A DEEP NEURAL NETWORK-BASED NUMERICAL METHOD FOR SOLVING CONTACT PROBLEMS

Article · June 2022

DOI: 10.23952/jnva.6.2022.5.04

5 authors, including:

Shen Xing
Zhejiang University
7 PUBLICATIONS **114** CITATIONS

SEE PROFILE

J. Nonlinear Var. Anal. 6 (2022), No. 5, pp. 483-498
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.6.2022.5.04

A DEEP NEURAL NETWORK-BASED NUMERICAL METHOD FOR SOLVING CONTACT PROBLEMS

XING SHEN, XIAOLIANG CHENG*, KEWEI LIANG, XILU WANG, ZHENGHUA WU

School of Mathematical Sciences, Zhejiang University, Hangzhou, Zhejiang, China

Abstract. In this paper, we propose a deep neural network-based numerical method for solving contact problems. Focusing on a static frictionless unilateral contact problem, we derive its weak formulation and prove that the solution of the weak formulation is also the minimizer of the corresponding energy functional. By converting the original contact problem into a minimization problem, a deep neural network is adopted to approximate the solution and solve the minimization problem. Numerical results demonstrate the effectiveness and accuracy of our method.

Keywords. Deep neural networks; Frictionless unilateral contact problem; Hemivariational inequality.

1. INTRODUCTION

A contact problem occurs when at least two bodies not mechanically joined touch each other without becoming rigidly attached. Contact processes appear in industry and everyday life, such as brake pads in contact with wheels, tires on roads, and pistons with skirts. Due to the importance of contact processes in structural and mechanical systems, the contact problem has received a lot of attention. After Panagiotopoulos' pioneering work on hemivariational inequalities [1], mathematical theories and numerical analysis of contact models in the form of hemivariational inequality were established extensively and thoroughly in recent decades; see, e.g., [2, 3, 4, 5] and the references therein. In [6], the authors discussed a bilateral contact problem with nonmonotone friction by an elliptic hemivariational inequality and derived the error estimate for its finite element solutions. In [7], a class of variational-hemivariational inequalities was studied theoretically and numerically. For the linear finite element solutions of the problem, an optimal first-order error estimate was derived. In [8], three representative mathematical models for the contact problem with elastic and viscoelastic materials were studied, and the proofs of existence and uniqueness are given, as well as the convergence analysis and error estimation of the numerical solutions.

Recently, developments in deep learning led to an explosive growth of data-driven solutions for PDEs via deep neural networks; see, e.g., [9, 10, 11, 12, 13, 14]. In these methods, DNNs are applied as a universal approximator to parametrize PDE solutions, and appropriate parameters are identified by minimizing an optimization problem constructed from the given PDE and

*Corresponding author.
E-mail addresses: shenxingsx@zju.edu.cn (X. Shen), xiaoliangcheng@zju.edu.cn (X. Cheng), matlkw@zju.edu.cn (K. Liang), xiluwang@zju.edu.cn (X. Wu), zhenhuawu@zju.edu.cn (Z. Wu).

Received May 10, 2022; Accepted May 31, 2022.

corresponding boundary condition. Karniadakis and his team proposed physics-informed neural networks (PINNs), which solve supervised learning tasks respecting the incorporate physical information. They designed several models for different types of PDEs [11, 15]. In [14], Weinan and Yu proposed the Deep Ritz method for the numerical solution of variational problems based on the Ritz method. In [16], DeepOnets was proposed to learn nonlinear operators to identify differential equations based on the universal approximation theorem. In [17], residual networks were adopted to approximate the evolution operator for solving and recovering unknown time-dependent PDEs.

Although many deep learning-based methods have been proposed for solving different types of PDEs and have achieved remarkable results, there are few works on contact problems and the corresponding hemivariational inequalities. In [18], the authors proposed a deep learning-based method to solve the elliptic hemivariational inequalities and compared the numerical performance of three different training strategies for updating the parameters. To the best of our knowledge, there are no other deep learning-based numerical methods for solving contact problems. In this paper, we propose a deep neural network-based method for solving contact problems. We focus on a static frictionless unilateral contact problem and propose the corresponding hemivariational inequality. We prove that the solution of the hemivariational inequality is also the minimizer of a corresponding energy functional. Then we relax the restriction of boundary by adding two Lagrange terms to the energy functional. In this way, we convert the original contact problem into a minimization problem and use a deep neural network to approximate the solution. We propose an asynchronous iterative strategy to make it easier for the model to converge. Our method has the advantages of being unsupervised, meshless, and easy to implement. Finally, two numerical experiments demonstrate the effectiveness and precision of our method.

This paper is organized as follows. In Section 2, we introduce some basic notions and definitions. In Section 3, we propose the contact problem and the corresponding hemivariational inequality and prove that the solution of the hemivariational inequality is also the minimizer of a corresponding energy functional. In Section 4, the deep neural network framework is presented in detail. In Section 5, two numerical examples are presented to illustrate the performance of the proposed method.

2. PRELIMINARIES

We first present some necessary notations. Let X be a Banach space and $\varphi : X \to \mathbb{R}$ be a locally Lipschitz function. From [19], the generalized (Clarke) directional derivative of φ at $x \in X$ in the direction $v \in X$ is defined by

$$\varphi^0(x;v) = \limsup_{y \to x, t \downarrow 0} \frac{\varphi(y+tv)-\varphi(y)}{t}.$$

The generalized gradient of φ at x is a subset of a dual space X^*, given by $\partial\varphi(x) = \{\, \zeta \in X^* \mid \varphi^0(x;v) \geq \langle \zeta, v\rangle_{X^*\times X} \text{ for all } v \in X\}$. Let $\Phi : X \to \mathbb{R}\cup\{+\infty\}$ be proper, convex, and l.s.c. (lower semicontinuous). Then the (convex) subdifferential of Φ at $x \in X$ is

$$\partial\Phi(x) = \{\, \xi \in X^* \mid \Phi(v) - \Phi(x) \geq \langle \xi, v-x\rangle_{X^*\times X} \text{ for all } v \in X\}.$$

Let $\mathbb{S}^d$ be the linear space of second-order symmetric tensors on $\mathbb{R}^d$. The inner product and the corresponding norm on $\mathbb{S}^d$ are:

$$\begin{aligned} u\cdot v &= u_i v_i, \qquad \|v\| = (v\cdot v)^{1/2} \quad \text{for all } u=(u_i), v=(v_i)\in\mathbb{R}^d.\\ \sigma\cdot\tau &= \sigma_{ij}\tau_{ij}, \quad \|\tau\| = (\tau\cdot\tau)^{1/2} \quad \text{for all } \sigma=(\sigma_{ij}), \tau=(\tau_{ij})\in\mathbb{S}^d. \end{aligned}$$

We consider a domain Ω, which is open, bounded, and connected in $\mathbb{R}^d$. The boundary $\Gamma = \partial\Omega$ is partitioned into three disjoint and measurable parts Γ_1, Γ_2, and Γ_3 such that $\text{meas}(\Gamma_1) > 0$ and $\text{meas}(\Gamma_3) > 0$. We assume that the boundary $\Gamma = \partial\Omega$ is Lipschitz continuous. Then the unit outward normal vector ν is defined a.e. on Γ. For a vector field v, the normal and tangential components of v are denoted by $v_\nu = v\cdot\nu$ and $v_\tau = v - v_\nu\nu$. Similarly, for a tensor field σ, the normal and tangential components are denoted by $\sigma_\nu = (\sigma\nu)\cdot\nu$ and $\sigma_\tau = \sigma\nu - \sigma_\nu\nu$. The corresponding space is defined as follows

$$\begin{aligned} &H = L^2\left(\Omega;\mathbb{R}^d\right), \quad \mathscr{H} = \left\{\tau=\{\tau_{ij}\} : \tau_{ij}=\tau_{ji}\in L^2(\Omega)\right\} = L^2\left(\Omega;\mathbb{S}^d\right).\\ &H_1 = \{u\in H : \varepsilon(u)\in\mathscr{H}\} = H^1\left(\Omega;\mathbb{R}^d\right), \quad \mathscr{H}_1 = \{\tau\in\mathscr{H} : \text{div}\,\tau\in H\}, \end{aligned}$$

where the deformation operator ε and divergence operator Div are defined as

$$\varepsilon_{ij}(u) = (\varepsilon(u))_{ij} = \frac{1}{2}\left(u_{i,j}+u_{j,i}\right),$$
$$\text{Div}\,\sigma = \left(\sum_{j=1}^{d}\sigma_{ij,j}\right).$$

The index following comma indicates a partial derivative. Let V be a Hilbert space with the inner product

$$(u,v)_V := \int_\Omega \varepsilon(u):\varepsilon(v)dx, \quad \forall u,v\in V,$$

and the associated norm $\|\cdot\|_V$ is equivalent to the standard H^1 norm over V. For $v\in V$, we use the same symbol v for its trace on Γ. The dual of V is denoted by V^*, and the duality pairing of V and V^* is denoted by $\langle\cdot,\cdot\rangle$.

Let $Q = L^2(\Omega;\mathbb{S}^d)$. It is a Hilbert space with the canonical inner product

$$(\sigma,\tau)_Q := \int_\Omega \sigma_{ij}(x)\tau_{ij}(x)dx, \quad \forall\ \sigma,\tau\in\mathbb{S}^d$$

and the associated norm $\|\cdot\|_Q$.

3. Contact Problems

In this section, we focus on a static frictionless unilateral contact problem. Followed by [20, 21], we present the weak form of the contact model and the unique solvability, and we prove that the weak form is equivalent to a minimization problem.

Now, we consider the following contact model. The dependence on the spatial variable x is not always indicated to simplify the notation.

Problem *P Find a displacement field* $u : \Omega \to \mathbb{R}^d$, *a stress field* $\sigma : \Omega \to \mathbb{S}^d$ *such that*

$$\sigma = \mathscr{A}\varepsilon(u) \quad \text{in} \quad \Omega, \tag{3.1}$$

$$\mathrm{Div}\,\sigma + f_0 = 0 \quad \text{in} \quad \Omega, \tag{3.2}$$

$$u = 0 \quad \text{on} \quad \Gamma_1, \tag{3.3}$$

$$\sigma\nu = f_2 \quad \text{on} \quad \Gamma_2, \tag{3.4}$$

$$u_\nu \le g, \quad \sigma_\nu + \xi_\nu \le 0, \quad (u_\nu - g)(\sigma_\nu + \xi_\nu) = 0, \quad \xi_\nu \in \partial j(u_\nu) \quad \text{on} \quad \Gamma_3, \tag{3.5}$$

$$\sigma_\tau = 0 \quad \text{on} \quad \Gamma_3.$$

Equation (3.1) represents the constitutive law of elastic material. Equation (3.2) is the normalized equilibrium equation for the static process. A total body force of density f_0 is applied in the body Ω. Boundary conditions (3.3)-(3.4) indicate that the body is fixed on Γ_1, and is in equilibrium under the action of a surface traction of density f_2 on Γ_2. The relations (3.5) model a frictionless contact with a foundation made of a rigid body covered by a layer of an elastic material with thickness $g > 0$. The penetration is allowed but is restricted by the relation $u_\nu \le g$. This contact model is a simplified version of that in [8] with the friction term on Γ_3 being dropped here.

Then we present the weak form of Problem P and its unique solvability. Since it is the result proved in [8], we skip the details. The hypotheses on the data are listed.

H($\mathscr{A}$): The elasticity tensor $\mathscr{A} : \Omega \times \mathbb{S}^d \to \mathbb{S}^d$ is in the form of $\mathscr{A} = (A_{ijkl})_{1\le i,j,k,l\le d}$ and satisfies:

(i) $A_{ijkl} = A_{jikl} = A_{klij},\ 1 \le i,j,k,l \le d$;

(ii) $A_{ijkl} \in L^\infty(\Omega),\ 1 \le i,j,k,l \le d$;

(iii) there eixsts $m_{\mathscr{A}} > 0$ such that $(\mathscr{A}\varepsilon)\ \varepsilon \ge m_{\mathscr{A}}|\varepsilon|^2,\ \forall \varepsilon \in \mathbb{S}^d$.

H(j): The function $j : \Gamma_3 \times \mathbb{R} \to \mathbb{R}$ satisfies:

(i) $j(\cdot, r)$ is measurable on Γ_3 for all $r \in \mathbb{R}$ and $j(\cdot, 0) \in L^1(\Gamma_3)$;

(ii) $j(x, \cdot)$ is locally Lipschitz on $\mathbb{R}$ for a.e. $x \in \Gamma_3$;

(iii) $|\partial j(x, r)| \le \bar{c}_0 + \bar{c}_1|r|$ for a.e. $x \in \Gamma_3$, for all $r \in \mathbb{R}$ with $\bar{c}_0, \bar{c}_1 \ge 0$;

(iv) there eixsts $m_j \ge 0$ such that $j^0(x, r_1; r_2 - r_1) + j^0(x, r_2; r_1 - r_2) \le m_j|r_1 - r_2|^2$ for a.e. $x \in \Gamma_3$, all $r_1, r_2 \in \mathbb{R}$.

H(f): The densities of body forces and surface tractions satisfy:

$$f_0 \in L^2(\Omega; \mathbb{R}^d), \quad f_2 \in L^2(\Gamma_2; \mathbb{R}^d).$$

We also need the space V and set K

$$\begin{aligned} V &= \{v \in H^1(\Omega; \mathbb{R}^d) \mid v = 0 \text{ a.e. on } \Gamma_1\}, \\ K &= \{v \in V \mid v_\nu \le g \text{ on } \Gamma_3\}. \end{aligned}$$

Then we define the function $f \in V^*$ by

$$\langle f, v\rangle = \int_\Omega f_0 \cdot v dx + \int_{\Gamma_2} f_2 \cdot v d\Gamma, \quad \forall v \in V.$$

Through a standard derivation, we have the weak form of Problem P.

Problem P_V. *Find a displacement field $u \in K$ such that*

$$(\mathscr{A}\varepsilon(u), \varepsilon(v-u))_Q + \int_{\Gamma_3} j^0(u_\nu; v_\nu - u_\nu) d\Gamma \geq \langle f, v-u\rangle, \quad \forall v \in K.$$

Let $c_j = \lambda_j^{-1/2}$, where $\lambda_j > 0$ is the smallest eigenvalue of the eigenvalue problem

$$u \in V, \quad \int_\Omega \varepsilon(u)\varepsilon(v) dx = \lambda_j \int_{\Gamma_3} u_\nu v_\nu da, \quad \forall v \in V.$$

Then

$$\|v_\nu\|_{L^2(\Gamma_3)} \leq c_j \|v\|_V, \ \forall v \in V.$$

The unique solvability for Problem P_V is provided in [8].

Theorem 3.1. *Assume H($\mathscr{A}$), H(j), H(f), and*

$$m_{\mathscr{A}} > m_j c_j^2. \tag{3.6}$$

Then Problem P_V has a unique solution.

Now we consider a minimization problem, which is equivalent to Problem P_V. For this purpose, we further define the operator $A : V \to V^*$ by

$$\langle Au, v\rangle = (\mathscr{A}\varepsilon(u), \varepsilon(v))_Q,$$

and the functional $J : V \to \mathbb{R}$ by

$$J(u) = \int_{\Gamma_3} j(u_\nu) d\Gamma, \quad \forall u \in V.$$

We have the following results.

Lemma 3.1. *The operator $A : V \to V^*$ is symmetric, Lipschitz continuous, and strongly monotone. $\langle Av_1 - Av_2, v_1 - v_2\rangle \geq m_{\mathscr{A}} \|v_1 - v_2\|_V^2$, $\forall v_1, v_2 \in V$. The functional $J : V \to \mathbb{R}$ is locally Lipschitz continuous, and*

$$J^0(v_1; v_2 - v_1) + J^0(v_2; v_1 - v_2) \leq m_j c_j^2 \|v_1 - v_2\|_V^2, \quad \forall v_1, v_2 \in V. \tag{3.7}$$

Proof. To prove (3.7), we apply the property ([5], Section 3.3)

$$J^0(u; v) \leq \int_{\Gamma_3} j^0(u_\nu; v_\nu) d\Gamma, \quad \forall u, v \in V. \tag{3.8}$$

Then we can obtain

$$\begin{aligned} J^0(v_1; v_2 - v_1) + J^0(v_2; v_1 - v_2) &\leq \int_{\Gamma_3} j^0(v_{1\nu}; v_{2\nu} - v_{1\nu}) + j^0(v_{2\nu}; v_{1\nu} - v_{2\nu}) d\Gamma \\ &\leq \int_{\Gamma_3} m_j |v_{1\nu} - v_{2\nu}|^2 d\Gamma \\ &\leq m_j c_j^2 \|v_1 - v_2\|_V^2. \end{aligned}$$

The other results are easy to check, and we omit the proof. □

Note that (3.7) is equivalent to the condition

$$\langle \xi_1 - \xi_2, v_1 - v_2 \rangle \geq -m_j c_j^2 \|v_1 - v_2\|_V^2, \quad \forall\, \xi_i \in \partial J(v_i),\ v_i \in V,\ i = 1,2. \tag{3.9}$$

We quote two necessary lemmas about the strong convexity of a locally Lipschitz continuous function ([20]).

Lemma 3.2. *Let V be a real Banach space, K be a non-empty convex set in V, and $g : K \to \mathbb{R}$ be locally Lipschitz continuous. Then g is strongly convex on K with a constant $\alpha > 0$, i.e.,*

$$g(tu + (1-t)v) \leq tg(u) + (1-t)g(v) - \alpha t(1-t)\|u - v\|_V^2,\ \forall u, v \in K,\ t \in [0,1],$$

if and only if ∂g is strongly monotone on K with a constant 2α, that is,

$$\langle \xi - \eta, u - v \rangle \geq 2\alpha \|u - v\|_V^2, \quad \forall u, v \in K,\ \xi \in \partial g(u),\ \eta \in \partial g(v).$$

Lemma 3.3. *Let V be a real Hilbert space, K be a non-empty closed convex set in V , and $g : K \to \mathbb{R}$ be a locally Lipschitz continuous and strongly convex functional on K with a constant $\alpha > 0$. Then g is coercive on K.*

The following lemma shows the unique solvability for a minimization problem ([22]).

Lemma 3.4. *Assume that V is a reflexive Banach space, and $K \subset V$ is convex and closed. If $F : K \to \mathbb{R}$ is convex, l.s.c, and coercive on K, then the minimization problem*

$$\inf_{v \in K} F(v)$$

has a solution. Furthermore, if F is strictly convex, then the solution is unique.

Now, we present the minimization problem.

Problem P_M. *Find a displacement field $u \in K$ such that $u = \arg\min_{v \in K} E(v)$,* where

$$E(v) = \frac{1}{2}\langle Av, v \rangle + J(v) - \langle f, v \rangle.$$

For the equivalence between Problems P_V and Problem P_M, we have the following important theorem.

Theorem 3.2. *Problem P_M has a unique solution, which is also the unique solution of Problem P_V.*

Proof. We first prove that E is locally Lipschitz and strongly convex on V. Since each term in E is locally Lipschitz on V, so is E. For the strong convexity of E, by Lemma 3.2, it is sufficient to prove the strong monotonicity of ∂E. For any $v_i \in V$, let $\zeta_i \in \partial E(v_i),\ i = 1,2$. Since

$$\partial E(v) = Av + \partial J(v) - f, \quad \forall v \in V,$$

we can write

$$\zeta_i = Av_i + \xi_i - f, \quad \xi_i \in \partial J(v_i).$$

Using Lemma 3.1 and (3.9), we have

$$\begin{aligned} \langle \zeta_1 - \zeta_2, v_1 - v_2 \rangle &= \langle Av_1 - Av_2, v_1 - v_2 \rangle + \langle \xi_1 - \xi_2, v_1 - v_2 \rangle \\ &\geq (m_{\mathscr{A}} - m_j c_j^2)\|v_1 - v_2\|_V^2. \end{aligned}$$

Combining with the smallness condition (3.6), we obtain that ∂E is strongly monotone on V. Thus E is strongly convex on V. Applying Lemma 3.3, we know that E is also coercive on K. By Lemma 3.4, we conclude that Problem P_M has a unique solution u_M. The unique solution u_M is characterized by the condition

$$0 \in \partial(E(u_M) + I_K(u_M)) \subset Au_M + \partial J(u_M) - f + \partial I_K(u_M),$$

where I_K is the indicator function of the set K. Then we deduce

$$(Au_M, v - u_M) + J^0(u_M; v - u_M) - \langle f, v - u_M \rangle + I_K(v) - I_K(u_M) \geq 0, \quad \forall v \in V,$$

which implies

$$u_M \in K, \quad (Au_M, v - u_M) + J^0(u_M; v - u_M) - \langle f, v - u_M \rangle \geq 0, \quad \forall v \in K.$$

Using property (3.8) again, we have

$$u_M \in K, \quad (Au_M, v - u_M) + \int_{\Gamma_3} j^0(u_{Mv}; v_v) d\Gamma \geq \langle f, v - u_M \rangle, \quad \forall v \in K.$$

It means that u_M solves Problem P_V. By Theorem 3.1, Problem P_V has a unique solution. Thus we conclude that the unique solution u_M of Problem P_M is also the unique solution to Problem P_V. □

4. Deep Neural Network Framework

In this section, we propose a deep learning-based framework for solving the contact problem mentioned above. Our framework consists of the following three parts:

- neural network architecture for constructing the objective function;
- loss function derived from the contact problem;
- optimization algorithm for the loss function.

4.1. Neural network architecture.

Our aim is to train a deep neural network $u^\theta(x)$ to approximate the displacement field $u(x)$ for $\forall x \in \Omega$, where θ denotes the trainable parameters of the network, including weights and biases.

In this work, we adopt the residual network architecture, which is inspired by the Deep Ritz Method [14]. It consists of several residual blocks, and each block is constructed by two linear layers, two nonlinear activation functions and a skip connection. The i-th block is described as:

$$block_i(s) = \phi(W_{i,2} \cdot \phi(W_{i,1}x + b_{i,1}) + b_{i,2}) + s,$$

where $W_{i,1}, W_{i,2} \in R^{m \times m}, b_{i,1}, b_{i,2} \in R^m$, and ϕ is the activation function. We visualize the residual block in Figure 1.

Note that the input of the model is $x \in R^d$, so we add another fully-connect layer to adjust the output of the first layer to R^m, i.e.,

$$block_0(x) = W_0 \cdot x + b_0,$$

where $W_0 \in R^{m \times d}$. The last layer of the last layer of the network is a fully connected layer,

$$block_n(s) = W_n \cdot s + b_n,$$

where $W_n \in R^{d \times m}, b_n \in R^d$.

In summary, the deep neual network $u^\theta(\cdot) : \Omega \to \mathbb{R}^d$ can be formulated as

$$u^\theta(x) = block_n \circ \ldots \circ block_0(x).$$

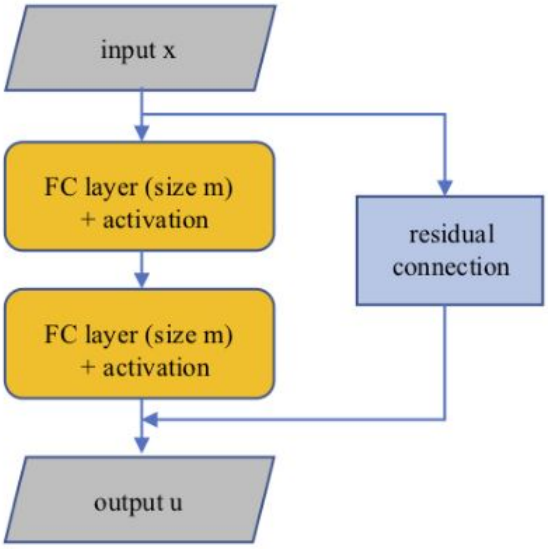


FIGURE 1. Residual Block.

In this work, we adopt a variant of ReLU, LeakyReLU, as the activation function

$$\phi(x) = \begin{cases} x & if\ x \geq 0, \\ ax & if\ x < 0. \end{cases}$$

The parameter a in LeakyReLU is a small positive number, and we set $a = 0.1$ by default. In numerical experiments, we find that LeakyReLU significantly outperforms ReLU, tanh, and $ReLU^3$. Furthermore, our method is not limited to the current model architecture, other model architectures are also feasible, such as [23].

4.2. **Loss function.** As we discussed in Section 3, contact problem P is equivalent to Problem P_M. Note that Problem P_M can be regarded as a constrained optimization problem

$$\begin{aligned} \min \quad & E(v) = \tfrac{1}{2}\langle Av, v\rangle + J(v) - \langle f, v\rangle. \\ \text{subject to} \quad & v = 0 \text{ on } \Gamma_1 \\ & v_\nu \leq g \text{ on } \Gamma_3. \end{aligned}$$

Inspired by [24], we convert the above problem into the following minimization problem by introducing two Lagrange terms

$$\min_{v \in H^1(\Omega;\mathbb{R}^d)} \mathbb{L}(v) = \frac{1}{2}\langle Av, v\rangle + J(v) - \langle f, v\rangle + \lambda_1 \int_{\Gamma_1} |v|^2 dx + \lambda_2 \int_{\Gamma_3} \max(0, v_\nu - g) dx,$$

where λ_1 and λ_2 are the corresponding weights of Lagrange terms. So far, we use deep neural networks to solve the above problem, a.e.

$$\begin{aligned} \min_\theta \mathbb{L}(\theta) = & \frac{1}{2}\langle Au^\theta, u^\theta\rangle + J(u^\theta) - \langle f, u^\theta\rangle \\ & + \lambda_1 \int_{\Gamma_1} |u^\theta|^2 dx + \lambda_2 \int_{\Gamma_3} \max(0, u^\theta_\nu - g) dx. \end{aligned}$$

We then introduce the discrete form of $\mathbb{L}$

$$\min_{\theta} \hat{\mathbb{L}}(\theta) = \hat{\mathbb{L}}_{main}(\theta) + \hat{\mathbb{L}}_{lagrange}(\theta), \tag{4.1}$$

where

$$\hat{\mathbb{L}}_{main}(\theta) = \frac{1}{2N_\Omega}\sum_{i=1}^{N_\Omega} \mathscr{A}\varepsilon(u_i^{\theta,2}) + \frac{1}{N_3}\sum_{i=1}^{N_3} j(u_{\nu,i}^{\theta}) - \frac{1}{N_2}\sum_{i=1}^{N_2}(f_2 \cdot u_i^{\theta}) - \frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega}(f_0 \cdot u_i^{\theta}),$$

and

$$\hat{\mathbb{L}}_{lagrange}(\theta) = \frac{\lambda_1}{N_1}\sum_{i=1}^{N_1} ||u^{\theta}||_2^2 + \frac{\lambda_2}{N_3}\sum_{i=1}^{N_3} \max(u_{\nu,i}^{\theta} - g)$$

where $(\cdot)_i$ represents the value at i-th sampled point, and N_Ω, N_1, N_2, N_3 are the number of sampled points in the domain Ω, on the boundary Γ_1, Γ_2 and Γ_3, respectively, and $(\cdot)_\nu$ represents the outer normal component.

In practical computation, the above loss function includes the calculation of the partial derivative with respect to u^θ, we use the automatic differentiation package in PyTorch to calculate it.

4.3. **Optimization algorithm.** Through the above analysis, we only need to train the neural network and update the parameter θ to make the loss function (4.1) small enough to obtain a numerical solution $u(\theta)$ that is sufficiently close to the solution of the original contact problem. The stochastic gradient descent(SGD) method is a common choice in machine learning to update the model parameters. The SGD algorithm is described as follows:

$$\theta^{k+1} = \theta^k - \eta \frac{\partial \hat{\mathbb{L}}\gamma^k}{\partial \theta},$$

where η is the learning rate, and $\{\gamma^k\}$ is a subset of the sampled training points. It is worth mentioning that the training data of our method can be randomly sampled which does not require fixed grids or measurement data, so our method is mesh-less and unsupervised. In this work, training data is randomly sampled from a uniform distribution on $\Omega, \Gamma_1, \Gamma_2$, and Γ_3. We use the Adam optimizer version of the SGD algorithm and take 'mini-batch' to compute the gradient of the loss function at each iteration. In this work, the batch size on $\Omega, \Gamma_1, \Gamma_2$, and Γ_3 defaults to $2048, 256, 256$, and 256, respectively.

During the numerical experiments, we find that the gradients of $\hat{\mathbb{L}}_{main}$ and the Lagrangian terms $\hat{\mathbb{L}}_{Lagrange}$ conflict with each other, which makes the optimization process unstable. In addition, adjusting the parameters λ_1 and λ_3 does not overcome this problem. Motivated by asynchronous optimization in reinforcement learning, we propose an asynchronous iteration strategy for this minimization problem.

Instead of directly updating the parameters according to 4.3, we update the main part and Lagrange part separately. That is, every time the Lagrangian part is updated, the main part is updated N times. At the beginning of the training process, with a small value of N, more updates to the Lagrange part ensure that the model converges in the right direction. As the training process progresses, after the loss value corresponding to the Lagrange part is small enough, we gradually increase N from 10 to 1200, so that the main part receives more updates, to ensure that the model can converge better. In this way, the choice of the values of λ_1 and λ_2 has little effect on the result. We choose $\lambda_1 = \lambda_2 = 1000$ by default. In addition, the total

number of updated iterations $K = 8000$, and the value of N increases by 100 every 50 iterations. The whole process is shown in the Algorithm 4.1.

Algorithm 4.1 Optimization algorithm

Require: Initial weights θ, number of iterations K, number of sampled points N, asynchronous iteration interval parameter $d(j)$, learning rate η,

1: **for** $j = 1, \ldots, K$ **do**
2: Randomly sample $\gamma_j = \{x_i\}_{i=1}^{N}$
3: Compute $\hat{\mathbb{L}}_{main}(\theta)_{\gamma_j}$
4: $\theta \longleftarrow \theta - \eta \cdot \frac{\partial \hat{\mathbb{L}}_{main}(\theta)_{\gamma_j}}{\partial \theta}$
5: **if** j mod d(j) = 0 **then**
6: compute $\hat{\mathbb{L}}_{lagrange}(\theta)_{\gamma_j}$
7: $\theta \longleftarrow \theta - \eta \cdot \frac{\partial \hat{\mathbb{L}}_{lagrange}(\theta)_{\gamma_j}}{\partial \theta}$
8: **return** $\hat{\theta} \longleftarrow \theta$

5. Numerical Examples

In this section, two numerical examples are presented to demonstrate the efficiency and accuracy of the proposed method.

5.1. **Example 1.** As shown in Figure 2, we first consider a simple contact problem where the thickness of the elastic material $g = 0$.

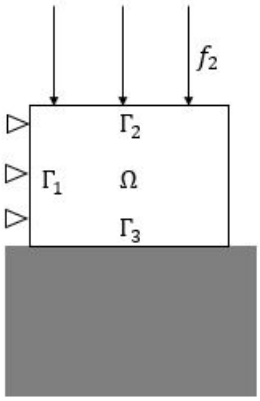


Figure 2. Contact model with elastic material thickness $g = 0$.

The boundarys are:

$$\Gamma_1 = \{0\} \times [0,1], \Gamma_2 = (0,1) \times \{1\}, \Gamma_3 = (0,1] \times \{0\},$$

and the function $j(\cdot)$ has the following form

$$j(u_\nu) = \begin{cases} 0 & if \quad u_\nu < 0, \\ 150u_\nu^2 & if \quad 0 \le u_\nu < 0.006, \\ -50u_\nu^2 + 2.4u_\nu - 0.0072 & if \quad 0.0006 \le u_\nu < 0.01, \\ 150u_\nu^2 - 1.6u_\nu + 0.0128 & if \quad u_\nu \ge 0.01. \end{cases}$$

The elasticity tensor operator $\mathscr{A}$ is

$$(\mathscr{A}\tau)_{ij} = \frac{E\kappa}{1-\kappa^2}(\tau_{11}+\tau_{22})\delta_{ij} + \frac{E}{1+\kappa}\tau_{ij}, 1 \le i,j \le 2,$$

where the Young's modulus $E = 1000N/m^2$ and the Poisson's ratio of the material $\kappa = 0.4$. The other parameters in this example are

$$\mathbf{f}_0(t) = (0,0)N/m^2,$$

$$\mathbf{f}_2(t) = \begin{cases} (0,f_2)N/m^2 & on \quad (0,1)\times\{1\}, \\ (0,0)N/m^2 & on \quad \{1\}\times(0,1). \end{cases}$$

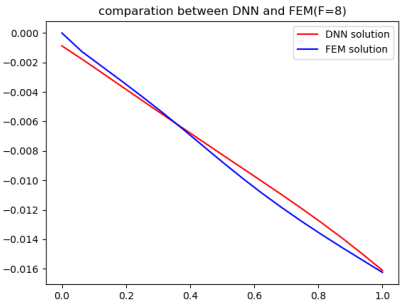
In this example, we use 5 residual blocks and the hidden size $m = 128$. We randomly sample 10000 points from a uniform distribution over the whole domain Ω, and sample 256 points on each of the three boundaries $\Gamma_1, \Gamma_2, \Gamma_3$. The learning rate is 5×10^{-3}. We train the model with one NVIDIA Tesla V100 GPU, and each experiment takes about 8 minutes.

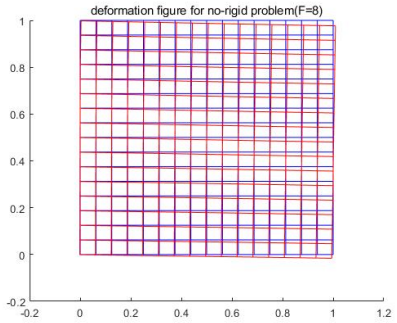
Because the contact problem usually does not have an analytical solution, we take the finite element solution with a sufficiently small grid size as the reference solution and compare it with the numerical solution of our method. As shown in Table 1, our method achieves accurate numerical solutions under different surface traction forces f_2. It is worth to mention that our method requires retraining a neural network for each input force f_2, but it is not necessary to retrain from scratch. For example, we can use the model parameters with $f = 8$ as initialization parameters for $f = 4$, which can reduce the training time by more than half.

TABLE 1. The mean square error and the maximum error of numerical solution with different f_2 when $g = 0$.

f_2	mean error	max error
8	4.6650e-06	1.4814e-05
16	3.8676e-06	2.2666e-05
32	2.7904e-05	9.8911e-05
64	5.2915e-05	0.0003

Furthermore, we visualize the outer normal component of the displacement field at the boundary Γ_3, and the displacement field u^θ obtained by our method in the whole domain Ω, see Figure (3,4,5,6).

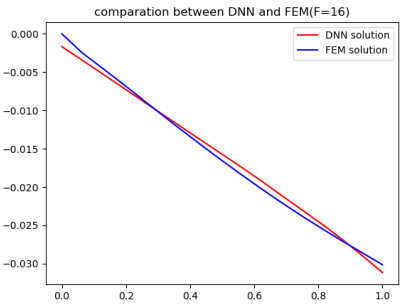


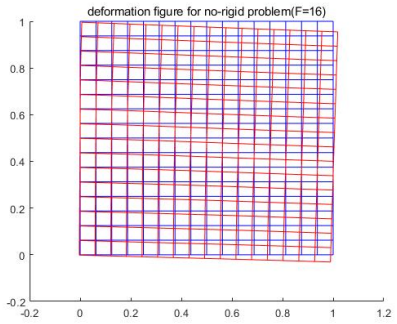


(a) Outer normal component on boundary Γ_3.

(b) Displacement field on domain Ω.

FIGURE 3. Surface traction force $f = 8$ when $g = 0$.

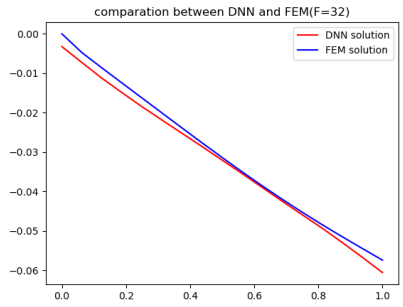


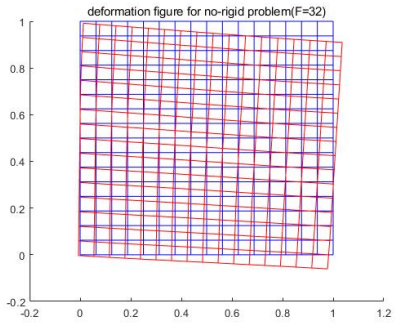


(a) Outer normal component on boundary Γ_3.

(b) Displacement field on domain Ω.

FIGURE 4. Surface traction force $f = 16$ with $g = 0$.

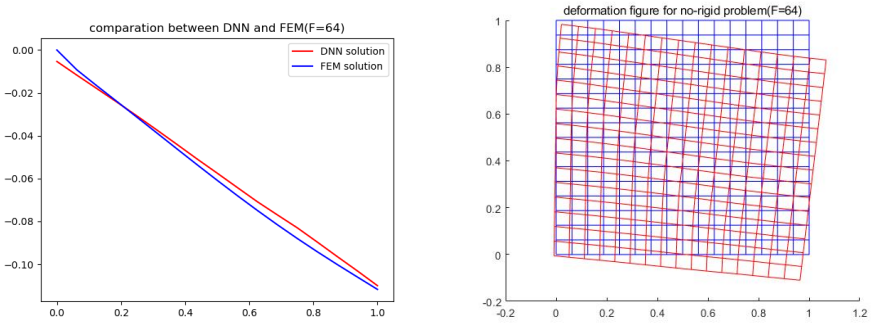




(a) Outer normal component on boundary Γ_3.

(b) Displacement field on domain Ω.

FIGURE 5. Surface traction force $f = 32$ with $g = 0$.



(a) Outer normal component on boundary Γ_3.

(b) Displacement field on domain Ω.

FIGURE 6. Surface traction force $f = 64$ with $g = 0$.

5.2. **Example 2.** In this experiment, the thickness of the elastic material is $g = 0.01$, see Figure 7.

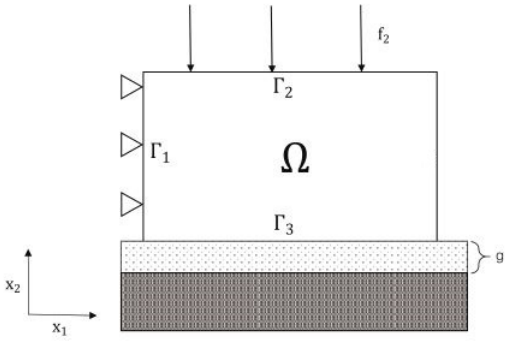


FIGURE 7. Contact model with elastic material thickness $g = 0.01$.

The function $j(\cdot)$ is

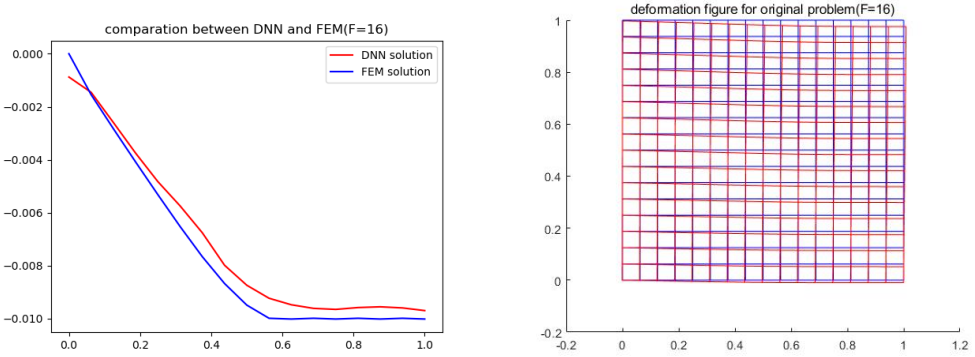
$$j(u_\nu) = \begin{cases} 0 & if \quad u_\nu < 0, \\ 150u_\nu^2 & if \quad 0 \le u_\nu < 0.006, \\ -50u_\nu^2 + 2.4u_\nu - 0.0072 & if \quad u_\nu > 0.006. \end{cases}$$

where other settings are the same as Example 1. We use the same deep neural network architecture and training strategy as Example 1. Similarity, we demonstrate the accuracy of our method with different surface traction forces $f_2 = 16, 32, 64$ on the boundary Γ_2. As shown in Table 2, our method still achieves high accuracy.

Furthermore, we visualize the outer normal component of the displacement field on the boundary Γ_3, and the displacement field u^θ on the whole domain Ω, see Figure (8,9,10).

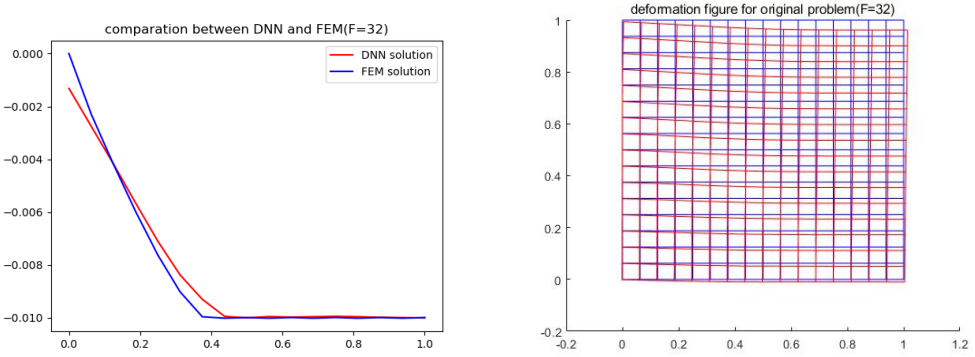
TABLE 2. The mean square error and the maximum error of numerical solution with different f_2 when $g = 0.01$.

f_2	mean error	max error
16	2.5756e-06	2.6900e-05
32	7.3964e-06	6.0122e-05
64	2.4126e-05	0.0003



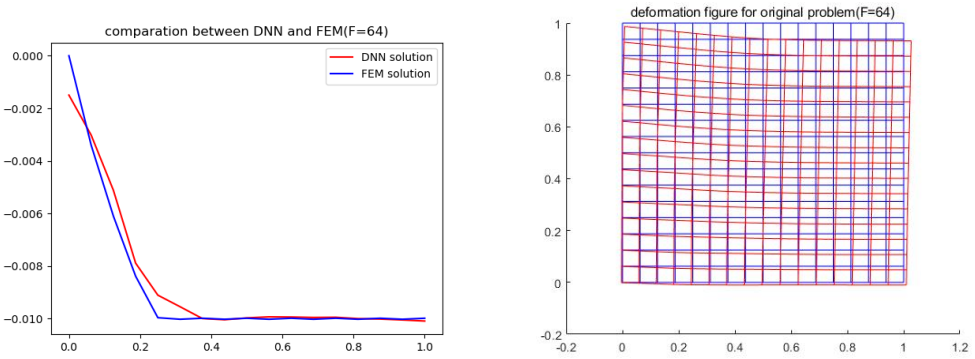
(a) Outer normal component on boundary Γ_3. (b) Displacement field on domain Ω.

FIGURE 8. Surface traction force $f = 16$ with $g = 0.01$.



(a) Outer normal component on boundary Γ_3. (b) Displacement field on domain Ω.

FIGURE 9. Surface traction force $f = 32$ with $g = 0.01$.



(a) Outer normal component on boundary Γ_3. (b) Displacement field on domain Ω.

FIGURE 10. Surface traction force $f = 64$ with $g = 0.01$.

6. CONCLUSION

In this work, we focused on a static frictionless unilateral contact problem. We first presented its weak form and the unique solvability, and then we proved the weak form is equivalent to a minimization problem. Based on the minimization problem, we proposed a deep learning framework for solving the contact problem. Numerical experiments present the efficiency and accuracy of our method. Our method has the advantages of being unsupervised, meshless, and easy to implement. We made a preliminary exploration of using neural networks to solve the contact problem. Our method has the potential to solve more complex contact problems, we will discuss more types of contact problem in future studies. One limitation of this study is that it needs to retrain the deep neural networks for different forces and thicknesses of elastic layers. Although we can speed up training with transfer learning techniques, it still requires tedious training and fine-tuning. Our further research might explore to overcome it.

REFERENCES

[1] P. D. Panagiotopoulos, Hemivariational inequalities: Applications in Mechanics Engineering, Springer-Verlag, 1993.
[2] M. Jureczka, A. Ochal, A nonsmooth optimization approach for hemivariational inequalities with applications to contact mechanics, Appl. Math. Optim. 83 (2021), 1465–1485.
[3] M.M. Makel, M. Miettinen, L. Luksan, J. Vlcek, Comparing nonsmooth nonconvex bundle methods in solving hemi-variational inequalities, J. Global Optim. 14 (1999), 117-135.
[4] M. Miettinen, M. Mkel, J. Haslinger, On numerical solution of hemivariational inequalities by nonsmooth optimization methods, J. Global Optim. 6 (1995), 401-425.
[5] S. Migorski, A. Ochal, M. Sofonea, Nonlinear inclusions and hemivariational inequalities: Models and analysis of contact problems, Vol. 26, Springer Science & Business Media, 2012.
[6] M. Barboteu, K. Bartosz, P. Kalita, An analytical and numerical approach to a bilateral contact problem with nonmonotone friction, Int. J. Appl. Math. Comput. Sci. 23 (2013), 263-276.
[7] W. Han, S. Migorski, M. Sofonea, A class of variational-hemivariational inequalities with applications to frictional contact problems, SIAM J. Math. Anal. 46 (2014), 3891-3912.
[8] W. Han , M. Sofonea, Numerical analysis of hemivariational inequalities in contact mechanics, Acta Numer. 28 (2019), 175-286.
[9] J. Han, A. Jentzen, E. Weinan, Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning, arXiv preprint arXiv: 1707.02568, 2017.

[10] Z. Long, Y. Lu, B. Dong, PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network, J. Comput. Phys. 399 (2019), 108925.
[11] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019), 686-707.
[12] J. Sirignano, K. Spiliopoulos, DGM:A deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018), 1339-1364.
[13] Y. Wang et al, Learning to discretize: Solving 1D scalar conservation laws via deep reinforcement learning, arXiv preprint arXiv: 1905.11079, 2019.
[14] E. Weinan, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, Commun. Math. Statis. 6 (2018), 1-12.
[15] L. Lu et al. DeepXDE: A deep learning library for solving differential equations, SIAM Rev. 63 (2021), 208-228.
[16] L. Lu, P. Jin, G.E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv preprint arXiv: 1910.03193, 2019.
[17] K. Wu, D. Xiu, Data-driven deep learning of partial differential equations in modal space, J. Comput. Phys. 408 (2020), 109307.
[18] J. Huang, C. Wang, H. Wang, Adaptive learning on the grids for elliptic hemivariational inequalities, arXiv preprint arXiv:2104.04881, 2021.
[19] F. Clarke, Optimization and Nonsmooth Analysis, Canad, Math. Soc. Ser. Monogr. Adv. Texts Wiley-Intersci. Publ. John Wiley & Sons, Inc., New York, 1983.
[20] W. Han, Minimization principles for elliptic hemivariational inequalities, Nonlinear Anal. 54 (2020), 103114.
[21] W. Han, M. Sofonea, M. Barboteu, Numerical analysis of elliptic hemivariational inequalities, SIAM J. Numer. Anal. 55 (2017), 640-663.
[22] W. Han, K. E. Atkinson, Theoretical Numerical Analysis: A Functional Analysis Framework, Springer, 2009.
[23] Z. Wang, Z. Zhang, A mesh-free method for interface problems using the deep learning approach, J. Comput. Phys. 400 (2020), 108963.
[24] M. Rong, D. Zhang, N. Wang, A Lagrangian dual-based theory-guided deep neural network, Complex & Intelligent Systems, (2022). https://doi.org/10.1007/s40747-022-00738-1.