The Institute of Mathematics and
Computer-Based Simulations

Deep Learning in Computational Mechanics

Predicting the Beam Deflection using Physics Informed Neural Networks (PINNs)

Tarik Sahin

A part of the XXX Manuscript 2022

Universität der Bundeswehr München

Contents

List of Figures

CHAPTER 1

Introduction

Machine Learning and Deep Learning approaches require usually a large amount of simulation or experimental data which might not be always feasible to obtain due to the complexity of the simulations and the expensiveness of the experiments. Presence of the limited data often causes data-driven approaches to fail to deliver adequate accuracy since they are purely data-dependent. Thus, the data-driven methods with non-matching real-world observations (sensor or experimental data) or wrong labeled datasets are prone mostly to make the wrong predictions, which can lead the irrevocable failures as there is no physics-based control mechanism to validate the predictions. To compensate for the lack of adequate data and unknown black-box behavior of the data-driven approaches, Raissi et al. proposed an approach that combines the data-driven methods with the existing physics knowledge of the given problem [RPK19]. With this approach, the data-driven method can learn even with less data since the governing equations of the problem are involved in the learning process.

As depicted in Figure 1.1, obtaining more data is more expensive due to the need for more simulation or experiments, while involving more physics increases the complexity of the problem. Herein, physics-informed neural networks (PINNs) come into play, which requires some data and some physics. PINNs enforce the neural network architecture to learn the pattern by using the limited data with the help of the governing equations of the problem [Kol+].

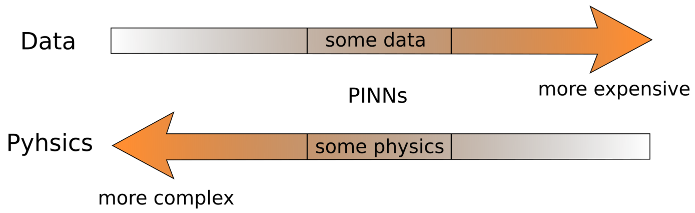


Figure 1.1: Trade-off between data and physics

CHAPTER 2

Physics-Informed Neural Networks (PINNs)

2.1 Overview

Physics-informed neural networks can be considered as an alternative approach to conventional partial differential equation (PDE) solvers. The general idea is to solve problems by minimizing the loss function constructed from PDEs where data is limited. Compared to conventional solvers, PINNs

- are mesh-free.
- can efficiently handle unstructured or irregular domains [Lu+21].
- can be applied to solve also *inverse* problems.

2.2 Formulation of a PINN

Consider a partial differential equation as

$$\frac{\partial u}{\partial t} + \mathcal{N}[u;\lambda] = 0, x \in \Omega, t \in [0, \mathcal{T}] \tag{2.1}$$

where $u(t, x)$ is the latent solution with time $t \in [0, \mathcal{T}]$ and a spatial variable $x \in \Omega$, $\mathcal{N}[u;\lambda]$ is the nonlinear differential operator with the coefficient λ, and Ω is a subset of $\mathbb{R}^D$.

The conventional methods to solve Eq. 2.1 are either analytical or numerical. The analytical solutions are not always feasible and numerical methods are expensive. The PINNs approach can be considered more feasible than analytical methods and less expensive than numerical methods. The physics-informed neural network estimates the solution $u(t, x)$ and the physics-enhanced part evaluates the given partial differential equation using the estimated solution. As shown in Fig. 2.1, the physics-informed neural network predicts the solution $u(x, t)$ by using fully connected layers with nonlinear activation functions, denoted as σ. The physics-enhanced part evaluates the chosen PDE, denoted as $f(x)$, using the higher derivatives of the solution $u(x, t)$, denoted here d_x, d_{xx} and d_t, with respect to space (x) and time (t), and the identity matrix of u. Since the solution is known on the boundaries and the approximated solution must fulfill the governing PDE, the loss can be computed using a cost function, i.e. mean squared error. If the solution is not converged, the weights of the neural network are updated using back-propagation to estimate the new solution, until the given convergence limit is reached.

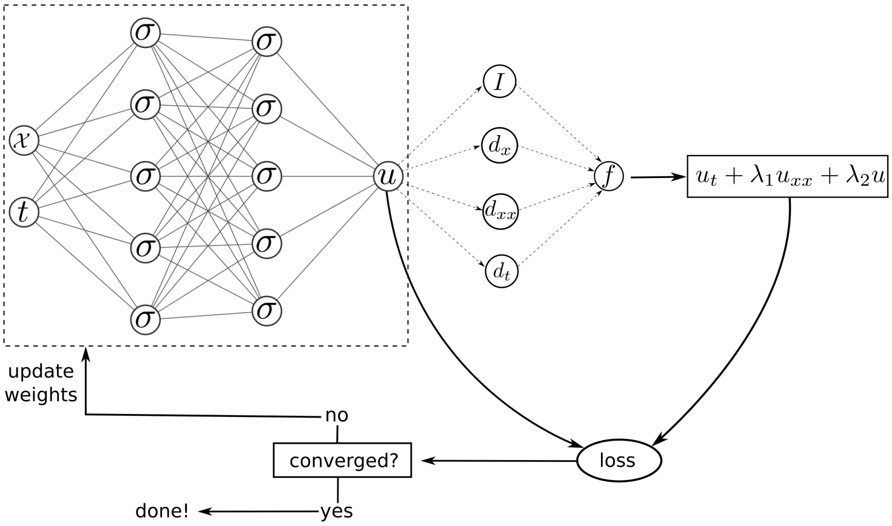


Figure 2.1: The visionary representation of physics-informed neural network for an arbitrary space-time dependent partial differential equation.

Compared to classical neural networks, the loss function of a PINN has three terms:

$$L = E_u + E_0 + E_f \tag{2.2}$$

The first term, denoted as E_u, calculates the error for the approximated solution on the known boundaries. In the case of a one dimensional non-linear heat equation subjected to the following homogenous Neumann boundary condition

$$\frac{\partial u(x=1,t)}{\partial x} = h, \tag{2.3}$$

the Dirichlet boundary condition

$$u(x=0,t) = g, \tag{2.4}$$

E_u term is calculated as

$$E_u = E_{Neumann} + E_{Dirichlet}, \tag{2.5}$$

where

$$E_{\text{Neumann}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left(\frac{\partial}{\partial x}u_P\left(x_b^i, t_b^i\right) - h\right)^2 \tag{2.6}$$

enforces the homogenous Neumann boundary condition (refer to 2.3) by penalizing the error between the derivative of the predicted solution, denoted

as $\frac{\partial}{\partial x}u_P\left(x_b^i, t_b^i\right)$, and the given Neumann boundary condition h at N_b random points $\left\{x_b^i, t_b\right\}_{i=1}^{N_b}$ on the boundary $x_b = 1$.
The term, $E_{Dirichlet}$ of the Eq. 2.5

$$E_{\text{Dirichlet}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left(u_P\left(x_b^i, t_b^i\right) - g\right)^2 \tag{2.7}$$

enforces the Dirichlet boundary condition according Eq. 2.4 by penalizing the error between approximated solution $u_P(x_b^i, t_b^i)$ and prescribed Dirichlet boundary condition g at N_b random points $\left\{x_b^i, t_b\right\}_{i=1}^{N_b}$ on the boundary $x_b = 0$.

The initial condition loss, E_0 of the Eq. 2.2

$$E_0 = \frac{1}{N_0}\sum_{i=1}^{N_0}\left(u_P\left(x_0^i, 0\right) - u_0^i\right)^2, \tag{2.8}$$

with the following initial condition

$$u(x, t = 0) = u_0, \tag{2.9}$$

enforces the initial conditions at N_0 random points $\left\{x_0^i, t_b\right\}_{i=1}^{N_0}$ at initial time $t_b = 0$

The loss functions E_u and E_0 contain only the loss on the known boundaries and on the known initial conditions, respectively.

On the other hand, the final term of the Eq. 2.2

$$E_{\text{f}} = \frac{1}{N_f}\sum_{i=1}^{N_f}\left(\frac{\partial}{\partial t}u_P\left(x_f^i, t_b^i\right) - \mathcal{N}[u_P(x_b^i, t_b^i)]\right)^2 \tag{2.10}$$

imposes the given partial differential equation at every random collocation point inside the domain by penalizing the estimated left-hand side and the known right-hand side of the governing equation. Since PINNs are mesh-free, the distribution of collocation points can be uniform or random.

After calculation of the loss terms, weights of the architecture is updated using back-propagation with the help of a optimization function, i.e. stochastic gradient descent, Adam etc.

2.3 General remarks regarding PINNs

- A PINN is mesh-free so it can handle unstructured or irregular domains.
- PINNs can be deployed on both forward and inverse problems. In forward problems, they can find the hidden solution $u(x, t)$ where the coefficients λ are known (Eq. 2.1) or discover the parameters λ using the provided solution, which are known as inverse problems. This approach is called also as data-driven identification.

- Depending on the type of the governing equation, they can be used on a wide variety of problems ranging from biological systems (reconstruction clinical magnetic resonance imaging (MRI) data) to the governing equations of continuum mechanics.

- Since the governing equations of the problem are involved in the learning process, PINNs can learn even data is limited.

- The derivatives of the network prediction u_P with respect to space (x) and time (t) can be easily and efficiently computed by the automatic differentiation capabilities implemented in many deep learning tool kits, i.e. Tensorflow [Aba+16] and Pytorch [Pas+19].

CHAPTER 3

The Euler-Bernoulli Beam Theory with Governing Equations

A beam is a structural element having one dimension that is much larger than the other two and capable of resisting the loads applied laterally to the beam axis. Since the further investigations are pursued for the Euler-Bernoulli beam theory, known as thin beam theory, the main assumptions and the governing equations of the Euler-Bernoulli are given in more detail.

3.1 Assumptions of the Euler-Bernoulli Beams

The main hypothesis of the Euler-Bernoulli beam theory is that plane cross-sections which are orthogonal to the beam axis (neutral axis) stay still plane and orthogonal to the beam axis after deformation. The overall assumptions of the Euler beam theory:

- Plane cross-sections remain plane and perpendicular to the neutral axis of the beam after deformation.
- The deformations are small, so the equilibrium is computed on the reference configuration (Small deformation theory).
- Behavior of the beam is linear elastic and isotropic.
- Shear strains are neglected.

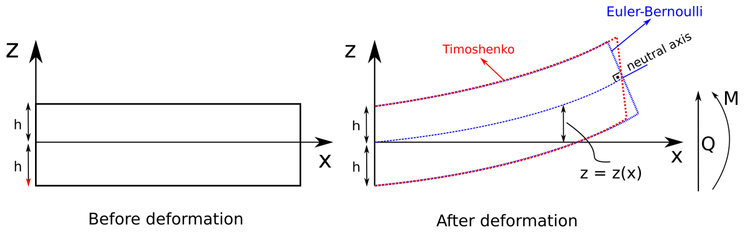


Figure 3.1: The Euler-Bernoulli and Timoshenko beam deflections

As depicted in Fig. 3.1, the plane cross-sections stay plane and perpendicular to the beam axis in the Euler-Bernoulli beam. In the Timoshenko beam, plane cross-sections still remain plane but are no longer perpendicular to the beam axis. The difference between normal to the beam axis and plane cross-section is known as shear deformation. Here, shear strains will be ignored so the beam Formulation is considered under the Euler-Bernoulli beam theory, which is known also as thin beam theory.

3.2 Governing Equations

Time-dependent beam equation

Consider the governing equation of an Euler-Bernoulli beam under the space-time dependent distributed load as (Fig. 3.2)

$$\rho A \frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left(EI \frac{\partial^2 w(x,t)}{\partial x^2} \right) + q(x,t) = 0, x \in \Omega, t \in [0,T] \qquad (3.1)$$

where $w(x,t)$ is the unknown solution, Ω is a subset of $\mathbb{R}^D$, E is the uniform elastic modulus of the material, I is the moment of the inertia of the cross-section of the beam, ρ is the material density, A is the cross-sectional area, and $q(x,t)$ is distributed space-time varying load. Eq. 3.1 is known as also *Euler-Langrange* equation [WQ07] used mainly in vibration problems. The first term describes the dissipated kinetic energy where ρA is area density or the mass per unit length, and the second term characterizes the potential energy due to internal forces.

The concentrated load q can be considered as a special case of the distributed load as follows:

$$q(x,t) = P(t)\delta\left(x - x_0\right) \qquad (3.2)$$

where δ is the *Dirac* delta function, and $P(t)$ is the time-dependent load applied at position x_0.

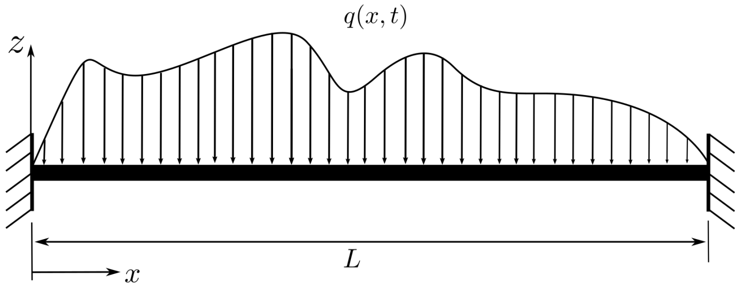


Figure 3.2: A fixed-fixed supported beam under a space-time dependent distributed load.

Static beam equation

Eliminating time terms in Eqs. 3.1 and 3.2, the governing ordinary differential equation is obtained for a static model

$$\frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 w(x)}{\partial x^2}\right)+q(x)=0, x\in\Omega, \tag{3.3}$$

and, similarly if $q(x)$ is a concentrated load

$$q(x)=P\delta\left(x-x_0\right). \tag{3.4}$$

3.3 Boundary Conditions and Beam Supports

To estimate the unknown solution $w(x,t)$, the governing equation of the Euler-Bernoulli beam must be solved using the initial and boundary conditions. Since the governing equation is of the $4th$ order in space (Eq. 3.1), four different boundary conditions are introduced:

- $w(x,t)$ → represents the displacement in the z direction,
- $w_x(x,t)$ → indicates the slope of the beam,
- $w_{xx}(x,t)$ → measures the bending moment,
- $w_{xxx}(x,t)$ → measures the shear force,

at location $x\in\Omega$ and time $t\in[0,T]$.

Similary, and two initial conditions are needed as Eq. 3.1 is of the 2^{th} order in time

- $w(x,0)$ → represents the displacement in the z direction,
- $w_t(x,0)$ → indicates the time derivative of the displacement,

at location $x\in\Omega$ and at time $t=0$. Initial conditions are not taken into account in the static beam equation since the time-dependent derivative term vanishes.

Two main supports will be further investigated:

- fixed supported beams
- simply supported beams

Cantilever beam

A cantilever beam is fixed supported at one end and free at the other as depicted in Fig. 3.3. No displacement and rotation are allowed at the fixed edge. No bending moment and no shear force at the free edge are generated.

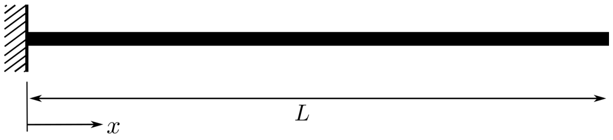


Figure 3.3: An illustration of a fixed supported beam

Boundary conditions of a cantilever beam

$$\begin{aligned} &w(x,t) = 0 \quad and \quad w_x(x,t) = 0, \quad at\ x = 0, t = [0,T] \\ &w_{xx}(x,t) = 0 \quad and \quad w_{xxx}(x,t) = 0, \quad at\ x = L, t = [0,T] \end{aligned} \tag{3.5}$$

Simply supported beam

Simply supported beam is pinned at the ends for two edges as shown in Fig. 3.4. At both pinned ends, no displacement but rotation is allowed which means that no bending moments are generated.

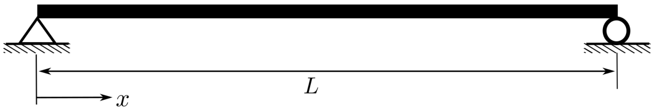


Figure 3.4: An illustration of a simply supported beam.

Boundary conditions of a simply supported beam

$$\begin{aligned} &w(x,t) = 0 \quad and \quad w_{xx}(x,t) = 0, \quad at\ x = 0, t = [0,T] \\ &w(x,t) = 0 \quad and \quad w_{xx}(x,t) = 0, \quad at\ x = L, t = [0,T] \end{aligned} \tag{3.6}$$

Clamped beam

A clamped beam is fixed supported at both ends as depicted in Fig. 3.5. At both fixed ends, no displacement and rotation is allowed.

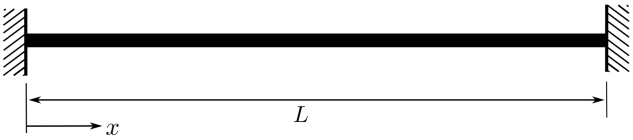


Figure 3.5: An illustration of a clamped beam.

Boundary conditions of a clamped beam

$$\begin{aligned} &w(x,t) = 0 \quad and \quad w_x(x,t) = 0, \quad at\ x = 0, t = [0,T] \\ &w(x,t) = 0 \quad and \quad w_x(x,t) = 0, \quad at\ x = L, t = [0,T] \end{aligned} \tag{3.7}$$

CHAPTER 4

Application of PINNs on 1D Beams

In Sec. 2.2, a general formulation for physics-informed neural networks is explained. Here, one-dimensional Euler-Bernoulli beams under the distributed load with uniform geomerical and material properties are conducted, which are governed by an ordinary differential equation (Eq. 3.3) in static models and governed by a partial differential equation (Eq. 3.1) in continuous-time models.

4.1 Static Models

Consider the governing equations of one-dimesional linear elastic Euler-Bernoulli beam (Fig. 3.5) with the corresponding boundary conditions of clamped beam as

$$\frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 w(x)}{\partial x^2}\right) + q(x) = 0 \quad \text{on } \Omega \tag{4.1}$$

$$\frac{\partial w(x)}{\partial x} = h \quad \text{on } \Gamma_N, \tag{4.2}$$

$$w(x) = g \quad \text{on } \Gamma_D. \tag{4.3}$$

The Neumann and Dirichlet boundary conditions are denoted as Γ_N and Γ_D. To give a better understanding, the spatial domain is defined $\Omega = [0, L]$, with $\Gamma_D = \{x | x = 0, x = L\}$ and $\Gamma_N = \{x | x = 0, x = L\}$. If the length of the beam is set to $L = 1$, the following boundary conditions of a clamped support Euler-Bernoulli beam are obtained

$$\begin{aligned} w(x = 0) = 0 \text{ and } \quad w_x(x = 0) = 0, \\ w(x = 1) = 0 \text{ and } \quad w_x(x = 1) = 0. \end{aligned} \tag{4.4}$$

The physics-informed neural network approximates the unknown solution $w(x)$ through the hidden layers enhanced with non-linear activation functions σ using the spatial domain x as the input as depicted in Fig. 4.1. Since only the solution on the boundaries are known, the predicted solution for every location inside the domain must fulfill the Eq. 4.1 such that

$$f := \frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 w(x)}{\partial x^2}\right) + q(x). \tag{4.5}$$

Here the f term represents the residuum or physics-informed residual which has to be fulfilled at every point inside the domain. The main purpose of PINNs is to minize the residuum enforcing the boundary conditions via the loss function.

Combining Eqs. 4.4 and 4.5 with 2.6, 2.7, 2.10, 2.5 and 2.2, the loss function of static clamped Euler-Bernoulli beam is defined

$$
\begin{aligned}
L &= E_u + E_f \\
&= E_{Neumann} + E_{Dirichlet} + E_f \\
&= \frac{1}{N_b}\sum_{i=1}^{N_b}\left[\left(\frac{\partial}{\partial x} w_P\left(x_b^i\right) - h\right)^2 + \left(w_P\left(x_b^i\right) - g\right)^2\right] + \frac{1}{N_f}\sum_{i=1}^{N_f}\left(f_P(x_f^i)\right)^2 \\
&= \frac{1}{N_b}\sum_{i=1}^{N_b}\left[\underbrace{\left(\frac{\partial}{\partial x} w_P\left(x_b^i\right)\right)^2}_{\text{Neumann term}} + \underbrace{\left(w_P\left(x_b^i\right)\right)^2}_{\text{Dirichlet term}}\right] + \frac{1}{N_f}\sum_{i=1}^{N_f}\left(f_P(x_f^i)\right)^2
\end{aligned}
\tag{4.6}
$$

The h and g terms are the prescribed Neumann and Dirichlet boundary conditions at N_b boundary points $\left\{x_b^i\right\}_{i=1}^{N_b} = \{0, 1\}$. Since the beam is considered as a clamped beam, $h = 0$ and $g = 0$. The last term of Eq. 4.6 minizes the error at every point inside the spatial domain, called as the collocation points $\left\{x_f^i\right\}_{i=1}^{N_f}$. The collocation points can be generated using random or uniform distributions.

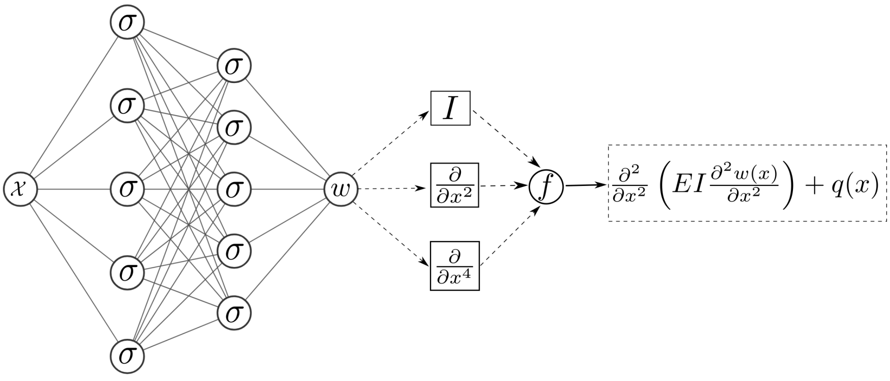

x
σ
w
I
$\frac{\partial}{\partial x^2}$
$\frac{\partial}{\partial x^4}$
f
$\frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 w(x)}{\partial x^2}\right) + q(x)$

Figure 4.1: An illustration of physics-informed neural network for the governing ordinary differential equation of the one-dimensional Euler-Bernoulli beam.

For demonstrating the application of PINNs on the Euler-Bernoulli beams, three cases are further investigated to involve the higher-order derivatives in the loss function as depicted in Fig. 4.2. For instance, the clamped beam contains only the first order, while the cantilever beam contains the first, second and third-order derivatives (cf. 3.3, 3.4, 3.5). For simplicity, Young's modulus and the moment of inertia terms are set to $AE = 1$, the beam length and the magnitude of the distributed loading are set to $L = 1$ and $q = 1$, relatively.

The following PINN architecture is used to compute the approximated solution:

- Architecture contains 1 input layer, 1 output layer and 3 hidden layers containing 30 nodes.
- Hyperbolic tangent activation functions *tanh* are used to non-linearize the outputs of the layers.
- The *Glorot Uniform* initializer is applied to initialize the weights randomly to avoid dying out of the nodes.
- The *Adam* optimizer with learning rate *0.0005* is chosen as the optimization function.
- The number of collocation points is set $N_f = 20$ and the Sobol sampling is performed to have a non-uniform distribution.

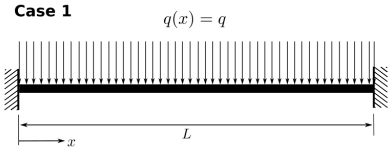
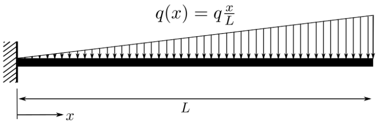
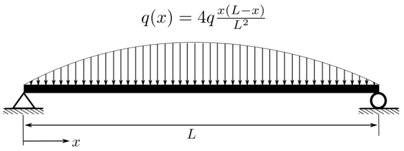
Beam model	**Description**
Case 1  	Clamped beam under a uniformly distributed load q(x) with the magnitude q. The analytical solution: $w(x) = -\frac{qx^2}{24EI}(x-L)^2$
Case 2  	Cantilever beam under a triangular distributed load q(x) with the magnitude q at x=L. The analytical solution: $w(x) = -\frac{qx^2}{120EI}(20L^3 - 10L^2x + x^3)$
Case 3  	Simply supported beam under a quadratically distributed load q(x) with the magnitude q at x=L/2. The analytical solution: $w(x) = -\frac{qx}{90EI}(x^5 - 6x^4L + 15x^3L^2 - 15x^2L^3 + 5L^5)$

Figure 4.2: Three investigated beam models with their descriptions and analytical solutions.

The results of the investigated cases are given in Fig. 4.3. Figures on the left show that the predicted displacements are almost identical to the analytical solutions at each collocation point as can be seen in the magnitude of the loss function. On the other hand, there is a difference in terms of the order of magnitudes of the loss functions between the differential equation loss and

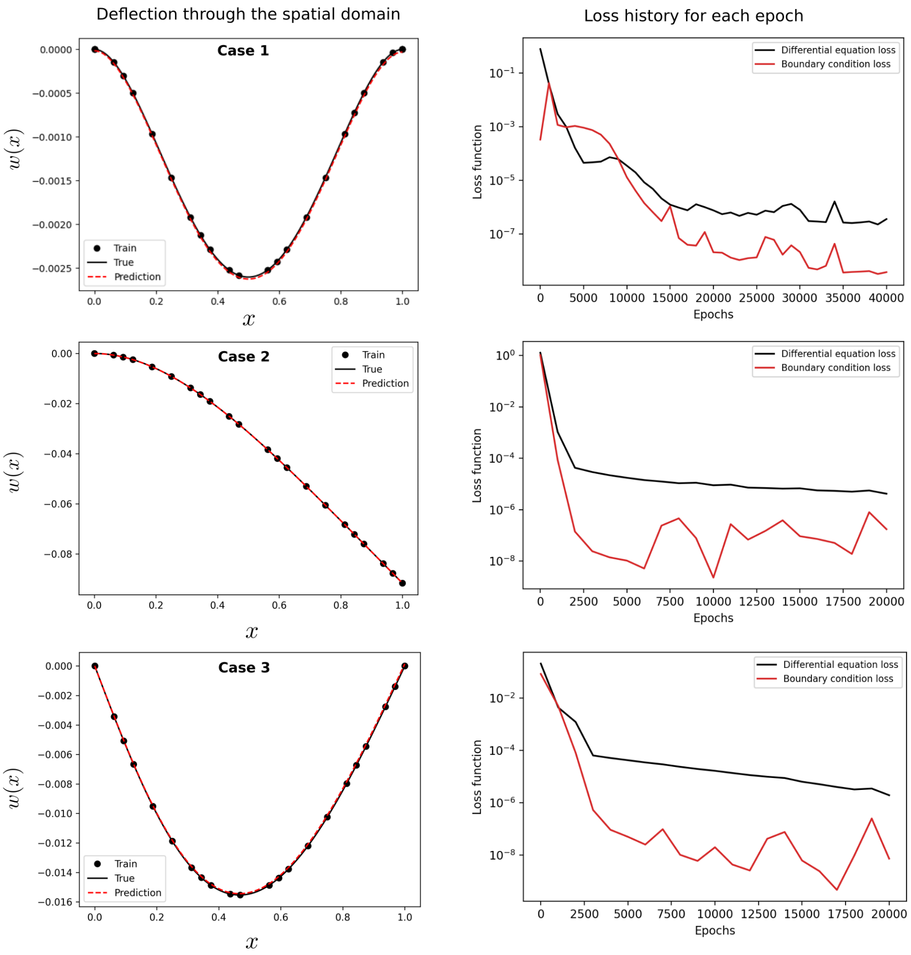


Figure 4.3: On the figures left, the predicted and true deflections through the spatial domain are compared for three different cases. On the figures right, the loss history containing the differential equation and boundary condition losses are given for each epoch.

the total boundary losses. The reason is that the boundary conditions are not explicitly enforced. In Eq. 4.6, the boundary condition losses are added to overall lost and this overall loss is minimized. To enforce the boundary conditions, the loss function can be reformulated to a soft constraint problem or penalty methods can be considered [Lu+21].

A complex consideration

So far, simple beam structures with uniform material and geometrical properties under non-complex loadings have been investigated. Consider a simply supported beam having the following non-uniform cross-section

$$I(x) = \left(\frac{x}{L}\right)^2 \tag{4.7}$$

under the following loading

$$p(x) = \frac{8\pi^2 \left(\left(2\pi^2 x^2 - L^2\right) \sin(\frac{2\pi x}{L}) - 4\pi x \cos(\frac{2\pi x}{L}) \right)}{L^4}, \tag{4.8}$$

as depicted in Fig. 4.4. Applying the boundary conditions (Eq. 3.6) and using the governing equation of the beam theory (4.1), the analytical solution is obtained

$$w(x) = \sin(\frac{2\pi x}{L}). \tag{4.9}$$

For simplicity, Young's modulus is set to $E = 1$ and the length of the beam is set to $L = 1$.

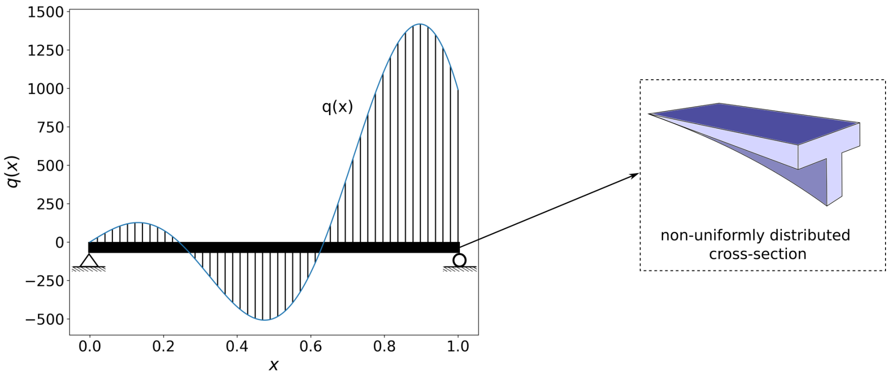


Figure 4.4: An illustration of a simple supported beam having non-uniformly distributed cross-section under an arbitrary load

The results of the investigated complex model is given Fig. 4.5. Similar to the previously conducted simpler models, figure on the left shows that the predicted displacements are almost identical to the true solution. On the contrary, the difference between differential equation loss and boundary equation loss is large compared to the simple models (see 4.3). The difference is due to the magnitude of the quantities. Since the applied load has a large magnitude compared to the predicted displacement, the initial differential loss was 10^8 larger than the boundary loss. During the training of the network, the difference becomes smaller but they don't converge the same value since the boundary condition loss is not enforced explicitly which turns into a unconstrained optimization problem if it is forced explicitly. Moreover, as a result of the complexity of the model, the loss has more oscillations and does not smoothly converge.

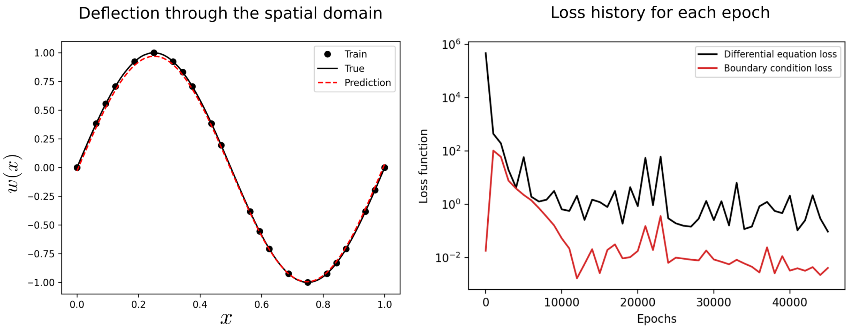


Figure 4.5: On the figure left, the predicted and true deflection through the spatial domain are represented. On the figure right, the loss history containing the differential equation and boundary condition losses are given for each epoch.

4.2 Continuous-Time Model

Involving the time term to the Euler beam theory transforms the problem into a vibration problem and the behavior of the beam is governed by the Euler-Lagrange equation. The partial differential equation of the one-dimensional Euler-Lagrange beam including Neumann and Dirichlet boundary conditions with initial conditions is formulated as

$$\rho A \frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left(EI \frac{\partial^2 w(x,t)}{\partial x^2}\right) + q(x,t) = 0 \quad \text{on } \mathcal{T} \otimes \Omega \tag{4.10}$$

$$\frac{\partial w(x,t)}{\partial x} = h \quad \text{on } \mathcal{T} \otimes \Gamma_N, \tag{4.11}$$

$$w(x,t) = g \quad \text{on } \mathcal{T} \otimes \Gamma_D, \tag{4.12}$$

$$w(x,0) = g_0 \quad \text{and} \quad w_t(x,0) = g_{t,0} \quad \text{on } \Omega. \tag{4.13}$$

Here, in addition to the spatial domain Ω in the static models, the temporal domain $\mathcal{T}$ is introduced. To give a better understanding, the spatial domain is defined $\Omega = [0, L = 1]$, with $\Gamma_D = \{x | x = 0, x = 1\}$ and $\Gamma_N = \{x | x = 0, x = 1\}$ and the temporal domain i set to $\mathcal{T} = [0, 1]$. Since the beam is particularly considered as a clamped support Euler-Lagrange beam, the following homogenous boundary conditions are subjected to the problem

$$\begin{aligned} w(0,t) = 0 \quad &\text{and} \quad w_x(0,t) = 0, \\ w(1,t) = 0 \quad &\text{and} \quad w_x(1,t) = 0, \end{aligned} \tag{4.14}$$

with the following non-homogenous initial conditions

$$w(x,0) = \sin(\pi x) \quad \text{and} \quad w_t(x,0) = 0. \tag{4.15}$$

Similarly, physics-informed neural network approximates the unknown solution $w(x,t)$ using the spatial domain x and temporal domain t as inputs as illustrated in Fig. 4.6. Since the predicted solution has to fulfill the governing equation of Euler-Lagrange problem, residuum f is defined as

$$f := \rho A \frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 w(x,t)}{\partial x^2}\right) + q(x,t). \tag{4.16}$$

Combining Eqs. 4.14, 4.15, 4.16 with 2.2, 2.5, 2.6, 2.7, 2.8, 2.10, the loss function L of a dynamic clamped supported Euler-Lagrange beam is derived as

$$\begin{aligned}
L &= E_{Neumann} + E_{Dirichlet} + E_0 + E_f \\
&= \frac{1}{N_b}\sum_{i=1}^{N_b}\left[\left(\frac{\partial}{\partial x}w_P\left(x_b^i, t_b^i\right) - h\right)^2 + \left(w_P\left(x_b^i, t_b^i\right) - g\right)^2\right] + \\
&\quad \frac{1}{N_0}\sum_{i=1}^{N_0}\left[\left(\frac{\partial}{\partial t}w_P\left(x_0^i, 0\right) - g_{t,0}\right)^2 + \left(w_P\left(x_0^i, 0\right) - g_0\right)^2\right] + \frac{1}{N_f}\sum_{i=1}^{N_f}\left(f_P(x_f^i, t_f^i)\right)^2 \\
&= \frac{1}{N_b}\sum_{i=1}^{N_b}\left[\underbrace{\left(\frac{\partial}{\partial x}w_P\left(x_b^i, t_b^i\right)\right)^2}_{\text{Neumann BC.}} + \underbrace{\left(w_P\left(x_b^i, t_b^i\right)\right)^2}_{\text{Dirichlet BC.}}\right] + \\
&\quad \frac{1}{N_0}\sum_{i=1}^{N_0}\left[\underbrace{\left(\frac{\partial}{\partial t}w_P\left(x_0^i, 0\right) - \sin(\pi x)\right)^2 + \left(w_P\left(x_0^i, 0\right)\right)^2}_{\text{Initial C.}}\right] + \frac{1}{N_f}\sum_{i=1}^{N_f}\left(f_P(x_f^i, t_f^i)\right)^2
\end{aligned} \tag{4.17}$$

While the Neumann and Dirichlet loss terms enforces the boundary conditions at N_b random points $\left\{x_b^i, t_b^i\right\}_{i=1}^{N_b}$ where $x = 0$ and $x = 1$, the initial loss terms enforce the initial conditions at $\left\{x_0^i, 0\right\}_{i=1}^{N_0}$, since the temporal domain is set to $t_b = 0$. Finally, the last term assures that the predicted displacement fulfills the governing equation of the problem at random N_f collocation points $\left\{x_f^i, t_f^i\right\}_{i=1}^{N_f}$ where the spatial domain $x \in [0,1]$ and the temporal domain $t \in [0,1]$.

The further calculation of the continuous-time model of the Euler-Lagrange beam, Young's modulus and the moment of inertia terms are set to $AE = 1$ and additionally the density and the area of the cross-section terms are set to $\rho A = 1$. Applying the following distributed load

$$q(x,t) = sin(\pi x)e^{-t}\left(\pi^4(t+1) + t - 1\right), \tag{4.18}$$

results in the analytical solution of the clamped Euler-Lagrange beam with the given boundary and initial conditions (see 4.14 and 4.15) as follows

$$w(x,t) = \sin(\pi x)(t+1)e^{-t}. \tag{4.19}$$

Compared to the PINN architecture of static models, a slightly more advanced model is generated since the input has two dimensions. The PINN architecture to predict the space-time solution $w(x,t)$ has the following features

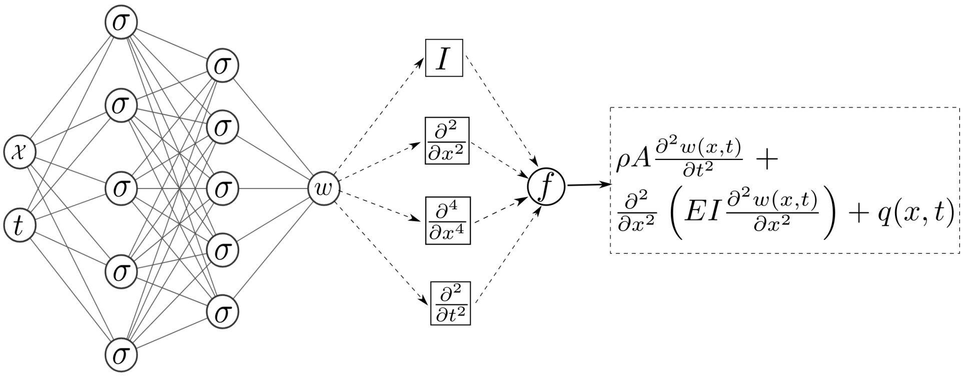

x
t
σ
w
I
$\frac{\partial^2}{\partial x^2}$
$\frac{\partial^4}{\partial x^4}$
$\frac{\partial^2}{\partial t^2}$
f
$\rho A \frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left(EI \frac{\partial^2 w(x,t)}{\partial x^2}\right) + q(x,t)$

Figure 4.6: An illustration of physics-informed neural network for the governing partial differential equation of the one-dimensional Euler-Lagrange beam.

- Architecture contains 2 input layers, 1 output layer and 3 hidden layers containing 100 nodes.
- Similarly Hyperbolic tangent activation functions *tanh* are used with $GlorotUniform$ initializer.
- The *Adam* optimizer with learning rate $l_r = 0.0001$ is chosen as the optimization method. The number of *epochs* is set as 20000.
- The number of collocation points inside domain is set to $N_f = 20$, the number of boundary condition points is set to $N_b = 50$ and the number of initial condition point is set to $N_0 = 50$. Additionally, the *Sobol* sampling is performed have a random distribution.

The results of the Euler-Lagrange beam is given in Fig. 4.8. The comparison of the predicted and analytical solutions at different space and time intervals shows that the PINN architecture can predict the solution quite well if the time term is kept constant. On the other hand, the predicted and analytical solutions have a small gap if the space term is taken as constant. To fill this gap, the model is trained longer or the parameters of the PINN architecture are optimized using hyper-parameter optimization techniques.

In contrast to the static models, the overall loss contains also the initial condition loss as depicted in Fig. 4.7. The loss on the known conditions have a smooth decay to the convergence level, while the differential equation loss has local oscillations due to the difference in magnitudes of order since the small changes in the predicted solution can lead to relatively large oscillations during the training.

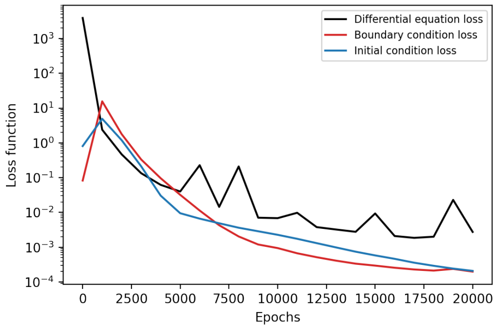


Figure 4.7: The loss history containing the differential equation and boundary condition losses are given for each epoch.

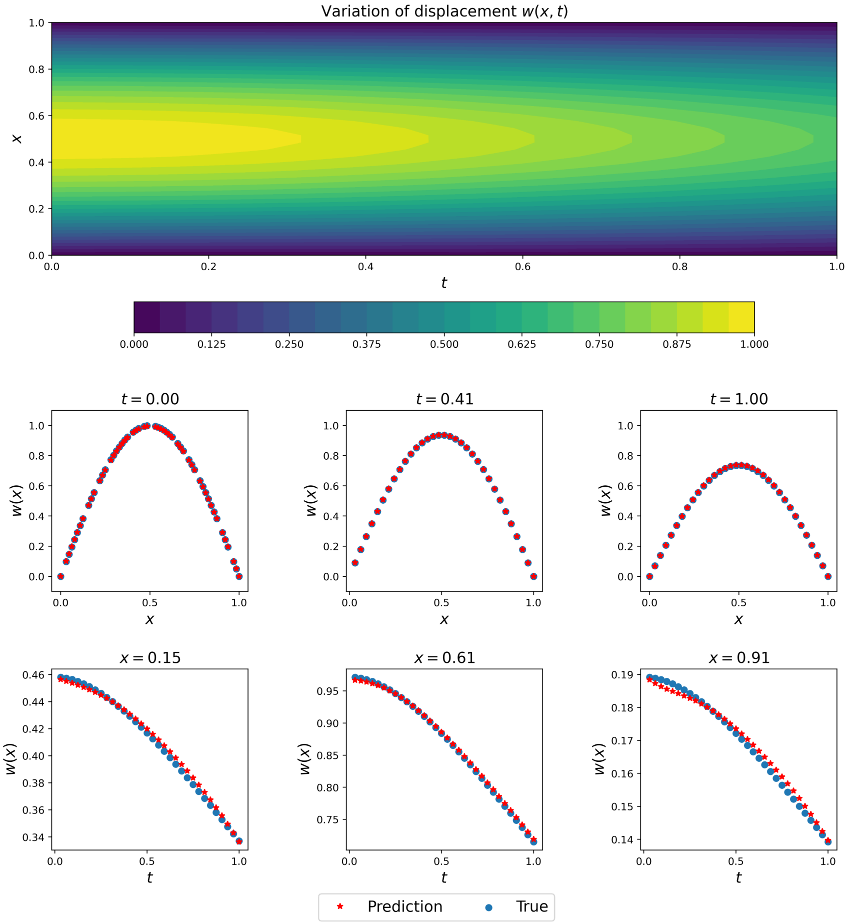


Figure 4.8: The results of the Euler-Lagrange beam. The figure with the colorbar gives the predicted solution through the space-time domain.

Bibliography

[Aba+16] Abadi, M. et al. 'Tensorflow: A system for large-scale machine learning'. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 2016, pp. 265–283.

[Kol+] Kollmannsberger, S. et al. *Deep Learning in Computational Mechanics: An Introductory Course.*

[Lu+21] Lu, L. et al. 'Physics-informed neural networks with hard constraints for inverse design'. In: *arXiv preprint arXiv:2102.04626* (2021).

[Pas+19] Paszke, A. et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Ed. by Wallach, H. et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[RPK19] Raissi, M., Perdikaris, P. and Karniadakis, G. E. 'Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations'. In: *Journal of Computational Physics* vol. 378 (2019), pp. 686–707.

[WQ07] Wang, J. and Qiao, P. 'Vibration of beams with arbitrary discontinuities and boundary conditions'. In: *Journal of Sound and Vibration* vol. 308, no. 1-2 (2007), pp. 12–27.