations means that one can expect univariate ReLU networks to be capable of delivering uniformly accurate approximations of functions exhibiting boundary layers. The results in Figure 1 confirm this expectation. While such a result shows that neural networks are capable of delivering uniformly accurate approximations for direct approximation of functions exhibiting boundary layers in the univariate case, it remains to be seen whether similar results can be achieved in higher dimensions and when the function is

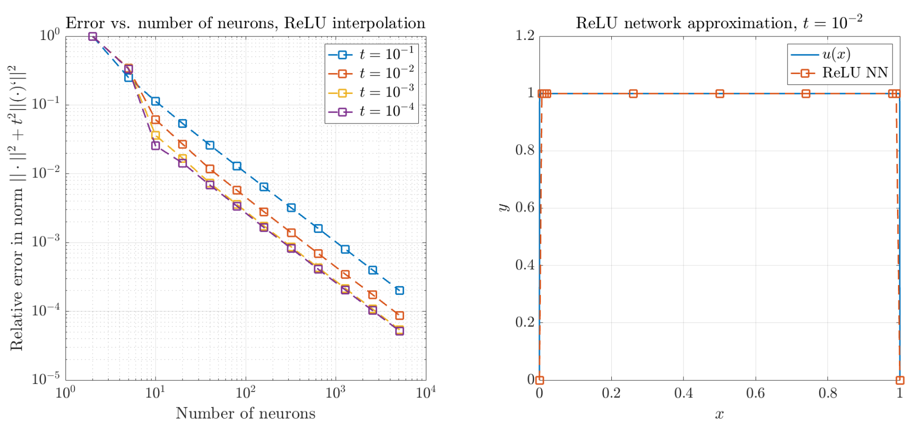


Figure 1: Left: Numerical approximation rate of the function $u(x) = (1 - e^{(x-1)/t} - e^{-x/t} + e^{-1/t})/(1 + e^{-1/t})$ for various t using a ReLU network with one hidden layer. Right: The function $u(x)$ alongside the ReLU approximation. The hidden weights are set to 1, the biases are graded to account for the boundary layers, and the activation coefficients are chosen so that the network interpolates the function u at the points where the ReLU basis functions activate.

Algorithm 1: Galerkin Neural Network Framework.

Input: Data L, bilinear operator a, network widths $\{n_i\}$, initial approximation $u_0 \in X$, tolerance $\texttt{tol} > 0$, optimization subroutine AugmentBasis (any optimization procedure for approximating (7)).

Output: Numerical approximation u_N to the variational problem: $u \in X$ such that $a(u, v) = \langle f, v\rangle$ for all $v \in X$; basis functions $\{\varphi_i^{NN}\}_{i=0}^N$.

Set $i = 1$ and $\varphi_0^{NN} = u_0$.
$\varphi_1^{NN} \leftarrow$ AugmentBasis(u_0).
while $\langle r(u_{i-1}), \varphi_i^{NN}\rangle / |||\varphi_i^{NN}||| > \texttt{tol}$ **do**
  Form $S_i := \text{span}\{u_0, \varphi_1^{NN}, \ldots, \varphi_i^{NN}\}$ and seek $u_i \in S_i \; : \; a(u_i, v) = L(v)$ for all $v \in S_i$.
  $\varphi_{i+1}^{NN} \leftarrow$ AugmentBasis(u_i).
  Set $i \leftarrow i + 1$.
end
Return u_N and $\{\varphi_j^{NN}\}_{j=0}^N$.

implicitly defined to be the solution of a singularly perturbed system of elliptic PDEs.

Accordingly, we investigate the neural network approximation of Reissner-Mindlin plates. As alluded to earlier, the nonlinear nature of neural networks should mean that they are naturally adaptive to multiscale features without the need for specialized grids or elements. In addition, it is easy to increase the smoothness of the neural network approximation by choosing appropriate activation functions, which gives the flexibility to consider natural variational formulations posed on smoother Sobolev spaces than might otherwise be practical were standard finite elements to be used. We introduce and analyze several new, least squares-type, variational formulations for the Reissner-Mindlin plate and use them in conjunction with the Galerkin Neural Network approach developed in [1]. The relative performance of the various formulations and, in particular, whether they exhibit locking, together with their ability to resolve multiscale features is illustrated for a class of new benchmark (closed form) solutions of the Reissner-Mindlin problem which exhibits boundary layers.

The rest of this work is structured as follows. In Section 2, we review the Galerkin Neural Network framework and its underlying theory which will be used to approximate the model problem. In Section 3, we introduce a new realistic benchmark problem which exhibits multiscale features and also consider the capacity of neural networks to approximate such features. In Section 4, we apply the Galerkin Neural Network method to several new variational formulations for the Reissner-Mindlin plate and demonstrate the robustness of our approach with respect to the plate thickness. Conclusions follow in Section 5.

2 Galerkin Neural Network Framework

In this section, we briefly summarize the main features of the Galerkin Neural Network approach developed in [1] that we will later use. Let

$$V_n^\sigma := \{v : v(x) = \sum_{i=1}^{n} c_i \sigma(x \cdot W_i + b_i),\ b_i, c_i \in \mathbb{R},\ W_i \in \mathbb{R}^d,\ x \in \Omega\} \tag{1}$$

be the set of all functions which are the realizations of a feedforward neural network consisting of a single hidden layer of n neurons and nonlinear, continuous activation function $\sigma : \mathbb{R} \to \mathbb{R}$. Here, W and b are the weights and biases, respectively, of the hidden layer, and c are the activation coefficients of the network. Additionally, we let $V_{n,C}^\sigma$ be the subset consisting of realizations with bounded parameters:

$$V_{n,C}^\sigma := \{v \in V_n^\sigma \ :\ ||(W, b, c)|| < C\}, \tag{2}$$

where $||(W, b, c)|| := \max_{ij} |W_{ij}| + \max_i |b_i| + \max_i |c_i|$.

A key property of neural networks is that they are universal approximators [21] in the sense that, for any given function $f \in H^s(\Omega)$ and $\tau > 0$, there exist a network of width n and a function $\tilde{f} \in V_n^\sigma$ such that $||f - \tilde{f}||_{H^s(\Omega)} < \tau$. The universal approximation property suggests that neural networks might be used to approximate the solutions of PDEs. To this end, consider the following variational problem:

$$u \in X \ :\ a(u, v) = L(v) \quad \forall v \in X, \tag{3}$$

where $X \subset H^s(\Omega)$, $L(\cdot)$ is a bounded linear operator on $H^s(\Omega)$, and $a(\cdot, \cdot)$ is a symmetric, positive-definite bilinear operator on $H^s(\Omega)$ which is continuous and coercive with respect to $H^s(\Omega)$, i.e. there exist constants $M, \alpha > 0$ such that $|a(u, v)| \leqslant M||u||_{H^s(\Omega)}||v||_{H^s(\Omega)}$ and $\alpha||v||_{H^s(\Omega)}^2 \leqslant a(v, v)$ for all $u, v \in X$. The bilinear form $a(\cdot, \cdot)$ induces a norm denoted by $|||\cdot|||_a := \sqrt{a(\cdot, \cdot)}$.

In previous work [1], we used neural networks to iteratively construct a sequence of basis functions $\{\varphi_i^{NN} \in V_{n_i}^{\sigma_i},\ i \in \mathbb{N}\}$ that were used to define a Galerkin scheme for (3) based on an initial approximation $u_0 \in X$ and the functions φ_i^{NN} as follows:

$$u_i \in S_i := \text{span}\{u_0, \varphi_1^{NN}, \ldots, \varphi_i^{NN}\} \ :\ a(u_i, v) = L(v) \quad \forall v \in S_i. \tag{4}$$

Céa's Lemma [12] provides the following error estimate for u_i:

$$|||u - u_i|||_a \leqslant \sqrt{\frac{M}{\alpha}} |||u - v|||_a \quad \forall v \in S_i, \tag{5}$$

which means that the approximation u_i is, up to a constant, the best possible approximation from the subspace S_i.

The basis functions φ_i^{NN} are constructed iteratively as follows. Given u_{i-1}, $i \geqslant 1$, the weak residual $r(u_{i-1}) : X \to \mathbb{R}$ is defined by

$$\langle r(u_{i-1}), v\rangle = L(v) - a(u_{i-1}, v), \quad v \in X, \tag{6}$$

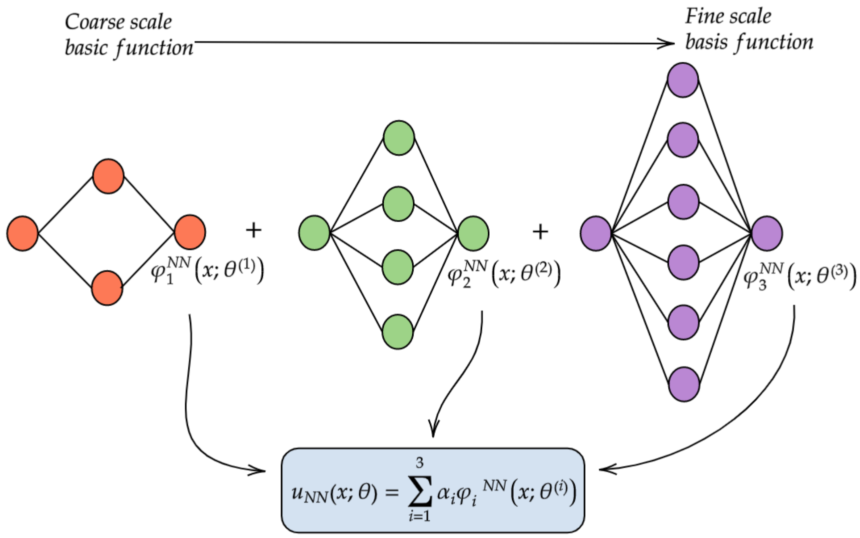


Figure 2: A visualization of the Galerkin Neural Network framework using three shallow networks consisting of two, four, and six neurons each.

where $\langle r(u_{i-1}), v\rangle$ denotes the duality pairing on X. Using the weak residual as the loss function, the ith basis function φ_i^{NN} is constructed by training a neural network to find the maximizer as follows:

$$\varphi_i^{NN} \in V_{n_i,C_i}^{\sigma} : \langle r(u_{i-1}), \varphi_i^{NN}\rangle = \max_{v\in V_{n_i,C_i}^{\sigma}\cap B} \langle r(u_{i-1}), v\rangle, \tag{7}$$

where B is the closed unit ball in X. Any standard training procedure may be used to solve (7); we use the approach given in Algorithm 2 of [1].

The residual may be rewritten using (3) as

$$\langle r(u_{i-1}), v\rangle = a(u, v) - a(u_{i-1}, v) = a(u - u_{i-1}, v), \quad v \in X$$

which, thanks to the Cauchy-Schwarz inequality, is maximized when $v \propto u - u_{i-1}$. This means that the basis function $\varphi_i^{NN} \in V_{n_i,C_i}^{\sigma} \cap B$ is an approximation to the normalized error $(u-u_{i-1})/|||u-u_{i-1}|||$. Consequently, we may view $\{\varphi_i^{NN}\}$ as a sequence of increasingly fine scale corrections to the initial approximation u_0. A pictorial representation of the Galerkin Neural Network scheme is provided in Figure 2.

The basic properties of the Galerkin Neural Network scheme are summarized in the following theorem [1].

Theorem 2.1. *For $i \geqslant 1$, let $\tau_i \in (0, 1)$ and u_i be defined as in (4). Then there exist network widths $n_i = n(\tau_i, u_{i-1})$ and bounds $C_i = C(\tau_i, u_{i-1})$ depending on τ_i and u_{i-1} such that*

$$|||u - u_i|||_a \leqslant |||u - u_0|||_a \cdot \Pi_{j=1}^{i} 2\tau_j/(1 - \tau_j) \tag{8}$$

Moreover, if $0 < \tau_i < 1/3$ then

$$\frac{1-\tau_i}{1+\tau_i}\eta_i \leqslant |||u - u_{i-1}|||_a \leqslant \frac{1-\tau_i}{1-3\tau_i}\eta_i \tag{9}$$

where $\eta_i := \langle r(u_{i-1}), \varphi_i^{NN}\rangle$.

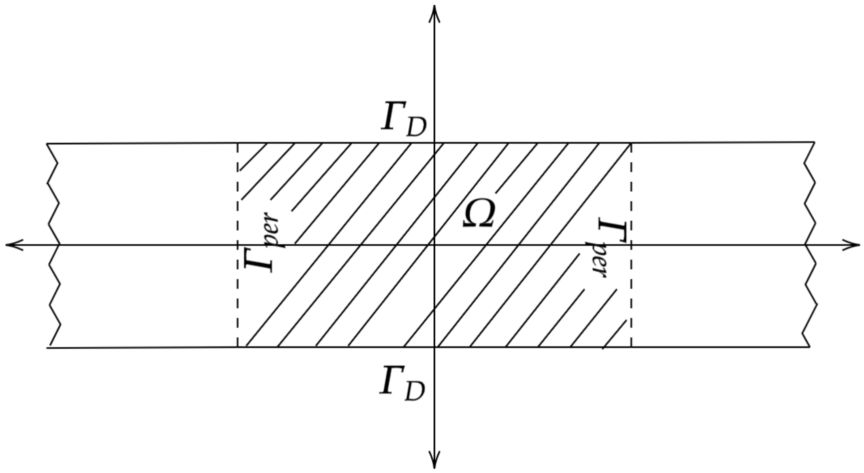


Figure 3: Domain for the test problem in Section 3.

The estimate (9) shows that the (fully computable) quantity η_i provides an a posteriori error estimator [2,37] for the true error $|||u - u_{i-1}|||_a$ which can be used as a stopping criterion (as in Algorithm 1).

3 Boundary Layer Resolution using Neural Networks

Physical problems involving small (or large) parameters often exhibit localized features on a length scale defined by the parameter. One such example is the Reissner-Mindlin plate model which describes the bending of a thin plate while taking into account shear deformation. Let $\boldsymbol{\beta}$ be the rotation of the fibers normal to the midplane of the plate and let ω be the transverse displacement of the midplane itself. The Reissner-Mindlin model takes the form of a system of elliptic PDEs:

$$\begin{cases} -\Delta\boldsymbol{\beta} + t^{-2}(\boldsymbol{\beta} - \nabla\omega) = 0 & \text{in } \Omega \\ t^{-2}\text{div}\,(\boldsymbol{\beta} - \nabla\omega) = g, & \text{in } \Omega \\ \boldsymbol{\beta} = \mathbf{0},\ \omega = 0 & \text{on } \partial\Omega. \end{cases} \tag{10}$$

Here, $\Omega \subset \mathbb{R}^2$ is the midplane of the plate, $t > 0$ is the plate thickness, g is the applied transverse load, and, for ease of exposition, we use a simplified stress-displacement law which nevertheless retains the essential character of the Reissner-Mindlin model. The boundary conditions correspond to a hard, simple support [5], which means that, in the limit where the plate thickness t tends to 0, the Reissner-Mindlin model gives rise to boundary layers which manifest most strongly in the shear stress [7] defined by

$$\boldsymbol{\sigma} := t^{-2}(\boldsymbol{\beta} - \nabla\omega). \tag{11}$$

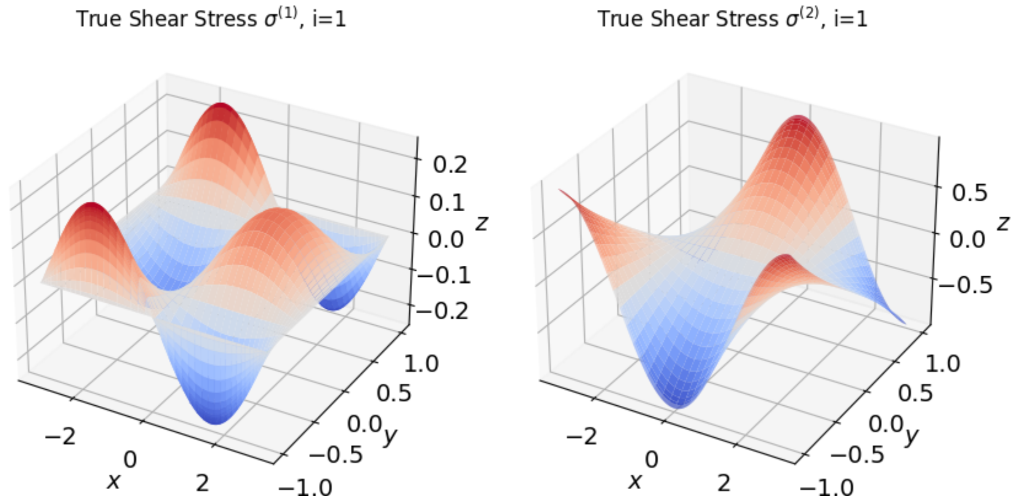


Figure 4: Left: True x-component of shear stress with $n = 1$ and $t = 10^{-2}$. Right: True y-component of the shear stress with $n = 1$ and $t = 10^{-2}$.

3.1 Benchmark Solution of Reissner-Mindlin Plate Exhibiting Boundary Layers

In order to illustrate the above points more concretely, we derive a class of closed form solutions of the Reissner-Mindlin plate model which will later be used to benchmark the performance of numerical schemes. Consider an infinite strip along the x-axis subject to a univariate 2π-periodic transverse load g. Due to periodicity, it suffices to consider a single period $\Omega = (-\pi, \pi) \times (-1, 1)$ as shown in Figure 3. We partition the boundary of Ω into disjoint sets $\Gamma_D = (-\pi, \pi) \times \{-1\} \cup \{1\}$ and $\Gamma_{\text{sym}} = \{-\pi\} \cup \{\pi\} \times (-1, 1)$ with homogeneous Dirichlet boundary conditions applied on Γ_D and periodic boundary conditions applied on Γ_{per}. The periodic transverse load $g(x)$ is written as a Fourier series given by

$$g(x) = \frac{1}{2} g_0 + \sum_{n=1}^{\infty} g_n \cos(nx) + \sum_{n=1}^{\infty} \tilde{g}_n \sin(nx).$$

For $n \in \mathbb{N}_0$, let $(\boldsymbol{\beta}_n, \omega_n)$ be given by

$$\begin{aligned}
\boldsymbol{\beta}_n(x, y) &= \begin{bmatrix} -(\Phi_n'(y,t) + n \cdot \Psi_n(y,t) - n \cdot \Upsilon_n(y,t)) \sin(nx) \\ (n \cdot \Phi_n(y,t) + \Psi_n'(y,t) - \Upsilon_n'(y,t)) \cos(nx) \end{bmatrix} \\
\omega_n(x, y) &= [(1 + t^2)\Psi_n(y,t) - \Upsilon_n(y,t)] \cos(nx),
\end{aligned} \tag{12}$$

where Φ_n, Ψ_n, and Υ_n are given by

$$\Phi_n(y,t) = \frac{A_n(t) t \sinh(\lambda_n y)}{\sinh(\lambda_n)}, \quad \Psi_n(y,t) = \begin{cases} -y^2/2 + D_0(t), & n = 0 \\ n^{-2} - D_n(t) \cdot \dfrac{\cosh(ny)}{\cosh(n)}, & n > 0 \end{cases} \tag{13}$$

$$\Upsilon_n(y,t) = \begin{cases} -y^2/2 - y^4/24 + B_0(t) y^2, & n = 0 \\ \dfrac{B_n(t) y \cdot \sinh(ny)}{\sinh n} - (n^{-4} - n^{-2}) + \dfrac{C_n(t) \cdot \cosh(ny)}{\cosh n}, & n > 0, \end{cases}$$

with $A_n(t)$, $B_n(t)$, $C_n(t)$, and $D_n(t)$ coefficients depending on n and t, and $\lambda_n := (n^2 + 1/t^2)^{1/2}$. The values of $A_n(t)$, $B_n(t)$, $C_n(t)$, and $D_n(t)$ are determined by the boundary conditions and have series expansions valid for small t given by

$$\begin{aligned} A_n(t) &= \left(\frac{2}{n} - 4\xi_n\right) t^2 + \mathcal{O}(t^3) \\ B_n(t) &= \xi_n \frac{\tanh(n)}{n}\left[-\frac{1}{n^2} + 2(-1+2\xi_n)t^2 + \mathcal{O}(t^3)\right] \\ C_n(t) &= \frac{\xi_n}{n^2}\left[\frac{n\coth(n) - 2n^2 + 1}{n^2} + \left(2(\coth(n) - 2n^2 + 1) - 2\xi_n \tanh(n)\right) t^2 + \mathcal{O}(t^3)\right] \\ D_n(t) &= \xi_n \left[\frac{\tanh(n)}{n^2} + 4\left(1 - 2\xi_n\right) t^2 + \mathcal{O}(t^3)\right], \end{aligned} \tag{14}$$

where $\xi_n = \sinh(2n)/(2n + \sinh(2n))$. Full details leading to (14) will be found in the Appendix.

Straightforward manipulation reveals that the functions defined in (12) satisfy

$$\begin{cases} -\Delta \boldsymbol{\beta}_n + t^{-2}(\boldsymbol{\beta}_n - \nabla\omega_n) = 0 & \text{in } \Omega \\ t^{-2}\text{div}\,(\boldsymbol{\beta}_n - \nabla\omega_n) = \cos(nx), & \text{in } \Omega \\ \boldsymbol{\beta}_n = \mathbf{0},\ \omega_n = 0 & \text{on } \Gamma_D \\ \boldsymbol{\beta}_n, \omega_n \text{ periodic} & \text{on } \Gamma_{\text{per}}. \end{cases} \tag{15}$$

In turn, thanks to linearity and the translation identity $\sin(nx) = \cos(nx - \pi/2)$, the solution to (10) is given by

$$\boldsymbol{\beta}(x,y) = \frac{1}{2} g_0 \boldsymbol{\beta}_0(x,y) + \sum_{n=1}^{\infty} g_n \boldsymbol{\beta}_n(x,y) + \sum_{n=1}^{\infty} \tilde{g}_n \boldsymbol{\beta}_n(x - \frac{\pi}{2n}, y)$$

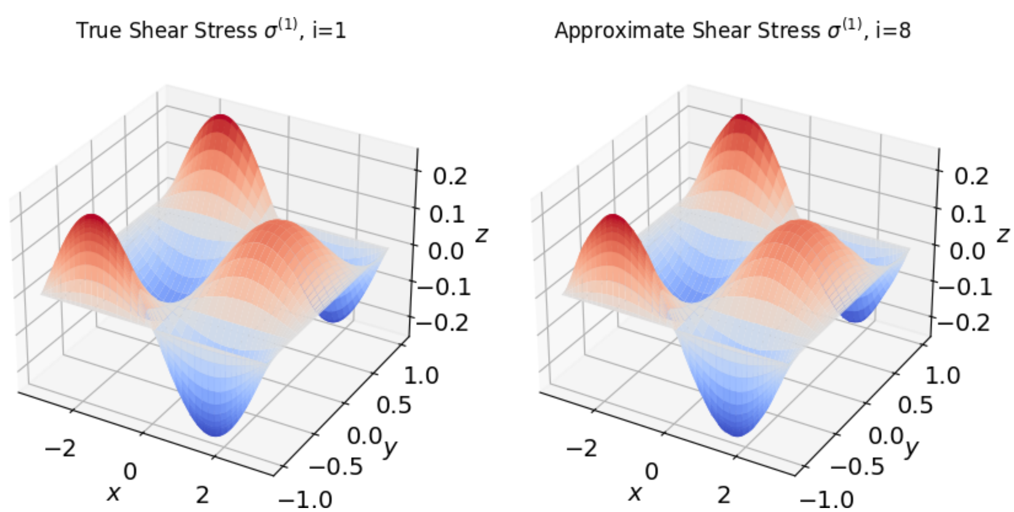


Figure 5: Left: True x-component of shear stress with $n = 1$ and $t = 10^{-2}$. Right: Neural network approximation of x-component of shear stress with $n = 1$ and $t = 10^{-2}$.

$$\omega(x,y) = \frac{1}{2} g_0 \omega_0(x,y) + \sum_{n=1}^{\infty} g_n \omega_n(x,y) + \sum_{n=1}^{\infty} \tilde{g}_n \omega_n(x - \frac{\pi}{2n}, y).$$

This solution was given in [3] in the particular case when $n = 1$; here, we generalize to any $n \in \mathbb{N}_0$. The shear stress $\boldsymbol{\sigma}_n = t^{-2}(\boldsymbol{\beta}_n - \nabla\omega_n)$ associated with the nth Fourier mode is given by

$$\boldsymbol{\sigma}_n(x,y) = \begin{bmatrix} [-t^{-2}\Phi_n'(y,t) + n \cdot \Psi_n(y,t)] \sin(nx) \\ [nt^{-2} \cdot \Phi_n(y,t) - \Psi_n'(y,t)] \cos(nx) \end{bmatrix}, \tag{16}$$

which, for $t \ll 1$, exhibits a boundary layer on the upper and lower edges of the plate:

$$\boldsymbol{\sigma}_n(x,y) \sim \frac{A_n(t)}{t^2 \sinh \lambda_n} \begin{bmatrix} -\lambda_n \cosh(\lambda_n y) \sin(nx) \\ n \sinh(\lambda_n y) \cos(nx) \end{bmatrix},$$

or, thanks to (14):

$$\boldsymbol{\sigma}_n(x,y) \sim \left(\frac{1}{n} - \frac{2\sinh(2n)}{2n + \sinh(2n)} + \mathcal{O}(t^3) \right) \frac{2}{\sinh(\lambda_n)} \begin{bmatrix} -\lambda_n \cosh(\lambda_n y) \sin(nx) \\ n \sinh(\lambda_n y) \cos(nx) \end{bmatrix}.$$

Figure 4 shows the shear stress in the case $n = 1$ and $t = 10^{-2}$. The factors $\cosh(\lambda_n y)$ and $\sinh(\lambda_n y)$ each give rise to boundary layers in both components of the shear stress, which is more pronounced in the x-component of $\boldsymbol{\sigma}_n$ due to the presence of the factor $\lambda_n \gg n$ when $t \ll 1$.

3.2 Function Fitting of Reissner-Mindlin Shear Stress

We are interested in the approximability of the boundary layer in the shear stress $\boldsymbol{\sigma}$ using neural networks. In order to isolate the issue of approximability, we utilize Algorithm 1 in order to "learn" the x-component of the shear stress, $\boldsymbol{\sigma}^{(1)}$, with $n = 1$. The bilinear operator is given by $a(u,v) = (u,v)_\Omega + (\nabla u, \nabla v)_\Omega + \varepsilon^{-1}(u,v)_{\partial\Omega}$ and the data is given by $L(v) = (\boldsymbol{\sigma}^{(1)}, v)_\Omega + (\nabla\boldsymbol{\sigma}^{(1)}, \nabla v)_\Omega$. The training data consists of 128×128 Gauss-Legendre quadrature points. The width of the network for each basis function is $n_i = 20 \cdot 2^{i-1}$ while the activation function for each basis function is $\sigma_i(z) = \tanh((1+0.25i)z)$. The hyperplanes of the network for each basis function are initialized so that they are parallel to the x-axis, the y-axis, $y = x$, or $y = -x$ (see [1], §3.2.2).

Figure 5 shows the neural network approximation to the shear stress, from which it is apparent that the sequence of networks is capable of approximating the boundary layer. These results demonstrate that the Galerkin Neural Network procedure is capable of resolving boundary layers without issue when applied to a simple function fitting of the shear stress $\boldsymbol{\sigma}_n$.

4 Variational Formulations of the Reissner-Mindlin Model

Encouraged by the results of Section 3 showing the capability of neural networks to approximate the shear stress, we now turn to the question of which choice of variational formulation

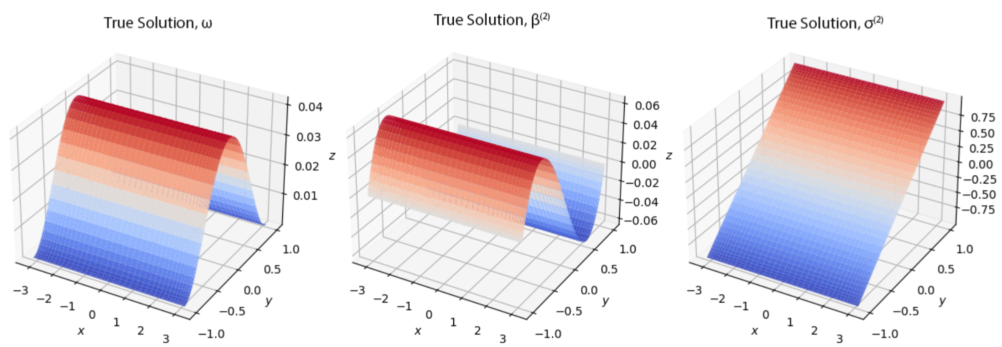


Figure 6: True solution ω (left), $\boldsymbol{\beta}$ (middle), and $\boldsymbol{\sigma}$ (right) with $t = 10^{-6}$ and $n = 0$.

for the Reissner-Mindlin plate model to use in conjunction with neural networks. Ideally, we seek a variational formulation for which the associated bilinear form is continuous and coercive with constants M and α (see [12]) that are independent of t which, thanks to Céa's Lemma, means that by establishing control of the ratio M/α, we have a quasi-optimal solution. To this end, we present and analyze several variational formulations and provide numerical results demonstrating their efficacy or lack thereof.

In all numerical examples that follow, unless otherwise stated, the neural network architecture for the ith Galerkin Neural Network basis function consists of a single hidden layer of width $n_i = 20 \cdot 2^{i-1}$. The activation function for the ith basis function is $\sigma_i(z) = \tanh((1 + 0.25i)z)$. The weights and biases are initialized so that the hyperplanes $x \cdot W_j + b_j$ of the hidden layer are either parallel to the x-axis, y-axis, $y = x$, or $y = -x$ as described in [1]. The training data consists of 128×128 Gauss-Legendre quadrature nodes, while the validation data used to compute the loss function and true errors where such a computation is possible consists of 150×150 Gauss-Legendre quadrature nodes.

4.1 Natural Variational Formulation

The natural variational formulation of (10) is to seek $(\boldsymbol{\beta}, \omega) \in X := \mathbf{H_0^1}(\Omega) \times H_0^1(\Omega)$, where X is equipped with the norm $||(\boldsymbol{\beta}, \omega)||_X := (||\boldsymbol{\beta}||^2_{H^1(\Omega)} + ||\omega||^2_{H^1(\Omega)})^{1/2}$, such that

$$\mathfrak{B}_0((\boldsymbol{\beta}, \omega); (\boldsymbol{\varphi}, v)) := (\nabla\boldsymbol{\beta}, \nabla\boldsymbol{\varphi})_\Omega + t^{-2}(\boldsymbol{\beta} - \nabla\omega, \boldsymbol{\varphi} - \nabla v)_\Omega = (g, v) =: \mathfrak{L}_0(\boldsymbol{\varphi}, v) \qquad (17)$$

for all $(\boldsymbol{\varphi}, v) \in X$. The operator $\mathfrak{B}_0$ is symmetric and positive-definite as well as continuous and coercive according to the following result:

Proposition 4.1. *Let* $(\boldsymbol{\beta}, \omega) \in X$ *and* $0 < t \leqslant 1$. *There exist constants* $C_1 > 0$ *and* $C_2 > 0$ *such that*

$$\begin{aligned} \mathfrak{B}_0((\boldsymbol{\beta}, \omega); (\boldsymbol{\varphi}, v)) &\leqslant C_1 t^{-2} ||(\boldsymbol{\beta}, \omega)||_X \cdot ||(\boldsymbol{\varphi}, v)||_X \quad && \forall (\boldsymbol{\beta}, \omega), (\boldsymbol{\varphi}, v) \in X \\ C_2 ||(\boldsymbol{\beta}, \omega)||_X^2 &\leqslant \mathfrak{B}_0((\boldsymbol{\beta}, \omega); (\boldsymbol{\beta}, \omega)) && \forall (\boldsymbol{\beta}, \omega) \in X. \end{aligned}$$

Proof. First, we apply the Cauchy-Schwarz inequality with respect to L^2 to obtain

$$\mathfrak{B}_0((\boldsymbol{\beta}, \omega); (\boldsymbol{\varphi}, v)) \leqslant ||\nabla\boldsymbol{\beta}||_\Omega ||\nabla\boldsymbol{\varphi}||_\Omega + t^{-2} ||\boldsymbol{\beta} - \nabla\omega||_\Omega ||\boldsymbol{\varphi} - \nabla v||_\Omega$$

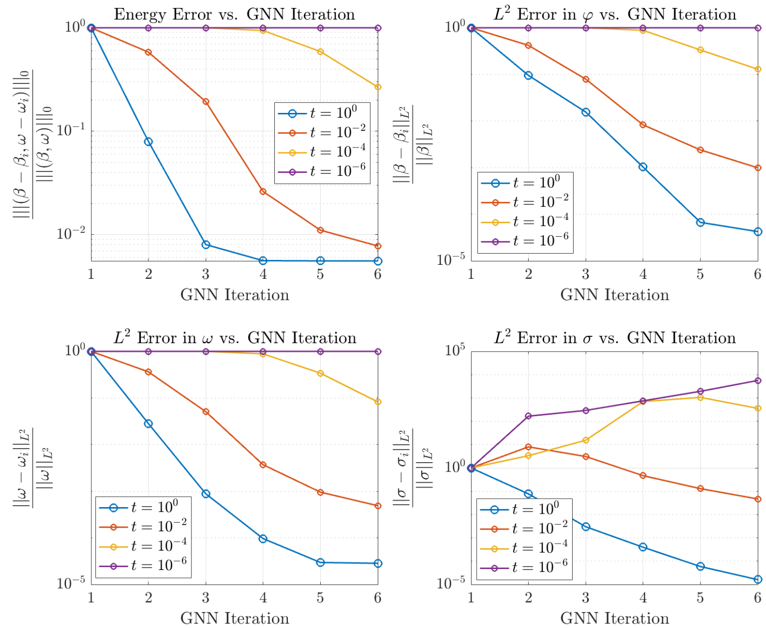


Figure 7: Relative error after each iteration of Algorithm 1 for the case when $n = 0$ with variational formulation on $\mathfrak{B}_0, \mathfrak{L}_0$.

$$\leqslant ||\nabla \boldsymbol{\beta}||_\Omega ||\nabla \boldsymbol{\varphi}||_\Omega + t^{-2}(||\boldsymbol{\beta}||_{H^1(\Omega)} + ||\omega||_{H^1(\Omega)})(||\boldsymbol{\varphi}||_{H^1(\Omega)} + ||v||_{H^1(\Omega)}).$$

Applying the Cauchy-Schwarz inequality with respect to ℓ^2 as well as Young's inequality yields

$$\begin{aligned}
\mathfrak{B}_0((\boldsymbol{\beta}, \omega); (\boldsymbol{\varphi}, v)) &\leqslant [||\boldsymbol{\beta}||^2_{H^1(\Omega)} + 2t^{-2}||\boldsymbol{\beta}||^2_{H^1(\Omega)} + 2t^{-2}||\omega||^2_{H^1(\Omega)}]^{1/2} \\
&\quad \cdot [||\boldsymbol{\varphi}||^2_{H^1(\Omega)} + 2t^{-2}||\boldsymbol{\varphi}||^2_{H^1(\Omega)} + 2t^{-2}||v||^2_{H^1(\Omega)}]^{1/2} \\
&\leqslant C_1 t^{-2}(||\boldsymbol{\beta}||^2_{H^1(\Omega)} + ||\omega||^2_{H^1(\Omega)})^{1/2}(||\boldsymbol{\varphi}||^2_{H^1(\Omega)} + ||v||^2_{H^1(\Omega)})^{1/2}
\end{aligned}$$

where $C_1 = 3$. Similarly, applying Poincare's inequality to $\boldsymbol{\beta}$ and ω and the triangle inequality yields

$$\begin{aligned}
||\boldsymbol{\beta}||^2_{H^1(\Omega)} + ||\omega||^2_{H^1(\Omega)} &\leqslant C_p||\nabla \boldsymbol{\beta}||^2_\Omega + C_p||\nabla \omega||^2_\Omega \leqslant C_2(||\nabla \boldsymbol{\beta}||^2_\Omega + ||\boldsymbol{\beta} - \nabla \omega||^2_\Omega) \\
&\leqslant C_2(||\nabla \boldsymbol{\beta}||^2_\Omega + t^{-2}||\boldsymbol{\beta} - \nabla \omega||^2_\Omega).
\end{aligned}$$

□

Proposition 4.1 means that Algorithm 1 is applicable to (17). However, we note that the ratio M/α is unbounded as $t \to 0$, which suggests that the approximation to the variational problem may not be quasi-optimal. In order to illiustrate this, we begin by considering the simplest case of (15) in which the load $g(x, y) = 1$. This problem corresponds to (15) with $n = 0$. In this case, the true solution is univariate and does not vary in the x-coordinate, nor is it sensitive to the value of t. However, even though the solution is univariate in y, we do not explicitly enforce this in the neural network structure. Figure 6 shows the y-components of the true solutions ω and $\boldsymbol{\beta}$, and $\boldsymbol{\sigma}$ when $t = 10^{-6}$.

Figure 7 shows a convergence plot of the error in the energy norm ($\mathfrak{B}_0$-norm) after each iteration of Algorithm 1 as well as the L^2 error in the stress $\boldsymbol{\sigma} = t^{-2}(\boldsymbol{\beta} - \nabla\omega)$ and rotation and displacement $\boldsymbol{\beta}$ and ω for $t = 1, 10^{-2}, 10^{-4}, 10^{-6}$. We denote by $\boldsymbol{\varphi}_i^{NN}$ the basis functions for approximating $\boldsymbol{\beta}$ and by v_i^{NN} the basis functions for approximating ω. Figure 8 shows the approximate shear stress obtained by Algorithm 1 by computing $t^{-2}(\varphi_i - \nabla\omega_i)$ after the 2nd, 4th, 6th, and 8th iterations with $t = 10^{-6}$. The hyperparameters are as given at the beginning of Section 4.

Since the Galerkin Neural Network framework is not a direct Galerkin method, a natural question to ask is whether it exhibits locking. As the results show, it is immediately evident that as t decreases, the convergence rate of both the energy error and the L^2 errors are greatly reduced. Even worse, while the true shear stress is a simple linear function, the neural network approximation exhibits large spurious oscillations which dampen slowly as more iterations of Algorithm 1 are taken. This stalled convergence as t is decreased is reminiscent of the locking phenomenon observed in finite element approximation [8, 36] and suggests that the natural variational formulation (17), like with the finite element method, is not suitable for the Galerkin Neural Network framework.

4.2 Mixed Least Squares Variational Formulation

One common approach to help reduce, or even eliminate, the effects of locking consists of introducing the shear stress $\boldsymbol{\sigma}$ as a primal variable [4]. That is, the system

$$\begin{cases} -\Delta\boldsymbol{\beta} + \boldsymbol{\sigma} = \mathbf{0} & \text{in } \Omega \\ \operatorname{div} \boldsymbol{\sigma} = g & \text{in } \Omega \\ t^2\boldsymbol{\sigma} - (\boldsymbol{\beta} - \nabla\omega) = \mathbf{0} & \text{in } \Omega \\ \boldsymbol{\beta} = \mathbf{0},\ \omega = 0 & \text{on } \partial\Omega \end{cases} \tag{18}$$

is considered in lieu of (10). One disadvantage of this approach is that the resulting mixed

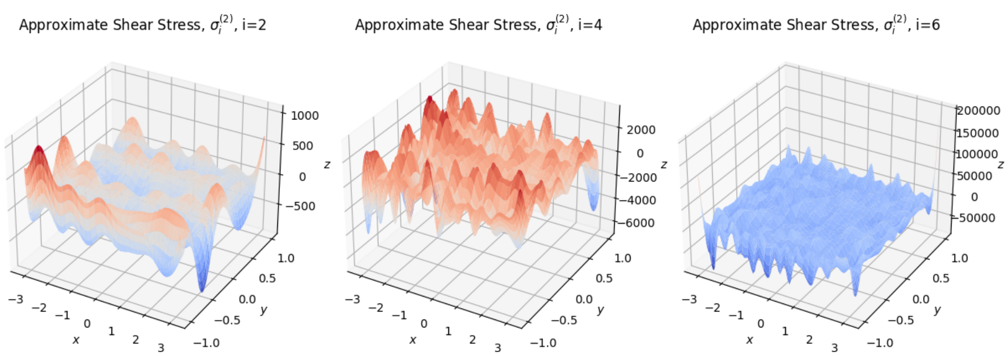


Figure 8: Approximate x-component of the shear stress for the 2nd, 4th, and 6th iterations of the Galerkin Neural Network algorithm for the univariate problem with variational formulation on $\mathfrak{B}_0, \mathfrak{L}_0$.

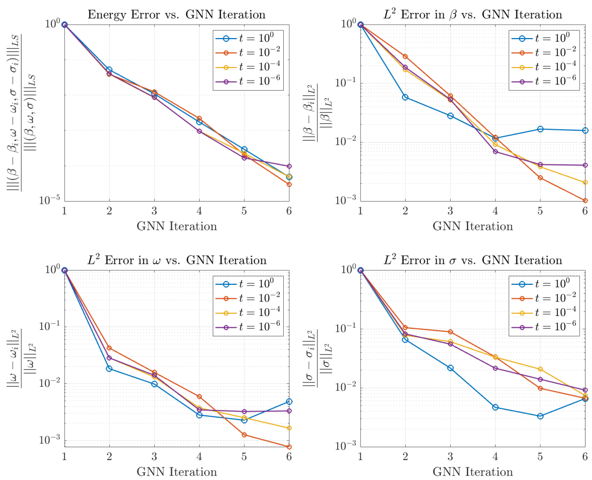


Figure 9: Relative errors after each iteration of Algorithm 1 for the univariate problem with variational formulation on $\mathfrak{B}_{LS}, \mathfrak{L}_{LS}$.

variational formulation corresponds to the bilinear form

$$(\nabla\boldsymbol{\beta}, \nabla\boldsymbol{\varphi})_\Omega + (\boldsymbol{\sigma}, \boldsymbol{\varphi} - \nabla v)_\Omega + (\boldsymbol{\beta} - \nabla\omega, \boldsymbol{\tau})_\Omega - t^2(\boldsymbol{\sigma}, \boldsymbol{\tau})_\Omega$$

which, although symmetric, is not positive definite (e.g. take $\boldsymbol{\beta} = \boldsymbol{\varphi} = \mathbf{0}$, $\omega = v = 0$, and $\boldsymbol{\sigma} = \boldsymbol{\tau} \neq \mathbf{0}$), so Algorithm 1 is not applicable. Instead, we shall consider an alternative mixed formulation based on least squares variational principles. Setting $X_{LS} := \mathbf{H}^1_0(\Delta;\Omega) \times H^1_0(\Omega) \times \mathbf{H}(\mathrm{div};\Omega)$ where $\mathbf{H}^1_0(\Delta;\Omega) := \{\mathbf{v} \in \mathbf{H}^1_0(\Omega) : \Delta\mathbf{v} \in \mathbf{L}^2(\Omega)\}$ with norm

$$||(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma})||_{LS} := (||\boldsymbol{\beta}||^2_{H^1(\Omega)} + ||t\Delta\boldsymbol{\beta}||^2_\Omega + ||\omega||^2_{H^1(\Omega)} + ||t\boldsymbol{\sigma}||^2_\Omega + ||\nabla\cdot\boldsymbol{\sigma}||^2_\Omega)^{1/2},$$

we define the bilinear operator $\mathfrak{B}_{LS} : X_{LS} \times X_{LS} \to \mathbb{R}$ given by

$$\begin{aligned}\mathfrak{B}_{LS}((\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}); (\boldsymbol{\varphi}, v, \boldsymbol{\tau})) := t^2(-\Delta\boldsymbol{\beta} + \boldsymbol{\sigma}, -\Delta\boldsymbol{\varphi} + \boldsymbol{\tau})_\Omega + (\mathrm{div}\,\boldsymbol{\sigma}, \mathrm{div}\,\boldsymbol{\tau})_\Omega \\ + (t^2\boldsymbol{\sigma} - (\boldsymbol{\beta} - \nabla\omega), t^2\boldsymbol{\tau} - (\boldsymbol{\varphi} - \nabla v))_\Omega \end{aligned} \tag{19}$$

and the linear operator $\mathfrak{L}_{LS} : X_{LS} \to \mathbb{R}$ given by

$$\mathfrak{L}_{LS}(\boldsymbol{\varphi}, v, \boldsymbol{\tau}) := (g, \mathrm{div}\,\boldsymbol{\tau})_\Omega. \tag{20}$$

The bilinear operator $\mathfrak{B}_{LS}$ is both continuous and coercive, allowing us to apply Algorithm 1. However, we note that the coercivity estimate is again degenerate in t, which suggests we might expect deterioration of the numerical approximation as the plate thickness is reduced.

Proposition 4.2. *Let $(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}) \in X_{LS}$ and $0 < t \leqslant 1$. There exist constants $C_1 > 0$ and $C_2 > 0$ independent of t such that*

$$\begin{aligned}\mathfrak{B}_{LS}((\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}); (\boldsymbol{\varphi}, v, \boldsymbol{\tau})) &\leqslant C_1||(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma})||_{LS} \cdot ||(\boldsymbol{\varphi}, v, \boldsymbol{\tau})||_{LS} \quad \forall(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}), (\boldsymbol{\varphi}, v, \boldsymbol{\tau}) \in X_{LS} \\ C_2 t^2||(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma})||^2_{LS} &\leqslant \mathfrak{B}_{LS}((\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}); (\boldsymbol{\beta}, \omega, \boldsymbol{\sigma})) \quad \forall(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}) \in X_{LS}.\end{aligned}$$

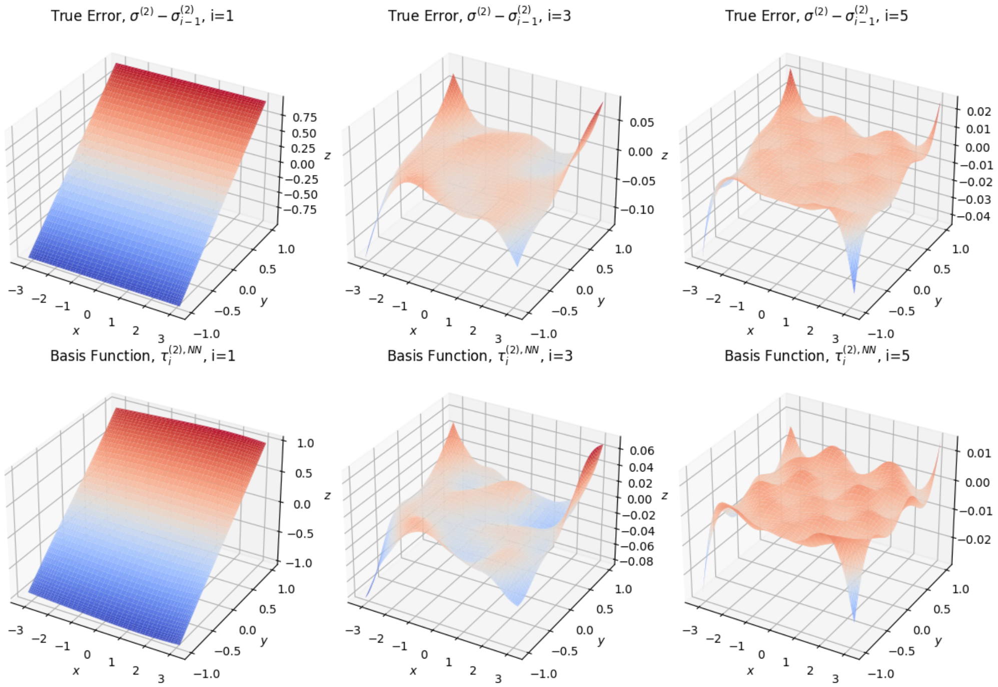


Figure 10: For $t = 10^{-6}$, the y-component of the true error $\boldsymbol{\sigma} - \boldsymbol{\sigma}_i$ (top row) and the y-component of the basis function $\boldsymbol{\tau}_i^{NN}$ (bottom row) for the 1st, 3rd, and 5th iterations of Algorithm 1 for the univariate problem with variational formulation on $\mathfrak{B}_{LS}, \mathfrak{L}_{LS}$.

Proof. For the first inequality, we apply the Cauchy-Schwarz inequality with respect to L^2 to obtain

$$\begin{aligned}
\mathfrak{B}_{LS}((\boldsymbol{\beta},\omega,\boldsymbol{\sigma});(\boldsymbol{\varphi},v,\boldsymbol{\tau})) &\leqslant ||t(-\Delta\boldsymbol{\beta}+\boldsymbol{\sigma})||_\Omega \cdot ||t(-\Delta\boldsymbol{\varphi}+\boldsymbol{\tau})||_\Omega + ||\nabla\cdot\boldsymbol{\sigma}||_\Omega \cdot ||\nabla\cdot\boldsymbol{\tau}||_\Omega \\
&\quad + ||t^2\boldsymbol{\sigma} - (\boldsymbol{\beta}-\nabla\omega)||_\Omega \cdot ||t^2\boldsymbol{\tau} - (\boldsymbol{\varphi}-\nabla v)||_\Omega \\
&\leqslant (||t\Delta\boldsymbol{\beta}||_\Omega + ||t\boldsymbol{\sigma}||_\Omega)\cdot(||t\Delta\boldsymbol{\varphi}||_\Omega + ||t\boldsymbol{\tau}||_\Omega) + ||\nabla\cdot\boldsymbol{\sigma}||_\Omega \cdot ||\nabla\cdot\boldsymbol{\tau}||_\Omega \\
&\quad + (||t\boldsymbol{\sigma}||_\Omega + ||\boldsymbol{\beta}||_{H^1(\Omega)} + ||\omega||_{H^1(\Omega)})\cdot \\
&\qquad (||t\boldsymbol{\tau}||_\Omega + ||\boldsymbol{\varphi}||_{H^1(\Omega)} + ||v||_{H^1(\Omega)}).
\end{aligned}$$

An application of the Cauchy-Schwarz inequality with respect to ℓ^2 yields the result.

As for the second inequality, given $(\boldsymbol{\beta},\omega,\boldsymbol{\sigma}) \in X_{LS}$, we form the system

$$\begin{cases}
\mathbf{g}_1 := -\Delta\boldsymbol{\beta} + \boldsymbol{\sigma} & \text{in } \Omega \\
g_2 := \nabla\cdot\boldsymbol{\sigma} & \text{in } \Omega \\
t^2\mathbf{g}_3 := t^2\boldsymbol{\sigma} - (\boldsymbol{\beta}-\nabla\omega) & \text{in } \Omega \\
\boldsymbol{\beta} = \mathbf{0},\ \omega = 0 & \text{on } \partial\Omega.
\end{cases} \tag{21}$$

We can form the following variational formulation: $(\boldsymbol{\beta},\omega,\boldsymbol{\sigma}) \in X_{LS}$ s.t.

$$\begin{cases}
(\nabla\boldsymbol{\beta},\nabla\boldsymbol{\varphi})_\Omega + (\boldsymbol{\sigma},\boldsymbol{\varphi})_\Omega & = (\mathbf{g}_1,\boldsymbol{\varphi})_\Omega \quad \forall\boldsymbol{\varphi}\in X_{\boldsymbol{\beta}} \\
(\nabla\cdot\boldsymbol{\sigma}, v)_\Omega & = (g_2, v)_\Omega \quad \forall v \in X_\omega \\
t^2(\boldsymbol{\sigma},\boldsymbol{\tau})_\Omega - (\boldsymbol{\beta},\boldsymbol{\tau})_\Omega + (\nabla\omega,\boldsymbol{\tau})_\Omega & = t^2(\mathbf{g}_3,\boldsymbol{\tau})_\Omega \quad \forall\boldsymbol{\tau}\in X_{\boldsymbol{\sigma}}.
\end{cases} \tag{22}$$

Setting $\boldsymbol{\varphi} = \boldsymbol{\beta}$, $v = \omega$, and $\boldsymbol{\tau} = \boldsymbol{\sigma}$, integrating the third equation of (22) by parts, and adding all three equations together yields

$$||\nabla\boldsymbol{\beta}||_\Omega^2 + t^2||\boldsymbol{\sigma}||_\Omega^2 \leqslant ||\mathbf{g}_1||_\Omega \cdot ||\boldsymbol{\beta}||_\Omega + ||g_2||_\Omega \cdot ||\omega||_\Omega + t||\mathbf{g}_3||_\Omega \cdot t||\boldsymbol{\sigma}||_\Omega. \tag{23}$$

For the second term on the RHS of (23), we observe by the Poincare inequality that

$$||\omega||_\Omega \leqslant C||\nabla\omega||_\Omega \leqslant C||t^2\mathbf{g}_3 - t^2\boldsymbol{\sigma} + \boldsymbol{\beta}||_\Omega \leqslant C(t^2||\mathbf{g}_3||_\Omega + t^2||\boldsymbol{\sigma}||_\Omega + ||\boldsymbol{\beta}||_\Omega),$$

from which we obtain

$$\begin{aligned}||\nabla\boldsymbol{\beta}||_\Omega^2 + t^2||\boldsymbol{\sigma}||_\Omega^2 \leqslant C(||\mathbf{g}_1||_\Omega + ||g_2||_\Omega) \cdot ||\boldsymbol{\beta}||_\Omega + C||g_2||_\Omega \cdot t^2||\mathbf{g}_3||_\Omega \\ + C(t||g_2||_\Omega + t||\mathbf{g}_3||_\Omega) \cdot t||\boldsymbol{\sigma}||_\Omega.\end{aligned}$$

Now, the first term on the RHS can be dealt with using Poincare's inequality and the ϵ-Young's inequality while the second and third terms on the RHS can be dealt with using the ϵ-Young's inequality:

$$||\nabla\boldsymbol{\beta}||_\Omega^2 + t^2||\boldsymbol{\sigma}||_\Omega^2 + ||\nabla\omega||_\Omega^2 \leqslant C(||\mathbf{g}_1||_\Omega^2 + ||g_2||_\Omega^2 + ||t^2\mathbf{g}_3||_\Omega^2).$$

As for $||\nabla \cdot \boldsymbol{\beta}||_\Omega$, we have $||\nabla \cdot \boldsymbol{\beta}||_\Omega = ||g_2||_\Omega$. Finally, for the second-order term, we consider

$$-t\Delta\boldsymbol{\beta} + t\boldsymbol{\sigma} = t\mathbf{g}_1$$

to obtain $||t\Delta\boldsymbol{\beta}||_\Omega \leqslant ||t\boldsymbol{\sigma}||_\Omega + ||t\mathbf{g}_1||_\Omega$. Altogether, we have

$$\begin{aligned}||(\boldsymbol{\beta}, \omega, \boldsymbol{\sigma})||_{LS}^2 &\leqslant C(||\mathbf{g}_1||_\Omega^2 + ||t\mathbf{g}_1||_\Omega^2 + ||g_2||_\Omega^2 + ||t^2\mathbf{g}_3||_\Omega^2) \\ &\leqslant Ct^{-2}(||t\mathbf{g}_1||_\Omega^2 + ||g_2||_\Omega^2 + ||t^2\mathbf{g}_3||_\Omega^2) \leqslant Ct^{-2} \cdot \mathfrak{B}_{LS}((\boldsymbol{\beta}, \omega, \boldsymbol{\sigma}); (\boldsymbol{\beta}, \omega, \boldsymbol{\sigma})).\end{aligned}$$

□

While the least squares formulation based on (18) seems a natural choice, hitherto it has not been employed in practice. One reason is that a direct application of the least squares functional to second-order problems requires that the approximation space be a conforming subspace of H^2, i.e. continuously differentiable elements are required when using finite element methods, which is often viewed as unattractive by finite element practitioners. To circumvent this issue, several approaches exist in the literature. The first is to reduce second order problems to first order problems by introducing auxiliary variables, which increases the size of the corresponding linear system. The second more sophisticated approach is to recast the least squares formulation in terms of a negative norm (i.e. H^{-1}) residual, which in the context of finite elements allows one to retain the advantages of least squares formulations while only requiring continuous basis functions, as in [11, 16].

One advantage of applying the Galerkin Neural Network framework to the H^2 least squares formulation is that the regularity and global nature of functions in the set $V_{n,C}^\sigma$ is determined solely by the regularity of the activation function σ. In other words, choosing $\sigma \in C^2(\bar{\Omega})$ is sufficient to ensure that the resulting neural network functions are H^2-conforming.

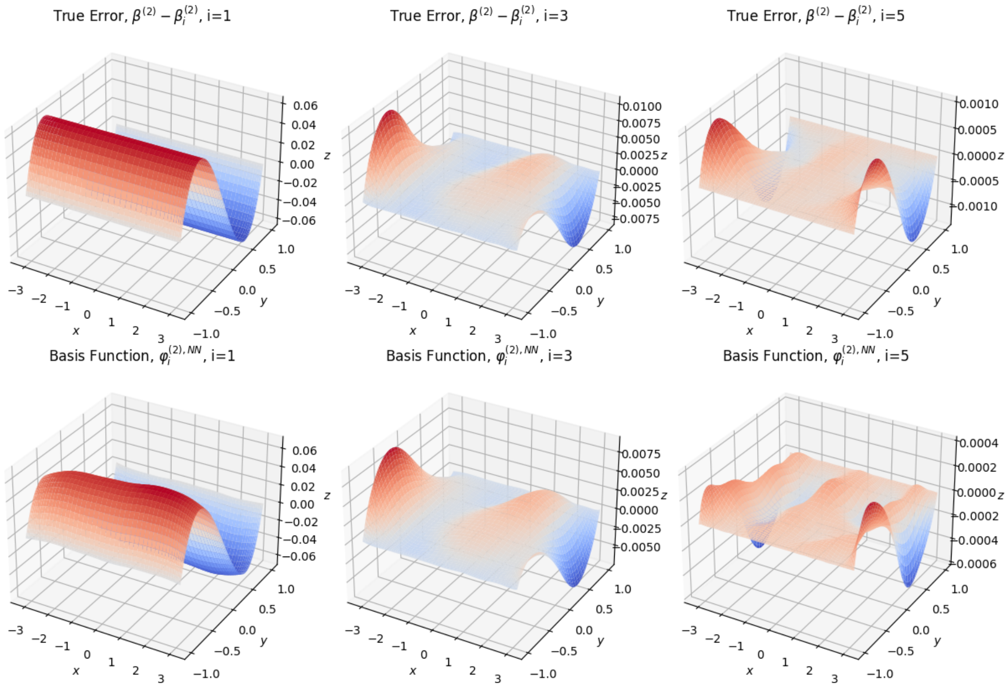


Figure 11: For $t = 10^{-6}$, the y-component of the true error $\boldsymbol{\beta} - \boldsymbol{\beta}_i$ (top row) and the y-component of the basis function $\boldsymbol{\varphi}_i^{NN}$ (bottom row) for the 1st, 3rd, and 5th iterations of Algorithm 1 for the univariate problem with variational formulation on $\mathfrak{B}_{LS}, \mathfrak{L}_{LS}$.

4.2.1 Benchmark Problem with Constant Load

We return to the constant load univariate problem described in Section 4.1, this time applying Algorithm 1 to the variational formulation involving $\mathfrak{B}_{LS}$ and $\mathfrak{L}_{LS}$. The hyperparameters are chosen as described at the beginning of Section 4.

Figures 9-12 show the analogous results to Figures 7-8 for the least squares approach described by (19). We denote by $\boldsymbol{\varphi}_i^{NN}$, v_i^{NN}, and $\boldsymbol{\tau}_i^{NN}$ the basis functions used to approximate $\boldsymbol{\beta}$, ω, and $\boldsymbol{\sigma}$, respectively. Figures 11-12 show comparisons of the basis functions $\boldsymbol{\varphi}_i^{NN}$ and v_i^{NN} with the errors in $\boldsymbol{\beta}$ and ω, respectively. We observe that no locking effect nor any spurious oscillations are present in the shear stress.

4.2.2 Benchmark Problem with Sinusoidal Load and Boundary Layer

We again consider the problem described in Section 3, this time with $n = 1$. In this case, the solution is fully two-dimensional and the shear stress contains a boundary layer of width $\mathcal{O}(t^{-1})$. Figure 13 shows the true shear stress in the x-coordinate when $t = 10^{-2}$ as well as the neural network approximation to the shear stress when $\mathfrak{B}_{LS}$ and $\mathfrak{L}_{LS}$ are used in the variational formulation with Algorithm 1. Figure 14 shows the errors in the $\mathfrak{B}_{LS}$-norm and L^2-norm for each primal variable. The hyperparameters are chosen as described at the beginning of Section 4.

We observe that while the mixed least squares variational formulation on X_{LS} is capable of providing a uniformly accurate approximation in t when the solution is smooth, the same cannot be said for problems with boundary layers. In particular, we observe that most of

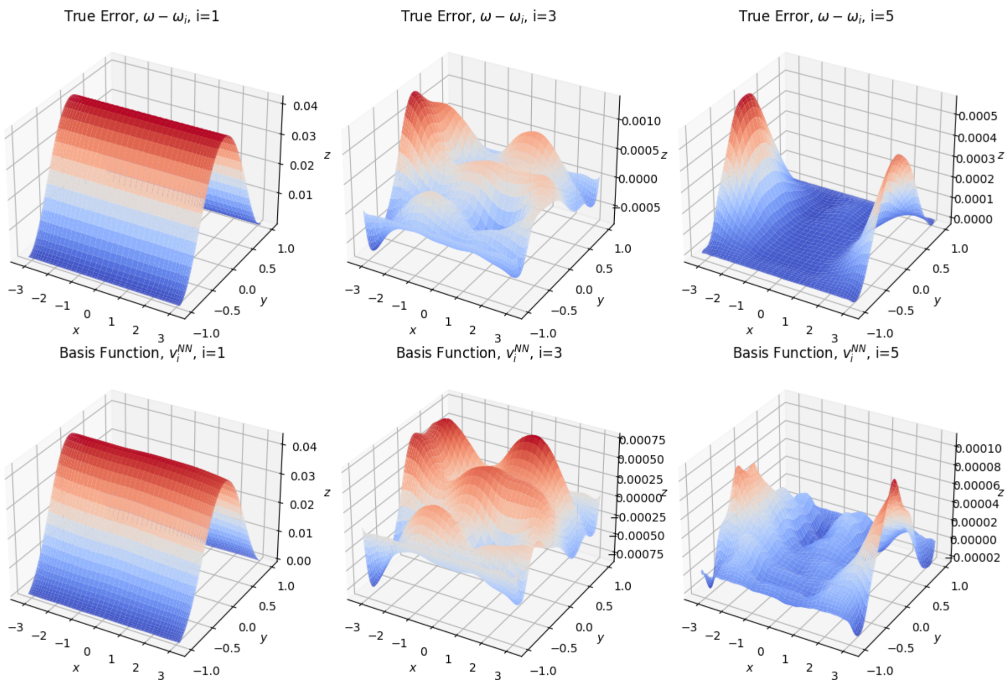


Figure 12: For $t = 10^{-6}$, the true error $\omega - \omega_i$ (top row) and the basis function v_i^{NN} (bottom row) for the 1st, 3rd, and 5th iterations of Algorithm 1 for the univariate problem with variational formulation on $\mathfrak{B}_{LS}, \mathfrak{L}_{LS}$.

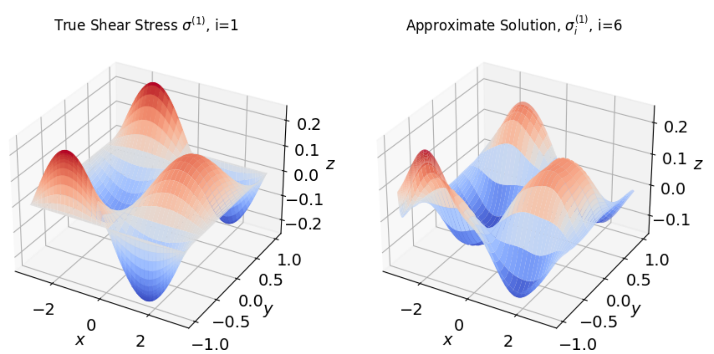


Figure 13: Left: True x-component of shear stress. Right: Neural network approximation of x-component of shear stress with variational formulation on $\mathfrak{B}_{LS}$, $\mathfrak{L}_{LS}$.

the approximation error in the x-coordinate of the shear stress is encoded in its y-derivative – $\partial\boldsymbol{\sigma}^{(1)}/\partial y$ – while the variational formulation posed on $\mathfrak{B}_{LS}$, $\mathfrak{L}_{LS}$ only contains information about $\partial\boldsymbol{\sigma}^{(1)}/\partial x$.

4.3 Least Squares Based on Brezzi-Fortin Formulation

A third variational formulation of the Reissner-Mindlin problem starts by considering the Helmholtz decomposition of the shear stress which explicitly accounts for both the irrotational and solenoidal components of the shear stress. By writing the Helmholtz decomposition of the shear stress as $\boldsymbol{\sigma} = \nabla^{\perp} p - \nabla r$, Brezzi and Fortin arrived at the following equivalent formulation to (17) [14]: seek $(r, \boldsymbol{\beta}, p, \omega) \in H_0^1(\Omega) \times \mathbf{H_0^1}(\Omega) \times H^1(\Omega)/\mathbb{R} \times H_0^1(\Omega)$ such that

$$\begin{cases} (\nabla r, \nabla \mu)_\Omega = (g, \mu)_\Omega & \forall \mu \in H_0^1(\Omega) \\ (\nabla \boldsymbol{\beta}, \nabla \boldsymbol{\varphi})_\Omega + (\nabla^\perp p, \boldsymbol{\varphi})_\Omega = (\nabla r, \boldsymbol{\varphi})_\Omega & \forall \boldsymbol{\varphi} \in \mathbf{H_0^1}(\Omega) \\ (\boldsymbol{\beta}, \nabla^\perp q)_\Omega - t^2 (\nabla^\perp p, \nabla^\perp q)_\Omega = 0 & \forall q \in H^1(\Omega)/\mathbb{R} \\ (\nabla \omega, \nabla v)_\Omega = (\boldsymbol{\beta} + t^2 \nabla r, \nabla v)_\Omega & \forall v \in H_0^1(\Omega). \end{cases} \tag{24}$$

Thus, the system (24) may be solved in three stages: first, a straightforward solution of the Poisson equation with data g to obtain r; second, a solution of the perturbed, rotated Stokes problem with data $(\nabla r, 0)$ to obtain $\boldsymbol{\beta}$ and p; and finally, another straightforward solution of the Poisson equation with data $-\nabla \cdot (\boldsymbol{\beta} + t^2 \nabla r)$ to obtain ω.

In this work, we shall apply the Galerkin Neural Network algorithm based on a least squares formulation of each of the equations of (24). We shall focus the brunt of our attention on the inner perturbed Stokes problem. As stated in (24), the inner product described by

$$((\boldsymbol{\beta}, p), (\boldsymbol{\beta}, p)) \mapsto (\nabla \boldsymbol{\beta}, \nabla \boldsymbol{\varphi})_\Omega + (\nabla^\perp p, \boldsymbol{\varphi})_\Omega + (\boldsymbol{\beta}, \nabla^\perp q)_\Omega - t^2 (\nabla^\perp p, \nabla^\perp q)_\Omega$$

is not positive definite. In particular, the choice $\boldsymbol{\beta} = \boldsymbol{\varphi} = (1,1)^T$ and $p = q = x + y$ yields $((\boldsymbol{\beta}, p), (\boldsymbol{\beta}, p)) \mapsto -2t^2|\Omega| < 0$. Instead, we consider a least squares formulation of the perturbed Stokes problem, which consists of the variational problem posed on $X_{BF} := (\mathbf{H^2}(\Omega) \cap \mathbf{H_0^1}(\Omega)) \times (H^1(\Delta; \Omega)/\mathbb{R})$ where $H^1(\Delta; \Omega) := \{v \in H^1(\Omega) : \Delta v \in L^2(\Omega)\}$ and X_{BF} is endowed with the norm

$$||(\boldsymbol{\varphi}, q)||_{BF} := (||\boldsymbol{\varphi}||^2_{H^2(\Omega)} + ||\frac{1}{t}(\nabla \times \boldsymbol{\beta})||^2_\Omega + ||q||^2_{H^1(\Omega)} + ||t\Delta q||^2_{L^2(\Omega)} + ||t\partial_n q||^2_{H^{-1/2}(\partial\Omega)})^{1/2}$$

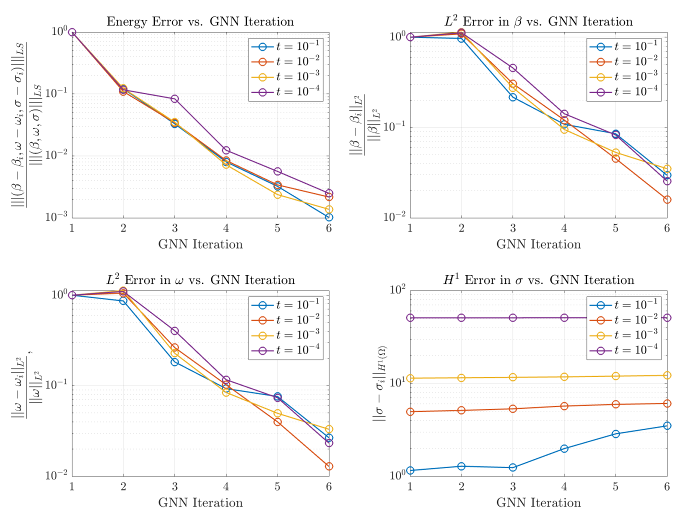


Figure 14: Relative energy error, relative L^2 error in $\boldsymbol{\beta}$ and ω, and H^1 error in $\boldsymbol{\sigma}$ after each iteration of Algorithm 1 for the Reissner-Mindlin plate with boundary layer and variational formulation on $\mathfrak{B}_{LS}, \mathfrak{L}_{LS}$.

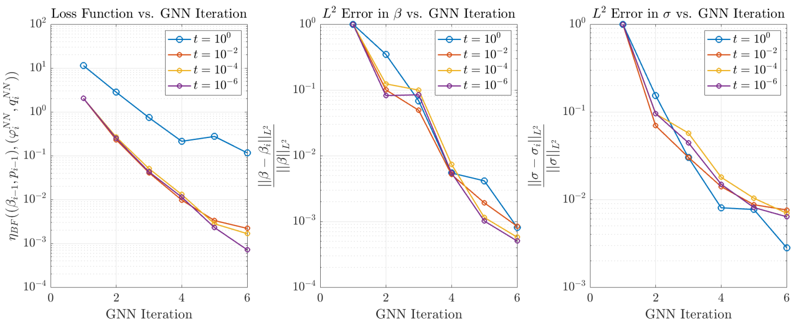


Figure 15: Errors after each iteration of Algorithm 1 for the univariate problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}, \tilde{\mathfrak{L}}_{BF}$.

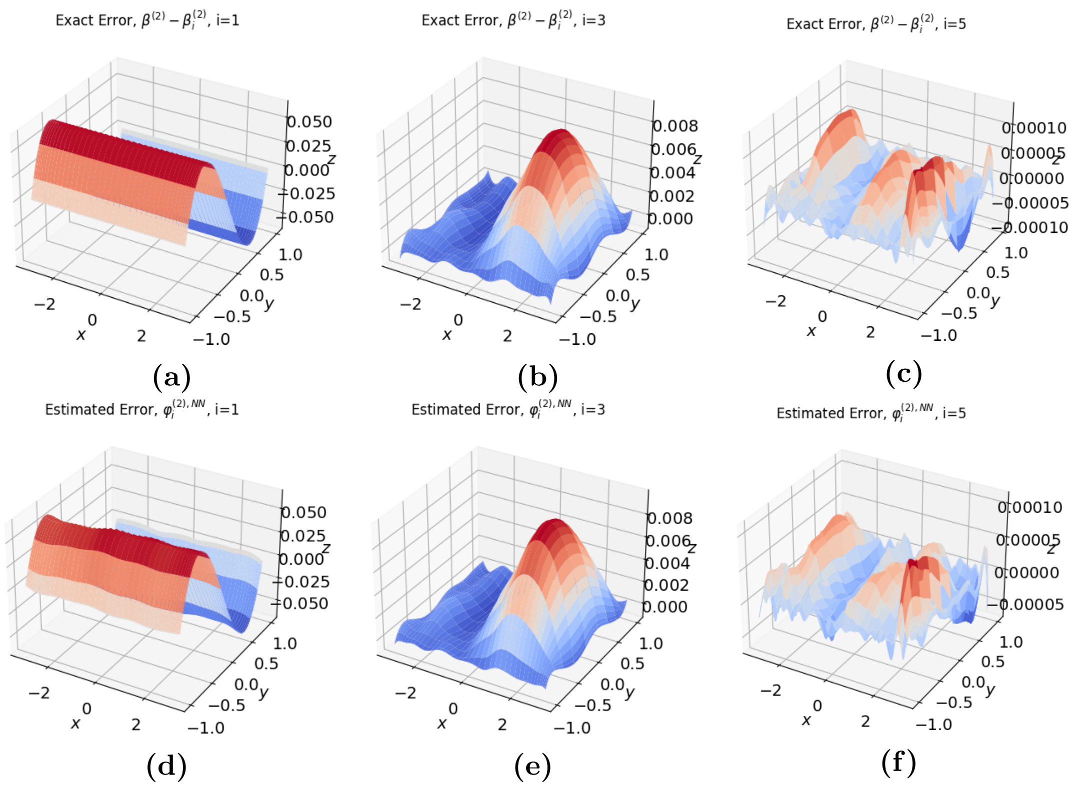
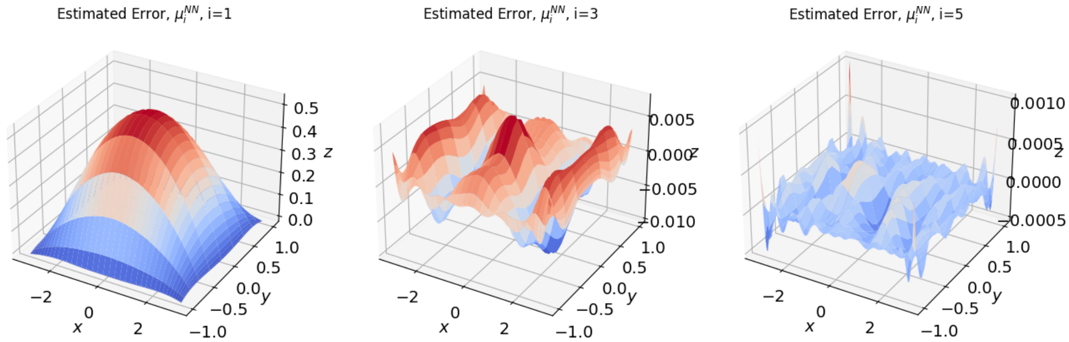


Figure 16: For $t = 10^{-6}$ in the univariate problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: (a)-(c) True error $\boldsymbol{\beta}^{(2)} - \boldsymbol{\beta}_i^{(2)}$. (d)-(f) Basis function $\boldsymbol{\varphi}_i^{(2),NN}$.

with

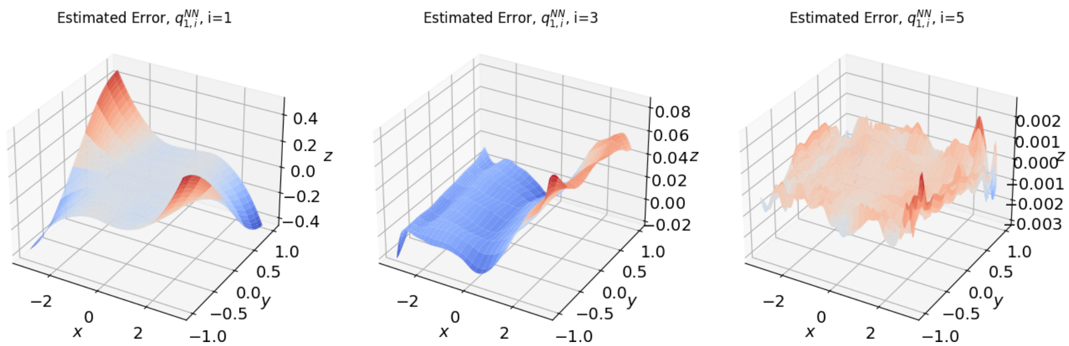
$$\begin{aligned} \mathfrak{B}_{BF}((\boldsymbol{\beta}, p); (\boldsymbol{\varphi}, q)) &:= (-\Delta\boldsymbol{\beta} + \nabla^{\perp} p, -\Delta\boldsymbol{\varphi} + \nabla^{\perp} q)_{\Omega} + t^{-2}(\nabla \times \boldsymbol{\beta} + t^2 \Delta p, \nabla \times \boldsymbol{\varphi} + t^2 \Delta q)_{\Omega} \\ &\quad + t^2 (\partial_n p, \partial_n q)_{H^{-1/2}(\partial\Omega)} \\ \mathfrak{L}_{BF}((\boldsymbol{\varphi}, q)) &:= (\nabla r, -\Delta\boldsymbol{\varphi} + \nabla^{\perp} q)_{\Omega}. \end{aligned}$$

Here, $H^{-1/2}(\Omega)$ is the dual space to the Sobolev-Slobodeckij space [19] $H^{1/2}(\Omega)$ and $(\cdot,\cdot)_{H^{-1/2}(\Omega)}$ is its associated inner product.



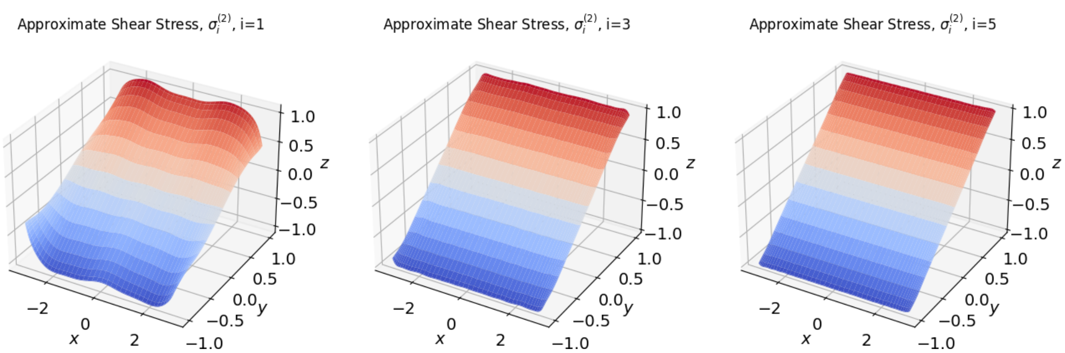
(a) (b) (c)

Figure 17: For $t = 10^{-6}$ in the univariate problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: basis function μ_i^{NN}.



(a) (b) (c)

Figure 18: For $t = 10^{-6}$ in the univariate problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: basis function $q_i^{(2),NN}$.



(a) (b) (c)

Figure 19: For $t = 10^{-2}$ in the univariate problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: approximation $\boldsymbol{\sigma}_i^{(2)} = -\partial p_i/\partial x - \partial r_i/\partial y$.

The following result demonstrates that the Helmholtz least squares variational formulation for $\boldsymbol{\beta}$ and p is both continuous and coercive, thus allowing us to apply Algorithm 1.

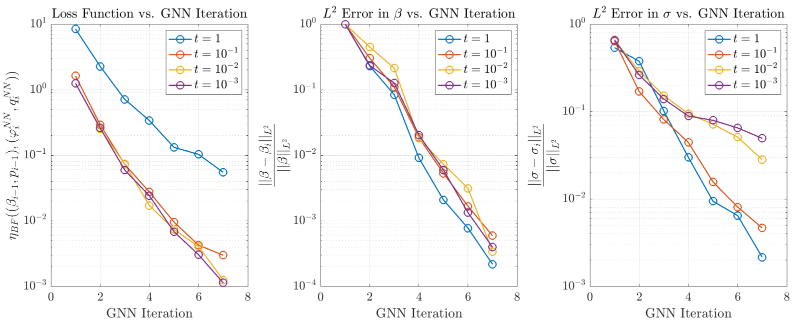


Figure 20: Energy error and L^2 error in $\boldsymbol{\beta}$ and ω for the problem with sinusoidal load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$.

Proposition 4.3. *Let* $(\boldsymbol{\beta}, p) \in X_{BF}$ *and suppose* $0 < t \leqslant 1$. *There exist constants* $C_1 > 0$ *and* $C_2 > 0$ *independent of* t *such that*

$$\begin{aligned} \mathfrak{B}_{BF}(\boldsymbol{\beta}, p; \boldsymbol{\varphi}, q) &\leqslant C_1 ||(\boldsymbol{\beta}, p)||_{BF} ||(\boldsymbol{\varphi}, q)||_{X_{BF}} \qquad && \forall (\boldsymbol{\beta}, p),\ (\boldsymbol{\varphi}, q) \in X_{BF} \\ C_2 ||(\boldsymbol{\beta}, p)||_{BF}^2 &\leqslant \mathfrak{B}_{BF}(\boldsymbol{\beta}, p; \boldsymbol{\beta}, p) && \forall (\boldsymbol{\beta}, p) \in X_{BF}. \end{aligned}$$

Proof. We begin by applying the Cauchy-Schwarz inequality with respect to L^2 and $H^{-1/2}$:

$$\begin{aligned} \mathfrak{B}_{BF}((\boldsymbol{\beta}, p); (\boldsymbol{\varphi}, q)) \leqslant\ & || - \Delta \boldsymbol{\beta} + \nabla^{\perp} p||_{\Omega} \cdot || - \Delta \boldsymbol{\varphi} + \nabla^{\perp} q||_{\Omega} \\ & + ||\frac{1}{t}(\nabla \times \boldsymbol{\beta} + t^2 \Delta p)||_{\Omega} \cdot ||\frac{1}{t}(\nabla \times \boldsymbol{\varphi} + t^2 \Delta q)||_{\Omega} \\ & + t^2 ||\partial_n p||_{H^{-1/2}(\partial\Omega)} \cdot ||\partial_n q||_{H^{-1/2}(\partial\Omega)} \\ \leqslant\ & (||\boldsymbol{\beta}||_{H^2(\Omega)} + ||p||_{H^1(\Omega)})(||\boldsymbol{\varphi}||_{H^2(\Omega)} + ||q||_{H^1(\Omega)}) \\ & + (||\frac{1}{t}(\nabla \times \boldsymbol{\beta})||_{\Omega} + ||tp||_{H^2(\Omega)}) \cdot (||\frac{1}{t}(\nabla \times \boldsymbol{\varphi})||_{\Omega} + ||tq||_{H^2(\Omega)}) \\ & + t^2 ||\partial_n p||_{H^{-1/2}(\partial\Omega)} \cdot ||\partial_n q||_{H^{-1/2}(\partial\Omega)}. \end{aligned}$$

An application of the Cauchy-Schwarz inequality with respect to ℓ^2 yields the first result.

As for the second inequality, given $(\beta, p) \in X_{BF}$, we form the system

$$\begin{cases} \mathbf{g}_1 := -\Delta \boldsymbol{\beta} + \nabla^{\perp} p & \text{in } \Omega \\ t \cdot g_2 := \nabla \times \boldsymbol{\beta} + t^2 \Delta p & \text{in } \Omega \\ \boldsymbol{\beta} = \mathbf{0} & \text{on } \partial\Omega \\ g_3 := t \cdot \partial_n p & \text{on } \partial\Omega. \end{cases} \tag{25}$$

From [15], we have the a priori estimate

$$||\boldsymbol{\beta}||_{H^2(\Omega)}^2 + ||p||_{H^1(\Omega)}^2 + ||t\Delta p||_{\Omega}^2 \leqslant C(||\mathbf{g}_1||_{\Omega}^2 + ||g_2||_{\Omega}^2).$$

from which we further obtain

$$||\boldsymbol{\beta}||_{H^2(\Omega)}^2 + ||p||_{H^1(\Omega)}^2 + ||t\Delta p||_{\Omega}^2 \leqslant C(||\mathbf{g}_1||_{\Omega}^2 + ||g_2||_{\Omega}^2 + ||tg_3||_{H^{-1/2}(\partial\Omega)}^2).$$

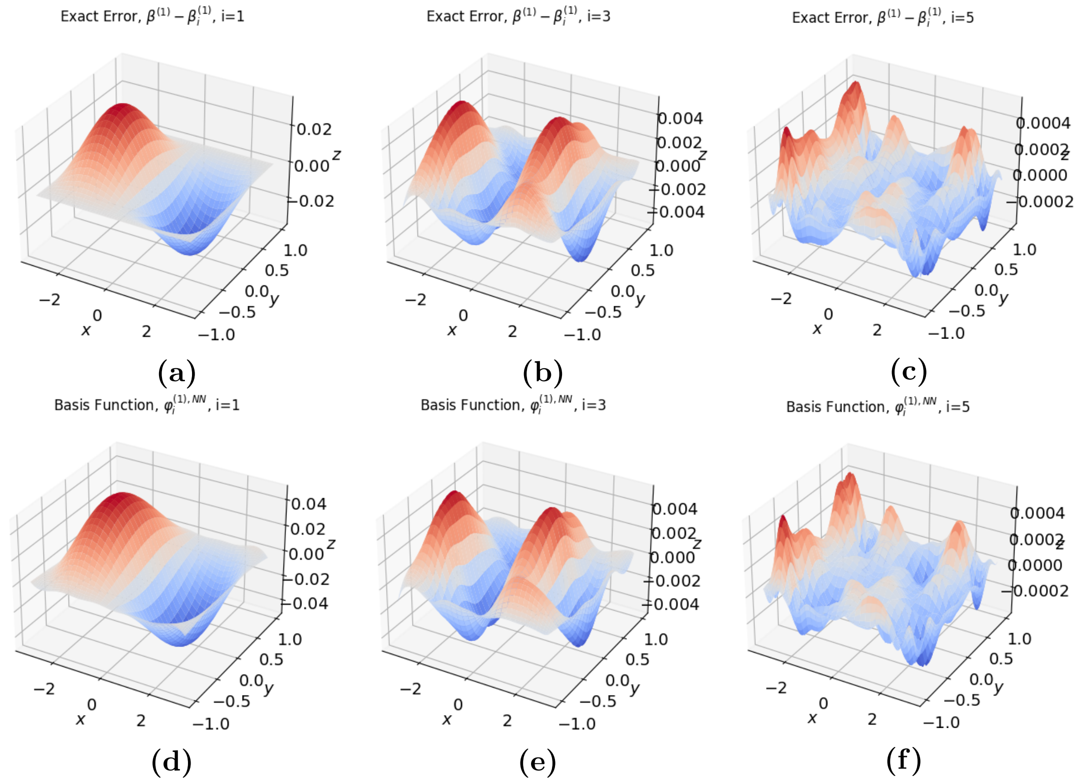


Figure 21: For $t = 10^{-2}$ in the problem with sinusoidal load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: (a)-(c) True error $\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}_i^{(1)}$. (d)-(f) Basis function $\boldsymbol{\varphi}_i^{(1),NN}$.

Additionally, we have

$$\begin{aligned} ||\frac{1}{t}(\nabla \times \boldsymbol{\beta})||_\Omega &\leqslant ||t\Delta p||_\Omega + ||g_2||_\Omega \\ &\leqslant C(||\mathbf{g}_1||_\Omega + ||g_2||_\Omega + ||tg_3||_{H^{-1/2}(\partial\Omega)}) \end{aligned}$$

which completes the proof. □

Approaches based on the H^{-1} norm of the residual of the second equation in (24) and the L^2 norm of the residual of the third equation have again been explored in [10, 15], but approaches based fully on the L^2 norm of the interior residuals, as described by $\mathfrak{B}_{BF}$, have not been used hitherto in practice due to the necessity of H^2 regularity. Nevertheless, Proposition 4.3 shows that the ratio M/α for the formulation based on $\mathfrak{B}_{BF}$ and $\mathfrak{L}_{BF}$ is independent of t and a quasi-optimal approximation should be expected.

We note that the term $t^2(\partial_n p, \partial_n q)_{H^{-1/2}(\partial\Omega)}$ corresponds to the weak enforcement of the boundary condition $\partial_n p = 0$. The computation of $H^{-1/2}$ inner products is not straightforward but could, in principle, be achieved using singular integrals [31]. Instead, for simplicity and to avoid unnecessary technical distractions, we elect to impose the boundary condition on p more strongly by considering the penalization $t^2(\partial_n p, \partial_n q)_{\partial\Omega}$. More specifically, we consider the modified bilinear form

$$\tilde{\mathfrak{B}}_{BF}((\boldsymbol{\beta}, p); (\boldsymbol{\varphi}, q)) := (-\Delta\boldsymbol{\beta} + \nabla^\perp p, -\Delta\boldsymbol{\varphi} + \nabla^\perp q)_\Omega + t^{-2}(\nabla \times \boldsymbol{\beta} + t^2\Delta p, \nabla \times \boldsymbol{\varphi} + t^2\Delta q)_\Omega$$

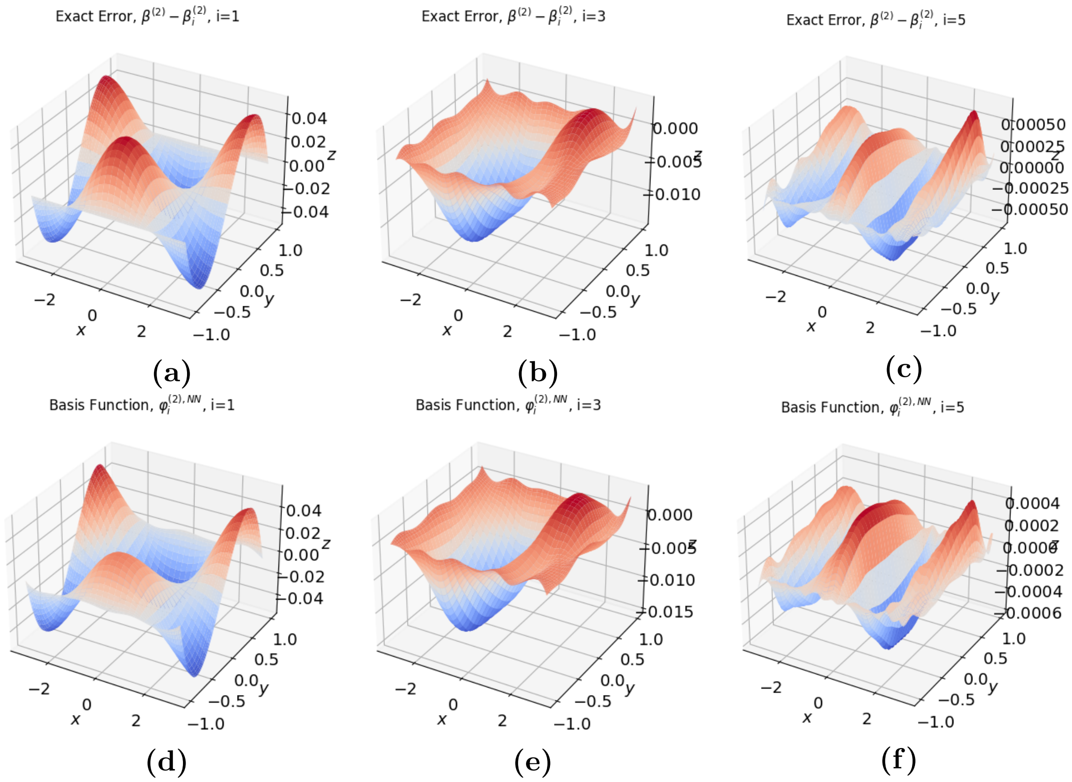


Figure 22: For $t = 10^{-2}$ in the problem with sinusoidal load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: (a)-(c) True error $\boldsymbol{\beta}^{(2)} - \boldsymbol{\beta}_i^{(2)}$. (d)-(f) Basis function $\boldsymbol{\varphi}_i^{(2),NN}$.

$$+ t^2(\partial_n p, \partial_n q)_{\partial\Omega}.$$

In order to apply this formulation to the examples in 4.2.1 and 4.2.2, we must determine the appropriate boundary conditions on r and p. We impose $r = 0$ on $\partial\Omega$. Due to the periodic nature of $\boldsymbol{\beta}$ and ω along $x = -\pi$ and $x = \pi$, $\boldsymbol{\sigma}$ is also periodic along $x = -\pi$ and $x = \pi$. Since we have

$$\begin{aligned}(\boldsymbol{\sigma} \cdot \mathbf{t})(x, y) &= (\nabla r \cdot \mathbf{t})(x, y) - (\nabla^{\perp} p \cdot \mathbf{t})(x, y) \\ &= -(\nabla^{\perp} p \cdot \mathbf{t})(x, y) = -(\nabla p \cdot \mathbf{n})(x, y) \quad \forall (x, y) \in \partial\Omega\end{aligned}$$

where $\mathbf{t}$ is the unit counterclockwise tangent vector, we must have $\nabla p \cdot \mathbf{n}|_{x=-\pi} + \nabla p \cdot \mathbf{n}|_{x=\pi} = 0$ on $\partial\Omega$.

4.3.1 Benchmark Problem with Constant Load

Figure 15 shows the loss function (estimated energy error with respect to $\mathfrak{B}_{BF}$) per Galerkin Neural Network iteration as well as the L^2 error of the rotation $\boldsymbol{\beta}$ and shear stress $\boldsymbol{\sigma}$ for $t = 1, 10^{-2}, 10^{-4}, 10^{-6}$ in the case when $n = 0$. We denote by $\boldsymbol{\varphi}_i^{NN}$ the basis functions for approximating $\boldsymbol{\beta}$ and by q_i^{NN} the basis functions for approximating p. The hyperparameters for this example and all remaining examples are chosen as described at the beginning of Section 4 with the exception that the activation function for each basis function for $\boldsymbol{\beta}$ and

p is $\sigma_i(z) = \tanh((1 + 0.55i)z)$. Figure 16-19 show the true errors and basis functions for $i = 1, 3, 5$. We observe no issues with locking.

4.3.2 Benchmark Problem with Sinusoidal Load and Boundary Layer

We next turn our attention to the model problem when $n = 1$. Figure 20 shows the loss function per Galerkin Neural Network iteration as well as the L^2 error of the rotation $\boldsymbol{\beta}$ and shear stress $\boldsymbol{\sigma}$ for $t = 1, 10^{-1}, 10^{-2}, 10^{-3}$ in the case when $n = 1$. Again, we observe no issues with locking. More importantly, we observe good resolution of the boundary layer as seen in Figure 25. Figures 21-24 show the true errors and basis functions for $i = 1, 3, 5$.

4.3.3 Triangular Wave Forcing Term

We next consider the case when the forcing term is the triangular wave shown in Figure 26. The domain and boundary conditions considered are the same as in 4.2.2. The triangular wave function has a Fourier series representation given by

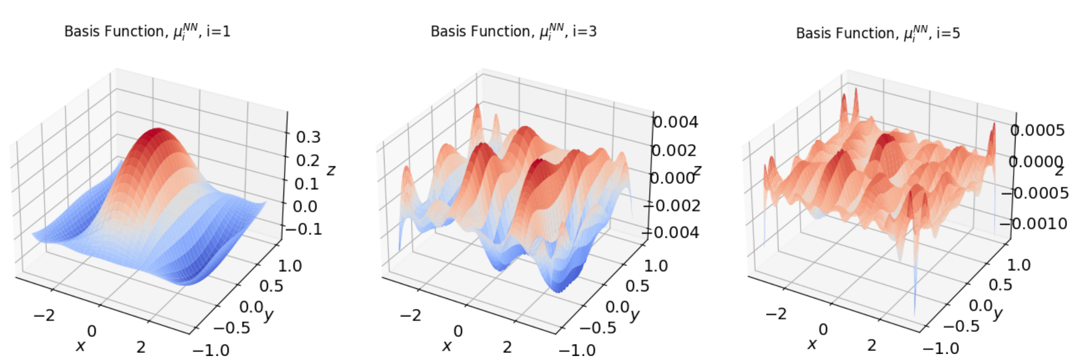
$$g_T(x) = \frac{8}{\pi^2} \sum_{n=1}^{\infty} \frac{\sin^2(n\pi/4)}{n^2} \cos(nx),$$

from which the exact solution may be obtained by superimposing the solutions corresponding to each n, as in Section 3.

Figure 27 shows the loss and L^2 error in $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}$ while Figures 28-31 show the basis functions $\boldsymbol{\varphi}_i^{NN}$, μ_i^{NN}, and q_i^{NN} as well as the approximations $\boldsymbol{\beta}_i$ and p_i. We again observe that the boundary layer is resolved correctly.

5 Conclusions

We have presented a neural network approach to approximating Reissner-Mindlin plates which is uniformly accurate in the plate thickness. The main contributions of this work are as follows. The neural network framework utilized is oblivious to the nature of the PDE and



(a) (b) (c)

Figure 23: For $t = 10^{-2}$ in the problem with sinusoidal load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: basis function μ_i^{NN}.

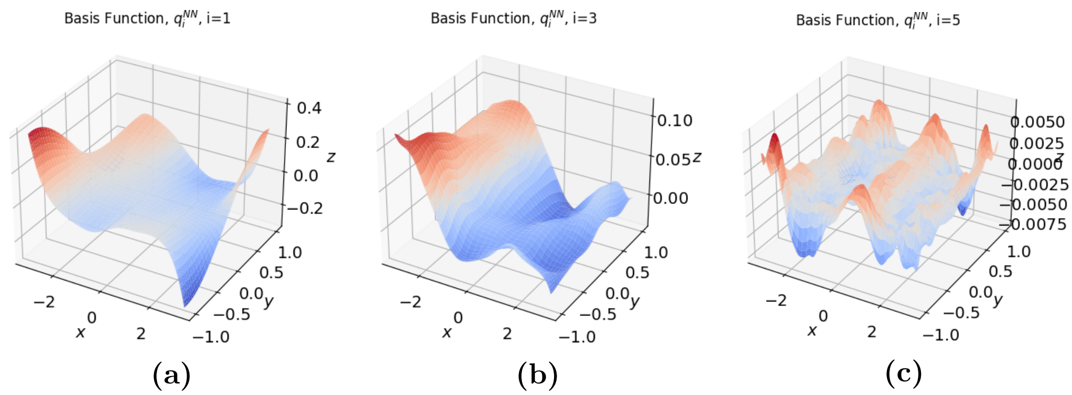


Figure 24: For $t = 10^{-2}$ in the problem with sinusoidal load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: basis function q_i^{NN}.

thus requires no structural modifications in order to be applied to the Reissner-Mindlin model other than a continuous, coercive, symmetric, positive-definite bilinear operator. In presenting results for two new least squares variational formulations of the Reissner-Mindlin plate, we have demonstrated even for neural networks the importance of selecting a variational formulation which does not exhibit degenerate behavior as the plate thickness is reduced.

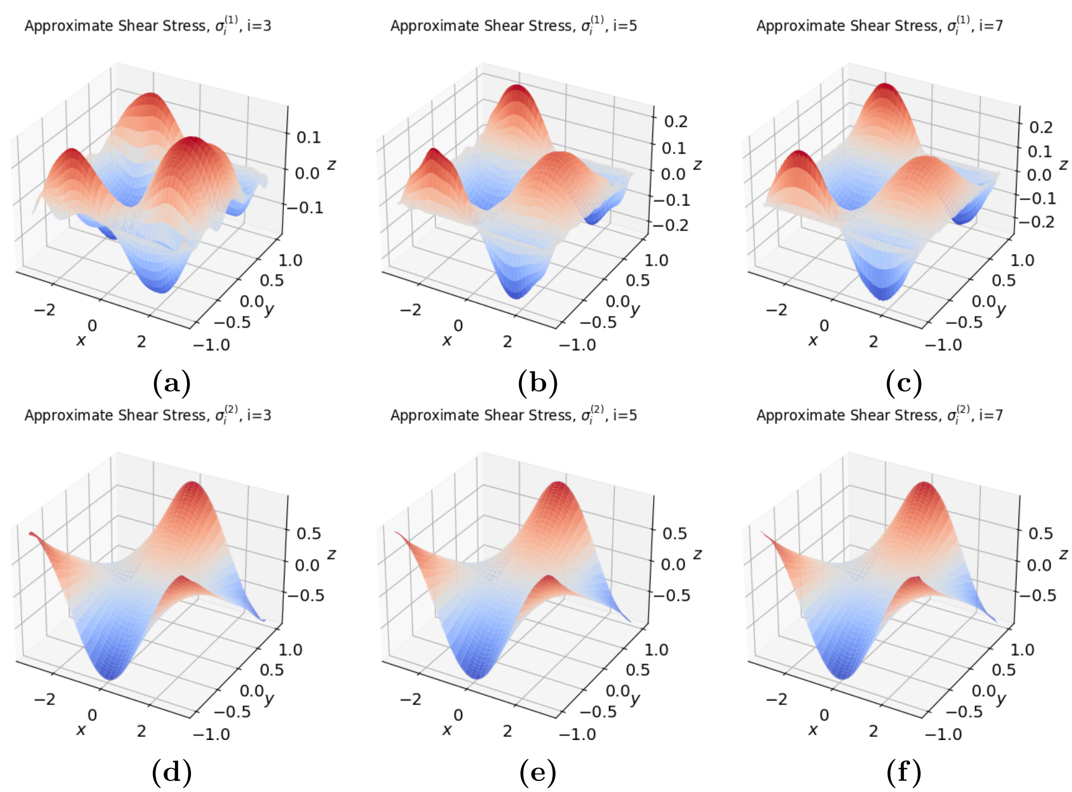


Figure 25: For $t = 10^{-2}$ in the problem with sinusoidal load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: (a)-(c) Approximation $\boldsymbol{\sigma}_i^{(1)} = \partial p_i/\partial y - \partial r_i/\partial x$. (d)-(f) Approximation $\boldsymbol{\sigma}_i^{(2)} = -\partial p_i/\partial x - \partial r_i/\partial y$.

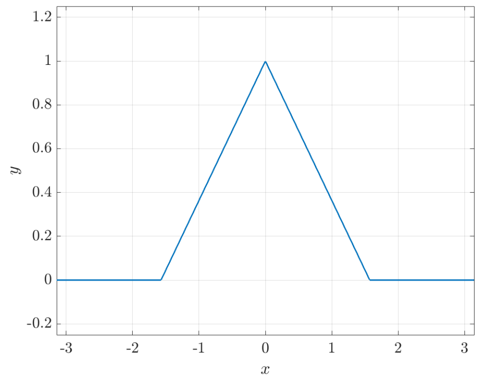


Figure 26: The triangular wave function g_T.

Moreover, in comparison to traditional finite element methods, a reduction of the PDE to a first order system of least squares (FOSLS) is unnecessary with Galerkin neural networks as the activation functions are global without element continuity constraints and are only required to be elements of H^k, where k is the order of the PDE. Additionally, the accurate resolution boundary layer problems using finite element approaches typically requires the use of graded meshes around the boundary layer. Our approach in comparison does not utilize any a priori knowledge of the location of the boundary layer. Finally, numerical results are provided for a complex benchmark problem which exhibits a boundary layer in the shear stress, and we also provide a framework for synthesizing a large class of test problems with analytic solutions which are crucial for evaluating performance of numerical methods – even beyond neural network approaches – applied to Reissner-Mindlin plates.

6 Acknowledgement

We thank the NSF (Grant No. 1644760) and the PhILMs Center (DE-SC0019453) for support.

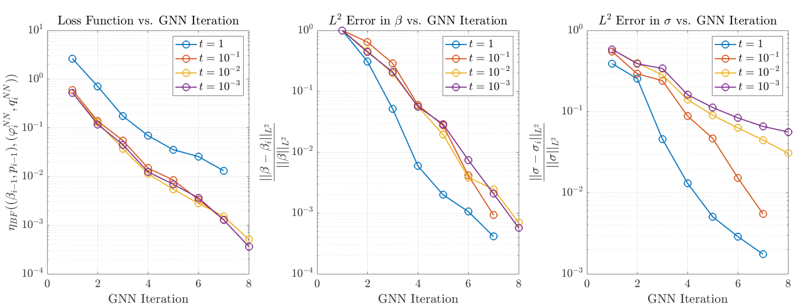


Figure 27: Energy error and L^2 error in $\boldsymbol{\beta}$, ω, and $\boldsymbol{\sigma}$ for the problem with triangular wave load with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$.

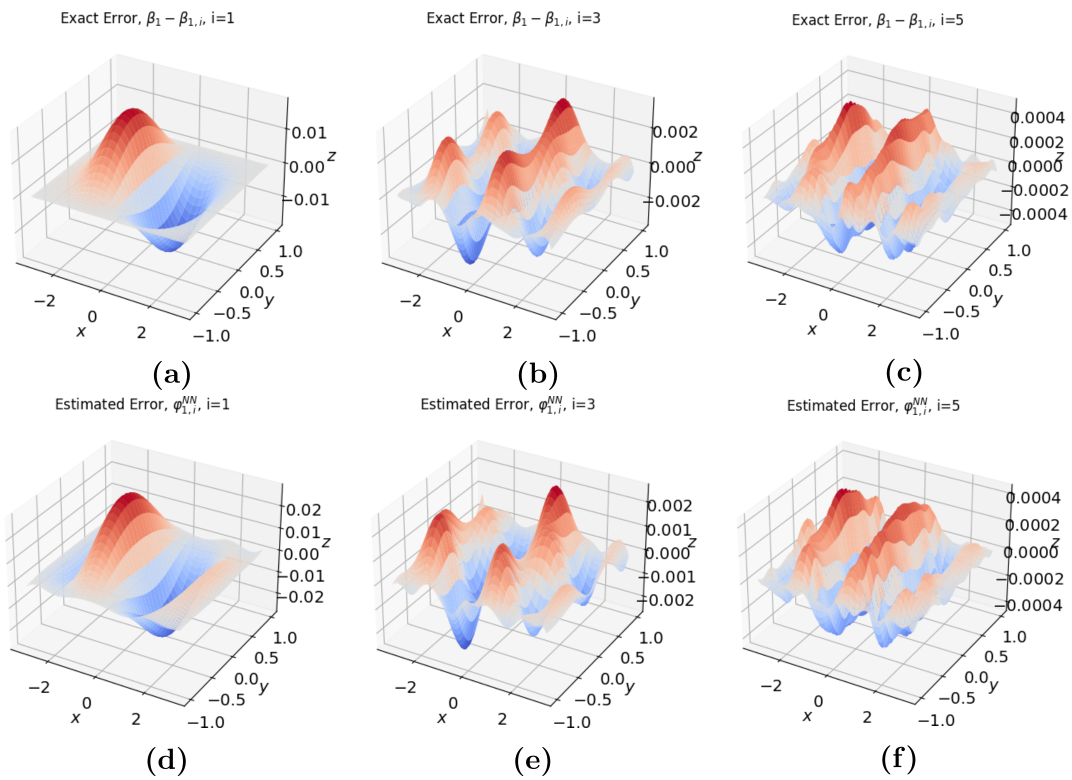


Figure 28: For $t = 10^{-6}$ in the triangular wave forcing term problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$ (a)-(c) True error $\boldsymbol{\beta}^{(1)} - \boldsymbol{\beta}_i^{(1)}$. (d)-(f) Basis function $\boldsymbol{\varphi}_i^{(1),NN}$.

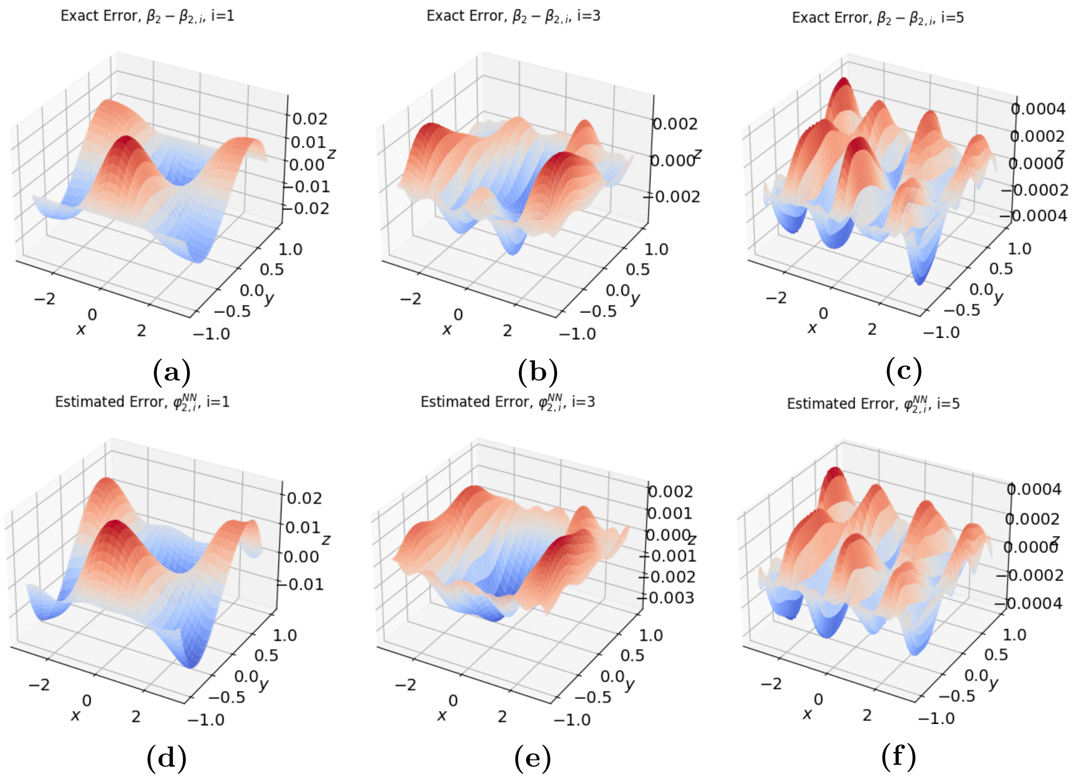


Figure 29: For $t = 10^{-6}$ in the triangular wave forcing term problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: (a)-(c) True error $\boldsymbol{\beta}^{(2)} - \boldsymbol{\beta}_i^{(2)}$. (d)-(f) Basis function $\boldsymbol{\varphi}_i^{(2),NN}$.

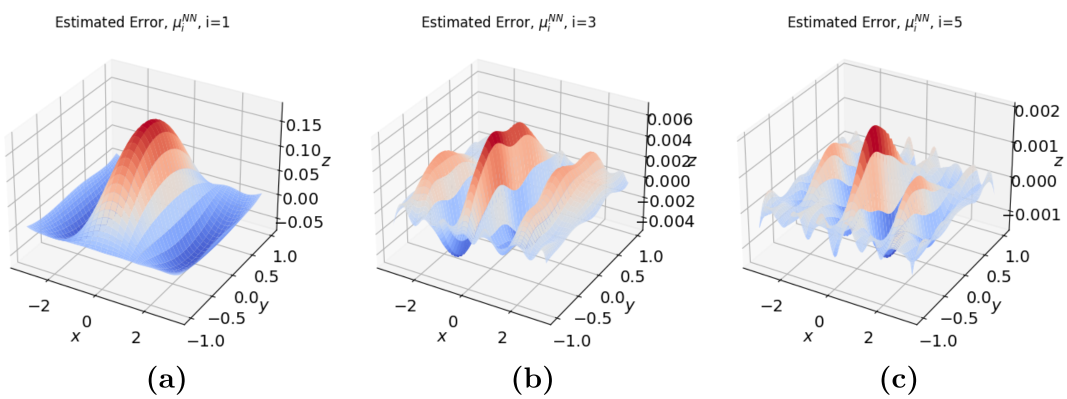


Figure 30: For $t = 10^{-2}$ in the triangular wave forcing term problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: basis function μ_i^{NN}.

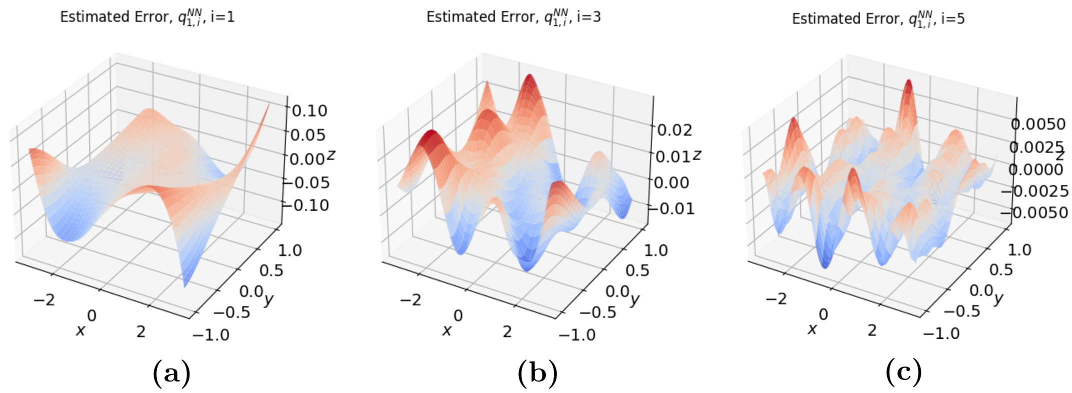


Figure 31: For $t = 10^{-2}$ in the triangular wave forcing term problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: basis function q_i^{NN}.

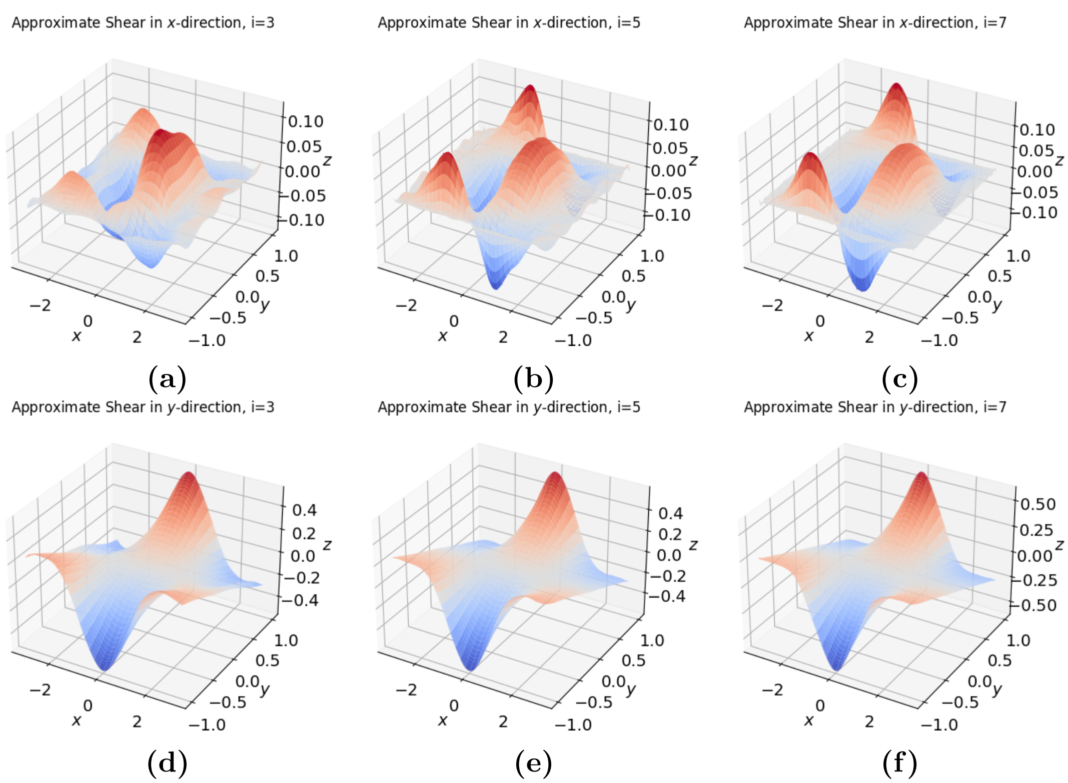


Figure 32: For $t = 10^{-2}$ in the triangular wave forcing term problem with variational formulation on $\tilde{\mathfrak{B}}_{BF}$, $\tilde{\mathfrak{L}}_{BF}$: (a)-(c) Approximation $\boldsymbol{\sigma}_i^{(1)} = \partial p_i/\partial y - \partial r_i/\partial x$. (d)-(f) Approximation $\boldsymbol{\sigma}_i^{(2)} = -\partial p_i/\partial x - \partial r_i/\partial y$.

7 Appendix

The coefficients $A_n(t)$, $B_n(t)$, $C_n(t)$, and $D_n(t)$ appearing in (13) are determined by the boundary conditions: $\boldsymbol{\beta}_n = \mathbf{0}$ and $\omega_n = 0$ on Γ_D with periodic conditions on Γ_{per}. Namely, we require that $\boldsymbol{\beta}_n(x,1) = \mathbf{0}$ and $\omega_n(x,1) = 0$ while the periodic boundary conditions on Γ_{per} are automatically satisfied thanks to the structure of $\boldsymbol{\beta}_n$ and ω_n. Additionally, satisfying the PDE $-\Delta\boldsymbol{\beta}_n + t^{-2}(\boldsymbol{\beta}_n - \nabla\omega_n) = \mathbf{0}$ gives rise to an additional constraint $\Upsilon_n''(y) - n^2\Upsilon_n(y,t) = \Psi_n(y,t) - 1$.

These requirements lead to a linear system of equations which determine the coefficients given by

$$\begin{cases} \lambda_n t\coth(\lambda_n)\cdot A_n(t) - n\cdot B_n(t) - n\cdot C_n(t) - n\cdot D_n(t) = -\dfrac{1}{n^3} \\ nt\cdot A_n(t) - (n\coth(n)+1)\cdot B_n(t) - n\tanh(n)\cdot C_n(t) - n\tanh(n)\cdot D_n(t) = 0 \\ -B_n(t) - C_n(t) - (1+t^2)\cdot D_n(t) = -\dfrac{t^2}{n^2} - \dfrac{1}{n^4} \\ 2n\cdot B_n(t) + \tanh(n)\cdot D_n(t) = 0 \end{cases} \tag{26}$$

The solution to (26) can be computed directly but has a rather complicated form. However, one can instead seek a series approximation valid for $t \ll 1$ of the form

$$\vec{\alpha}_n(t) = \vec{\gamma}_0 + \vec{\gamma}_1 t + \vec{\gamma}_2\frac{t^2}{2} + \mathcal{O}(t^3),$$

where $\vec{\alpha}_n(t)$ is the vector of coefficients $\vec{\alpha}_n(t) = (A_n(t), B_n(t), C_n(t), D_n(t))^T$. Let $M_n(t)$ denote the coefficient matrix and $\vec{F}_n(t)$ the right-hand side of (26), respectively. Then expanding $M^{-1}(t)$ as a Taylor series about $t=0$ yields the series expansion

$$\begin{aligned}\vec{\alpha}_n(t) = [&M^{-1}(0) - M^{-1}(0)M'(0)M^{-1}(0)t + (-M^{-1}(0)M''(0)M^{-1}(0) + \\ &2M^{-1}(0)M'(0)M^{-1}(0)M'(0)M^{-1}(0))\frac{t^2}{2} + \mathcal{O}(t^3)]\vec{F}_n(t).\end{aligned}$$

The vectors $\vec{\gamma}_0$, $\vec{\gamma}_1$, and $\vec{\gamma}_2$ are thus given by

$$\begin{aligned}
\vec{\gamma}_0 &= M^{-1}(0)\vec{F}_n(0) = \frac{\sinh(2n)}{n^4(2n+\sinh(2n))}\begin{bmatrix} 0 \\ -n\tanh(n) \\ n\coth(n) - 2n^2 + 1 \\ 2n^2 \end{bmatrix} \\
\vec{\gamma}_1 &= -M^{-1}(0)M'(0)M^{-1}(0)\vec{F}_n'(0) + M^{-1}(0)\vec{F}_n'(0) = \vec{0} \\
\vec{\gamma}_2 &= M^{-1}(0)\vec{F}_n''(0) - \left(M^{-1}(0)M''(0)M^{-1}(0) + 2M^{-1}(0)M'(0)M^{-1}(0)M'(0)M^{-1}(0)\right)\vec{F}_n(0) \\
&= \begin{bmatrix} 2/n \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{\sinh(2n)}{2n+\sinh(2n)}\begin{bmatrix} -4 \\ -2\tanh(n)/n \\ 2(n\coth(n) - 2n^2 + 1)/n^2 \\ 4 \end{bmatrix}
\end{aligned}$$

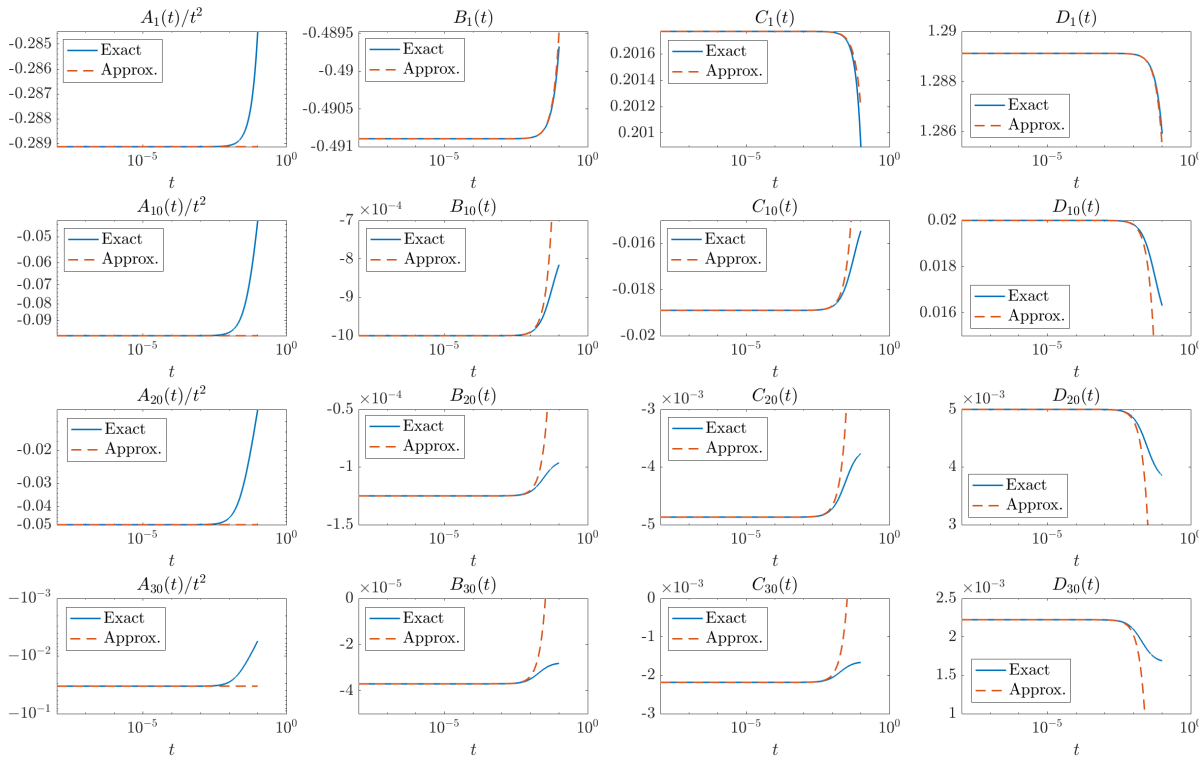


Figure 33: Coefficients $A_n(t)$, $B_n(t)$, $C_n(t)$, $D_n(t)$ and their series expansions for $n = 1, 10, 20, 30$.

$$+\left(\frac{\sinh(2n)}{2n+\sinh(2n)}\right)^2 \begin{bmatrix} 0 \\ 4\tanh(n)/n \\ -4(n\coth(n)-2n^2+1)/n^2 \\ -8 \end{bmatrix}.$$

We note in particular that $A_n(t) = \mathcal{O}(t^2)$ and therefore, the factor $A_n(t)/t^2$ which appears in (16) remains bounded as $t \to 0$.

Moreover, Figure 33 shows plots of $A_n(t)$, $B_n(t)$, $C_n(t)$, and $D_n(t)$ versus t along with their approximations for $n = 1, 10, 20, 30$. It is clear that the series expansions are highly accurate for $t < 10^{-2}$. As n is increased, discrepancies between the coefficients and their series expansions appear to also increase for $t > 10^{-2}$. However, since the Fourier coefficients of $\boldsymbol{\beta}_n$, ω_n, and $\boldsymbol{\sigma}_n$ generally decrease in magnitude as n increases, the net effect of these discrepancies is negligible. Indeed, Figure 34 shows the error in $\boldsymbol{\beta}$, ω, and $\boldsymbol{\sigma}$ for the problem in Section 4.3.3 with $t = 10^{-2}$ when the exact coefficients are used compared to the series expansions. We thus use the simpler series expansions when computing the exact solutions for all model problems.

Lastly, we note that the evaluation of quantities such as $\sinh(\lambda_n y)$ and $\cosh(\lambda_n)$ for small t in (13) requires the use of high-precision arithmetic libraries, such as [24] in Python, in order to accurately evaluate the true solution.

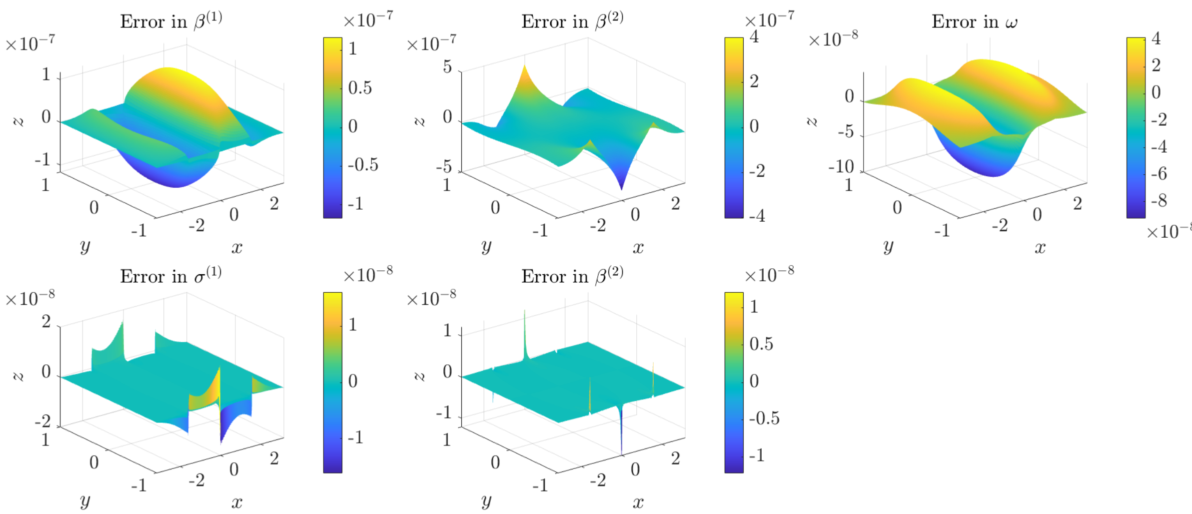


Figure 34: Errors in the true solution of the model problem in Section 4.3.3 with $t = 10^{-2}$ when using exact coefficients compared to their series expansions.

References

[1] M. Ainsworth and J. Dong, *Galerkin neural networks: a framework for approximating variational equations with error control*, SIAM J. Sci. Comput., 43 (2021), pp. A2474–A2501.

[2] M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Computer methods in applied mechanics and engineering, 142 (1997), pp. 1–88.

[3] M. Ainsworth and K. Pinchedez, *The hp-mitc finite element method for the reissner–mindlin plate problem*, Journal of computational and applied mathematics, 148 (2002), pp. 429–462.

[4] D. N. Arnold, *Discretization by finite elements of a model parameter dependent problem*, Numerische Mathematik, 37 (1981), pp. 405–421.

[5] D. N. ARNOLD and R. S. FALK, *Edge effects in the reissner-mindlin plate theory*, in Presented at the winter annual meeting of the American Society of Mechanical Engineers, vol. 1, 1989, p. I2.

[6] D. N. Arnold and R. S. Falk, *A uniformly accurate finite element method for the reissner–mindlin plate*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 1276–1290.

[7] ——, *The boundary layer for the reissner–mindlin plate model*, SIAM Journal on Mathematical Analysis, 21 (1990), pp. 281–312.

[8] I. Babuška and M. Suri, *Locking effects in the finite element approximation of elasticity problems*, Numerische Mathematik, 62 (1992), pp. 439–463.

[9] J. Berg and K. Nyström, *A unified deep artificial neural network approach to partial differential equations in complex geometries*, Neurocomputing, 317 (2018), pp. 28–41.

[10] J. Bramble and T. Sun, *A negative-norm least squares method for reissner-mindlin plates*, Mathematics of computation, 67 (1998), pp. 901–916.

[11] ——, *A negative-norm least squares method for reissner-mindlin plates*, Mathematics of computation, 67 (1998), pp. 901–916.

[12] S. C. Brenner, L. R. Scott, and L. R. Scott, *The mathematical theory of finite element methods*, vol. 3, Springer, 2008.

[13] F. Brezzi, K.-J. Bathe, and M. Fortin, *Mixed-interpolated elements for reissner–mindlin plates*, International Journal for Numerical Methods in Engineering, 28 (1989), pp. 1787–1801.

[14] F. Brezzi and M. Fortin, *Numerical approximation of mindlin-reissner plates*, Mathematics of computation, 47 (1986), pp. 151–158.

[15] Z. Cai, *Least squares for the perturbed stokes equations and the reissner–mindlin plate*, SIAM Journal on Numerical Analysis, 38 (2000), pp. 1561–1581.

[16] Z. Cai, X. Ye, and H. Zhang, *Least-squares finite element approximations for the reissner-mindlin plate*, Numerical linear algebra with applications, 6 (1999), pp. 479–496.

[17] D. Chenais and J.-C. Paumier, *On the locking phenomenon for a class of elliptic problems*, Numerische Mathematik, 67 (1994), pp. 427–440.

[18] T. De Ryck, S. Lanthaler, and S. Mishra, *On the approximation of functions by tanh neural networks*, Neural Networks, 143 (2021), pp. 732–750.

[19] E. DiNezza, G. Palatucci, and E. Valdinoci, *Hitchhiker's guide to the fractional sobolev spaces*, Bulletin des sciences mathematiques, 136 (2012), pp. 521–573.

[20] R. Durán and E. Lieberman, *On mixed finite element methods for the reissner-mindlin plate model*, Mathematics of computation, 58 (1992), pp. 561–573.

[21] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural networks, 4 (1991), pp. 251–257.

[22] T. J. Hughes and L. P. Franca, *A mixed finite element formulation for reissner-mindlin plate theory: Uniform convergence of all higher-order spaces*, Computer Methods in Applied Mechanics and Engineering, 67 (1988), pp. 223–240.

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

[24] F. Johansson et al., *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.14)*, February 2010. `http://code.google.com/p/mpmath/`.

[25] E. Kharazmi, Z. Zhang, and G. E. M. Karniadakis, *hp-VPINNs: variational physics-informed neural networks with domain decomposition*, Comput. Methods Appl. Mech. Engrg., 374 (2021), pp. Paper No. 113547, 25.

[26] M. Liu and Z. Cai, *Adaptive two-layer relu neural network: Ii. ritz approximation to elliptic pdes*, arXiv preprint arXiv:2107.06459, (2021).

[27] C. Ma, L. Wu, et al., *The barron space and the flow-induced function spaces for neural network models*, Constructive Approximation, (2021), pp. 1–38.

[28] J. A. Opschoor, P. C. Petersen, and C. Schwab, *Deep relu networks and high-order finite element methods*, Analysis and Applications, 18 (2020), pp. 715–770.

[29] M. L. Piscopo, M. Spannowsky, and P. Waite, *Solving differential equations with neural networks: Applications to the calculation of cosmological phase transitions*, Physical Review D, 100 (2019), p. 016002.

[30] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics, 378 (2019), pp. 686–707.

[31] S. A. Sauter and C. Schwab, *Boundary element methods*, in Boundary Element Methods, Springer, 2010, pp. 183–287.

[32] J. W. Siegel and J. Xu, *Approximation rates for neural networks with general activation functions*, Neural Networks, 128 (2020), pp. 313–321.

[33] J. Sirignano and K. Spiliopoulos, *Dgm: A deep learning algorithm for solving partial differential equations*, Journal of computational physics, 375 (2018), pp. 1339–1364.

[34] R. Stenberg, *A new finite element formulation for the plate bending problem*, Asymptotic methods for elastic structures, (1995), pp. 209–221.

[35] R. Stenberg and M. Suri, *An hp error analysis of mitc plate elements*, SIAM journal on numerical analysis, 34 (1997), pp. 544–568.

[36] M. Suri, I. Babuška, and C. Schwab, *Locking effects in the finite element approximation of plate models*, Mathematics of computation, 64 (1995), pp. 461–482.

[37] R. Verfürth, *A posteriori error estimation and adaptive mesh-refinement techniques*, Journal of Computational and Applied Mathematics, 50 (1994), pp. 67–83.

[38] E. Weinan and B. Yu, *The deep ritz method: A deep learning-based numerical algorithm for solving variational problems*, Communications in Mathematics and Statistics, 6 (2018).

[39] X. Yang, S. Zafar, J.-X. Wang, and H. Xiao, *Predictive large-eddy-simulation wall modeling via physics-informed neural networks*, Physical Review Fluids, 4 (2019), p. 034602.

[40] Y. Zang, G. Bao, X. Ye, and H. Zhou, *Weak adversarial networks for high-dimensional partial differential equations*, Journal of Computational Physics, 411 (2020), p. 109409.