



Energy-based physics-informed neural network for frictionless contact problems under large deformation

Jinshuai Bai ^{a,b}, Zhongya Lin ^a, Yizheng Wang ^{c,e}, Jiancong Wen ^{b,d}, Yinghua Liu ^c, Timon Rabczuk ^e, YuanTong Gu ^b,^{*}, Xi-Qiao Feng ^{a,*}

^a Institute of Biomechanics and Medical Engineering, AML, Department of Engineering Mechanics, Tsinghua University, Beijing, 100084, China

^b School of Mechanical, Medical and Process Engineering, Queensland University of Technology, Brisbane, 4000, Queensland, Australia

^c Department of Engineering Mechanics, AML, Tsinghua University, Beijing, 100894, China

^d Institute of Aeronautics and Astronautics, Nanchang University, Nanchang, 330031, China

^e Institute of Structural Mechanics, Bauhaus Universität Weimar, Weimar, 99423, Thuringia, Germany



ARTICLE INFO

Keywords:

Contact mechanics
Physics-informed neural network
Nonlinear computational mechanics
Minimal of potential energy

ABSTRACT

Numerical methods for contact mechanics are of great importance in engineering applications, enabling the prediction and analysis of complex surface interactions under various conditions. In this work, we propose an energy-based physics-informed neural network (PINN) framework for solving frictionless contact problems under large deformation. Inspired by microscopic Lennard-Jones potential, a surface contact energy is used to describe the contact phenomena. To ensure the robustness of the proposed PINN framework, relaxation, gradual loading and output scaling techniques are introduced. In the numerical examples, the well-known Hertz contact benchmark problem is conducted, demonstrating the effectiveness and robustness of the proposed PINN framework. Moreover, challenging contact problems with the consideration of geometrical and material nonlinearities are tested. It has been shown that the proposed PINN framework provides a reliable and powerful tool for nonlinear contact mechanics. More importantly, the proposed PINN framework exhibits competitive computational efficiency to the commercial FEM software when dealing with those complex contact problems. The codes used in this manuscript are available at https://github.com/JinshuaiBai/energy_PINN_Contact.

1. Introduction

Contact and interaction between objects are ubiquitous in nature and industrial production. Accurately simulating the deformation of bodies during contact is crucial for understanding and predicting mechanical behaviour. Contact problems are inherently nonlinear and are a fundamental aspect of solid mechanics [1]. Over the years, significant efforts have been dedicated to addressing these problems, with mesh-based methods emerging as a predominant approach [2–5]. Additionally, meshless methods, employing both weak and strong formulations, have been proposed to tackle challenges in contact mechanics [6–10].

The complexities of contact mechanics, such as contact discretisation and boundary smoothness, make it a particularly challenging area of study [11]. To overcome these challenges, isogeometric analysis (IGA) frameworks based on non-uniform rational B-splines (NURBS) have been developed. These frameworks enable exact representations of complex geometries and provide superior continuity properties [12–16]. Beyond issues of contact smoothness and discretisation, the enforcement of inequality constraints

* Corresponding authors.

E-mail addresses: yuantong.gu@qut.edu.au (Y. Gu), fengxq@tsinghua.edu.cn (X.-Q. Feng).

represents another critical aspect of contact mechanics. Various numerical techniques, such as the Lagrange multiplier method, the augmented Lagrangian formulation, and mortar methods, have been extensively developed to address this issue [17,18].

In recent years, a novel kind of deep learning (DL) framework [19,20], namely the physics-informed neural networks (PINNs), has merged as a powerful and promising tool for solving partial differential equations [21–24], and therefore attracted significant attention in the computational mechanics [25–28]. Numerous numbers of PINN-based computational solid mechanics frameworks have been proposed [29–32]. Among those frameworks, neural networks regulated by the energy form loss provide the most robust results with favourable efficiency. Those energy-based PINN frameworks are also known as the deep energy method (DEM) [31]. Great performances of those PINN frameworks have been witnessed in a variety of applications, including linear elasticity [31,33], hyperelasticity [34], elastoplasticity [35,36], fracture [37–39], large deformation problems [40], and topology optimisation [41–44]. Besides, adaptive sampling strategies [45,46] have been proposed to avoid the overfitting issue and enhance the accuracy of predictions. In addition, traditional PDE solving techniques have been introduced to the energy-based PINN frameworks, ending up with establishing novel methods for solid mechanics challenges. For example, complementary energy form and boundary integration form lead to novel deep complementary energy method [47] and boundary-informed neural network frameworks [48], respectively. Novel neural network structures are also implemented. Bai et al. [40] applied radial basis networks to deal with problems involving both material and geometrical nonlinearity. It has been demonstrated that such a framework can easily capture instability phenomena and is locking-free when modelling nearly incompressible materials. Wang et al. [49] implement the Kolmogorov Arnold network (KAN) with physics knowledge. By doing so, the KAN shows exceptional performances for multi-scale and material discontinuity problems.

Despite its notable success in computational mechanics, only limited exploration regarding PINNs has been conducted on contact problems. Sahin et al. [50] proposed a PINN framework under the strong form formulation with the classical Karush–Kuhn–Tucker constraint to solve the forward and inverse contact problems. Efforts have also been directed towards addressing interfacial discontinuities arising from the consideration of multi-material systems [51]. However, in those existing and limited literature, either small deformation problems are discussed or the contact surfaces remain unchanged throughout the training, e.g., neither new contact area generated nor sliding between contact surfaces is considered. It has been demonstrated that the PINN framework is very effective for material nonlinearities and large deformation problems [34,40]. Besides, the energy-based PINNs are computationally more efficient and robust than the vanilla PINNs [32]. Therefore, it is of great interest to implement energy-based PINNs to solve contact problems. Moreover, the effectiveness and performance of the energy-based PINNs for contact problems with complex nonlinearities are also worth investigating.

To address this research gap, in this work, we propose an energy-based PINN framework for contact problems with both material and geometrical nonlinearities. The contact behaviour is described by a contact potential and together optimised by gradient descendant algorithms. Moreover, additional numerical treatments, including the relaxation, gradual loading and output scaling are also introduced to improve the robustness of the proposed framework. The Hertz contact benchmark problems under small deformation assumption are examined to show the performances of the proposed framework. More importantly, various contact examples considering both the material and the geometrical nonlinearities are tested, including the rubber ironing, the rubber ring contact instability and the compression of two rings. The deformed configurations and stress contours demonstrate the effectiveness and substantial potential of the proposed PINN framework for complex nonlinear mechanics. Notably, the proposed framework is very straightforward to implement numerically and can easily incorporate experimental data. Additionally, the PINN framework demonstrates competitive computational efficiency compared to commercial FEM software when addressing complex contact problems.

The remainder of the paper is organised as follows: In Section 2, the basic conceptions of energy-based PINN for solid mechanics are briefly introduced. In Section 3, the frictionless contact achieved by using the surface potential and its implementation based on the PINN framework are proposed. The Hertz contact benchmark problem is also examined and discussed in detail. In Section 4, numerical examples involving large deformation and material nonlinearities are presented, including the rubber ironing, the rubber ring contact instability and the compression of two rings. In Section 5, the conclusions of this work are summarised.

2. Energy-based PINN for nonlinear solid mechanics

2.1. Loss function informed by potential energy

We first recap the basic conceptions of the energy-based PINN in solid mechanics. In this method, neural networks are used to approximate the admissible displacement solution $\mathbf{u}(\mathbf{x})$ satisfying the essential boundary condition:

$$\begin{aligned}\mathbf{u}(\mathbf{x}) &\approx \mathbf{F}(\mathbf{x}; \theta) \\ \mathbf{u}(\mathbf{x}) &= \bar{\mathbf{u}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma^u\end{aligned}\tag{1}$$

where $\mathbf{F}(\cdot)$ denotes a neural network mapping and θ denotes the trainable parameters in the network. $\bar{\mathbf{u}}(\mathbf{x})$ is the prescribed essential boundary conditions on essential boundary Γ^u . Note that the neural network structure has been introduced in many PINN-based computational mechanics literature, and different types of neural networks can be found in [52–54]. Referring to the principle of minimum potential energy, the solution to the solid mechanics problem can be obtained by finding a $\mathbf{u}(\mathbf{x})$ that minimises the potential energy of the solid system

$$\mathbf{u}(\mathbf{x}) = \arg \min_{\mathbf{u}(\mathbf{x})} \Pi(\mathbf{u}(\mathbf{x})),\tag{2}$$

where $\Pi(\cdot)$ is the overall potential energy functional and can be calculated by

$$\Pi = E_{\text{in}} - E_{\text{ex}}, \quad (3)$$

where E_{in} and E_{ex} are the strain energy and the potential energy of external forces, respectively. For elastic material, the strain energy can be calculated by

$$E_{\text{in}} = \int_{\Omega} \Psi(\mathbf{F}) d\Omega, \quad (4)$$

where $\Psi(\cdot)$ and \mathbf{F} are the strain energy density and deformation gradient tensor, respectively [55]. The potential energy of external forces can be obtained by

$$E_{\text{ex}} = \int_{\Gamma^t} \bar{\mathbf{t}} \cdot \mathbf{u} d\Gamma, \quad (5)$$

where $\bar{\mathbf{t}}$ is the given traction force on the natural boundary Γ^t . Many weak-form computational mechanics methods are established based on the principle of minimum potential energy, such as FEM, element free Galerkin (EFG) method [56] and point interpolation method (PIM) [57]. They are considered to be more stable than those based on the strong form formulation, and PINN-based computational mechanics frameworks are no exception [58]. It has been reported that energy-based PINNs are computationally more efficient and better at capturing the stress concentration [33,58].

2.2. Boundary condition imposition

As mentioned above, the natural (traction) boundary conditions are already embedded in the potential energy. Auxiliary techniques are required to impose the essential boundary conditions. Consider the essential boundary condition is defined on Γ^u

$$\mathbf{u}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma^u \quad (6)$$

where $\bar{\mathbf{u}}(\mathbf{x})$ is the prescribed essential boundary conditions on Γ^u . Currently, the essential boundary conditions in energy-based PINNs can be imposed in soft or hard manners.

The so-called “soft” boundary condition imposition technique in PINNs is also known as the penalty method [59]. It produces solutions that approximately satisfy essential boundary conditions. To implement it, a least-square functional, Π_{EBC} , is first formulated

$$\Pi_{\text{EBC}} = \int_{\Gamma^u} \|\mathbf{u}(\mathbf{x}) - \bar{\mathbf{u}}(\mathbf{x})\|_2^2 d\Gamma = \int_{\Gamma^u} ([\mathbf{u}(\mathbf{x}) - \bar{\mathbf{u}}(\mathbf{x})] \cdot [\mathbf{u}(\mathbf{x}) - \bar{\mathbf{u}}(\mathbf{x})]) d\Gamma, \quad (7)$$

where $\|\cdot\|_2$ is the 2-norm on the essential boundary. As observed, Π_{EBC} only reaches its minimal value when the predicted displacements exactly equal the given boundary conditions. It is then added to the overall potential energy functional with a penalty factor κ

$$\Pi^* = \Pi + \kappa \Pi_{\text{EBC}}, \quad (8)$$

where Π^* is the modified energy functional. Referring to the penalty method, κ should be a positive factor. With increasing κ , the solution obtained by Eq. (8) will recover to the solution of the original problem. In numerical implementation, residuals are inevitable on the essential boundaries, such penalty method is therefore called the soft boundary condition imposition technique.

To impose the essential boundary conditions in the “hard” way, the neural network output is tailored so that it can naturally satisfy the essential boundary conditions

$$\mathbf{u}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) \odot g(\mathbf{x}) + \bar{\mathbf{u}}(\mathbf{x}), \quad (9)$$

where \odot is the element-wise product, and the distance network $g(\mathbf{x})$ represents the shortest distance from a given point \mathbf{x} to the essential boundary:

$$g(\mathbf{x}) = \min_{\mathbf{y} \in \Gamma^u} \sqrt{\|\mathbf{x} - \mathbf{y}\|_2^2} = \min_{\mathbf{y} \in \Gamma^u} \sqrt{(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})}. \quad (10)$$

There are two main advantages of using the hard boundary condition imposition technique. First, the essential boundary condition can be exactly imposed. Second, the physics-informed loss function can include fewer loss terms. The neural network training can be regarded as a multi-task learning process if multiple loss terms exist in the final loss function. Using too many loss terms may induce the imbalance training issue, and thus hyperparameters are required to balance the residuals from different loss terms [29]. Tuning hyperparameters before each loss term can be annoying. For simple geometries, the explicit form of $g(\mathbf{x})$ can be obtained. For complex boundary geometries, the distance function can be constructed in replacement of $g(\mathbf{x})$. For more details, please refer to [60].

In this study, the hard boundary condition enforcement technique is directly utilised to fixed boundary conditions. For displacement loading, the soft boundary condition enforcement is initially applied, followed by the use of the hard boundary condition technique.

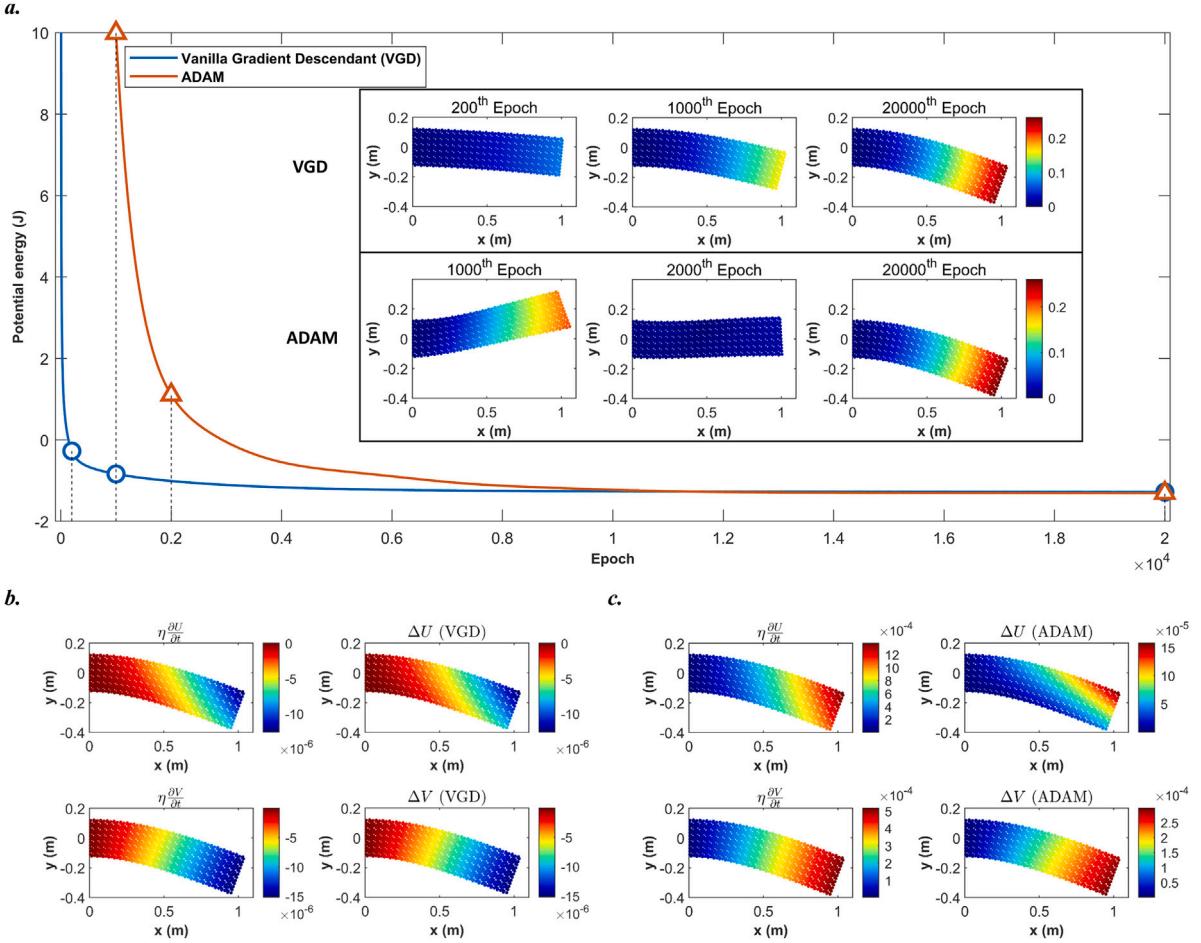


Fig. 1. (a) The training dynamics of energy-based PINNs for solving a cantilever beam ($0.25 \text{ m} \times 1 \text{ m}$) problem with a linear elastic material. The Young's modulus and the Poisson's ratio are $1 \times 10^4 \text{ Pa}$ and 0.3 , respectively. The left boundary of the beam is fixed. A parabolic distributed force of 10 N is downwardly applied on the right boundary. Two feedforward neural networks with 3 hidden layers and 5 neurons per layer are used for predicting displacements. When using gradient descendant algorithms, the overall potential energy decreases during the training of energy-based PINNs. The intermediate absolute displacement mappings are shown. Along with the decreasing energy functional, the beam seems to be dynamically bent to the equilibrium state. Note that the VGD in the figure refers to the vanilla gradient descendant algorithm. A learning rate of 1×10^{-4} is applied. (b) Comparisons between $\eta \frac{\partial u}{\partial t}$ and actual displacement increment when using the VGD at the 20000th epochs. (c) Comparisons between $\eta \frac{\partial v}{\partial t}$ and actual displacement increment when using the ADAM optimiser at the 20000th epochs.

2.3. Neural network training and its pseudo-dynamic nature

To solve the problem, optimisers are applied to minimise the loss function by iteratively adjusting those trainable parameters, θ . Among all, gradient descendant optimisers have earned great favour due to their effectiveness and efficiency. Currently, the ADAM optimiser is the most popular optimiser in PINN training [61]. In the energy-based PINNs, the energy functional at a given set of displacement mappings is treated as the loss function. Therefore, the training of the energy-based PINN can be regarded as a way to dissipate the overall potential energy. In other words, the energy-based PINNs are trained in a pseudo-dynamic manner. This can be verified by the example shown in Fig. 1(a), where both the vanilla gradient descendant (VGD) algorithm and the ADAM optimiser are applied to train neural networks for a 2D cantilever beam problem. After initialisation, neural networks map to random displacement fields. During the training, with the decreasing of the potential energy functional, the intermediate displacement mappings are shown. Along with the decreasing energy functional, the beam seems to be dynamically bent to the equilibrium state. Eventually, the functional loss stably converges around 2×10^4 epochs. Meanwhile, it can be noticed that the bending rates of the beam vary from the use of different optimisers. The ADAM optimiser incorporates the momentum of the gradient information. Hence, it can adjust the learning rate of each trainable parameter. By so doing, the use of the ADAM optimiser can significantly avoid loss oscillations during training and enhance the training robustness.

If gradient descendant algorithms are selected, the pseudo-velocity of the displacement mappings by neural networks during training can be approximately calculated by

$$\frac{\partial u}{\partial t} \approx -\frac{\partial F(\mathbf{x}; \theta)^T}{\partial \theta} \frac{\partial \Pi}{\partial \theta}, \quad (11)$$

where $F(\mathbf{x}; \theta)$ is the neural network displacement mapping, as presented in Eq. (1). Moreover, the increment of displacement can be obtained by

$$\Delta u \approx \eta \frac{\partial u}{\partial t}. \quad (12)$$

The derivation is given in Appendix A. These equations can be further verified by Fig. 1(b) and (c). As observed, the pseudo-velocity can effectively predict the displacement increment during training when the VGD algorithm is selected. Although small departures can be observed when using the ADAM optimiser, the overall pseudo-velocity is quite similar to the displacement increment after the energy loss is stabilised. It should be noted that the pseudo-dynamics nature exhibited by neural network training is not identical to second-order ordinary differential systems. In addition, the VGD converges faster at the beginning of the training. However, for complex nonlinear problems, the VGD may trap in local optima, while the ADAM optimiser can provide more reliable predictions (see Appendix B). We highlight that the pseudo-dynamics nature of the energy-based PINN can be utilised as a gradual loading process till the equilibrium state, equivalent to the iterative solvers in the traditional methods when dealing with nonlinear problems. Besides, the pseudo-velocity can help properly select the learning rate so that no penetration will happen during the training process. Those numerical techniques are explained in the following section.

3. Frictionless contact potentials and numerical implementations

3.1. Frictionless surface contact potential

In general contact problems, consider two bodies are getting close to each other, as shown in Fig. 2(a). From the microscopic point of view, it is the molecular and atom forces between two bodies that prevent them from penetrating each other. Such repulsive phenomenon is usually described by potentials, for example, the well-known Lennard-Jones (LJ) potential [62], as presented in Fig. 3(a). Therefore, bringing this idea to the macroscale, a penalty-like surface contact potential can be used to mimic the microscopic LJ potential between molecules and atoms. In this case, the overall potential energy of the conservation system can be re-written as

$$\Pi = E_{in} - E_{ex} + E_c, \quad (13)$$

where E_c is the surface contact potential energy and can be obtained by [63]

$$E_c = \int_{\Gamma_1} \int_{\Gamma_2} \beta_1 \beta_2 \phi(r) d\Gamma_1 d\Gamma_2, \quad (14)$$

where ϕ and r are the surface contact potential and the relative distance between contact surfaces, respectively. β is the surface density. The subscripts 1 and 2 denote the two contact surfaces. As observed, surface integrations are done on both surfaces of bodies. It is clear to find that the LJ potential is not a monotonic function, where it goes down first and then increases sharply when $r \rightarrow 0$. Nevertheless, given that this non-monotonic area only appears when the distance between the atom and molecular is close enough, which generally does not appear in the macroscopic contact problems. Therefore, a monotonic function can be applied as the surface contact potential, as long as the macroscopic potential increases steeply when $r \rightarrow 0$ and decays fast when $r \rightarrow \infty$. Here, an exponential form potential is applied in this work and defined as

$$\phi(r) = \phi_0 \exp\left(-\frac{r}{r_0}\right), \quad (15)$$

where ϕ_0 and r_0 are the pre-defined potential constant and the effective radius of the discrete points, respectively. The plot of the repulsive potential is given in Fig. 3(b). Therefore, the corresponding contact force can be calculated by

$$\begin{aligned} F_c &= \int_{\Gamma_1} \int_{\Gamma_2} \beta_1 \beta_2 \frac{\partial \phi(r)}{\partial r} d\Gamma_1 d\Gamma_2 \\ &= \int_{\Gamma_1} \int_{\Gamma_2} -\beta_1 \beta_2 \frac{\phi_0}{r_0} \frac{r}{r^2} \exp\left(-\frac{r}{r_0}\right) d\Gamma_1 d\Gamma_2. \end{aligned} \quad (16)$$

It is worth noting that the idea of using the surface potential for contact problems has been implemented in [63]. Besides, the rigorous derivations regarding the contact forces and variational formulation have been discussed in detail. However, in the framework of the proposed energy-based PINN, the potential energy functional is directly used and is minimised by powerful neural network optimisers. Meanwhile, we note that contact potentials can be in different forms and not necessarily identical to Eq. (15). Other contact potentials can be selected according to different contact problems. More contact potentials can be found in [63]. This work will only focus on demonstrating the effectiveness of the proposed PINN framework for contact problems. Therefore, only one contact potential is tested in the following numerical examples.

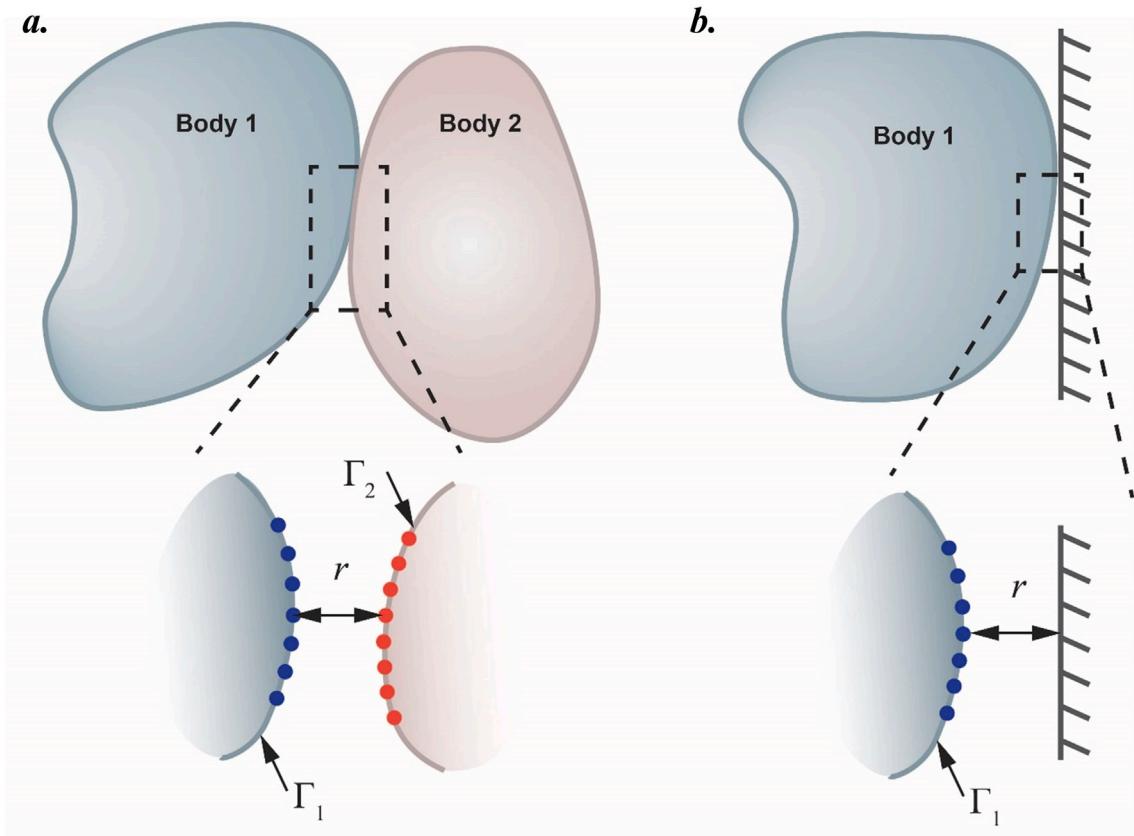


Fig. 2. Schematics of contact model discretisation [1]. (a) Point-to-point contact model. (b) Point-to-surface model.

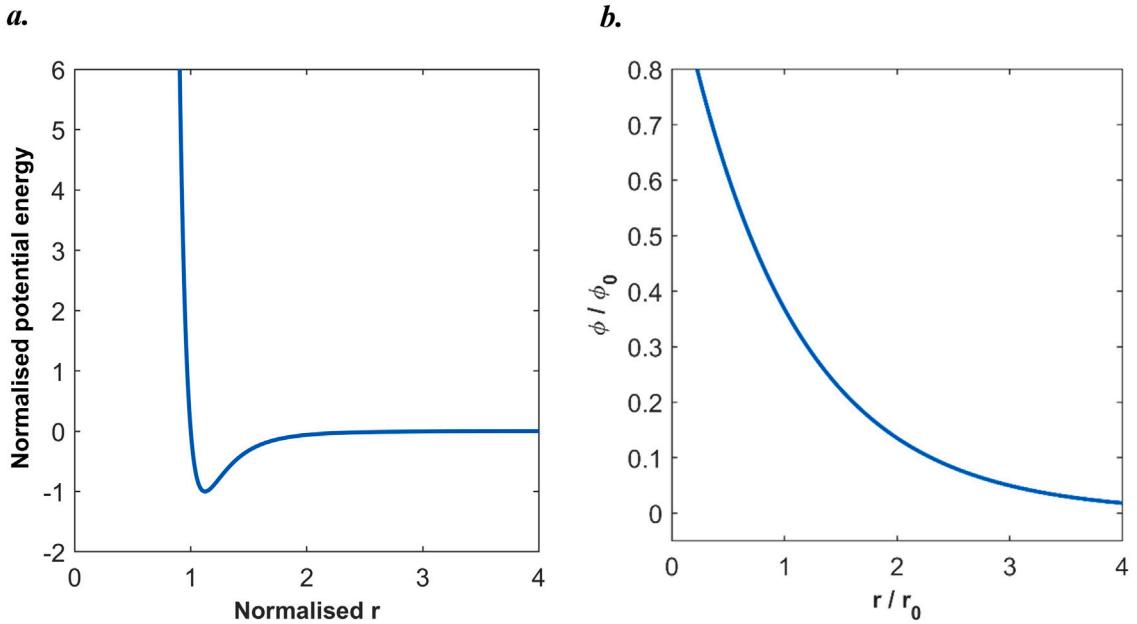


Fig. 3. (a) The Lennard-Jones potential plot; (b) The exponential form surface contact potential used in this work. Note that the ϕ_0 and r_0 are the pre-defined potential constant and the effective radius of the discrete points, respectively.

To numerically obtain the surface contact potential energy, sample points are placed to the possible contact area on both surfaces of the two bodies. In this work, this kind of contact model is called the point-to-point (PP) model, as shown in Fig. 2(a). Such kind of contact model can generally apply to different irregular geometries.

For some cases where regular geometry is used, as shown in Fig. 2(b). In these scenarios, only discrete sample points are placed on the irregular surface. In this work, we call this kind of contact model the point-to-surface (PS) model. Therefore, the surface contact potential energy of the PS contact model can be simplified as

$$E_c = \int_{\Gamma_1} \beta_1 \phi(r) d\Gamma_1, \quad (17)$$

where r is always perpendicular to the surface of the regular surface. Such a PS model is less generalised but can greatly improve both computational efficiency and accuracy. In the following numerical examples, both the PP and the PS models are applied to deal with different contact scenarios. More detailed discussions between those two contact models are available in Section 3.4.

It is worth noting that, such contact potential can also be used to describe the frictional contact phenomenon. However, in this work, we only use this potential to deal with frictionless contact problems. The friction force is a non-conservative force. Hence, frictional contact problems will be covered in our future work.

3.2. Numerical implementation

As shown in Eq. (15), the distances between the contact boundary nodes are necessary. Here, all the formulas are based on 2D problems and they can be easily extended to 3D scenarios. Consider the numbers of contact points on the potential contact surfaces are n and m , respectively. The relative distance between contact points can be stored in a matrix of size $n \times m$

$$r_{n \times m} = \sqrt{(\Delta u)^2 + (\Delta v)^2}, \quad (18)$$

where

$$\begin{aligned} \Delta u_{n \times m} &= x_{n \times 1} \otimes \mathbf{1}_{m \times 1} - (x_{m \times 1} \otimes \mathbf{1}_{n \times 1})^T \\ \Delta v_{n \times m} &= y_{n \times 1} \otimes \mathbf{1}_{m \times 1} - (y_{m \times 1} \otimes \mathbf{1}_{n \times 1})^T \end{aligned} \quad (19)$$

where $\mathbf{1}$ is the vector of ones, and the notation \otimes denotes the outer product. In Python, the outer product can be directly used as the multiplication by TensorFlow2. Therefore, the potential between each pair of points can be calculated as

$$\phi(r)_{n \times m} = \phi_0 \exp\left(-\frac{r_{n \times m}}{r_0}\right). \quad (20)$$

This equation is then substituted into Eq. (14) to calculate the overall contact potential. We note that the above implementation is compatible with the automatic differentiation technique. Thus, the contact energy added to the final loss can be directly minimised by neural network optimisers. The overall PINN framework proposed for contact problems is shown in Fig. 4.

3.3. Relaxation, gradual loading and output scaling

After initialising neural networks, the displacement mappings were established. However, those randomly built-up mappings may cause the multiple bodies to overlap with each other, as the example shown in Fig. 5(a–e). Therefore, to avoid the overlapping issue, relaxation can be applied first before solving problems. In relaxation, the contact potential is not activated in the loss function. In this manner, the neural network displacement mappings will be trained to dissipate the overall potential energy, and eventually reach their undeformed states, as shown in Fig. 5(f).

As mentioned in Section 2.3, neural network training has a pseudo-dynamics nature. If the increment of one training epoch goes too far, the contact bodies may penetrate each other. To address this problem, the learning rate of training should be properly selected based on the pseudo-velocity assessed by Eq. (11). Moreover, since the soft boundary condition imposition technique is applied for the moving boundary conditions, a gradual loading formula can be applied to circumvent the body penetration

$$\Pi_{EBC}(t) = \frac{1}{n} \sum \|u(x) - \frac{t}{t_{\max}} \bar{u}(x)\|_2^2, \quad (21)$$

where t is the current training epoch and t_{\max} is the maximum training epoch, n is the number of sample points on the essential boundary. As such, the soft boundary condition is gradually imposed through the training loop. We also highlight that, when the displacement boundary condition is roughly fulfilled with a small residual and the contact system is stabilised, the hard boundary imposition techniques can be then utilised instead.

In general, the initialising schemes for neural networks are likely to confine the initial output of neural networks within the range of $[-1, 1]$. Furthermore, given that bias is typically not assigned to the final hidden layer in neural networks, the significant responsibility for scaling the output value is placed on the weights of this layer when solely the tanh activation function is employed. Hence, when the displacement fields only have small deformation, the training of neural networks will be very difficult and time-consuming. To address this problem, the output scaling factors are added to the established neural network mapping (Eq. (9))

$$u(x) = \xi(F(x) \odot g(x) + \bar{u}(x)), \quad (22)$$

where ξ is the pre-defined scaling factor. The output scaling is used in the following benchmark problems (the Hertz contact problems).

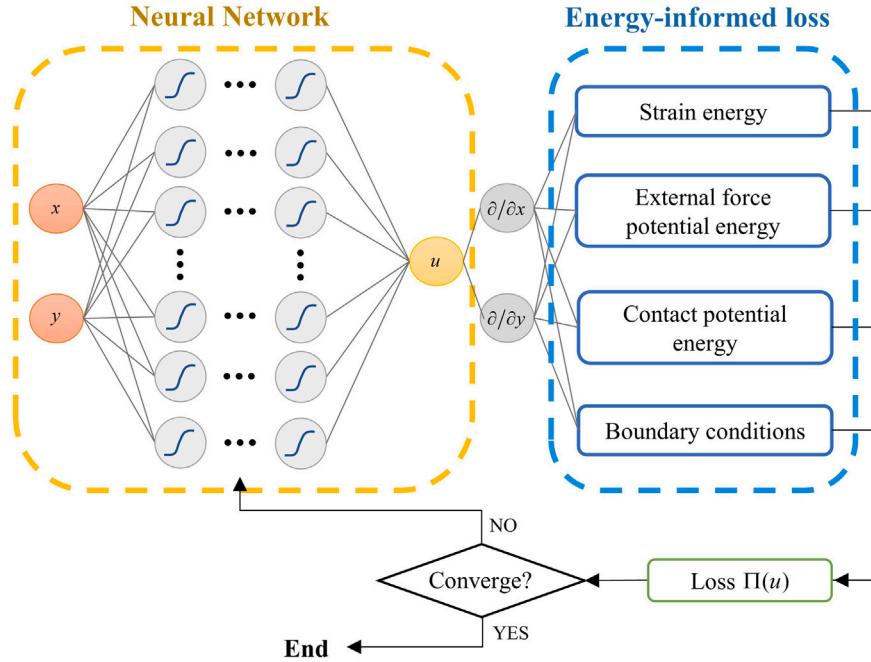


Fig. 4. The schematic of the proposed energy-based PINN framework for contact problems.

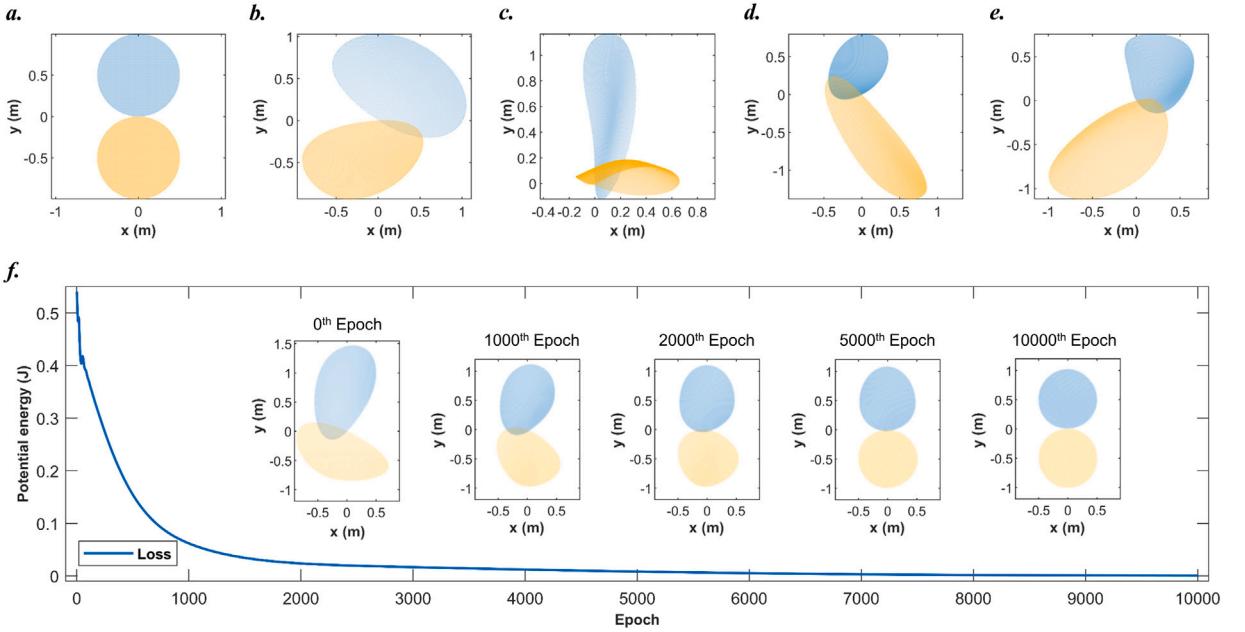


Fig. 5. The multiple bodies overlapping issue caused by neural network initialisations and the relaxation scheme. (a) Consider two circles (both radius is 0.5 m) placed at $(0, 0.5)$ and $(0, -0.5)$. The displacement fields u and v of each circle are predicted by two individual neural networks. (b–e) The initial configurations of circles predicted by neural networks after initialisation without training. Overlapping between two circles can be observed. (f) The loss histogram of two circles by the relaxation scheme. Note that the ADAM optimiser is used with a learning rate of 1×10^{-4} .

3.4. Benchmark tests and discussions

Using the aforementioned framework, the 2D Hertz contact problem is analysed. The problem is under the assumption of small deformation. A half cylinder is compressed on a rigid horizontal plane, as shown in Fig. 6(a). A uniform pressure of $F = 0.5$ N/m is subjected to be top of the half cylinder. A linear elastic material is considered, where Young's modulus $E = 200$ Pa and Poisson's

Table 1

The mean absolute percentage error of PINN predictions by using different r_0 , ϕ_0 and contact models. PS and PP refer to point-to-surface and point-to-point, respectively. For modelling with varying r_0 and ϕ_0 , ϕ_0 and r_0 are fixed to 1×10^4 and 1×10^{-5} , respectively.

Contact model	r_0				ϕ_0			
	1×10^{-3}	1×10^{-4}	1×10^{-5}	1×10^2	1×10^4	1×10^6		
PS	15.40%	4.40%	3.87%	5.11%	3.87%	3.39%		
PP	18.31%	11.71%	6.83%	\	6.83%	4.98%		

Table 2

The relative mean absolute error (RMAE) and GPU training time of the contact pressure predicted by the energy-based PINNs using different neural network sizes. The PP model is selected. ϕ_0 and r_0 are 1×10^4 and 1×10^{-5} , respectively. For each case, neural networks are trained 5 times and each training converges by 5×10^3 epochs. The network size is presented by the number of hidden layers times the number of neurons per layer.

Network size	RMAE P	GPU time (s)	Network size	RMAE P	GPU time (s)
2×30	4.46%	159.40	3×20	7.58%	222.62
3×30	4.02%	204.55	3×30	4.02%	204.55
4×30	2.82%	261.46	3×40	3.20%	210.47

ratio $\nu = 0.3$. The analytical solution is obtained from [64]. To solve this problem, two FNNs are built up to predict the displacements U and V , respectively. Each network contains 3 hidden layers and 30 neurons per layer. The scaling factor is $\xi = 1 \times 10^{-3}$. Thus, the U and V outputs of neural networks are obtained by

$$\begin{aligned} U &= \xi x F_x(x, y), \\ V &= \xi F_y(x, y), \end{aligned} \quad (23)$$

where the hard boundary condition is applied to the U prediction.

The sample points distribution is shown in Fig. 6(b). In this work, the sample points inside the computational domain are generated following the Gaussian quadrature. Since the contact area is located at the bottom of the cylinder, refined sample points are applied near the contact area, as shown in Fig. 6(c). The ADAM optimiser is selected with a learning rate of 1×10^{-4} . For each case, neural networks are trained 5 times and each training converges by 2×10^3 epochs. Both the point-plane and point-point contact models are tested.

Fig. 6(d) shows the results produced by the PINN-based framework and the FEM. Fig. 6(e) further shows the absolute error contours of displacement fields of the proposed PINN frameworks. The mean absolute error of the predicted U and V fields by using the PP model are respectively 3.89×10^{-6} N/m² and 2.15×10^{-4} N/m², while those of the PS model are respectively 4.36×10^{-6} N/m² and 1.28×10^{-4} N/m². All predicted contours are in good agreement with the reference results. Fig. 6(f) and (g) plots the contact pressure obtained by the PINN-based framework and Table 1 lists the mean relative percentage errors of PINN predictions with different contact models. The discussion starts with using different r_0 while a fixed ϕ_0 of 1×10^4 is applied. We note that r_0 here also controls the spacing of the contact sample points on the contact area, in other words, a decreasing r_0 will induce more contact points on the same length of a contact boundary. With decreasing selections of r_0 , the predicted contact pressure fits the analytical solution better, especially near the area of the transition point from contact to the free surface, as shown in Fig. 6(e). It is rational since it should converge to the analytical solution with decreasing selection of r_0 , analogous to the real atom model used in molecular dynamics. When changing the value of ϕ_0 and fixing $r_0 = 1 \times 10^{-5}$ m, it is found that the accuracy of the results is not very sensitive to different ϕ_0 , as shown in Fig. 6(f). It is also worth noting that, when using the PP contact model, the cylinder can penetrate the substrate plane if too small ϕ_0 is selected. Notably, ϕ_0 is similar to the penalty factors in the penalty method when imposing the essential boundary conditions. Moreover, it can be found that the framework with the PS model can achieve greater accuracy. Despite the relatively large discrepancy, the framework with the PP contact model is still effective enough to accurately capture the stress concentration patterns near the contact area, as presented in Fig. 6(d). Therefore, for irregular contact surfaces, the PP contact model is used instead of the PS contact model.

Finally, Table 2 presents the computational accuracy and efficiency of the energy-based PINNs with respect to different neural network sizes. With increasing numbers of layers and neurons per layer, neural networks can provide more accurate contact pressure predictions. However, the computational efficiency decreases due to the increasing number of trainable parameters in larger networks. Notably, training networks of the 3×20 size costs more time than larger networks. This is because a GPU is used. The GPU spends more time in accessing data rather than calculations. The computational efficiency should be increased with decreasing size of networks if a CPU is used.

4. Numerical examples

Herein, three frictionless contact problems with material and geometric nonlinearities are conducted to test the performance of the proposed framework. In all cases, the hyperelastic Neo-Hookean material is applied. Relaxation is also applied before all loading processes. The plain strain condition is considered. The neural network is built up by the TensorFlow 2 library in Python. The ADAM optimiser is selected with a learning rate of 1×10^{-5} . All the problems are modelled on a 12th Gen Intel(R) Core(TM) i5-12490F CPU (3.0 GHz) and with a Geforce 4060 Ti GPU.

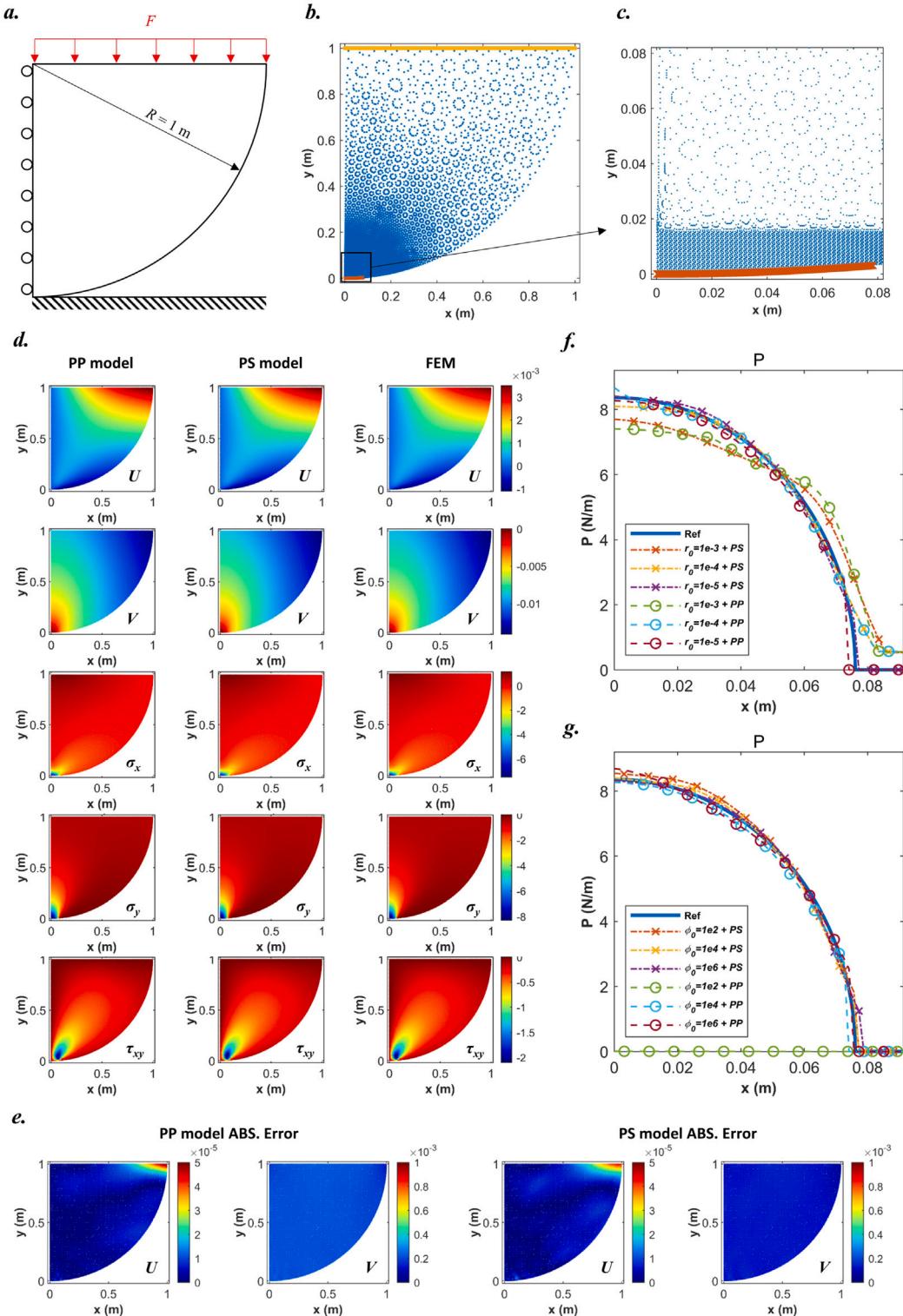


Fig. 6. Results for the Hertz contact problem. (a) The configuration of the Hertz contact problem; (b) The sample points distribution in the computational domain; (c) The refined sample points near the contact area. (d) Comparisons of the displacement and stress contours from the proposed PINN frameworks and the FEM. (e) The absolute error contours of the PINN frameworks. (f) Comparisons of the contact pressure plots with different r_0 ; (g) Comparisons of the contact pressure plots with different ϕ_0 . PS and PP refer to point-to-surface and point-to-point.

4.1. Rubber ironing

The ironing problem is one of the most prevailing benchmarks for frictionless contact problems under large deformation. The configuration of the problem is presented in Fig. 7(a), where a half cylinder is first compressed vertically on a slab and then滑动 horizontally. The sample points distribution for this problem is shown in Fig. 7(b). Throughout the whole modelling, the bottom boundary of the slab is fixed. For the half cylinder, during compressing, a vertical displacement of $V_t = -0.5$ m is applied by five uniform loading steps, while the horizontal displacement is fixed. During sliding, a horizontal displacement of $U_t = 2.5$ m is achieved by 25 uniform loading steps, while the compressing is remained. The Young's modulus for the half cylinder rubber and the slab are 3×10^2 Pa and 1×10^2 Pa, respectively, and their Poisson ratios are 0.3. Four FNNs are established with 3 hidden layers and 30 neurons per layer. To impose the displacement boundary conditions, the outputs of neural networks are constructed by

$$\begin{aligned} U_c &= (y - 3)u_c(x, y) + U_t, \\ V_c &= (y - 3)v_c(x, y) + V_t, \\ U_s &= yu_s(x, y), \\ V_s &= yv_s(x, y). \end{aligned} \quad (24)$$

We note that each loading step is trained by 10 training sessions and 2×10^3 epochs per session. The contact sample points on the half cylinder and the slab are placed with a spacing of $r_0 = 1 \times 10^{-2}$ m. The pre-defined potential constant $\phi_0 = 1 \times 10^2$. The penalty factor is $\kappa = 1 \times 10^4$.

Fig. 7(c) plots the vertical and horizontal reaction forces (RFs) during the whole loading process. Note that C and S in the legend refer to the RFs calculated on the upper boundary of the half cylinder and the bottom boundary of the slab. In the previous five loading steps, the vertical RF gradually increases due to the compression. The horizontal RFs remain zero due to the problem's symmetry. In the following loading steps, because the symmetry of the problem is broken, the horizontal RF slowly increases, while the vertical RF decreases. The contact pressure distribution and contour at the 5th loading step are given in Fig. 7(d). Both the contact pressure results obtained from the slab and cylinder align well with the FEM results. Moreover, compared to FEM results, the relative mean absolute differences of the contact pressure from the half cylinder and the slab are 6.81% and 6.41%, respectively. The σ_y stress contours at six selected loading steps are presented in Fig. 7(e). The FEM results are also presented for comparison. It can be observed that not only the RF plots obtained by the proposed PINN agree well with the FEM results, but the stress contours predicted by the PINN align closely to the FEM contours as well. Therefore, it is concluded that the frictionless condition is achieved during the whole modelling.

However, slight departures can still be found when comparing the stress contours, especially near the edges of the contact area at the previous 5 loading steps, as shown in Fig. 7(c) and (e). This is because the spacing of the contact sample points on the contact surfaces is still too large. As discussed in Section 3.4, the computational accuracy around the contact area can be further improved by decreasing the spacing of contact sample points.

4.2. Rubber ring contact instability

Consider a half rubber ring compressed on the rubber slab, as shown in Fig. 8(a). The Young's modulus of the ring and the slab are 100 Pa and 1 Pa, respectively. The Poisson's ratios of the ring and slab are 0.3. The thickness of the ring is 0.1 m. The displacement boundary condition, $V_t = 0.8$ m, is imposed on the rings, and the loading is done by 8 uniform loading steps. Relaxation is applied. The plain strain condition is considered. Four FNNs are established with 3 hidden layers and 50 neurons per layer. To impose the displacement boundary conditions, the outputs of neural networks are constructed by

$$\begin{aligned} U_r &= (y - 1.2)u_r(x, y), \\ V_r &= (y - 1.2)v_r(x, y) + V_t, \\ U_s &= yu_s(x, y), \\ V_s &= yv_s(x, y). \end{aligned} \quad (25)$$

Each loading step is trained by 5 training sessions and 5×10^4 epochs per session. The contact sample points on the half cylinder and the slab are placed with a spacing of $r_0 = 3 \times 10^{-4}$ m. The pre-defined potential constant $\phi_0 = 1 \times 10^5$. The penalty factor is $\kappa = 1 \times 10^6$.

The predicted von Mises stress contours are presented in Fig. 8(b). As observed, the rubber ring gradually attaches the slab with an increasing contact area when $V_t = 0.4$ m. The von Mises stress concentration mainly appears near the arc surfaces due to the bending of the ring. Interestingly, when $V_t = 0.4$ m, the ring bends inversely and the original contact area is divided into two parts. With further compression, the inverse bending appears more apparent. Besides, the von Mises stress concentration getting more severe at the middle of the half ring. We note that the contact of the two bodies also compresses the rubber slab, although it is not obvious in the stress contours. Fig. 8(c) further shows the σ_y contour at the 7th loading step. Moreover, the contact area contour is compared with the FEM result. The contours from the proposed PINN framework are in good agreement with those from the FEM and only small differences can be found.

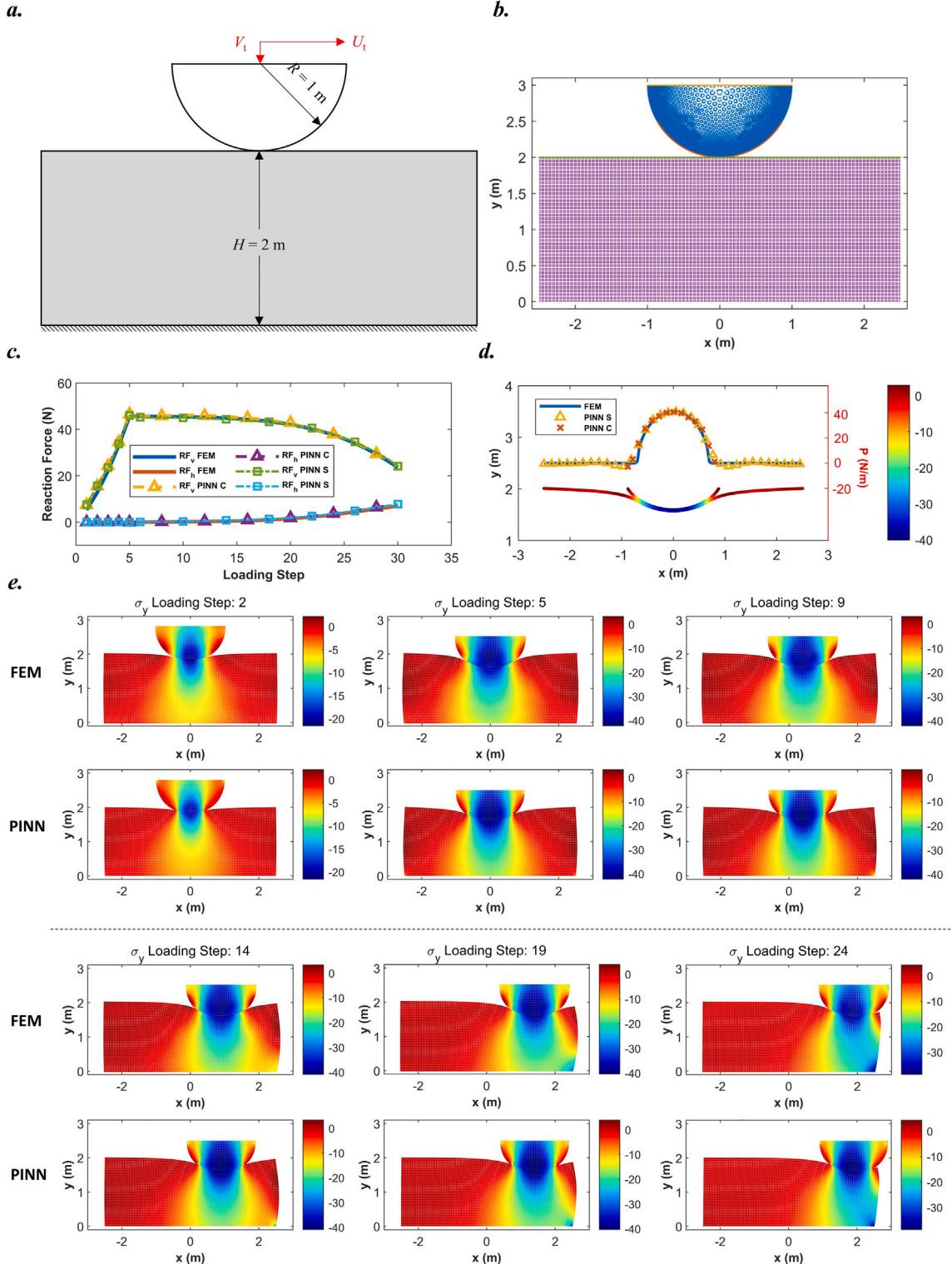


Fig. 7. The rubber ironing problem. (a) Configuration of the rubber ironing problem; (b) The sample point distribution in the computational domain; (c) Vertical and horizontal reaction forces (RF) during the loading process. C and S mean that the RFs are obtained from the boundary on the half cylinder and the slab; (d) The contact pressure plot and the contour at the 5th loading step; (e) Comparisons of σ_y stress contours from the proposed PINN framework at different loading steps.

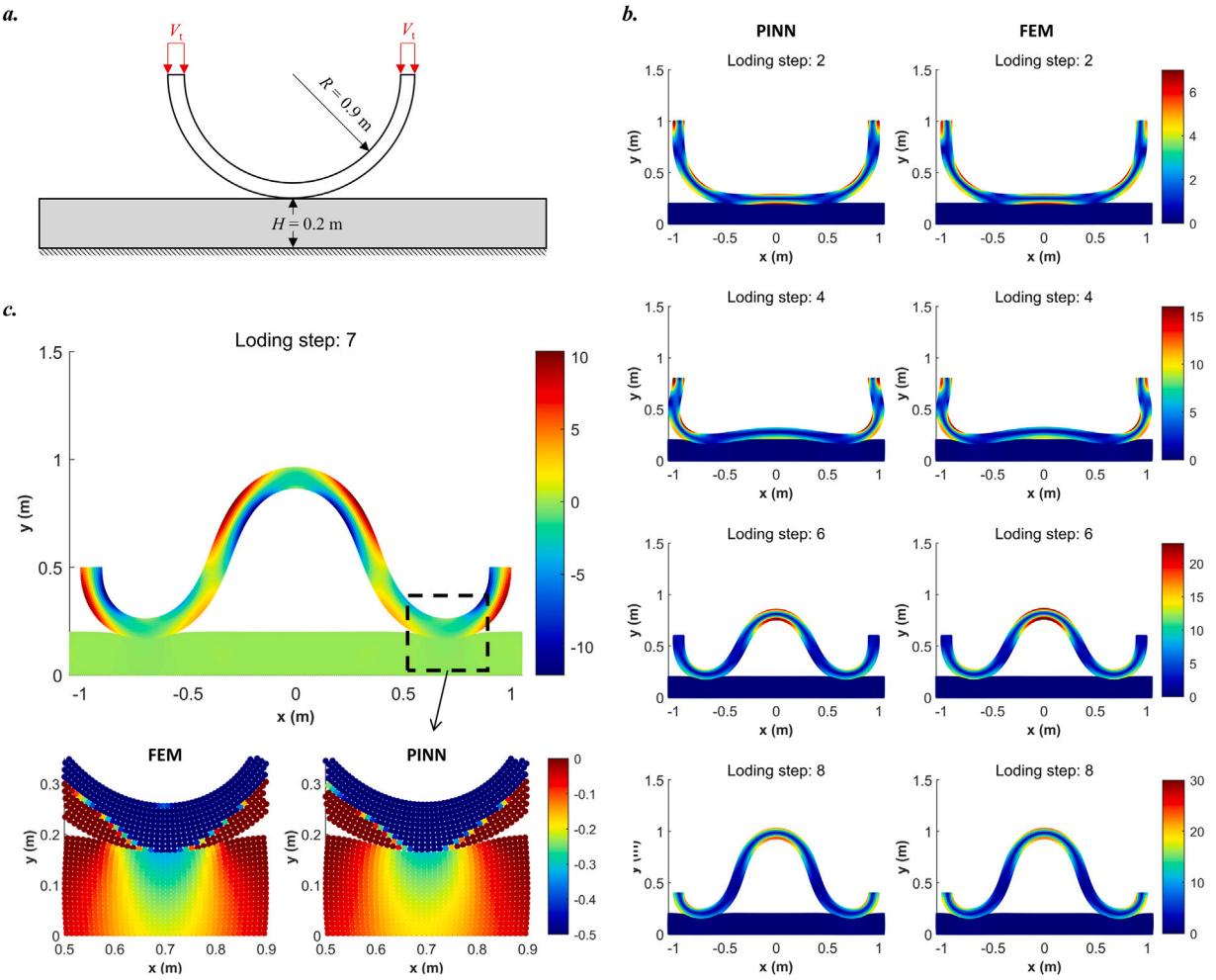


Fig. 8. The rubber ring contact instability problem. (a) Configuration of the ring and the slab; (b) Comparisons of the von Mises stress contours from the proposed PINN framework and the FEM at different loading steps; (c) σ_y stress contour obtained by the FEM and the proposed PINN at the 7th loading step.

4.3. Compression of two rubber rings

In this problem, two rubber rings are placed in a sink and compressed by a horizontal cap, as shown in Fig. 9(a). The Young's modulus and Poisson's ratio of the two rings are identically 100 Pa and 0.3, respectively. The inner radius and thickness of the rings are 0.3 m and 0.05 m, respectively. The centres of the two rings are located at (0.35, 0.35) and (0.9, 0.8). A vertical displacement, $V_t = -0.6\text{ m}$, is applied on the horizontal cap, compressing the two rubber rings to deform downward. The loading is done by 12 uniform loading steps. Four FNNs are established with 3 hidden layers and 50 neurons per layer. Since no explicit displacement boundary condition is required for the rings, the vanilla FNN structure is directly used. The PP model is applied for the ring-ring contact, while the PS contact model is applied for ring-sink and ring-cap contacts. Each loading step is trained 20 times and 2×10^3 epochs per training. The contact sample points on the rings are identically placed with a spacing of $r_0 = 1 \times 10^{-3}\text{ m}$, as shown in Fig. 9(b). The pre-defined potential constants for the PP model and the PS model are $\phi_0 = 1 \times 10^{-2}$ and 1×10^1 , respectively.

Fig. 9(c) shows the von Mises stress contour of the compressed rings at different loading steps. Results obtained from the ABAQUS are also presented for comparison. It is worth highlighting that the automatic stabilisation with a specific damping factor of 2×10^{-4} must be applied when using ABAQUS. This is not implemented in the proposed PINN framework. This additional robustness is contributed by the use of the ADAM optimiser. The Adam optimiser harnesses adaptive learning rates for each individual trainable parameter by assessing both the first and the second moments of loss gradients. Besides, incorporating momentum (i.e., maintaining an exponentially weighted moving average of past loss gradients) smoothens the optimisation process and avoids loss oscillations [61]. As observed, with the rigid cap moving downward, the upper ring is compressed and slides into the vacancy between the bottom ring and the sink. Those two rings squeeze each other and hold an equilibrium state. With increasing compression, the symmetry equilibrium condition is broken down. Due to the frictionless condition, rings exhibit chirality extrusion

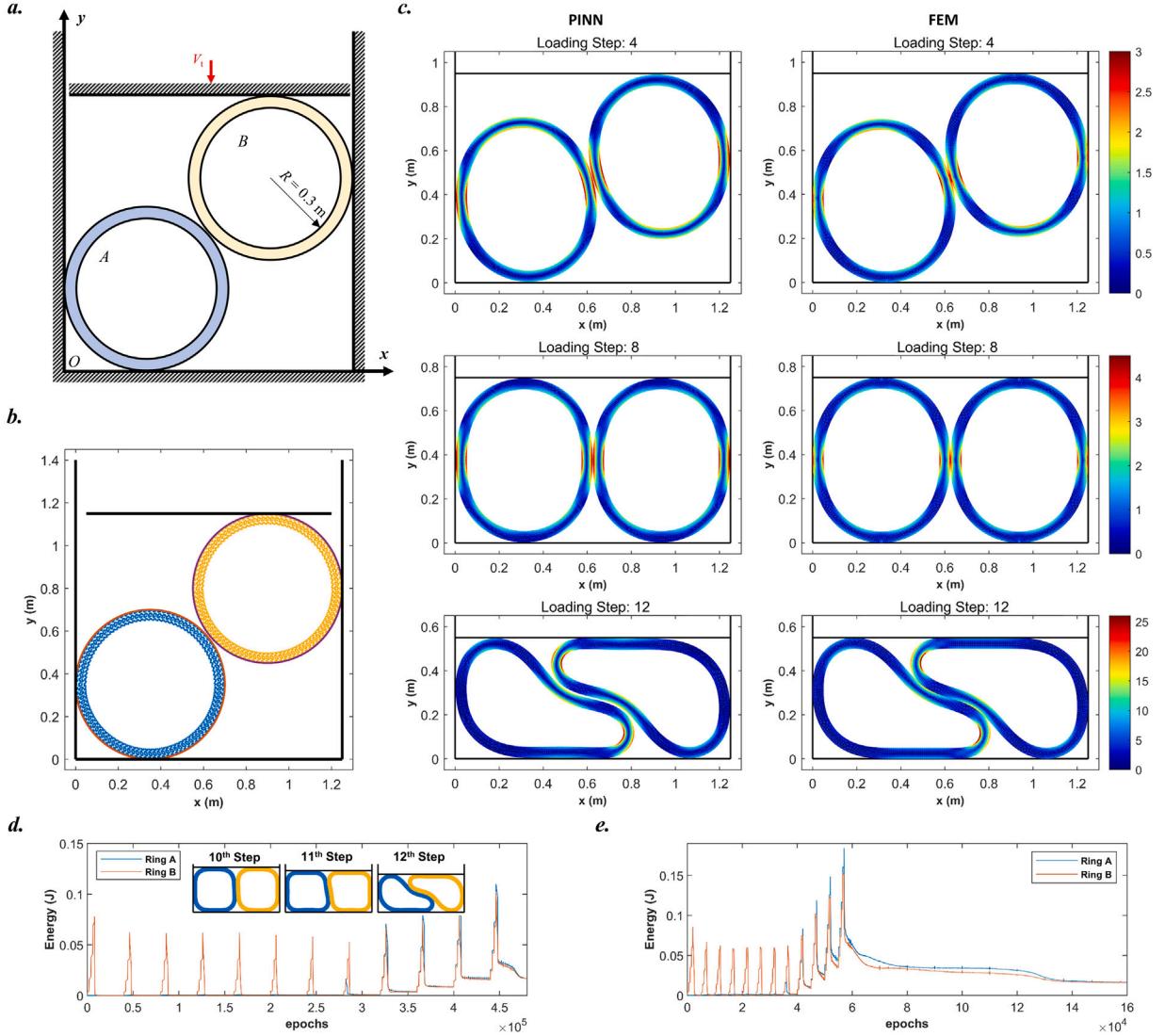


Fig. 9. The two rubber rings compression. (a) Configuration of the two rings, the rigid sink and the rigid cap; (b) The initialised sample points in the computational domain. (c) Comparisons of the von Mises stress contours from the proposed PINN framework and the FEM at different loading steps; (d) The potential energy of the two rings during the whole loading process (each loading step is trained 40 times and 2×10^3 epochs per training); (e) The potential energy of the two rings during the whole loading process. In this case, each loading step can be first trained 10 times and 5×10^2 epochs per training. After twelve-step loading, an additional 10 training sessions with 1×10^4 epochs per training are conducted to fine-tune the neural networks.

deformation. Such instability is captured in the loss histogram, as depicted in Fig. 9(d). Along with the loading processes, the stored strain energies of the two rings increase dramatically at the beginning of each loading step. For example, in the 12th loading step (started after 4.4×10^5 epochs), the stored strain energy of a single rubber ring increases to 0.12 J within the first 8×10^3 epochs. During the following training in a loading step, the strain energy gradually dissipates. It is obvious that the stored strain energy of the chirality configuration is lower than that of the symmetrical configuration, posing that the chirality deformation is the result of an instability phenomenon.

It is highlighted that, the modelling process is simpler and more straightforward while using the proposed PINN framework. Furthermore, once the neural network is well-trained, results in terms of displacements and stresses at any position can be directly exported by feeding the coordinate as an input. If only the final configuration of the compressed rings is required, it is not necessary to train the neural networks to converge at every loading step. For example, each loading step can be first trained 10 times and 5×10^2 epochs per training. After twelve-step loading, an additional 10 times training with 1×10^4 epochs per training are conducted to fine tune the neural networks. As the loss histogram shown in Fig. 9(e), the whole solving process can also stably converge after 1.4×10^5 epochs. In this manner, the CPU time of the proposed PINN framework for this compression case is 37 min (3767 sample points for each ring), while the ABAQUS costs 30 min (3468 nodes for each ring). Currently, our implementation leverages the GPU

functionality inherently integrated within the TensorFlow framework. The computational efficiency of the proposed energy-based PINNs could be enhanced through code optimisation, which will be addressed in future research endeavours.

5. Conclusions

In this work, we proposed an energy-based PINN framework for solving contact problems under large deformations. Thanks to the natural pseudo-dynamics of PINN training, the neural network training can be treated as a loading process, which ensures the framework's robustness. An exponential surface contact potential is applied to mimic the microscopic LJ potential, which intrinsically prevents solid bodies from penetrating each other. Besides, numerical schemes including relaxation, gradual loading and output scaling are proposed, further improving the stability and accuracy of the proposed framework for solving contact problems. The well-known Hertz contact benchmark problem is conducted, demonstrating the effectiveness of the proposed framework. Then, complex contact problems including the material nonlinearity as well as the large deformation are tested by using the proposed PINN framework.

It is worth highlighting that the proposed PINN framework offers a very easy way to model contact problems. Physics in terms of the energy functional, penalty function of essential BC and contact surface potential are straightforwardly integrated into the physics-informed loss function. It is very robust with respect to nonlinear problems. Besides, in the solving process, only one neural network training loop is required, while different iterative methods are needed to deal with different nonlinearities in the traditional computation methods. Moreover, as demonstrated by numerical examples, the energy-based PINN framework can easily capture the instability phenomenon without adding any artificial perturbation. In addition, for complex contact problems, the PINN framework can provide compatible computational efficiency when compared with commercial FEM software.

While the proposed Physics-Informed Neural Network (PINN) framework demonstrates excellent performance in addressing complex contact problems, it is important to acknowledge its limitations in frictionless scenarios, necessitating the consideration of frictional contact. The contact detection algorithms can also be integrated into the energy-based PINNs to alleviate the training expense. Besides, it should be noted that, due to the use of neural networks (i.e., nonlinear computing systems), neither the computational efficiency nor the convergence rate of the energy-based PINNs is comparable with traditional methods when dealing with simple solid mechanics problems. Hence, more attention should be paid to the complex mechanics problems that are very hard or even prohibitive for traditional methods. In addition, despite the competitive computational efficiency, the great potential of deep learning for scientific modelling has yet to be fully unleashed. Such physics-informed energy-based loss function can be integrated into more advanced deep learning models to further improve the computational efficiency, for example, the operator learning techniques [65–67]. These will be covered in our future works.

CRediT authorship contribution statement

Jinshuai Bai: Writing – original draft, Methodology, Formal analysis, Conceptualization. **Zhongya Lin:** Writing – review & editing, Methodology, Formal analysis. **Yizheng Wang:** Writing – review & editing, Methodology, Formal analysis. **Jiancong Wen:** Writing – review & editing, Formal analysis. **Yinghua Liu:** Writing – review & editing. **Timon Rabczuk:** Writing – review & editing, Formal analysis. **YuanTong Gu:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Xi-Qiao Feng:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Support from the National Natural Science Foundation of China (Grant nos. 11921002, 12032014 and T2488101) is acknowledged (J. Bai and X.-Q. Feng). Support from the Australian Research Council, Australia research grants (IC190100020 and DP200102546) is also acknowledged (J. Bai and Y. Gu). Support from the Key Project of the National Natural Science Foundation of China (12332005) is also acknowledged (Y. Wang and Y. Liu).

Appendix A. The pseudo-velocity of the energy-based PINNs during the training process

Recall Eq. (1), the displacement prediction before the $(t + 1)$ th epoch is calculated by

$$u^t = F(\mathbf{x}; \theta^t). \quad (\text{A.1})$$

Here, the superscript denotes the number of the current training epoch. After the $(t + 1)$ th epoch through gradient descendant optimisers, θ^t is updated by

$$\theta^{t+1} = \theta^t - \eta \frac{\partial \Pi}{\partial \theta} |_{\theta^t}. \quad (\text{A.2})$$

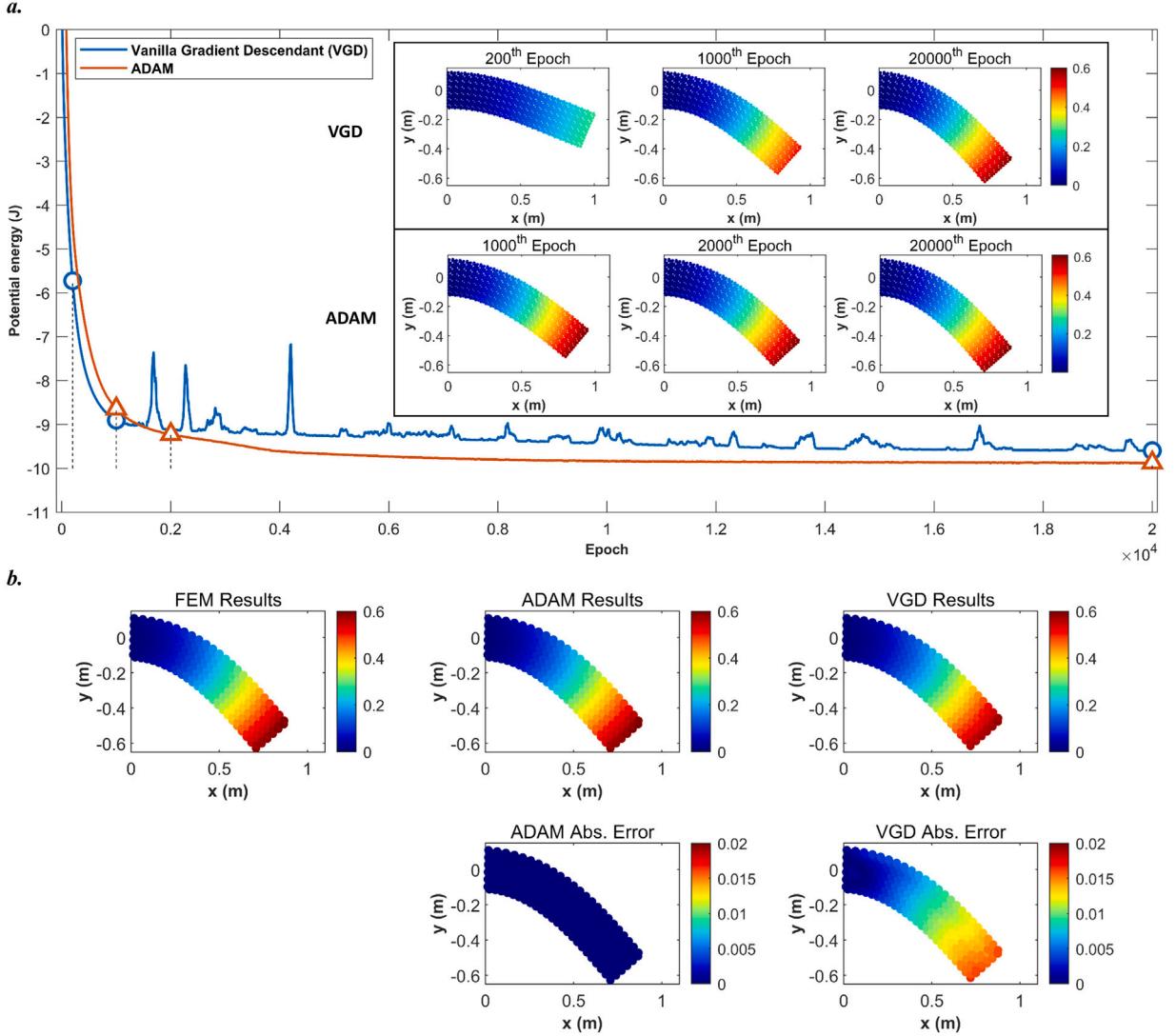


Fig. B.1. (a) The training dynamics of energy-based PINNs for solving a cantilever beam problem with neo-Hookean material and large deformation. A parabolic distributed force of 30 N is downwardly applied on the right boundary. (b) Absolute displacement contours from the FEM, the ADAM optimiser and the VGD algorithm and the corresponding absolute error contours.

Therefore, the increment of θ can be obtained as

$$\Delta\theta^{t+1} = \theta^{t+1} - \theta^t = -\eta \frac{\partial \Pi}{\partial \theta}|_{\theta^t}. \quad (\text{A.3})$$

Then, the Taylor expansion is performed

$$\begin{aligned} u^{t+1}(\mathbf{x}; \theta^{t+1}) &= F(\mathbf{x}; \theta^t) + \frac{\partial F(\mathbf{x}; \theta^t)^T}{\partial \theta} (\Delta\theta^{t+1}) + O(\Delta\theta^{t+1}) \\ &\approx F(\mathbf{x}; \theta^t) + \frac{\partial F(\mathbf{x}; \theta^t)^T}{\partial \theta} (\Delta\theta^{t+1}), \end{aligned} \quad (\text{A.4})$$

where $O(\Delta\theta^{t+1})$ represents the higher order small quantities in the expansion. Thus, the increment of displacement after one epoch of training gives

$$\Delta u^{t+1} = u^{t+1} - u^t \approx \frac{\partial F(\mathbf{x}; \theta^t)^T}{\partial \theta} (\Delta\theta^{t+1}). \quad (\text{A.5})$$

By substituting Eq. (A.3) into Eq. (A.5), one can obtain

$$\Delta u^{t+1} \approx -\eta \frac{\partial F(\mathbf{x}; \theta^t)^T}{\partial \theta} \frac{\partial \Pi}{\partial \theta}. \quad (\text{A.6})$$

The learning rate η controls the stepping length of displacement prediction. When the selection of η tend to be infinitely small, the flow of u with respect to the continuous training step t in a continues manner, e.g., a pseudo-velocity during the training process

$$\frac{\partial u}{\partial t} = \lim_{\eta \rightarrow 0} \frac{\Delta u^{t+1}}{\eta} = -\frac{\partial F(\mathbf{x}; \boldsymbol{\theta}^t)^T}{\partial \boldsymbol{\theta}} \frac{\partial \Pi}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}^t}. \quad (\text{A.7})$$

The same conception has also been applied to study dynamical property of $\boldsymbol{\theta}$ during training

$$\frac{\partial \boldsymbol{\theta}}{\partial t} \approx -\frac{\partial \Pi}{\partial \boldsymbol{\theta}}. \quad (\text{A.8})$$

where $\partial \boldsymbol{\theta}/\partial t$ is the gradient flow [68].

The pseudo-velocity approach is similar to the pseudo-time stepping method in computational fluid dynamics, which specifically transforms steady-state problems into transient problems. Pseudo-time stepping is a commonly used numerical method for nonlinear steady-state problems, and we apply this concept within energy-based PINNs. Consequently, a numerical pseudo-dynamic process can be obtained in the hyperelastic computation, allowing for a better understanding of the training process of energy-based PINNs.

Appendix B. Gradient descendant algorithms for the nonlinear cantilever beam problem

Herein, we consider the nonlinear version of the cantilever beam problem presented in Section 2.3. The neo-Hookean material and geometrical nonlinearity are applied. The Young's modulus, the Poisson's ratio and the geometry of the problem are the same. Given that the problem involves nonlinearities, two larger feedforward neural networks with 3 hidden layers and 10 neurons per layer are used to predict the displacements. Again, both the vanilla gradient descendant algorithm and the ADAM optimiser are used to train neural networks. The results are presented in Fig. B.1. It is obvious that the training of PINNs can be regarded as a pseudo-dynamic bending process. The training by using the ADAM optimiser can stably converge to the final solution, while that by using the VGD algorithm suffers oscillations, as shown in Fig. B.1(a). More importantly, the training by using the VGD algorithm is trapped by a local optimum and converges to a relatively high energy value, resulting in relatively larger discrepancies compared to the reference (FEM) result. This is further proofed by Fig. B.1(b). We also note that the training by using the VGD frequently suffers from numerical crashes caused by the training oscillations. Consequently, in this work, we apply the ADAM optimisers for all numerical examples.

Data availability

We have uploaded our data and code on Github. Anyone can freely access to them once the paper is accepted.

References

- [1] P. Wriggers, T.A. Laursen, *Computational Contact Mechanics*, vol. 2, Springer, 2006.
- [2] C. Sun, G.R. Liu, S.H. Huo, G. Wang, C. Yu, Z. Li, A novel node-to-segment algorithm in smoothed finite element method for contact problems, *Comput. Mech.* 72 (5) (2023) 1029–1057, <http://dx.doi.org/10.1007/s00466-023-02327-6>.
- [3] J. Yue, G.-R. Liu, M. Li, R. Niu, A cell-based smoothed finite element method for multi-body contact analysis using linear complementarity formulation, *Int. J. Solids Struct.* 141–142 (2018) 110–126, <http://dx.doi.org/10.1016/j.ijsolstr.2018.02.016>, URL <https://www.sciencedirect.com/science/article/pii/S0020768318300635>.
- [4] D. Liu, S.J. van den Boom, A. Simone, A.M. Aragón, An interface-enriched generalized finite element formulation for locking-free coupling of non-conforming discretizations and contact, *Comput. Mech.* 70 (3) (2022) 477–499, <http://dx.doi.org/10.1007/s00466-022-02159-w>.
- [5] P. Wriggers, J. Schröder, A. Schwarz, A finite element method for contact using a third medium, *Comput. Mech.* 52 (4) (2013) 837–847, <http://dx.doi.org/10.1007/s00466-013-0848-5>.
- [6] Y. Li, G.R. Liu, K.Y. Dai, M.T. Luan, Z.H. Zhong, G.Y. Li, X. Han, Contact analysis for solids based on linearly conforming radial point interpolation method, *Comput. Mech.* 39 (4) (2007) 537–554, <http://dx.doi.org/10.1007/s00466-006-0057-6>.
- [7] Y. Belaasilia, B. Braikat, M. Jamal, High order mesh-free method for frictional contact, *Eng. Anal. Bound. Elem.* 94 (2018) 103–112, <http://dx.doi.org/10.1016/j.enganabound.2018.06.003>, URL <https://www.sciencedirect.com/science/article/pii/S0955799718300997>.
- [8] A. Almasi, T.-Y. Kim, T.A. Laursen, J.-H. Song, A strong form meshfree collocation method for frictional contact on a rigid obstacle, *Comput. Methods Appl. Mech. Engrg.* 357 (2019) 112597, <http://dx.doi.org/10.1016/j.cma.2019.112597>, URL <https://www.sciencedirect.com/science/article/pii/S0045782519304736>.
- [9] A. Almasi, Y.-C. Yoon, T.-Y. Kim, T.A. Laursen, J.-H. Song, A strong-form meshfree collocation method for modeling stationary cracks with frictional contact, *Int. J. Non-Linear Mech.* 148 (2023) 104291, <http://dx.doi.org/10.1016/j.ijnonlinmec.2022.104291>, URL <https://www.sciencedirect.com/science/article/pii/S002074622200261X>.
- [10] A. Beel, J.-H. Song, Strong-form meshfree collocation method for multibody thermomechanical contact, *Eng. Comput.* 39 (1) (2023) 89–108, <http://dx.doi.org/10.1007/s00366-021-01513-5>.
- [11] T.A. Laursen, *Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*, Springer Science & Business Media, 2013.
- [12] I. Temizer, P. Wriggers, T. Hughes, Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS, *Comput. Methods Appl. Mech. Engrg.* 209 (2012) 115–128.
- [13] L. De Lorenzis, P. Wriggers, G. Zavarise, A mortar formulation for 3D large deformation contact using NURBS-based isogeometric analysis and the augmented Lagrangian method, *Comput. Mech.* 49 (2012) 1–20.
- [14] I. Temizer, P. Wriggers, T. Hughes, Contact treatment in isogeometric analysis with NURBS, *Comput. Methods Appl. Mech. Engrg.* 200 (9–12) (2011) 1100–1112.
- [15] J. Lu, Isogeometric contact analysis: Geometric basis and formulation for frictionless contact, *Comput. Methods Appl. Mech. Engrg.* 200 (5–8) (2011) 726–741.

- [16] L. De Lorenzis, İ. Temizer, P. Wriggers, G. Zavarise, A large deformation frictional contact formulation using NURBS-based isogeometric analysis, *Internat. J. Numer. Methods Engrg.* 87 (13) (2011) 1278–1300.
- [17] F. Fernandez, M.A. Puso, J. Solberg, D.A. Tortorelli, Topology optimization of multiple deformable bodies in contact with large deformations, *Comput. Methods Appl. Mech. Engrg.* 371 (2020) 113288, <http://dx.doi.org/10.1016/j.cma.2020.113288>, URL <https://www.sciencedirect.com/science/article/pii/S0045782520304734>.
- [18] B. Bozorgmehr, X. Yu, M.K. Matikainen, A.B. Harish, A. Mikkola, A study of contact methods in the application of large deformation dynamics in self-contact beam, *Nonlinear Dynam.* 103 (1) (2021) 581–616, <http://dx.doi.org/10.1007/s11071-020-05984-x>.
- [19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [20] G.-R. Liu, *Machine Learning with Python: Theory and Applications*, World Scientific, 2022.
- [21] M. Raissi, Deep hidden physics models: Deep learning of nonlinear partial differential equations, *J. Mach. Learn. Res.* 19 (25) (2018) 1–24.
- [22] 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.
- [23] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: a review, *Acta Mech. Sin.* 37 (12) (2021) 1727–1738, <http://dx.doi.org/10.1007/s10409-021-01148-1>.
- [24] Y. Wang, J. Bai, Z. Lin, Q. Wang, C. Anitescu, J. Sun, M.S. Eshaghi, Y. Gu, X.-Q. Feng, X. Zhuang, et al., Artificial intelligence for partial differential equations in computational mechanics: A review, 2024, arXiv preprint <arXiv:2410.19843>.
- [25] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, *Science* 367 (6481) (2020) 1026–1030, <http://dx.doi.org/10.1126/science.aaw4741>, arXiv:<https://www.science.org/doi/pdf/10.1126/science.aaw4741> URL <https://www.science.org/doi/abs/10.1126/science.aaw4741>.
- [26] H. Wessels, C. Weißfels, P. Wriggers, The neural particle method – An updated Lagrangian physics informed neural network for computational fluid dynamics, *Comput. Methods Appl. Mech. Engrg.* 368 (2020) 113127, <http://dx.doi.org/10.1016/j.cma.2020.113127>, URL <https://www.sciencedirect.com/science/article/pii/S0045782520303121>.
- [27] J. Bai, Y. Zhou, Y. Ma, H. Jeong, H. Zhan, C. Rathnayaka, E. Sauret, Y. Gu, A general neural particle method for hydrodynamics modeling, *Comput. Methods Appl. Mech. Engrg.* 393 (2022) 114740, <http://dx.doi.org/10.1016/j.cma.2022.114740>, URL <https://www.sciencedirect.com/science/article/pii/S0045782522000962>.
- [28] Z. Li, J. Bai, H. Ouyang, S. Martelli, M. Tang, Y. Yang, H. Wei, P. Liu, R. Wei, Y. Gu, Physics-informed neural networks for friction-involved nonsmooth dynamics problems, *Nonlinear Dynam.* 112 (9) (2024) 7159–7183, <http://dx.doi.org/10.1007/s11071-024-09350-z>.
- [29] J. Bai, T. Rabczuk, A. Gupta, L. Alzubaidi, Y. Gu, A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics, *Comput. Mech.* 71 (3) (2023) 543–562, <http://dx.doi.org/10.1007/s00466-022-02252-0>.
- [30] E. Haghghat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, *Comput. Methods Appl. Mech. Engrg.* 379 (2021) 113741.
- [31] E. Samaniego, C. Anitescu, S. Goswami, V.M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, *Comput. Methods Appl. Mech. Engrg.* 362 (2020) 112790.
- [32] Y. Wang, J. Sun, W. Li, Z. Lu, Y. Liu, CENN: Conservative energy method based on neural networks with subdomains for solving variational problems involving heterogeneous and complex geometries, *Comput. Methods Appl. Mech. Engrg.* 400 (2022) 115491, <http://dx.doi.org/10.1016/j.cma.2022.115491>, URL <https://www.sciencedirect.com/science/article/pii/S0045782522005096>.
- [33] W. Li, M.Z. Bazant, J. Zhu, A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches, *Comput. Methods Appl. Mech. Engrg.* 383 (2021) 113933, <http://dx.doi.org/10.1016/j.cma.2021.113933>, URL <https://www.sciencedirect.com/science/article/pii/S004578252100270X>.
- [34] V.M. Nguyen-Thanh, X. Zhuang, T. Rabczuk, A deep energy method for finite deformation hyperelasticity, *Eur. J. Mech. A Solids* 80 (2020) 103874, <http://dx.doi.org/10.1016/j.euromechsol.2019.103874>, URL <https://www.sciencedirect.com/science/article/pii/S0997753819305352>.
- [35] S. Niu, E. Zhang, Y. Bazilevs, V. Srivastava, Modeling finite-strain plasticity using physics-informed neural network and assessment of the network performance, *J. Mech. Phys. Solids* 172 (2023) 105177, <http://dx.doi.org/10.1016/j.jmps.2022.105177>, URL <https://www.sciencedirect.com/science/article/pii/S0022509622003532>.
- [36] J. He, D. Abueilda, R. Abu Al-Rub, S. Koric, I. Jasiuk, A deep learning energy-based method for classical elastoplasticity, *Int. J. Plast.* 162 (2023) 103531, <http://dx.doi.org/10.1016/j.ijplas.2023.103531>, URL <https://www.sciencedirect.com/science/article/pii/S0749641923000177>.
- [37] S. Goswami, C. Anitescu, T. Rabczuk, Adaptive phase field analysis with dual hierarchical meshes for brittle fracture, *Eng. Fract. Mech.* 218 (2019) 106608, <http://dx.doi.org/10.1016/j.engfracmech.2019.106608>, URL <https://www.sciencedirect.com/science/article/pii/S0013794419302814>.
- [38] S. Goswami, C. Anitescu, S. Chakraborty, T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, *Theor. Appl. Fract. Mech.* 106 (2020) 102447, <http://dx.doi.org/10.1016/j.tafmec.2019.102447>, URL <https://www.sciencedirect.com/science/article/pii/S016784421930357X>.
- [39] L. Zhao, Q. Shao, DEDEM: Discontinuity embedded deep energy method for solving fracture mechanics problems, 2024, arXiv preprint <arXiv:2407.11346>.
- [40] J. Bai, G.-R. Liu, T. Rabczuk, Y. Wang, X.-Q. Feng, Y. Gu, A robust radial point interpolation method empowered with neural network solvers (RPIM-NNS) for nonlinear solid mechanics, *Comput. Methods Appl. Mech. Engrg.* 429 (2024) 117159, <http://dx.doi.org/10.1016/j.cma.2024.117159>, URL <https://www.sciencedirect.com/science/article/pii/S0045782524004158>.
- [41] Z. Zhang, Y. Li, W. Zhou, X. Chen, W. Yao, Y. Zhao, TONR: An exploration for a novel way combining neural network with topology optimization, *Comput. Methods Appl. Mech. Engrg.* 386 (2021) 114083, <http://dx.doi.org/10.1016/j.cma.2021.114083>, URL <https://www.sciencedirect.com/science/article/pii/S004578252100414X>.
- [42] H. Jeong, J. Bai, C. Batuwatta-Gamage, C. Rathnayaka, Y. Zhou, Y. Gu, A physics-informed neural network-based topology optimization (PINNTO) framework for structural optimization, *Eng. Struct.* 278 (2023) 115484, <http://dx.doi.org/10.1016/j.engstruct.2022.115484>, URL <https://www.sciencedirect.com/science/article/pii/S0141029622015607>.
- [43] H. Jeong, C. Batuwatta-Gamage, J. Bai, Y.M. Xie, C. Rathnayaka, Y. Zhou, Y. Gu, A complete physics-informed neural network-based framework for structural topology optimization, *Comput. Methods Appl. Mech. Engrg.* 417 (2023) 116401, <http://dx.doi.org/10.1016/j.cma.2023.116401>, URL <https://www.sciencedirect.com/science/article/pii/S004578252300525X>.
- [44] H. Jeong, C. Batuwatta-Gamage, J. Bai, C. Rathnayaka, Y. Zhou, Y. Gu, An advanced physics-informed neural network-based framework for nonlinear and complex topology optimization, *Eng. Struct.* 322 (2025) 119194.
- [45] K. Tang, X. Wan, C. Yang, DAS-PINNs: A deep adaptive sampling method for solving high-dimensional partial differential equations, *J. Comput. Phys.* 476 (2023) 111868.
- [46] Z. Mao, X. Meng, Physics-informed neural networks with residual/gradient-based adaptive sampling methods for solving partial differential equations with sharp solutions, *Appl. Math. Mech.* 44 (7) (2023) 1069–1084.
- [47] Y. Wang, J. Sun, T. Rabczuk, Y. Liu, DCEM: A deep complementary energy method for linear elasticity, *Internat. J. Numer. Methods Engrg.* (2023) e7585.
- [48] J. Sun, Y. Liu, Y. Wang, Z. Yao, X. Zheng, BINN: A deep learning approach for computational mechanics problems based on boundary integral equations, *Comput. Methods Appl. Mech. Engrg.* 410 (2023) 116012, <http://dx.doi.org/10.1016/j.cma.2023.116012>, URL <https://www.sciencedirect.com/science/article/pii/S0045782523001366>.

- [49] Y. Wang, J. Sun, J. Bai, C. Anitescu, M.S. Eshaghi, X. Zhuang, T. Rabczuk, Y. Liu, Kolmogorov Arnold informed neural network: A physics-informed deep learning framework for solving PDEs based on Kolmogorov Arnold Networks, 2024, arXiv preprint [arXiv:2406.11045](https://arxiv.org/abs/2406.11045).
- [50] T. Sahin, M. von Danwitz, A. Popp, Solving forward and inverse problems of contact mechanics using physics-informed neural networks, *Adv. Model. Simul. Eng. Sci.* 11 (1) (2024) 11, <http://dx.doi.org/10.1186/s40323-024-00265-3>.
- [51] Y. Diao, J. Yang, Y. Zhang, D. Zhang, Y. Du, Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology, *Comput. Methods Appl. Mech. Engrg.* 413 (2023) 116120, <http://dx.doi.org/10.1016/j.cma.2023.116120>, URL <https://www.sciencedirect.com/science/article/pii/S004578252300244X>.
- [52] X. Zhuang, H. Guo, N. Alajlan, H. Zhu, T. Rabczuk, Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning, *Eur. J. Mech. A Solids* 87 (2021) 104225, <http://dx.doi.org/10.1016/j.euromechsol.2021.104225>, URL <https://www.sciencedirect.com/science/article/pii/S099775382100019X>.
- [53] J. Bai, G.-R. Liu, A. Gupta, L. Alzubaidi, X.-Q. Feng, Y. Gu, Physics-informed radial basis network (PIRBN): A local approximating neural network for solving nonlinear partial differential equations, *Comput. Methods Appl. Mech. Engrg.* 415 (2023) 116290, <http://dx.doi.org/10.1016/j.cma.2023.116290>, URL <https://www.sciencedirect.com/science/article/pii/S0045782523004140>.
- [54] X.-P. Zhou, K. Feng, MPNN based graph networks as learnable physics engines for deformation and crack propagation in solid mechanics, *Int. J. Solids Struct.* 291 (2024) 112695, <http://dx.doi.org/10.1016/j.ijsolstr.2024.112695>, URL <https://www.sciencedirect.com/science/article/pii/S0020768324000520>.
- [55] P.M. Schaefferkoetter, Y.-C. Yoon, J.-H. Song, Three-dimensional continuum point cloud method for large deformation and its verification, *Comput. Methods Appl. Mech. Engrg.* 432 (2024) 117307, <http://dx.doi.org/10.1016/j.cma.2024.117307>, URL <https://www.sciencedirect.com/science/article/pii/S0045782524005632>.
- [56] T. Zhu, S. Atluri, A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method, *Comput. Mech.* 21 (3) (1998) 211–222.
- [57] G.R. Liu, Y.T. Gu, K.Y. Dai, Assessment and applications of point interpolation methods for computational mechanics, *Internat. J. Numer. Methods Engrg.* 59 (10) (2004) 1373–1397, <http://dx.doi.org/10.1002/nme.925>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.925> URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.925>.
- [58] J. Bai, H. Jeong, C.P. Batuwatta-Gamage, S. Xiao, Q. Wang, C. Rathnayaka, L. Alzubaidi, G.-R. Liu, Y. Gu, An introduction to programming physics-informed neural network-based computational solid mechanics, *Int. J. Comput. Methods* 20 (10) (2023) 2350013.
- [59] J.W. Barrett, C.M. Elliott, Finite element approximation of the Dirichlet problem using the boundary penalty method, *Numer. Math.* 49 (4) (1986) 343–366, <http://dx.doi.org/10.1007/BF01389536>.
- [60] N. Sukumar, A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, *Comput. Methods Appl. Mech. Engrg.* 389 (2022) 114333, <http://dx.doi.org/10.1016/j.cma.2021.114333>, URL <https://www.sciencedirect.com/science/article/pii/S0045782521006186>.
- [61] D.P. Kingma, Adam: A method for stochastic optimization, 2014, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [62] J. Fischer, M. Wendland, On the history of key empirical intermolecular potentials, *Fluid Phase Equilib.* 573 (2023) 113876.
- [63] R.A. Sauer, L. De Lorenzis, A computational contact formulation based on surface potentials, *Comput. Methods Appl. Mech. Engrg.* 253 (2013) 369–395, <http://dx.doi.org/10.1016/j.cma.2012.09.002>, URL <https://www.sciencedirect.com/science/article/pii/S0045782512002769>.
- [64] D. Franke, A. Düster, V. Niibel, E. Rank, A comparison of the h-, p-, hp-, and rp-version of the FEM for the solution of the 2D Hertzian contact problem, *Comput. Mech.* 45 (5) (2010) 513–522, <http://dx.doi.org/10.1007/s00466-009-0464-6>.
- [65] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229, <http://dx.doi.org/10.1038/s42256-021-00302-5>.
- [66] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, 2020, arXiv preprint [arXiv:2010.08895](https://arxiv.org/abs/2010.08895).
- [67] Y. Wang, X. Li, Z. Yan, Y. Du, J. Bai, B. Liu, T. Rabczuk, Y. Liu, HomoGenius: a foundation model of homogenization for rapid prediction of effective mechanical properties using neural operators, 2024, arXiv preprint [arXiv:2404.07943](https://arxiv.org/abs/2404.07943).
- [68] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM J. Sci. Comput.* 43 (5) (2021) A3055–A3081, <http://dx.doi.org/10.1137/20M1318043>, arXiv:<https://doi.org/10.1137/20M1318043>.