Graduate thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Haga, Jakob Muan

Applications of physics-informed neural networks for uncertainty quantification and parameter estimation in poroelastic problems

Graduate thesis in Civil and Environmental Engineering
Supervisor: Depina, Ivan
June 2023

NTNU
Norwegian University of
Science and Technology

Haga, Jakob Muan

Applications of physics-informed neural networks for uncertainty quantification and parameter estimation in poroelastic problems

Graduate thesis in Civil and Environmental Engineering
Supervisor: Depina, Ivan
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This work is a master's thesis in geotechnical engineering at the Norwegian University of Science and Technology (NTNU) and the final part of an MSc program in Civil and Environmental Engineering. The work was carried out during the spring semester of 2023 and is the continuation of a previous project work conducted in the autumn semester of 2022.

The supervisor of the thesis has been Ivan Depina.

Trondheim, 11.06.2023

Acknowledgement

I would first and foremost like to thank my supervisor Ivan Depina for our constructive discussions and his inspiring enthusiasm for the subject.

It has been both educational and fun delving into a topic that combines geotechnical engineering and computer science, one of which was uncharted territory for me. Fortunately, the topic turned out to be right up my alley, with statistics and numerical computation serving as unifying connections.

Lastly, I want to thank my classmates throughout my five years at NTNU for being great friends, colleagues, and a source of constant support. Your camaraderie and collaboration have made this academic journey truly memorable and fulfilling. Thank you for being an integral part of my educational experience and for making it an unforgettable chapter in my life.

Abstract

This master thesis investigates the application of Physics Informed Neural Networks (PINNs) for uncertainty quantification in poroelastic problems. The main research question focuses on evaluating the accuracy and computational efficiency of PINNs, specifically in comparison to traditional numerical methods and commercial Finite Element Method (FEM) programs. The overarching objective is to assess the potential benefits of PINNs for uncertainty quantification and statistical analysis of high-dimensional transient problems and their suitability for solving forward and inverse problems.

The thesis presents the results obtained from testing PINNs on several poroelastic problems. The investigated problems include simple 1D and 2D uncoupled consolidation problems where the total stresses are assumed to be constant, as well as a more advanced coupled 2D problem. The coupled 2D consolidation problem was solved using a sequential model split, where the partial differential equations (PDEs) describing the flow problem of the fluid and the PDEs describing the equilibrium equations of the soil were solved iteratively. The PINN model demonstrates its capability to accurately capture the physical behavior of all three problems. Notably, the PINN model showcases its ability to learn and represent advanced poroelastic phenomena even in the absence of data.

While the accuracy of forward computations using PINNs is found to be slightly lower than that of commercial FEM programs, the overall results are promising. The PINN model successfully captures the essential physical behavior of the poroelastic problems tested, making it a valuable tool for simulations. The computational efficiency of PINNs significantly outperforms traditional numerical methods once the model is trained, providing substantial time savings.

In conclusion, this study establishes the potential of PINNs as a computational tool for expensive simulations in poroelastic problems. Despite minor differences in accuracy compared to commercial FEM programs, PINNs offer a sufficient level of accuracy for many practical applications. The exceptional computational efficiency of PINNs, combined with their ability to handle high-dimensional problems, positions them as a promising approach for uncertainty quantification, parameter estimation, and similar simulation tasks of poroelastic phenomena.

Keywords: Physics Informed Neural Networks (PINNs), uncertainty quantification, maximum likelihood estimation, poroelastic problems, computational efficiency.

Sammendrag

Denne masteroppgaven utforsker anvendeligheten til Fysikk informerte Neurale Nettverk (PINNs) for å kvantifisere usikkerhet og parameterestimere i typiske problemstillinger knyttet til poroelastisitet. Hovedfokuset til oppgaven er å evaluere nøyaktigheten til predikeringen av modellen, samt beregningsmessig effektivitet. Disse prestasjonsmålene har blitt brukt i en sammenligning med tradisjonelle numeriske metoder og kommersielle elementmetode (FEM) program. Det overordnede formålet med oppgaven er å vurdere potensielle fordeler med bruk av PINNs for flerdimensjonale transiente analyser, samt brukbarheten i vanlige beregninger og inverterte problemstillinger.

Oppgaven presenterer resultater fra bruk av PINNs for å løse flere klassiske problemer knyttet til poroelastisitet. De løste problemene varierer fra enkle en- og todimensjonale konsolideringsproblemer hvor totalspenningene er antatt å være konstante til en mer avansert todimensjonal problemstilling hvor vannstrømning og deformasjon av løsmassene påvirker hverandre. Det koblede problemet ble løst med en sekvensiell PINN modell, hvor de partielle differensialligningene som beskriver vannstrømningen og likevekstligningen for løsmassene ble løst iterativt. Resultatene fra modellenes prediksjoner viser at PINNs er kapabel til å lære alt fra en- og todimensjonal diffusjon, til likevektsligninger, til koblede likningssett med transiente partielle differensialligninger, helt uten data.

Nøyaktigheten til beregningene med bruk av PINNs ble vist å være litt lavere enn beregningene til kommersielle FEM program, men generelt er resultatene lovende. PINN modellen klarer å lære de grunnleggende fysiske fenomenene som beskriver de ulike problemene, noe som gjør det til et verdifullt verktøy for simulering av problemer med usikre parametre. Tidsbruken som går med for å gjøre en prediksjon med PINNs etter at den er ferdig trent er betydelig lavere enn tilsvarende beregning med tradisjonelle numeriske metoder. Dette gir betydelige tidsbesparelser for statistiske analyser av probabilistiske problemer, noe som resultatene og diskusjonen viser.

Konklusjonen er at resultatet av denne masteroppgaven viser potensialet til PINNs som beregningsverktøy for beregningskrevende simulering av probabilistiske poroelastiske problemer. Sett bort i fra små forskjeller i nøyaktighet sammenlignet med kommersielle FEM program, viser resultatene at PINNs nøyaktighet vil være godt nok for bruk i mange praktiske situsjoner. Den eksepsjonelle beregningseffektiviteten, kombinert med egenskaper som kan takle flerdimensjonale optimaliseringsproblemer, tilsier at PINNs er en lovende metode for usikkerhetskvantifisering, parameteroptimalisering og andre simuleringsoppgaver, innen poroelastiske og andre geotekniske problemstillinger.

Table of Contents

List of Figures

List of Tables

Abbreviations

1D One-dimensional

2D Two-dimensional

3D Three-dimensional

ANN Artificial Neural Networks

CPTU Cone Penetration Test with water pressure measurement

FDM Finite Difference Method

FEM Finite Element Method

FNN Feed-forward Neural Networks

MAE Mean absolute error

MCMC Markov Chain Monte Carlo

MLE Maximum Likelihood Estimation

MSE Mean squared error

MSLE Mean square logarithmic error

PDEs Partial differential equations

PDF Probability Density Function

PINNs Physics-Informed Neural Networks

List of symbols

Greek symbols

$\boldsymbol{\alpha}$	Biot coefficient
$\boldsymbol{\alpha_i}$	Nondimensional constant for variable **i**
$\boldsymbol{\gamma_f}$	Specific weight fluid
$\boldsymbol{\gamma_{sat}}$	Specific weight soil
$\boldsymbol{\gamma_w}$	Specific weight water
$\boldsymbol{\delta}$	Infinitesimal change
$\boldsymbol{\delta_{ij}}$	Kronecker delta
$\boldsymbol{\delta_v}$	Vertical displacement
$\boldsymbol{\Delta}$	Change
ε	Strain tensor
ε_v	Volumetric strain
$\boldsymbol{\theta}$	Unknown parameter
$\boldsymbol{\kappa_{ij}}$	Permeability tensor
$\boldsymbol{\mu}$	Mean
$\boldsymbol{\mu_f}$	Viscosity fluid
$\boldsymbol{\mu_w}$	Viscosity water
$\boldsymbol{\nu}$	Poisson ratio
$\boldsymbol{\nu_u}$	Undrained Poisson ratio
$\boldsymbol{\xi}$	Natural coordinate
$\boldsymbol{\rho}$	Mass density
$\boldsymbol{\sigma_{ij}}$	Total stress tensor
$\boldsymbol{\sigma'_{ij}}$	Effective stress tensor

Latin symbols

a_i	Input tensor
b_i^j	Bias tensor
c_{ijkl}	Stiffness tensor
C_v	Consolidation coefficient
E	Error
E	Young's modulus
E_{oed}	Odeometer modulus
f_i	Body force
G	Shear modulus
g_i	Gravitation vector
H	Height of soil column
k_{ij}	Hydraulic conductivity/permability
K_f	Bulk modulus fluid
K	Bulk modulus soil
K_f	Bulk modulus fluid
K_s	Bulk modulus solid
n	Porosity
p	Mean total stress
p_f	Pore fluid pressure
p_w	Pore water pressure
q	External load
q_i	Darcy velocity
S	Storativity
T	Time
u	Excess pore pressure
u_i	Displacement tensor
U	Uniform distribution
U_p	Degree of consolidation
v	Horisontal displacement
V	Volume
w	Vertical displacement
w_{ij}^k	Weight tensor
X	Random variable
y	Data
$\hat{y}$	Prediction
z_j^k	Input activation function

Other symbols

$\mathcal{L}$	Loss/Likelihood
$\mathcal{N}$	Normal distribution

1 Introduction

1.1 Background

Poroelasticity is the study of the interaction between a fluid and a porous solid, which has important applications in various fields, such as soil mechanics, engineering geology, reservoir engineering, biomechanics, and other related engineering fields. In many cases, understanding the behavior of poroelastic interaction requires solving complex partial differential equations (PDEs), which can be computationally expensive and time-consuming. Recently, there has been growing interest in using Artificial Neural Networks (ANN) to approximate the solutions of these PDEs, leading to the development of Physics-Informed Neural Networks (PINNs). Combining observed data with governing physical laws allows for efficient modeling of both forward and inverse problems. This is especially useful for scientific fields which require efficient surrogate models or engineering fields working with uncertain parameters.

The interaction between fluid flow and deformation in porous media gives rise to complex phenomena characterized by inherent uncertainties. Accurate estimation of parameters such as hydraulic conductivity, porosity, and elastic properties is essential for predicting the behavior of poroelastic systems. However, these parameters exhibit spatial and temporal variability, posing challenges in their characterization and quantification. Uncertainties arising from sources such as measurement errors and model assumptions further contribute to the overall uncertainty in poroelastic analysis. Therefore, rigorous quantification and propagation of uncertainty in poroelastic modeling are crucial for reliable predictions and informed decision-making in geotechnical engineering. Incorporating the temporal variability of parameters and addressing uncertainties could benefit the entire field of geotechnical engineering, enabling the design and assessment of geotechnical works under transient conditions with improved accuracy and robustness.

Dealing with uncertainty is a crucial aspect of geotechnical engineering, particularly in the context of parameter and model uncertainty. Accurate in-situ measurements and laboratory tests of soil behavior are both difficult and costly. In regards to poroelasticity, which is often associated with transient problems like consolidation or fluid injection, both parameter and model uncertainty are present. Additionally, uncertainties regarding changes in external factors like external loading, groundwater conditions, climate, and surrounding development may require extensive scenario analyses.

The importance of model accuracy in the context of uncertainty quantification and optimization problems cannot be overstated. In the domain of uncertainty quantification, precise and trustworthy estimations of uncertainties associated with input parameters or variables are of great value. Achieving a high degree of accuracy in the model ensures that the quantified uncertainties possess a commendable level of reliability. Consequently, decision-makers can make well-informed choices based on the confidence levels derived from the model. Likewise, in the field of optimization problems, accurate models facilitate the generation of precise predictions and evaluations of functions and constraints. Such accuracy proves instrumental in enabling efficient and effective optimization processes, as it guides the search for optimal solutions, mitigating the need for excessive iterations and minimizing computational costs. In both uncertainty quantification and optimization, the dependability and resilience of outcomes are dependent upon the accuracy of the underlying model, thereby making the accuracy a critical factor for the successful resolution of complex problems and the facilitation of well-informed decision-making.

1.2 Research objectives

The primary aim of this thesis is to investigate the applicability of PINNs as effective numerical approximators for uncertainty quantification in problems related to poroelasticity. To evaluate the performance of a numerical model, the main criterion is usually its accuracy in describing the physical behavior. However, computational efficiency is almost equally important for uncertainty quantification, particularly when dealing with extensive simulation and optimization tasks necessary for evaluating the statistical moments or finding the optimal solution with multiple variables.

To evaluate the viability of PINNs in this context, several performance criteria will be addressed. First and foremost, the accuracy of PINNs as numerical approximators will be assessed to ensure reliable predictions and realistic modeling of the physical phenomena. Additionally, precision, referring to the ability of PINNs to provide precise estimations and capture subtle variations, will be examined. Efficiency will also be considered, assessing the computational effectiveness of PINNs compared to traditional numerical methods and analytical approximation. Furthermore, the interpretability and explainability of the PINN-based approach will be frequently evaluated and discussed, with the objective of gaining insights into the model's learning ability and exploring potential benefits or shortcomings. By addressing these performance criteria, this research aims to provide a comprehensive understanding of the potential benefits and limitations of PINNs in tasks related to uncertainty quantification for poroelastic problems.

1.2.1 Sub objectives

To evaluate these performance criteria, the following tests were conducted

- PINNs performance in predicting problems related to Terzhagi's one-dimensional (1D) consolidation theory was tested by comparisons with traditional numerical methods and approximation of the analytical solution. Particularly the models' ability to interpolate and extrapolate the learning from the training was tested, to evaluate one of the core advantages of PINNs compared with FEM.
- PINNs behavior when exposed to probabilistic variables was tested and compared with simulations of the analytical solution, both in terms of accuracy and computational efficiency.
- PINNs efficiency and accuracy in solving inverse problems through parameter estimation was tested. Both for solving a deterministic problem and uncertainty quantification for noisy measurements.
- Lastly, PINNs performance in solving two-dimensional (2D) transient problems of coupled and uncoupled flow-deformation problems in consolidation theory was evaluated and compared with FEM results. The model's performance in the coupled problem serves the purpose as a benchmark for how advanced poroelastic models it is able to learn without data, and with limited computational power.

1.3 Limitations

While this thesis delves into the investigation of poroelasticity, it is important to acknowledge the limitations of the scope of this research. Firstly, the analysis is restricted to

the application of poroelasticity, which by definition is limited to elastic materials. The focus primarily revolves around materials with highly compressible pores, thereby placing relatively less emphasis on the compressibility of the solid and fluid. Moreover, this research predominantly addresses fully saturated problems, assuming the fluid to be water. Consequently, the findings and conclusions of this thesis should be interpreted within the confines of these limitations, which may impact the generalizability of the results to scenarios involving different material characteristics, nonlinear behavior, or diverse fluid compositions.

Due to limited computational power and memory capacity, only problems demanding less than 6GB RAM were possible to compute. This rules out most three-dimensional (3D) problems and some 2D problems because of the memory-demanding storage of weights and tensors during training. This is also the reason why the problems with additional input parameters were done for the 1D consolidation problem, and not 2D.

The primary aim of this thesis is to investigate the capability of PINN as a numerical tool. Therefore, the discussion will mainly focus on the relevant topics linking neural networks and numerical methods. PINNs and other machine learning models' efficacy in data-rich scenarios, such as image classification or unsupervised pattern recognition, is beyond the intended scope of this thesis.

2 Theory

The theory chapter is divided into two main subcategories: poroelasticity and PINNs.

The chapter on poroelasticity gives a brief overview of the relevant theory on soil mechanics, fluid flow in porous media, and how they are combined in poroelastic theory.

The chapter on PINNs gives a brief introduction to ANNs in general, to establish the theoretical foundation PINNs are built off of. This section emphasizes the primary distinctions between ANNs and PINNs, highlighting the unique aspects, advantages, and challenges of PINNs. Furthermore, the theory behind computational methods for uncertainty quantification and parameter estimation is presented, supplementing the theory on PINNs. Some basic aspects of the relationship between data or measurements and PINNs are also mentioned, with some relevant examples from poroelastic problems to bridge the gap between the sections.

2.1 Poroelasticity

Poroelasticity is commonly associated with consolidation processes, fluid injection, reservoir engineering, and other fields where fluid interacts with or flows through a porous media (Cheng, 2016). Consolidation theory is an important sub-field of soil mechanics because it is used to describe the long-term settlement of low-permeable soil. For soil mechanics purposes, the porousness and the elasticity of the soil material are the fundamental material properties in regard to poroelastic computations. The porousness of a soil material is dependent on grain size, grain shape, and degree of compaction. This implies that in regard to poroelasticity, fine-grained materials and coarse materials behave inherently differently.

2.1.1 Isotropic elasticity

Elasticity is the most common and basic mathematical framework describing the mechanical behavior of a material in both soil mechanics and continuum mechanics in general. The common form of elasticity assumes a linear stress-strain relationship, isotropic material, and small deformations (Irgens, 2008). All governing equations in this theory are linear partial differential equations, which means that the principle of superposition may be applied.

The general form of Hooke's law, relating stresses and strains for an elastic material, is

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl} \tag{1}$$

The strain tensor, ε_{ij} for infinitesimal strains, is defined as

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{2}$$

Isotropic conditions in elasticity theory refer to the assumption that the material's stiffness properties are uniform in all directions. The stiffness tensor c_{ijkl} for isotropic conditions is defined as

$$c_{ijkl} = \frac{3K - 2G}{3}\delta_{ij}\delta_{kl} + G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) \tag{3}$$

where K is the bulk modulus and G is the shear modulus, defined as

$$K = \frac{E}{3(1 - 2\nu)} \tag{4}$$

$$G = \frac{E}{2(1 + \nu)} \tag{5}$$

Inserting Equation (3) in Hooke's law from Equation (1) gives

$$\sigma_{ij} = \frac{3K - 2G}{3}\delta_{ij}\varepsilon_{kk} + G(\varepsilon_{ij} + \varepsilon_{ji}) \tag{6}$$

By assuming small strains and isotropic conditions, the normal strain components can be added together to approximate the volumetric strain (Cheng, 2016).

$$\varepsilon_v = \frac{\Delta V}{V} \approx \varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33} = \nabla \cdot u \tag{7}$$

The deviatoric part of the total strain tensor can followingly be defined as

$$\bar{\varepsilon}_{ij} = \varepsilon_{ij} - \delta_{ij}\frac{\varepsilon_v}{3} \tag{8}$$

where δ_{ij} is the Kronecker delta function defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases} \tag{9}$$

Similarly for stress, defining the mean isotropic total stress p as

$$p = \frac{\sigma_{11} + \sigma_{22} + \sigma_{33}}{3} \tag{10}$$

gives the deviatoric stress as

$$\bar{\sigma}_{ij} = \sigma_{ij} - \delta_{ij}p. \tag{11}$$

2.1.2 Effective stress

The first underlying assumption of poroelasticity, as first introduced by Terzahgi, is that the total stress affecting an isotropic soil body can be decomposed into effective stress caused by intergranular contact of particles in the soil skeleton and the pressure exerted by the fluid in the pores (Skempton, 1984).

$$\sigma_{ij} = \sigma'_{ij} + \alpha p_w \delta_{ij} \tag{12}$$

This calls for a secondary definition of mean isotropic stress, namely mean isotropic effective stress

$$\Delta p' = \frac{\Delta\sigma'_{11} + \Delta\sigma'_{22} + \Delta\sigma'_{33}}{3} \tag{13}$$

The volumetric stress-strain relation for small strains can then be described as

$$\Delta p' = K\Delta\varepsilon_v \tag{14}$$

For geotechnical engineering purposes, the relevant fluids are both liquid and gas. Therefore, soils are in general a three-phase material of solid grains surrounded by pores filled with air and water. The Biot coefficient α from equation (12) can be used to consider the effect of compressible pores in relation to the solid particles

$$\alpha = 1 - \frac{K}{K_s} \tag{15}$$

For soils with high porosity, the bulk modulus of the solid particles is much larger than the material as a whole. The Biot coefficient will therefore typically be set to one for soft soils like clay and silt, and close to one for granular soil.

2.1.3 Isotropic plane strain conditions

Plane strain conditions are perhaps the most common assumptions when modeling 3D problems in 2D. The assumption is that the out-of-plane strain is zero, i.e. $\varepsilon_{22} = 0$. This is realistic for problems with loading in one plane and a relatively large width in the direction of the zeroed strain component (Irgens, 2008). For problems where isotropic plane strain conditions can be assumed, Hooke's law reduces to

$$\sigma'_{ij} = \frac{3K - 2G}{3}\delta_{ij}\varepsilon_{kk} + 2G(\varepsilon_{ij}) \tag{16}$$

where $\sigma'_{12} = \sigma'_{21}$ because of symmetry. In matrix form, this is equivalent to

$$\begin{bmatrix} \sigma'_{11} \\ \sigma'_{22} \\ \sigma'_{12} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{12} \end{bmatrix} \tag{17}$$

Similarly, the strain vector reduces to

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{12} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u_1}{\partial x_1} \\ \dfrac{\partial u_2}{\partial x_2} \\ \dfrac{1}{2}\left(\dfrac{\partial u_1}{\partial x_2} + \dfrac{\partial u_2}{\partial x_1}\right) \end{bmatrix} \tag{18}$$

where $\varepsilon_{12} = \varepsilon_{21}$ because of symmetry.

2.1.4 Undrained conditions

Introducing external loading to a soil material in equilibrium will, in accordance with equation (12), be balanced by increased effective stress and pore pressure. For soils with a high Biot coefficient, the deformation of the soil is almost exclusively due to compaction of the pore space. This causes a high initial increase in pore pressure if the loading is applied faster than the velocity of the fluid in the pores. For an instant loading of soil with $\alpha = 1$, all strains depend solely on the pore fluid's bulk modulus (Nordal, 2020).

Pores filled with both air and water will, on the other hand, typically have $K_w << \text{inf}$ thus resulting in $\alpha < 1$, and therefore contract upon loading causing immediate volume reduction (Skempton, 1984). Pores filled with only water will have a large bulk modulus and therefore barely contract, causing in many cases negligible volume change. According to Hooke's law from equation (1), stress is dependent on strain, so declaring $\Delta\varepsilon_v = 0$ simultaneously declares a change in mean stress in the soil skeleton $\Delta p' = 0$. Fluids are on the other hand exerting pressure regardless of volume change. So for instant loading of a fully saturated soil body, the change in normal total stress at $t = 0$ is

$$\delta_{ij}\Delta\sigma_{ij} = \delta_{ij}\Delta p_w. \tag{19}$$

The assumption that water cannot take any shear stresses from equation 12 still holds, which implies that the grain skeleton takes all shear stresses

$$(1 - \delta_{ij})\Delta\sigma_{ij} = (1 - \delta_{ij})\Delta\sigma'_{ij}. \tag{20}$$

This short-term undrained condition can last from seconds in highly permeable soil to months in low permeable soil.

Numerical modeling of undrained conditions is often done with an alternative set of parameters. For realistic scenarios, $\Delta\varepsilon_v \neq 0$ due to somewhat compressible fluid and solid (Nordal, 2020). The FEM program Plaxis 2D offers multiple options for modeling the short-term stress-strain relationship. For undrained effective stress analysis, called undrained A in Plaxis, there are two primary ways to define the undrained parameters. One of the alternatives is to define the material's Biot parameter α and water bulk modulus K_w (Skempton, 1984). The alternative, which is the default option, is to define $\lim_{\nu \to 0.5}$ so that the bulk modulus $\lim_{K_u \to \infty}$. To avoid singularity in the stiffness matrix, the default value for ν is 0.495. Further is Skempton's B-parameter, relating the rate of change in excess pore pressure and rate of change in mean total stress, $\Delta u = B\Delta p$ defined as

$$B = \frac{\alpha}{\alpha + n\left(\frac{K}{K_w} + \alpha - 1\right)}. \tag{21}$$

Inserting Skempton's B-parameter in Equation (12) gives the relation between mean effective and total stress defined in Equation (13) and (10) (Skempton, 1984).

$$\Delta p' = (1 - \alpha B)\Delta p \tag{22}$$

2.1.5 Equation of motion

Based on the assumptions of balanced linear and angular momentum, the equation of motion for continuous materials is the Cauchy equation of motion (Irgens, 2008).

$$\sigma_{ij,j} + \rho b_i = \rho a_i \tag{23}$$

Equivalently in matrix notation

$$\nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{b} = \rho \boldsymbol{a} \tag{24}$$

Combining Hooke's law (1) and the relation between effective stress and total stress from Equation (12) gives a combined equation of stress, strains, and pore pressure

$$\Delta\sigma_{ij} = c_{ijkl}\varepsilon_{kl} + \delta_{ij}\alpha\Delta p_w \tag{25}$$

For isotropic plane strain conditions as Equation (16), this is simplified to

$$\sigma_{ij} = \frac{3K - 2G}{3}\delta_{ij}\varepsilon_{kk} + 2G(\varepsilon_{ij}) + \delta_{ij}\alpha\Delta p_w \tag{26}$$

which can be rewritten as

$$\sigma_{ij} = \delta_{ij}(K\varepsilon_{kk} + \alpha\Delta p_w) + 2G(\varepsilon_{ij} - \frac{1}{3}\delta_{ij}\varepsilon_{kk}) \tag{27}$$

Inserting the displacement tensors for the strains in the latter expression yields

$$\varepsilon_{ij} - \frac{1}{3}\delta_{ij}\varepsilon_{kk} = \frac{1}{2}(u_{i,j} + u_{j,i}) - \frac{1}{3}u_{i,i} = \delta_{ij}\frac{1}{6}\nabla \cdot \boldsymbol{u} + \frac{1}{2}\nabla^T \boldsymbol{u} \tag{28}$$

Inserting the expression of strain, displacement and pore pressure for the stress tensor in the equation of motion (23), and utilizing the Laplace-operator

$$\begin{gathered} \delta_{ij}\frac{1}{6}\nabla \cdot (\boldsymbol{\nabla} \cdot \boldsymbol{u}) = \delta_{ij}\frac{1}{6}\frac{\partial \varepsilon_{vol}}{\partial x_i} \\ \frac{1}{2}\nabla \cdot \boldsymbol{\nabla}^{\boldsymbol{T}}\boldsymbol{u} = \frac{1}{2}\nabla^2 u_i \end{gathered} \tag{29}$$

gives the Navier equations for the displacement field

$$\left(K + \frac{G}{3}\right)\frac{\partial \varepsilon_{vol}}{\partial x_i} + \alpha\frac{\partial p_w}{\partial x_i} + G\nabla^2 u_i + \rho b_i = \rho a_i. \tag{30}$$

By enforcing equilibrium, i.e. the acceleration a is prescribed to zero, the equations of equilibrium in the differential form are obtained (Cheng, 2016). Written out, the 2D equilibrium equations for isotropic plane strain conditions are (Verruijt, 2016).

$$\begin{aligned}\left(K+\frac{G}{3}\right)\frac{\partial \varepsilon_{vol}}{\partial x_1}+\alpha\frac{\partial p_w}{\partial x_1}+G\nabla^2 u_1+\rho b_1=0\\ \left(K+\frac{G}{3}\right)\frac{\partial \varepsilon_{vol}}{\partial x_2}+\alpha\frac{\partial p_w}{\partial x_2}+G\nabla^2 u_2+\rho b_2=0.\end{aligned} \tag{31}$$

Equivalently, the equations in regard to stresses are

$$\begin{aligned}\frac{\partial \sigma_{11}}{\partial x_1}+\frac{\partial \sigma_{12}}{\partial x_2}+f_1=0\\ \frac{\partial \sigma_{22}}{\partial x_2}+\frac{\partial \sigma_{12}}{\partial x_1}+f_2=0\end{aligned} \tag{32}$$

2.1.6 Flow in porous media

Fluid flow in porous media is highly dependent on the degree of saturation, the permeability of the soil, and the pressure distribution. The governing equation for incompressible fluid flow through porous media is Darcy's law (Whitaker, 1986).

$$q_i=-\frac{\kappa_{ij}}{\mu_f}(\nabla p_f-\rho_f g_i) \tag{33}$$

where q_i is the Darcy velocity, κ_{ij} is the permeability tensor of the soil, μ_f is the viscosity of the fluid, ρ_f is the mass density of the fluid and g_i is the gravity tensor. In soil mechanics, the permeability k_{ij} is commonly given as hydraulic conductivity

$$k_{ij}=\kappa_{ij}\frac{\rho_f g_i}{\mu_f} \tag{34}$$

which reduces Darcy's law to

$$q_i=-\frac{k_{ij}}{\gamma_f}(\nabla p_f-\rho_f g_i)=-k_{ij}\nabla h \tag{35}$$

by assuming gravity in one direction and introducing the hydraulic head $h=p_f\gamma_f^{-1}+z$, where z is the elevation head in the direction of gravity. By enforcing mass conservation for both the fluid and the solid, the net change of mass in a soil body can be described by the storage equation (Verruijt, 2016).

$$\nabla q_i+\frac{\partial \varepsilon_v}{\partial t}+n\left(\frac{1}{K_f}-\frac{1}{K_s}\right)\frac{\partial p}{\partial t}+\frac{1}{K_s}\frac{\partial p_f}{\partial t}=0. \tag{36}$$

where p is the isotropic mean stress from equation (14) and n is the porosity. By substituting ∇q_i with Darcy's law from Equation (35), a second-order partial differential equation is obtained (Verruijt, 2016).

$$\alpha\frac{\partial \varepsilon_v}{\partial t}+S\frac{\partial p_f}{\partial t}=\nabla\left(\frac{k_{ij}}{\gamma_f}\nabla p_f\right) \tag{37}$$

Equation (37) is commonly known as the storage equation, which describes transient fluid flow in porous media. The storage equation is a coupled PDE, which involves solving a differential equation for two functions simultaneously. In this case, this means that the deformation of the soil material affects the fluid flow and vice versa. Coupled PDEs are notoriously hard to solve, and numerical approximations are computationally expensive since multiple equations have to be solved simultaneously. The methods for solving coupled flow-deformation problems can be divided into two categories, fully coupled methods and sequential methods (Kim et al., 2011). Fully coupled methods solve the coupled problem simultaneously for small time increments, whereas sequential methods divide the coupled problem into a set of smaller equations that can be solved separately and then couple the physical behavior by updating the overlapping parameters through iteration back and forth. The stability of two possible sequential methods, fixed-strain and fixed-stress split, were evaluated in a study by Kim et al. (2011). The findings showed that the fixed-stress split was stable for $\alpha \geq 0.5$, where α is the control parameter of the time integration via the generalized midpoint rule. The fixed-strain split, on the other hand, showed instability for fully coupled flow-deformation. Both methods involve solving the flow problem first, and then the equilibrium equations using the results from the flow problem.

Algorithm 1 Sequential solver

procedure FIXED-STRAIN SPLIT($u_{init}, \varepsilon_{v,init}$)
    $u = u_{init}$
    $\varepsilon_v = \varepsilon_{v,init}$
    for `n number of iterations` **do**
        `Solve storage equation for` $u(\delta\dot{\varepsilon_v} = 0)$
        $u \leftarrow$ `Storage equation`
        `Solve equilibrium equations for` $\varepsilon_v(u)$
        $\varepsilon_v \leftarrow$ `Equilibrium equations`

procedure FIXED-STRESS SPLIT(u_{init}, p_{init})
    $u = u_{init}$
    $p = p_{init}$
    for `n number of iterations` **do**
        `Solve storage equation for` $u(\delta\dot{p} = 0)$
        $u \leftarrow$ `Storage equation`
        `Solve equilibrium equations for` $p(u)$
        $p \leftarrow$ `Equilibrium equations`

2.1.7 Uncoupled consolidation

There are two main assumptions that can simplify the coupled storage equation to an uncoupled consolidation equation.

If only the vertical displacement of the surface is of interest, and the load covers a relatively large surface area compared with depth, the horizontal strains ε_{xx} and ε_{yy} can be assumed to be negligible (Nordal, 2020). Strain in one direction causes

$$\frac{\partial \varepsilon_v}{\partial t} = \frac{\partial \varepsilon_{zz}}{\partial t} \propto \frac{\partial (\sigma_{zz} - u)}{\partial t} \tag{38}$$

If the vertical total stress is constant in time, the storage equation (79) effectively reduces

to a 3D transient diffusion equation (Verruijt, 2016).

$$\frac{\partial u}{\partial t} = C_x \frac{\partial^2 u}{\partial x^2} + C_y \frac{\partial^2 u}{\partial y^2} + C_z \frac{\partial^2 u}{\partial z^2} \tag{39}$$

Another possible uncoupling assumption is that the total isotropic stress stays constant in time (Verruijt, 2016). The result is the same as for horizontally confined problems

$$\frac{\partial \varepsilon_v}{\partial t} \propto \frac{\partial (p - u)}{\partial t} = \frac{\partial (-u)}{\partial t} \tag{40}$$

With strain in two directions,

$$\Delta \varepsilon_v = \frac{\Delta u}{K} \tag{41}$$

Inserting (41) in the storage equation gives a similar equation to (39), but the bulk modulus replaces the consolidation coefficients

$$\left(\frac{1}{K} + S \right) \frac{\partial u}{\partial t} = \frac{k_x}{\gamma_w} \frac{\partial^2 u}{\partial x^2} + \frac{k_y}{\gamma_w} \frac{\partial^2 u}{\partial y^2} + \frac{k_z}{\gamma_w} \frac{\partial^2 u}{\partial z^2} \tag{42}$$

2.2 Physics-Informed Neural networks

The following chapter is based on a project work by Haga (2022). The chapter starts by introducing and defining hybrid modeling, an overarching discipline of various modeling techniques, which includes PINNs. Followingly, Artificial Neural Networks (ANNs) are introduced, providing a comprehensive discussion of their general aspects as well as the specific components relevant to PINNs.

2.2.1 Hybrid modelling

When developing or designing material products or structures, computational models and simulations are commonly used to simulate physical behavior. The classical method, physics-based models, uses numerical methods to solve problems defined by mathematical equations. The emerging alternative method is data-driven modeling, which utilizes data from measurements or simulations to create predictions for new scenarios through high dimensional interpolation (Kurz et al., 2022). Combining the two approaches creates a new third methodology, commonly called hybrid modeling.

As illustrated in Figure 1, hybrid modeling can be defined as models that combine physics-based computation and data-driven computation in one model architecture. This definition includes PINNs, so PINNs can be regarded as a methodology within hybrid modeling.

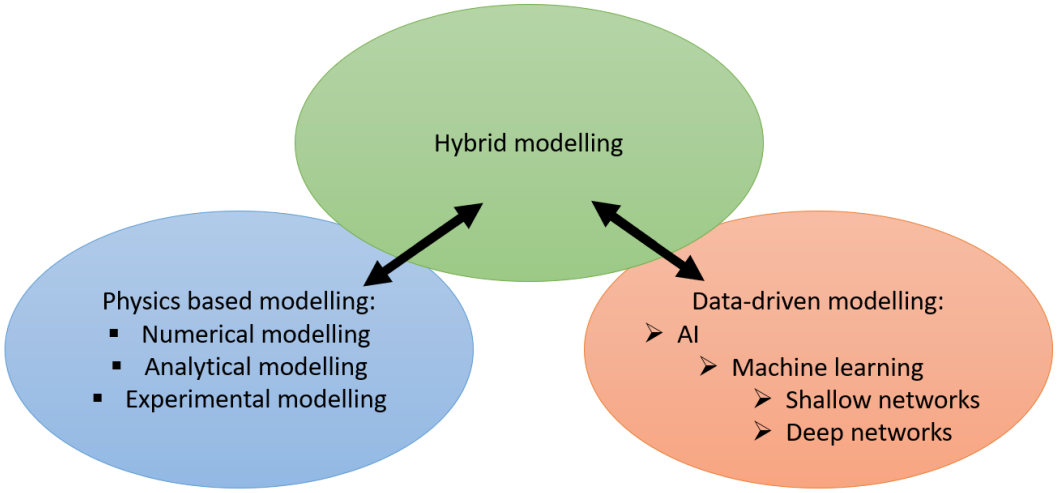


Figure 1: An illustration of hybrid modeling as the intersection between data-driven and physics-based modeling. Figure from Haga (2022).

The focus in this thesis going forward will be on PINNs specifically, but due to loose definitions, a lot of the theoretical basis overlaps with for instance scientific machine learning. Based on the definitions in Kurz et al. (2022), PINNs is a methodology within scientific machine learning, which, in turn, is a methodology within hybrid modeling, demonstrating its hierarchical nature within the broader field of computational modeling.

2.2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are among the most widespread types of machine learning due to their versatility and effectiveness (Engelbrecht, 2007). ANNs are in general algorithms whose functionality is designed to mimic that of the human brain. The primary objective for ANNs is to utilize the large processing power of modern-day computers with the human capability of pattern recognition and rationality.

ANNs are commonly categorized after functionality. For hybrid modeling, the most relevant methodology is Feedforward Neural Networks (FNN). FNNs are capable of serving as universal function estimators because of their abilities in dimension reduction and recognizing non-linear relations between input and output. FNNs have gained the most attention for use in regards to hybrid modeling, but some research exists on utilizing recursive neural networks for sequential time series analysis, like this research from Wu et al. (2022) for NVIDIA.

Due to the scope of this thesis, ANNs will imply FNN models for further reading.

2.2.3 Architecture

The architecture of an ANN algorithm is a design inspired by the biology of the human brain. It consists of linked neurons, where information flows in the form of mathematical matrix operations (Russell and Norvig, 2009). Figure 2 shows a simple network architecture where the inputs are connected to a network of neurons which again is connected to

the output. The illustration shows one sample with h number of inputs and h^* number of outputs, and a network of n x m neurons. Models with more than one hidden layer are often associated with the term deep learning, as opposed to shallow networks or perceptrons which respectively consist of one hidden layer and only one neuron.

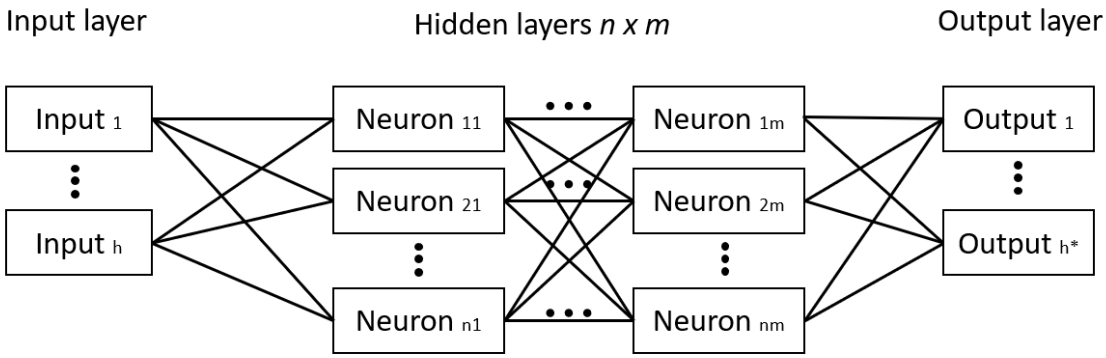


Figure 2: An illustrative example of a simple ANN architecture. Figure from Haga (2022).

2.2.4 Flow of information in forward propagation

Forward propagation or forward pass, are terms used to describe the information flow when the model predicts output from input (Russell and Norvig, 2009). The size of the input layer, h in Figure 2, is determined by the dimensionality of the problem. More rigorously, for an h-dimensional problem space, the input layer can be represented by an h x 1 vector. Each link, indicated by black lines in Figure 2, between inputs, neurons, and output consists of a scalar value, as illustrated in Figure 3. These links are commonly denoted weights. The weights determine the relative contribution each input and each neuron should have in determining the output of the next neuron or the entire model. The weights can effectively be represented using matrix or tensor notation, because of their tight structure and symmetry. This facilitates both effective computation and efficient data storage. The notation differs depending on the network architecture. If the number of neurons in the hidden layers is constant, one weight tensor can describe all connections between neurons. The first and last weight tensor is usually defined separately.

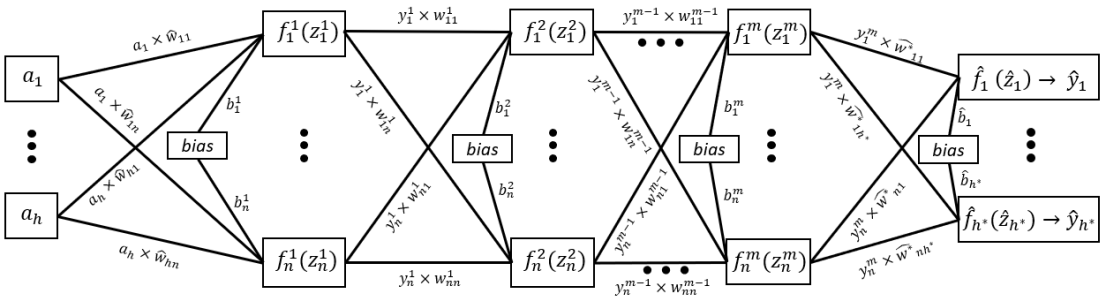


Figure 3: An undirected flowchart of information in a simple ANN model. Figure from Haga (2022).

The role of neurons in the information flow can be described mathematically as summing the incoming weighted information, adding bias, and calculating its output with an acti-

vation function. For the first hidden layer in a model with h inputs, this can be written as:

$$z_j^1 = \sum_{i=1}^{h} \hat{w}_{ij} \times a_i + b_j^1 \tag{43}$$

The neuron calculates its output based on an activation function with z_j^k as input. A specific activation function may be chosen for individual neurons, layers, or globally for the entire model.

$$Neuron_j^k : f_j^k(z_j^k) = y_j^k \tag{44}$$

For hidden layer k:

$$z_j^k = \sum_{i=1}^{n} w_{ij}^{k-1} \times y_i^{k-1} + b_j^k \tag{45}$$

For the output layer:

$$\hat{z_j} = \sum_{i=1}^{n} \hat{w}_{ij}^* \times y_i^m + \hat{b}_j \tag{46}$$

The activation function determines the size of the output and may deactivate the output altogether. The choice of activation functions is dependent on the nature of the problem, network size, and desired computational efficiency. Activation functions are further discussed in relation to PINNs in chapter 2.3.3.

2.2.5 Training process

The training process of an ANN is dependent on whether the learning is supervised or unsupervised (Goodfellow et al., 2016). Supervised learning involves feeding the model labeled input and output data, to train the model in seeing recurring patterns between the value of the inputs and the value of the belonging outputs. Unsupervised learning involves feeding the model unlabeled input and output, thus training the model to explore patterns by itself and not make it dependent on provided solutions. For PINNs, the training is more similar to supervised learning than unsupervised, mostly due to the data being labeled.

Before the training starts, the dataset is usually normalized and split into training, validation, and testing sets. The normalization of the dataset is usually making sure the inputs and outputs are of roughly the same magnitude. Preprocessing will be further elaborated in the chapter on PINNs, due to some unique challenges and possibilities. After the preprocessing, the network architecture and hyperparameters should be determined based on the dimensionality of the input and the complexity of the relation between input and output. Hyperparameters are the user-specified parameters in the network. The counterpart is parameters, which the model evaluates by itself during the training.

The training phase in supervised learning is similar to an optimization problem through trial and error. When training starts, the weights and biases are initialized, and the first batch of labeled input is sent through the forward pass. The forward pass ends with a predicted output, which is evaluated against the labeled output data with a loss function.

Relevant loss functions for regression analysis are:

$$\text{Mean Square Error: (MSE)} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{47}$$

$$\text{Mean Absolute Error: (MAE)} = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|) \tag{48}$$

$$\text{Mean Square Logarithmic Error: (MSLE)} = \frac{1}{n}\sum_{i=1}^{n}(log(y_i + 1) - log(\hat{y}_i + 1))^2 \tag{49}$$

MSE MSE is considered to be the default loss function because of its simplicity and history in classic statistical methods (Russell and Norvig, 2009). MSE computes the square of the error, which may be unfortunate for predictions of data with many outliers.MAE is a variant of MSE that works better with outliers but is only first-order differentiable. MSLE is another option, which provides a relative error estimate. It also introduces asymmetry where underestimation yields a bigger loss than overestimation. This can be useful in reliability analysis for engineering purposes.

ANNs use the total loss from the loss function to train the model through backpropagation (Russell and Norvig, 2009). After the first prediction, when the error is calculated, the system starts backcalculating the gradients of the loss function in regard to the model parameters. The gradients are calculated with automatic differentiation, which effectively calculates the derivatives based on the chain rule (Raissi et al., 2017).

The gradients of the weights in the output layer can be described mathematically as:

$$\frac{\partial E}{\partial \hat{w}_{ij}} = \frac{\partial E}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{z}_j}\frac{\partial \hat{z}_j}{\partial \hat{w}_{ij}} \tag{50}$$

where the loss function is defined as $E(y_i, \hat{y}_i)$.

The gradients in the hidden layer before the output layer can be described as:

$$\frac{\partial E}{\partial w_{ij}^m} = \frac{\partial E}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{z}_j}\frac{\partial \hat{z}_j}{\partial \hat{y}_j^m}\frac{\partial y_j^m}{\partial z_j^m}\frac{\partial z_j^m}{\partial w_{ij}^m} \tag{51}$$

The gradients in hidden layer r can be described as:

$$\frac{\partial E}{\partial w_{ij}^r} = \frac{\partial E}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{z}_j}\frac{\partial \hat{z}_j}{\partial y_j^m}\frac{\partial y_j^m}{\partial z_j^m}\frac{\partial z_j^r}{\partial w_{ij}^r}\prod_{l=r}^{m-1}\frac{\partial z_j^{l+1}}{\partial y_{ij}^l}\frac{\partial y_j^l}{\partial z_j^l} \tag{52}$$

The gradients of the loss in regard to the biases, $\frac{\partial E}{\partial b_j^r}$, are calculated in a similar pattern.

By definition, gradients quantify the impact of a small change in a system. In the context of backpropagation in ANNs, if parameters associated with large gradients undergo a small change, it will have a relatively large effect on the result of the loss function (Raissi et al., 2017). This suggests that optimizing these parameters is beneficial. These parameters, namely weights and biases, are updated through optimization algorithms.

Optimization algorithms seek to minimize the loss function, which in turn reduces the model error and increases performance. Numerous mathematical optimization techniques

exist, many of which are suitable for ANNs. The primary method, accompanied by various variants, is gradient descent. This approach utilizes gradients computed through automatic differentiation during backpropagation to iteratively update the model parameters. The gradient descent algorithm traverses in the negative direction within the space of $E(y_i, \hat{y}_i)$, as depicted in Figure 4. The learning rate η determines the magnitude of changes in weights and biases during this iterative process. A larger η may lead to faster, but inaccurate convergence. Smaller η, on the other hand, may get stuck in local minima.

$$\Delta w_{ij}^r = -\eta \frac{\partial E}{\partial w_{ij}^r} \tag{53}$$

$$w_{ij,(Updated)}^r = w_{ij,(Old)}^r + \Delta w_{ij}^r \tag{54}$$

$$\Delta b_j^r = -\eta \frac{\partial E}{\partial b_j^r} \tag{55}$$

$$b_{j,(Updated)}^r = b_{j,(Old)}^r + \Delta b_j^r \tag{56}$$

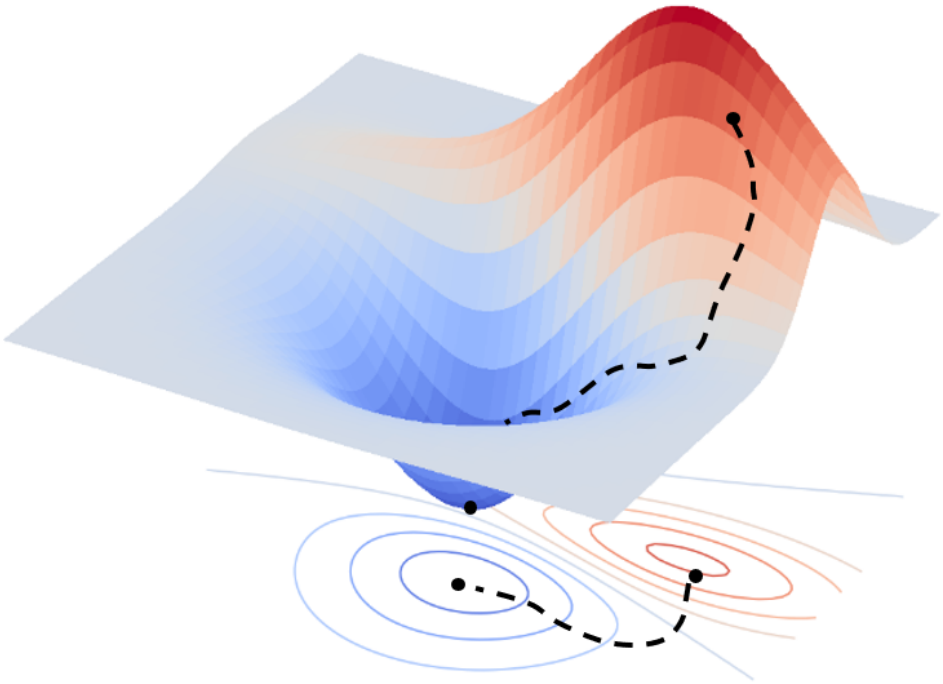


Figure 4: A simple illustration of gradient descent. Figure from Haga (2022)

Deep neural networks often consist of large architectures, resulting in complex gradients as depicted in equation 52. However, evaluating gradients for the entire dataset at each iteration is impractical due to computational cost. To address this, a common approach is to employ a variant of gradient descent known as stochastic gradient descent (Bottou, 2012). Stochastic gradient descent involves randomly selecting a single input instead of using

the entire training set, which significantly accelerates computation and reduces memory requirements. On the other hand, introducing randomness and using small samples may lead to noisy predictions.

To strike a balance between efficiency and accuracy, most optimizers based on stochastic gradient descent employ mini-batches, which involve using subsets of data rather than individual samples (Brownlee, 2018). This approach offers a trade-off between computational speed and accuracy. The size of the random subset, known as the batch size, can be adjusted as a hyperparameter.

Both Tensorflow and PyTorch provide built-in optimizers, with the most popular options being variants of stochastic gradient descent. One widely used optimizer is the Adam optimizer, which combines stochastic gradient descent with an adaptive learning rate (Kingma and Ba, 2014). Adam utilizes the first and second statistical moments of the gradients to dynamically adjust the learning rate based on the characteristics of the gradients. This adaptive approach enables larger steps at the beginning and smaller steps as the optimizer approaches a potential global minimum.

In addition to the mentioned hyperparameters, there are some tuning parameters related to the learning process. The number of epochs determines how many times the training dataset is processed by the network during training (Brownlee, 2018). Increasing the number of epochs often leads to better predictions of the training set and a decrease in the loss function. However, using a high number of epochs can result in overfitting, where the model becomes excessively specialized to the training data.

The iterations per epoch, calculated as the number of epochs divided by the batch size, is important to consider when fixing number of epochs and batch size. Understanding the relationship between epochs and batch size is particularly important in the context of the backpropagation mentioned earlier. Utilizing numerous epochs and small batch sizes results in a higher number of forward- and backpropagation iterations, in turn leading to increased computational costs. Consequently, the selection of the number of epochs and batch size should be made with consideration to their interdependence.

Another important aspect of training the ANN is how to initialize the weights and biases (Glorot and Bengio, 2010). A poor choice of weight initialization may cause slower convergence or even divergence. If the weights are initialized to small or large values, the gradients may vanish or explode during the backpropagation. Symmetric weight initialization should be avoided to reduce the risk of symmetric learning which could cause ineffective neurons. Glorot and Bengio (2010) proposed a popular weight initialization that ensures small values, non-symmetry and scales the standard deviation of the weights according to the input and output size of the network.

$$w_{ij}^{n} \sim \mathcal{N}\left(\mu = 0,\, \sigma = \sqrt{\frac{2}{h + h^{*}}}\right) \tag{57}$$

Or the similar uniformly distributed

$$w_{ij}^{n} \sim U\left(-\sqrt{\frac{6}{h + h^{*}}}, \sqrt{\frac{6}{h + h^{*}}}\right) \tag{58}$$

2.3 Physics Informed Neural Networks

The concept of Physics-Informed Neural Networks (PINNs) was first introduced in 2017 by Raissi et al. (2017). The primary objective was to establish a framework that combines the governing physical principles with data-driven computational methods. The underlying motivation of the objective was to utilize physics as a means to achieve faster convergence in data-driven predictions, and reversely utilize data-driven prediction to increase the efficiency of numerical modeling. In many instances, accurately predicting non-linear, high-dimensional physical phenomena solely through data-driven approaches requires a significant amount of data.

The idea behind PINNs was to incorporate physical constraints into the learning phase by penalizing the model when its predictions contradict established physical behavior.

$$\mathcal{L}_{total} = \mathcal{L}_{pde} + \mathcal{L}_{ic} + \mathcal{L}_{bc} + \mathcal{L}_{data} \tag{59}$$

This approach can be viewed as a form of regularization in traditional supervised learning. It also shares similarities with the concept of soft constraints from optimization theory.

In subsequent discussions, the physics integrated into the ANN network will be referred to as "physical constraints".

2.3.1 Applications

Raissi et al. (2017) highlighted three applications for PINNs: Forward solving, inverse solving, and data-driven discovery. The following definition of forward and inverse problems are used in this thesis:

- Forward problems are equations where the functional is the unknown
- Inverse problems or parameter estimation are problems where backcalculating the causal effect from observed data is the primary goal.

In comparison with purely data-driven approaches, there are other benefits besides faster convergence and increased accuracy. One of the main challenges with ANNs, is the lack of interpretability, which could lead to difficulties in trusting the output. Good implementation of PINNs can reveal parts of the machine learning process during the training phase that remains unknown in regular ANNs. For instance, the user may monitor how the losses are developing for each constraint. The losses after training the model may reveal if any of the constraints has a high residual error relative to the others, which could have been missed if only the total loss was calculated.

2.3.2 Preprocessing of the constraints

Data preprocessing plays a crucial role in various tasks related to ML. Preprocessing is usually viewed as a phase in the learning process, where different tasks in regard to data cleaning and preparation are done to ensure effective training of the model. For PINNs, the unique aspects when it comes to preprocessing are normalization and train-test split (Haghighat et al., 2022).

The purpose of data normalization is to address issues arising from the relative magnitudes of different data inputs. When certain inputs have significantly larger values compared to others, it can introduce bias in the model and impede learning convergence. In ANNs, normalization typically involves scaling all inputs and outputs to a range of 0 to 1 or -1 to 1, which is commonly referred to as min-max scaling.

For PINNs, the procedure of normalization is less straightforward due to the dependencies of the physical properties and their units. In order to normalize the PDEs, nondimensionalization has to be performed on both the variables occurring in the PDE and the PDE itself. Like standard normalization, this is usually done by min-max scaling, where the maximum values of the variables are used as the scale so the range of values is 0 to 1. This creates a set of new variables, usually denoted $\hat{z} = \frac{z}{H}$ or $t^* = \frac{t}{T}$. By inserting the nondimensionalized variables in the PDE, a nondimensionalized equation is obtained which the literature in unison promotes as being optimal to ensure efficient convergence.

The train-validation-test split can be done in two ways: randomly or user-picked (Xu and Goodacre, 2018). A random split of training and testing data ensures an unbiased selection of samples and is the default option for supervised learning. However, if the models' capability of interpolation or extrapolation is of interest, test data can be extracted from a specified area. This is especially relevant for PINNs, where the domain may be spatial or temporal. For instance, picking the last time steps in a transient problem to be in the test set will test the models' ability to extrapolate to a greater degree than random selection. If the problem domain is spatial, picking an entire section of an area to be the test data can test the models' ability to interpolate.

2.3.3 Activation functions

The choice of activation function for PINNs is limited to the functions that are differentiable up to the degree of the governing equations (Raissi et al., 2017). The most popular functions in the literature are listed in Table 1.

Table 1: Table of relevant activation functions for PINNs.

Name	Function	Characteristic
Hyperbolic Tangent	$f(x) = \frac{2}{1+e^{-2x}} - 1$	A continuous nonlinear function which is infinitely differentiable
Sigmoid	$f(x) = f(x) = \frac{1}{1+e^{-x}}$	A continuous nonlinear function which is infinitely differentiable
Swish	$f(x) = \frac{x}{1+e^{-x}}$	A continuous nonlinear function which is infinitely differentiable

Figure 5 shows the behavior of the listed activation function for relevant input ranges. Hyperbolic tangent (Tanh) and sigmoid behave similarly, but Tanh ranges from -1 to 1 whereas sigmoid ranges from 0 to 1. The activation happens in the same input range, so the gradient of Tanh is considerably steeper than the sigmoid which is relevant for the

gradient decent optimization. Swish has similar gradient steepness as Tanh, but does not have an upper bound and is asymmetrical around 0.

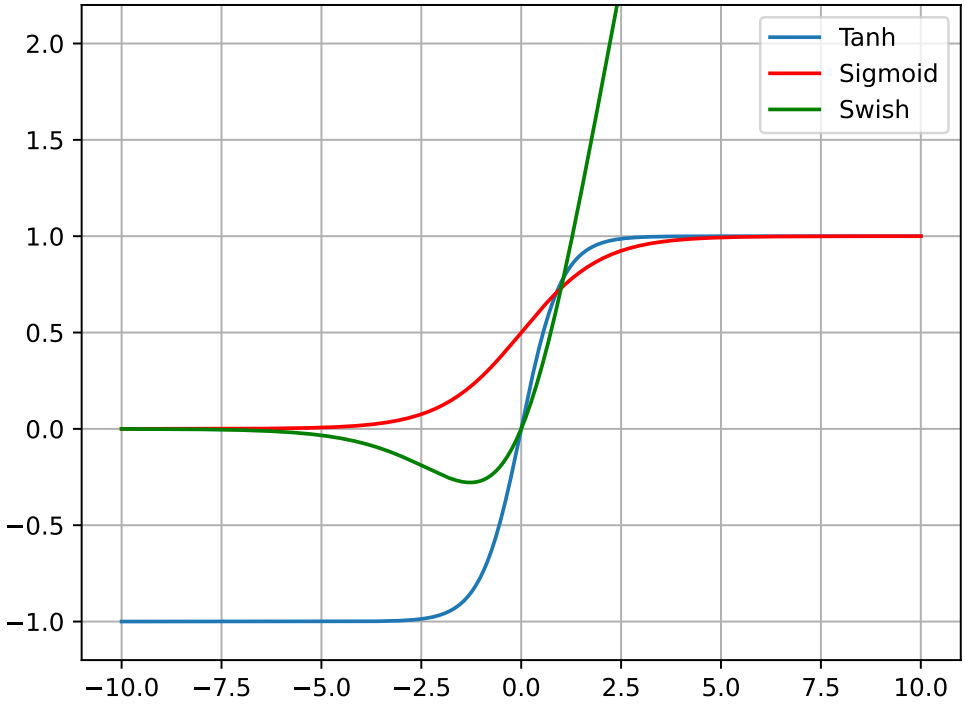


Figure 5: Plot of the activation functions behavior for relevant input values.

There are no agreed-upon guidelines for choosing an activation function, apart from the differentiability. The three mentioned activation functions are all infinitely differentiable, so they qualify as activation functions for any PDE.

2.3.4 PINNs for coupled problem

PINNs have previously been tested for related problems to those conducted in this project, both in solid mechanics, poromechanics, and other fields of physics and engineering. A paper by Bekele (2020) studied PINNs performance for forward and inverse calculation of the 1D consolidation problem. The researchers behind the computational framework SCIANN Haghighat and Juanes (2021) has explored PINNs applicability in coupled poroelasticity, in addition to several related fields like elasticity, elastoplasticity and fluid mechanics (Haghighat et al., 2022). The coupled flow-deformation problem discussed in Haghighat et al. (2022) was Mandel´s problem, and the sequential solver from Algorithm (1) was used to train the model.

2.4 Uncertainty quantification and statistical inference

Uncertainty quantification is the collective term of all attempts of quantifying the uncertainty of a system or a model (Smith, 2013). For modeling purposes, the uncertainty is

commonly classified into two categories, epistemic and aleatoric uncertainty. The definition states that epistemic uncertainty is uncertainty that can be reduced by more knowledge, whereas aleatoric uncertainty is nonreducible (Box and Tiao, 2011). Epistemic uncertainty is therefore often described as the uncertainty of the model, which can occur due to numerical errors like in FEM, simplifications like dimension reduction, or limited knowledge of parameters or variables. Aleatoric uncertainty occurs from inherent randomness in the nature of the problem like stochastic processes or random fields. The total uncertainty of a model can therefore be quantified by adding the epistemic uncertainty of the model and the aleatoric uncertainty of the problem. Estimating the uncertainty of a model can be done by comparison with a benchmark like analytical solutions, test measurements, or other similar models. PINNs provide some information about their confidence via the loss function. The loss function for PINNs gives information on not only the global error but also the local error in each boundary, PDE, or other constraints. Due to the inherent randomness in the stochastic gradient descent algorithm and the weight initialization, each training will give a different final weight matrix. For some aspects like reproducibility and interpretability, this inherent randomness is unwanted. However, randomness can be used to identify uncertainty and variability in the model output. Dropout regularization, a technique frequently used for regularizing ANNs to prevent overfitting, can be used for this purpose (Gal and Ghahramani, 2016). The term Dropout simply refers to a random deactivation of neurons, thus creating slightly different models for each training iteration, illustrated in Figure 6.

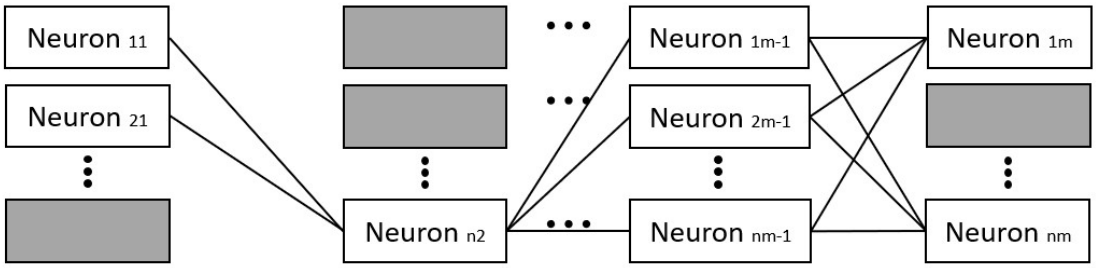


Figure 6: A simple illustration of the dropout concept.

This can help with preventing symmetrical training and inactive neurons, and the added randomness can also be used for measuring precision in the model output. By keeping the random deactivation of neurons after the training, the model prediction will be slightly different for the same input. The deviance of the predictions can be estimated through Monte Carlo simulations to determine the mean and standard deviation of the model. This method is particularly useful if there is a lack of benchmark methods to compare the output with.

2.4.1 Maximum likelihood Estimation

Maximum Likelihood Estimation (MLE) is a method commonly used for estimating the parameters of a probability distribution (Myung, 2003). The likelihood function $\mathcal{L}(\theta) = P(D|\theta)$ evaluates the likelihood that the observed data was caused by a fixed θ. $\mathcal{L}(\theta)$ can

be estimated by evaluating

$$\mathcal{L}(\theta) = P(\boldsymbol{D}|\theta) = \prod_{i=1}^{k} P(D_i|.\theta) \tag{60}$$

The MLE can then be found by finding

$$\hat{\theta} = \arg\max \mathcal{L}(\theta). \tag{61}$$

If the posterior probability distribution of the random variable is of interest, Bayes theorem can be used if prior knowledge of the random variable exists.

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \tag{62}$$

The likelihood is used as new evidence to update the prior belief of $P(\theta)$. The normalizing constant $P(D)$ can be evaluated as

$$P(D) = \int_{\theta} P(D|\theta)P(\theta)d\theta. \tag{63}$$

2.4.2 Monte Carlo sampling

Estimating the likelihood function $P(X|\boldsymbol{\theta})$ can be a computationally demanding task due to the wide range of possible θ values. To limit the number of computations needed for sufficient convergence, utilizing sampling techniques is common practice. Some of the easiest and most popular sampling techniques are based on Monte Carlo methods. Monte Carlo methods involve evaluating the numerical problem for random samples of the problem's random variables. For stochastic processes, this usually involves calculating the average outcome, the standard deviation of the outcomes, or logging the number of extreme events.

Monte Carlo methods can also be used in the inverse problem to estimate a probability function for the unknown parameters. Markov Chain Monte Carlo (MCMC) methods, a combination of Monte Carlo theory and Markov Chain theory, are often used for this task (Chib and Greenberg, 1995). Markov chains are commonly used for simplifying the degrees of freedom in stochastic processes, by assuming that the probability of each event is only dependent on the previous event. This simplification allows for multi-dimensional random walks which can be used to sample effectively from multi-dimensional probability distributions. An example of an MCMC algorithm is the Metropolis-Hasting algorithm. The Metropolis-Hastings algorithm performs what is essentially a random walk rejection sampling, where the direction is drawn from a chosen probability distribution and the rejection of samples is determined based on a random evaluation of the new state based on the previous.

Algorithm 2 Metropolis-Hastings

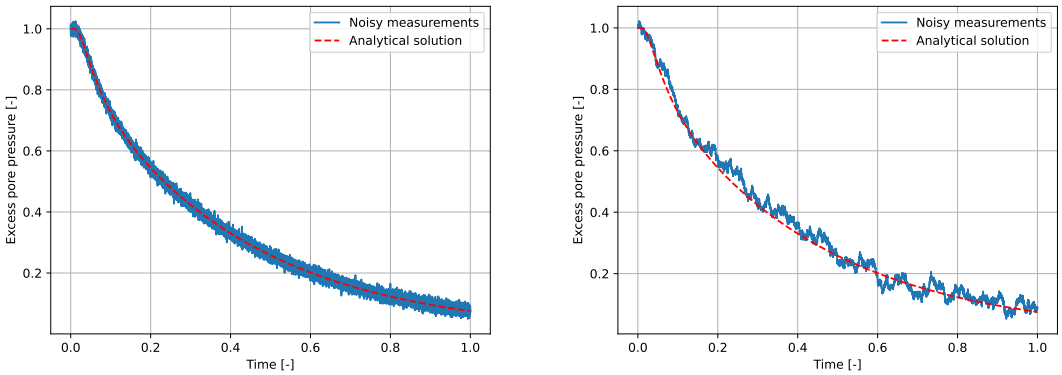
```
procedure METROPOLIS-HASTINGS(f, x, n)
    x = random.choice(x)
    f = f(x)
    x_list = []
    f_list = []
    for n number of iterations do
        x_new = random.normal(x, σ)
        f_new = f(x_new)
        α = f_new / f
        u = random.randint(0, 1)
        if α > u then
            x = x_new
            x_list.append(x)
            f = f_new
            f_list.append(f)
    return x_list, f_list
```

2.5 Data and measurements

It is always necessary to talk about data when dealing with ML models because its primary purpose is to extract, process, and utilize information contained in datasets. For PINNs, data is not obligatory for the model to learn, but the primary advantage lies in its ability to optimize in regard to physical equations and data simultaneously. An important feature of this is the fact that PINNs are mesh-free, so a model can easily be adapted to the domain of the data (Yang et al., 2021). For instance, data spread over time is more cumbersome to implement in a FEM algorithm due to limitations of time discretization. In regards to PINNs, the three relevant data types are measurement data, empirical data, and simulation data. The focus of this work is on the former, but the latter is also important for PINNs in general.

2.5.1 Measurements

Measurements from physical phenomena often contain unwanted noise that infers with the actual behavior. Noisy data can cause problems with outliers and skewed behavior resulting in increased aleatoric uncertainty. Noise can roughly be divided into two main categories, correlated and uncorrelated noise, both illustrated in Figure 7.



(a) Gaussian white noise

(b) Random walk noise

Figure 7: A visualization of noisy measurements.

Figure 7 shows an example of uncorrelated Gaussian noise and correlated Random walk noise. In many cases, both spatially and temporally, modeling measurement error as Gaussian white noise is realistic due to random fluctuations independent of time and space. For other problems, where for instance seasonal changes or other cyclic behaviors are relevant, the assumption of uncorrelated noise may be less accurate. Modeling with correlated noise demands knowledge of the expected dependencies of the noise, which is often unrealistic.

2.5.2 Hydraulic measurements

Subsurface hydraulic measurements are often conducted for geotechnical engineering purposes to determine both the mechanical, biological, and chemical properties of the site's groundwater conditions. The three most relevant characteristics for soil mechanics are groundwater level, groundwater flow, and groundwater pressure. To determine both groundwater level and groundwater pressure, the most common instruments are piezometers and Cone Penetration Test with water pressure measurement (CPTU). Piezometers are installed to a certain depth on-site to determine the hydraulic pressure. It is therefore common to install multiple piezometers at the same site to determine if there are differences in the water pressure between depths, other than simply decreased elevation head. CPTU tests are also conducted on-site, but the water pressure at the tip of the CPTU cone is measured for the entire borehole. CPTU measurements only tells the pressure at one point in time, whereas a piezometer can be installed and continue to measure for years. For surveying consolidation processes, piezometers are therefore the most relevant.

According to Rau et al. (2019), closed piezometer measurements denoted closed GMI in the article, are the method most commonly associated with geotechnical ground investigation. Closed piezometers are prone to multiple measurement errors due to factors like:

- Human error
- Equipment error
- Varying barometric conditions
- Varying fluid density

- Temperature fluctuations
- Time delay
- Sample frequency

Rau et al. (2019) highlights that estimating hydraulic head gradients or fluid velocity requires multiple measurements from varying spatial coordinates. Hydraulic pressure on the other hand requires only a single piezometer and is therefore cheaper to measure accurately compared with fluid flow.

2.5.3 Displacement measurements

Measuring the displacement of soil has changed drastically with the introduction of new technology. Today, automatic measurement systems like GPS, ground sensors, imagery, and radar continuously monitor displacements of soil in urban construction work, hazardous areas, or costly infrastructure projects. The choice of monitoring system often depends on:

- Cost
- Desired accuracy
- Number of dimensions
- Size of the area of interest
- Integration with other systems

Remote monitoring systems like InSAR or drone imagery are able to cover large areas, measure 3D displacements of the surface, but are only accurate on centimeter level . Ground sensors and GPS may be more accurate but are less effective for larger areas. For geotechnical engineering purposes, the expected displacement ranges from millimeters to meters, depending on the site's soil conditions and type of construction. It is therefore important to choose a monitoring system with a high enough resolution to capture the displacement rate.

3 Method

The methodology chapter is divided into the main subsections 1D Terzhagi's consolidation problem and 2D consolidation of a soil layer.

The first subsection introduces the problem and how the traditional solutions were obtained. Subsequently, the PINN implementation is described, both for the deterministic forward problem, uncertainty quantification, and parameter estimation.

The second subsection introduces the 2D problem and how the traditional solutions were obtained by both commercial software and numerical algorithms. Lastly, the PINN implementation is described, with emphasis on the differences between the coupled and uncoupled versions of the problem.

3.1 1D Terzhagi's consolidation theory

Terzhagis's theory for one-dimensional consolidation is one of the first theoretical frameworks for modeling consolidation (CRYER, 1963). The original theory is valid for one-dimensional displacement and flow of noncompressible fluid and solid. The resulting equation is a one-dimensional diffusion equation describing the dissipation of excess pore pressure after an initial undrained loading:

$$c_v \frac{\partial^2 u}{\partial z^2} = \frac{\partial u}{\partial t}. \tag{64}$$

The coefficient of consolidation cv determines the rate of pore pressure dissipation over time, and is proportional to the compressibility and permeability of the soil:

$$c_v = \frac{E_{oed} k_z}{\gamma_w}$$

The following parameters were used to normalize the PDE

$$z^* = \frac{z}{H}, \quad t^* = \frac{t}{T}, \quad u^* = \frac{u}{u_0}.$$

Combining the normalized parameters with equation (4.1) gives

$$C_v \frac{\partial^2 (u^* u_0)}{\partial (z^* H)^2} = \frac{\partial (u^* u_0)}{\partial (t^* T)}.$$

Moving all the constants to one side, and introducing the normalized consolidation coefficient

$$C_v{}^* = C_v{}^* \frac{T}{H^2}$$

gives the normalized 1D diffusion equation

$$c_v^* \frac{\partial^2 u^*}{\partial z^{*2}} = \frac{\partial u^*}{\partial t^*}. \tag{65}$$

The classical problem is usually solved for either closed or open water flow at the bottom end of the soil column. Fully drained ends give two Dirichlet boundaries $u(t, z = 1) = 0$ and $u(t, z = 0) = 0$, while closed water boundary at the bottom implies a Neumann boundary condition $\frac{\partial u}{\partial z}|_{z=0} = 0$. The initial condition is $u(t = 0, z) = q$ where q is the external load. The closed water boundary at the bottom gives two different types of boundary conditions, so this was chosen as the benchmark problem for testing the PINN method.

3.1.1 Analytical and numerical approximation

The analytical solution for the 1D consolidation problem was approximated with the following Fourier series as proposed by Churchill (1971)

$$u^*(t,z) \approx \frac{4}{\pi}\sum_{k=1}^{1000}\frac{(-1)^{k-1}}{2k-1}\cos[(2k-1)\frac{\pi z^*}{2}]\exp\left(-(2k-1)^2\frac{\pi^2 {C_v}^* t^*}{4}\right) \tag{66}$$

The associated vertical displacement was approximated by first calculating the final displacement by integrating the vertical strain. According to elasticity theory, combining equation (17) and (18), the vertical settlement is

$$\lim_{t\to\infty} w^*(t) = \int_z \Delta\varepsilon_{zz} dz = \frac{1}{E^*_{oed}}\int_z \Delta\sigma^*_{zz} = 1 \tag{67}$$

And then approximating the degree of consolidation with the following Fourier series

$$U^*_p \approx 1 - \frac{8}{\pi^2}\sum_{k=1}^{1000}\frac{1}{(2k-1)^2}\exp\left(-(2k-1)^2\frac{\pi^2 {C_v}^* t^*}{4}\right) \tag{68}$$

The finite difference method (FDM) approximation was calculated using the following Forward Time Central Space scheme. All variables are normalized.

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} = c_v\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta z^2}\begin{cases} t^n = n\Delta t, n = \{0,1,...,30000\}, \Delta t = \frac{1}{30000} \\ z_i = i\Delta z, i = \{0,1,...,100\}, \Delta z = \frac{1}{100}\end{cases} \tag{69}$$

The Neumann boundary condition was treated by introducing ghost nodes to approximate the closed flow boundary. The boundary condition in itself can be approximated by a first-order finite difference scheme

$$\frac{\partial u}{\partial z}\Big|_{z=0} \approx \frac{u_1^n - u_{-1}^n}{\Delta z} = 0$$

implying $u_1^n = u_{-1}^n$, where u_{-1}^n are ghost nodes.

The FEM approximation was calculated with linear elements and forward time integration. All variables are normalized

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \boldsymbol{C}^{-1}(\boldsymbol{f}^n - \boldsymbol{K}\boldsymbol{u}^n)\Delta t \left\{ t^n = 0 + n\Delta t, n = \{0,1,2,...,30000\}, \Delta t = \frac{1}{30000}\right. \tag{70}$$

where K is the stiffness matrix defined as

$$\boldsymbol{K_{el}} = \int_0^1 \boldsymbol{B^T} Ak \boldsymbol{B} d\xi \tag{71}$$

and C is the capasity matrix

$$\boldsymbol{C_{el}} = \int_0^1 \boldsymbol{N^T}\rho Ac\boldsymbol{N} d\xi \tag{72}$$

$$\boldsymbol{u}^n = \begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{100}^n \end{bmatrix}, \boldsymbol{f}^n = \begin{bmatrix} f_1^n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\boldsymbol{C} = 10^{-2} \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & & & \\ \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} \\ & & & \frac{1}{3} & \frac{2}{3} \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} 50 & -50 & & & \\ -50 & 100 & -50 & & \\ & \ddots & \ddots & \ddots & \\ & & -50 & 100 & -50 \\ & & & -50 & 50 \end{bmatrix} \tag{73}$$

3.1.2 PINN implementation

The training domain was established as a rectangular grid of t and z values, using the following discretization, which is visualized in Figure 8. All variables are normalized.

$$u(t^n, z_i) \in \mathbb{R} \begin{cases} t^n = 0 + n\Delta t, n = \{0, 1, 2, ..., 700\}, \Delta t = \frac{1}{700} \\ z_i = 0 + i\Delta z, i = \{0, 1, 2, ..., 100\}, \Delta z = \frac{1}{100} \end{cases} \tag{74}$$

The loss function was established based on the governing equation, initial condition, and boundary conditions. A user-defined scaling of the terms in the loss function was implemented, which is further discussed in the general discussion chapter on page 93.

$$\mathcal{L}(t^n, z_i) = \mathcal{L}(t^n, z_i)_{PDE} + 10 * \mathcal{L}(t^0, z_i)_{IC} + \mathcal{L}(t^n, z_0)_{BC1} + \mathcal{L}(t^n, z_{100})_{BC2} \tag{75}$$

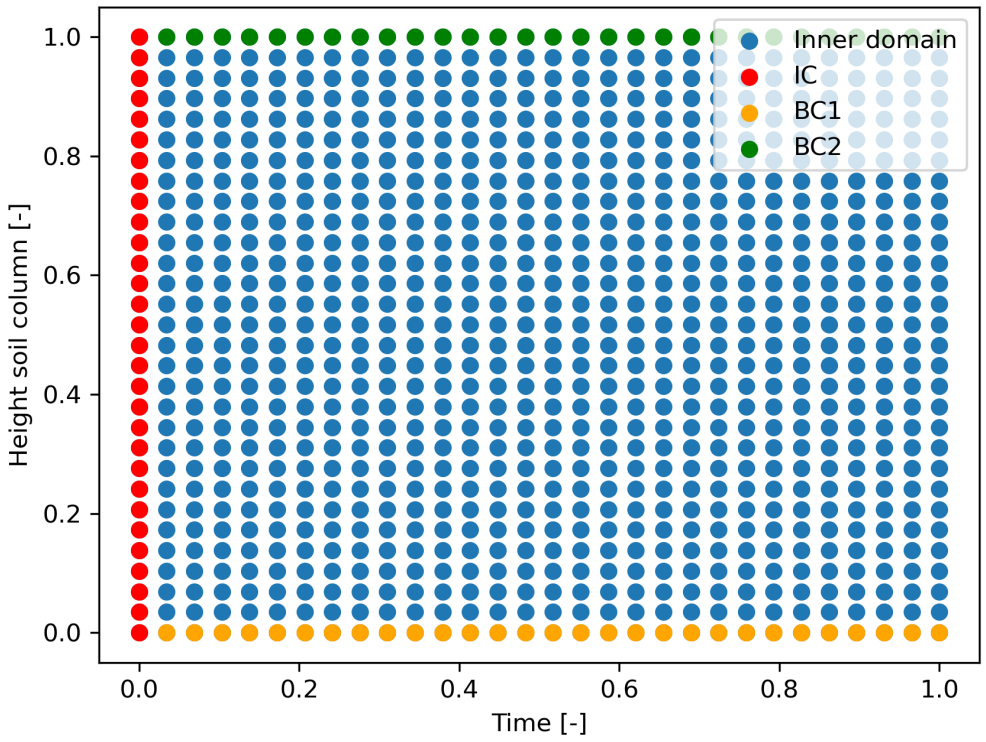


Figure 8: A visualization of the training domain and physical constraints.

The following hyperparameters were used for the forward problem:

Table 2: Hyperparameters forward problem.

Hyperparameter	Value
Number of hidden layers	5
Number of neurons per layer	75
Activation function	Swish
Number of epochs	5000
Batch size	100
Loss function	MSE
Optimizer	Adam
Initial learning rate	0.001
Weight initializer	Xavier
Bias initializer	Zeros

The forward problem of one-dimensional consolidation can be described by the graphical model presented in Figure 9.

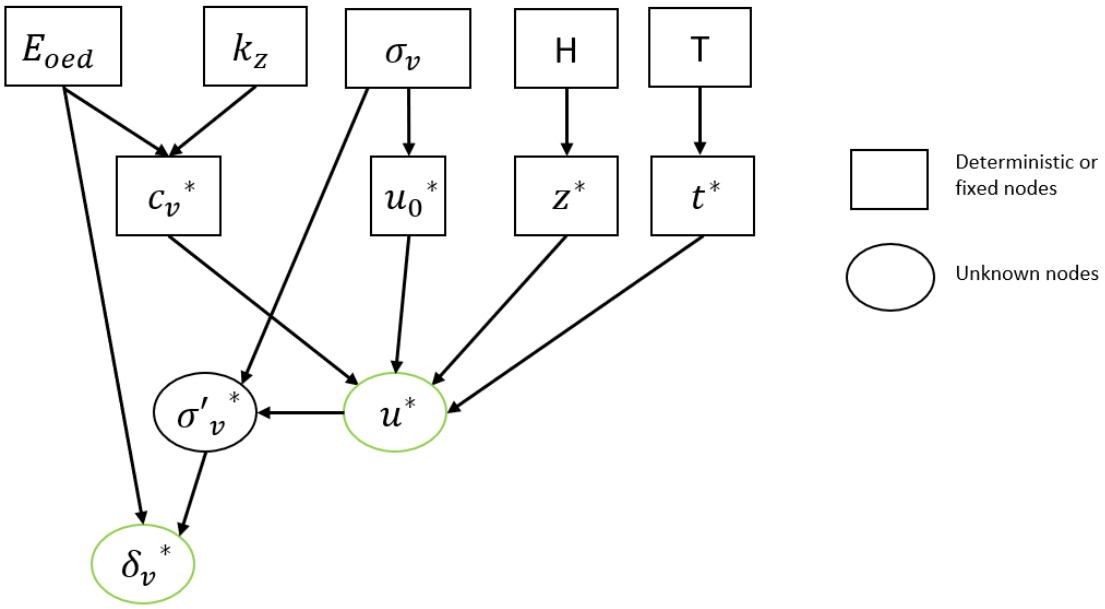


Figure 9: A flowchart of the forward problem. The green color indicates desired outputs.

The deterministic nodes are used as inputs to the model, both for training and prediction. The model prediction of excess pore pressure was used in post-processing to calculate the vertical displacement combining Hooke's law from Equation (1), the definition of strains

from Equation (2), and effective stresses from Equation (12).

$$\delta_v(t) = \frac{1}{E_{oed}} \int_0^1 \Delta\sigma_v^{'}(t,z)dz \approx \frac{\Delta z}{E_{oed}} \sum_0^{100} \Delta\sigma_v^{'}(t,z_i) \tag{76}$$

To test the accuracy of the mesh-free ability of the PINN model, different training grids were created. Figure 10 shows the testing grids compared with the training grid. The grids were created to test the efficiency of the interpolation and to check if the model was more sensitive to interpolation in the time or space domain.

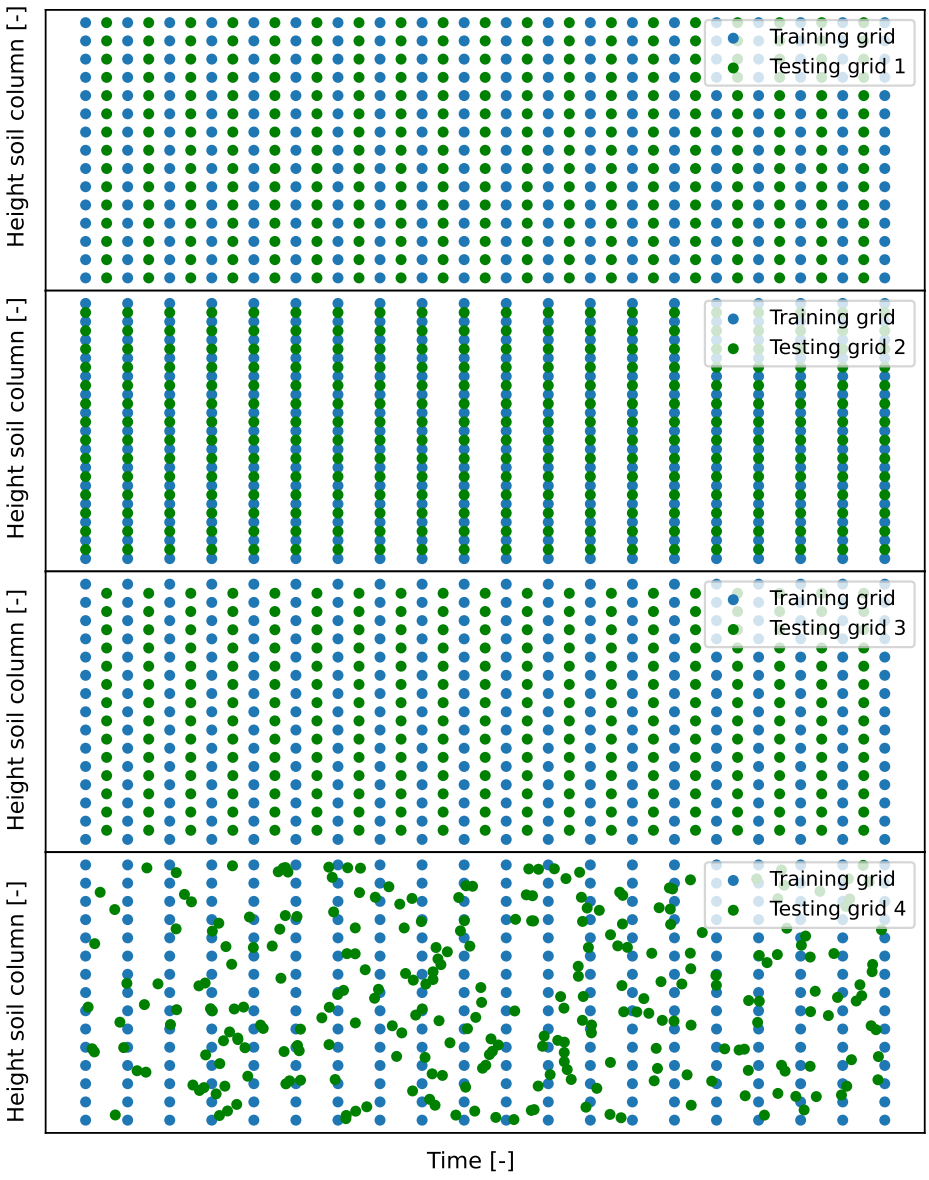


Figure 10: A visualization of the different grids used to test the model's ability to interpolate.

The accuracy of the model's ability to extrapolate was simply tested by feeding the model with increasing C_v values outside of the training domain. Since the C_v value effectively

controls the entire problem domain due to the nondimensionalization, this is equivalent to increasing the time period or increasing the height of the soil column.

For the forward probabilistic problem, the hydraulic conductivity k_z was treated as a random variable thus making C_v a random variable. The model was therefore trained on three inputs, $u(t, z, C_v)$, where $C_v \sim \mathcal{N}(\mu = 1, \sigma = 0.3)$. To evaluate the model, Monte Carlo simulations were used to evaluate the model prediction where C_v values were sampled randomly from its distribution. For each time step, Δt, the vertical settlement of the column top was evaluated for each sampled C_v value. This allowed for calculating the average of the vertical deformation each time step, as well as the standard deviation. In addition, histograms were produced for settlement at three different points in time to check the distribution of $P(\delta_v|C_v)$. The same procedure was afterward conducted for the analytical solution.

By utilizing measurements of the pore pressure, the consolidation coefficient can be determined by inverting the calculation flow. The partly observed excess pore pressure gives a unique solution for the consolidation coefficient. The calculated consolidation coefficient can then be used to calculate the forward problem which gives the unobserved pore pressure values and the unknown vertical displacement. The flowchart of the problem is visualized in Figure 11. The vertical displacement is dependent on both effective overburden pressure and oedometer stiffness. So this information flow is dependent on a known oedometer modulus but allows for unknown permeability.

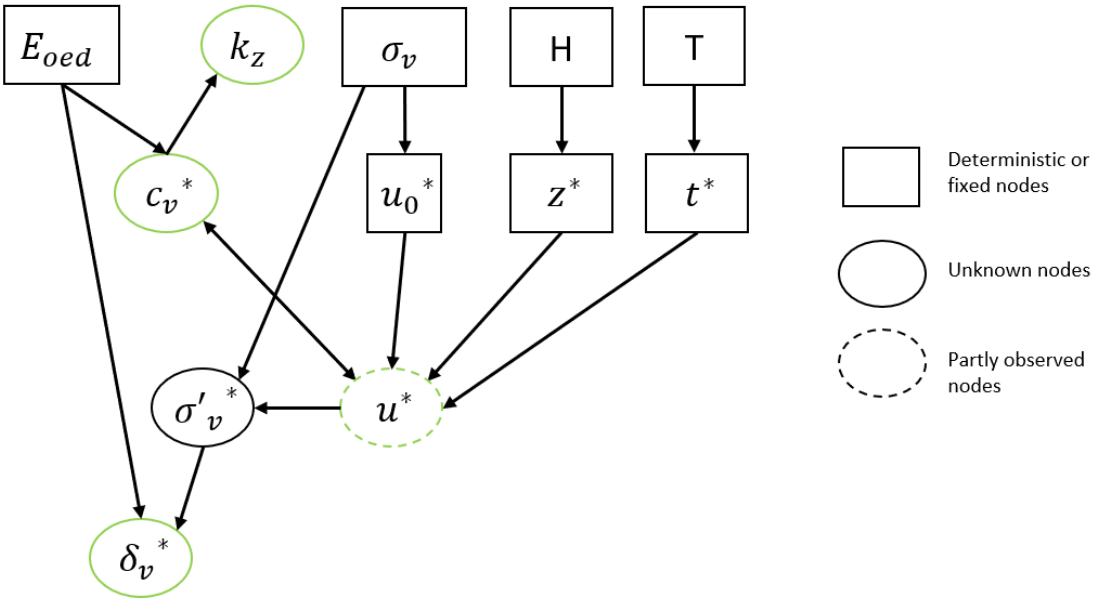


Figure 11: A flowchart of the inverse problem with one random variable. The green color indicates desired outputs.

The likelihood function $P(\boldsymbol{D}|\boldsymbol{C_v^*})$ was estimated using the Metropolis-Hastings Algorithm (2), with $P(\boldsymbol{D}|C_v^*)$ as

$$P(\boldsymbol{D}|C_v^*) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{pred(t_i, C_v^*) - D_i}{\sigma}\right)^2\right\} \tag{77}$$

The alternative option to MLE in postprocessing is to include the C_v parameter in the

optimization algorithm. This option provides less control over tested C_v values since the model may converge to one specific range of parameter values and stay there for the entire training period. this option is more prone to local minima issues and provides a narrower overview of the likelihood of possible values. On the other hand, this option is as accurate and reduces the number of inputs to the model. For comparison with the MLE in the postprocessing, the likelihood during training can be estimated with Algorithm 3. The likelihood during training is of course dependent on the training stage. The likelihood is mostly interesting for a well-trained model, so the values from the early stages of training were disregarded. Since the C_v value from epoch to epoch varies little due to relatively small learning rates, storing likelihood values for consecutive epochs can cause overlapping values.

Algorithm 3 Likelihood during training

```
1: procedure LIKELIHOOD(self,training-grid,measurements, σ)
2:     ŷ = self.predict(training-grid)
3:     if epoch > 300 and batch=last and epoch%10 == 0 then
4:         sum = 0
5:         for sample in ŷ do
6:             Product *= Normal.pdf(x =sample, μ = measurements, σ)
7:         self.list.append(Product)
```

To simulate realistic data from common geotechnical measurements relevant for soil parameters used in poroelastic computations, artificial data was created from the analytical solution. The relevant measurement types are piezometer measurements for monitoring changes in pore pressure over time, and displacement measurements to monitor the vertical deformation of the surface over time. Both piezometer measurements and displacement measurements have restricted spatial coverage, so the artificial data was only generated for surface displacement and one piezometer depth, as illustrated in Figure 12. The datasets with measurements were created with and without white Gaussian noise.

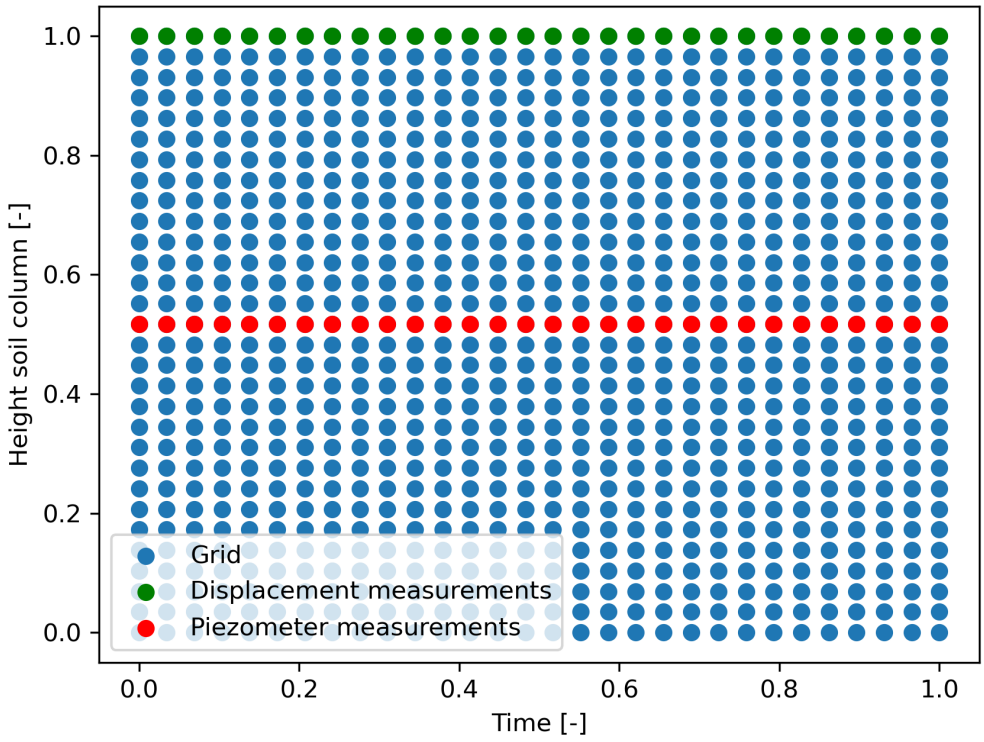


Figure 12: An illustration of the grid provided in the inverse problem.

By introducing measurements of the vertical displacement of the surface, the oedometer modulus can be uniquely defined. The oedometer modulus is conditionally independent of the permeability given the vertical displacement and consolidation coefficient. So both the components of the consolidation coefficient can be determined by these two measurements. The flowchart for this problem is presented in the graphical model in Figure 13.

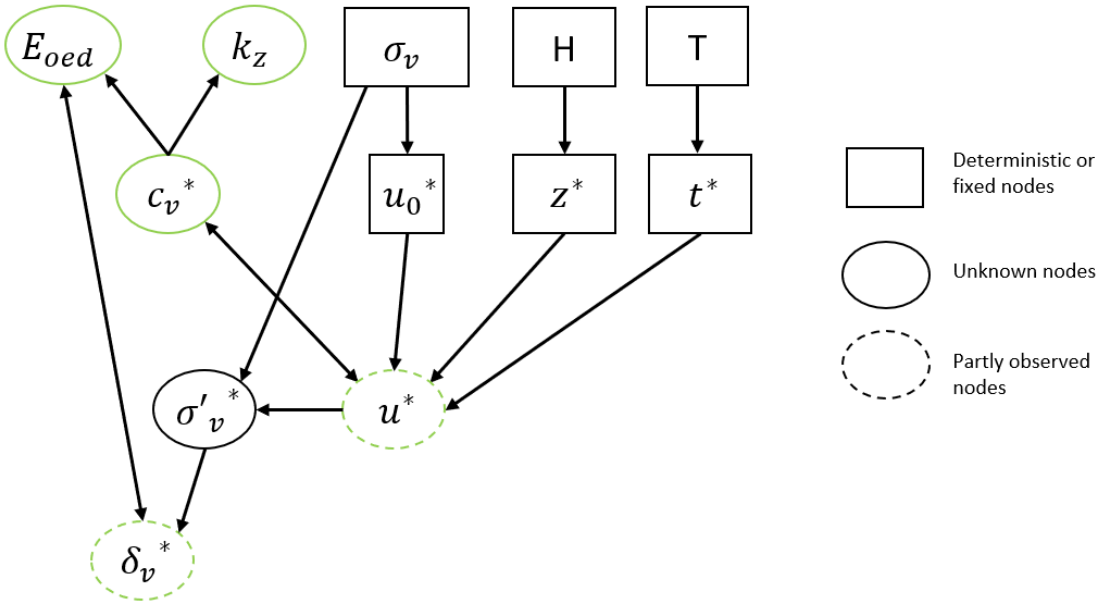


Figure 13: A flowchart of the inverse problem with two random variables. The green color indicates desired outputs.

3.2 Strip load on soil layer

The second problem is the two-dimensional consolidation of a soil layer, illustrated in Figure 14. This problem is generally a coupled flow-deformation problem, but several decoupling assumptions can be made.

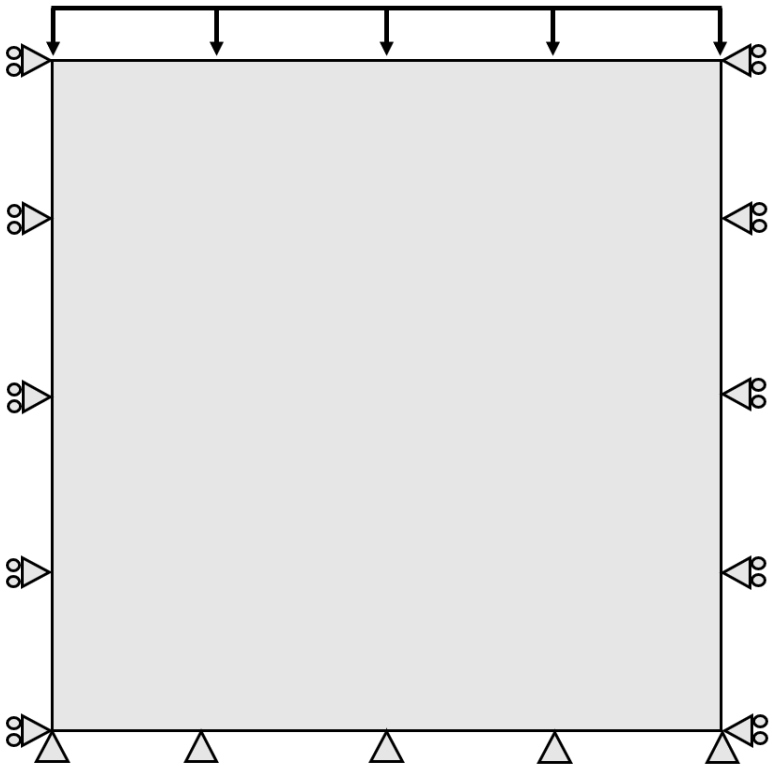


Figure 14: A simple illustration of the 2D consolidation problem.

The boundary conditions for the problem can be seen in Figure 14, where rollers indicate normally constrained movement and no rollers indicate constrained movement in all directions. The groundwater flow is closed at the bottom, and fully drained at the ends. The soil is loaded with a constant line load q.

The governing PDEs are the equilibrium equations for plane strain (32), in cartesian coordinates

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xz}}{\partial z} + f_x = 0 \\ \frac{\partial \sigma_{zz}}{\partial z} + \frac{\partial \sigma_{xz}}{\partial x} + f_z = 0 \end{aligned} \tag{78}$$

And the storage equation (37)

$$\frac{\partial \varepsilon_v}{\partial t} = \frac{k}{\gamma_w} \nabla^2 u \tag{79}$$

The problem is further divided into two parts, uncoupled and coupled flow.

By assuming negligible horizontal strains and constant isotropic total stress, the storage equation reduces to the 2D transient diffusion equation as seen in Equation (39). To evaluate the accuracy and efficiency of the model, the same problem was solved in Python using FDM. For comparison of the validity of the assumptions, the problem was also solved with Plaxis 2D without the assumption of negligible horizontal strain and constant total stress. The full procedure for all three methods was as follows:

1. Initial stress state from self-weight and hydrostatic pore pressure
2. Introduce external load as undrained loading
3. Consolidation analysis to a fixed time

The full problem of coupled flow was solved with PINNs and Plaxis 2D, but this time with the default Plaxis parameters for undrained behavior, see Table 3.

Table 3: Problem definition.

Properties	Uncoupled	Coupled
Soil model	Linear elastic	Linear elastic
E_{oed}	1000	1498
ν	0.33	0.33
ν_u	-	0.495
α	1	1
$\frac{K_w}{n}$	∞	$36.39E3$
B	1	0.9733
γ_w	0.5	0.5
γ_{sat}	1.0	1.0
K_0	0.5	0.5
$k_z[m/year]/k_z[m/day]$	$5 * 10^{-5}/1.37 * 10^{-7}$	$5 * 10^{-4}/1.37 * 10^{-6}$
$k_x[m/year]/k_x[m/day]$	$5 * 10^{-5}/1.37 * 10^{-7}$	$5 * 10^{-4}/1.37 * 10^{-6}$
q	1	1

The finite difference approximation for the uncoupled problem was calculated using the following Forward Time Central Space scheme, utilizing the symmetry of the problem to calculate only half of the grid.

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = K \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta z^2}$$

$$Grid \begin{cases} t^n = n\Delta t, n = \{0, 1, 2, ..., 20000\}, \Delta t = \frac{1}{20000} \\ x_i = i\Delta x, i = \{0, 1, 2, ..., 50\}, \Delta x = \frac{1}{50} \\ z_j = j\Delta z, j = \{0, 1, 2, ..., 50\}, \Delta z = \frac{1}{50} \end{cases} \tag{80}$$

The Neumann boundary conditions were approximated with ghost nodes, similar to the procedure for the 1D problem

$$\begin{aligned} \frac{\partial u}{\partial x}\Big|_{x=0} &\approx \frac{u_{1,j}^{n} - u_{-1,j}^{n}}{\Delta x} = 0 \\ \frac{\partial u}{\partial z}\Big|_{z=-1} &\approx \frac{u_{i,1}^{n} - u_{i,-1}^{n}}{\Delta z} = 0 \end{aligned} \tag{81}$$

3.2.1 PINN implementation uncoupled

The PINN implementation for the uncoupled problem can be split into three parts, one for each phase.

For the initial in-situ stress state, a two-dimensional grid of length $\frac{L}{2} = 1$ and depth $D = -1$ was created. The governing equations are the equilibrium equations (78), where $f_z = \gamma_{sat}$ and $f_x = 0$. The hydrostatic pore pressure was defined as $p_w = \gamma_w z$ The boundary and model conditions are:

1. Left: $\frac{\partial \sigma_{xx}}{\partial x} = 0$, $\frac{\partial \sigma_{zz}}{\partial x} = 0$ and $\sigma_{xz} = 0$
2. Right: $\frac{\partial \sigma_{xx}}{\partial x} = 0$, $\frac{\partial \sigma_{zz}}{\partial x} = 0$ and $\sigma_{xz} = 0$
3. Bottom: $\sigma_{xz} = 0$
4. Top: $\sigma_{zz} = 0$, $\sigma_{xx} = 0$ and $\sigma_{xz} = 0$
5. Constitutive relations: $\Delta\sigma'_{xx} = \Delta\sigma'_{zz} K'_0$ and $\Delta\sigma'_{yy} = \nu(\Delta\sigma'_{xx} + \Delta\sigma'_{zz})$

By definition, the effective stresses can be derived using Equation 15, after the total stress and pore pressure distribution are calculated.

The phase introducing external loading follows the same governing equations. The external load was modeled as a surface load along the boundary $z = 0$. With $\alpha = 1$ and $B = 1$, $\Delta\varepsilon_v = \Delta p' = 0$ for undrained loading. This further implies that $\Delta\sigma_{zz} = \Delta\sigma_{xx} = \Delta\sigma_{yy} = \Delta u = 1$, making the top boundaries:

4. Top: $\sigma_{zz} = 1$, $\sigma_{xx} = 1$ and $\sigma_{xz} = 0$

The pore pressure distribution is now $p_w = p_{steady} + u = \gamma_w z + 1$

The consolidation phase introduces a new variable t, making the training grid $20x20x50$. Furthermore, the uncoupled storage equation replaces the equilibrium equations in the loss function along with the boundary condition for the water flow. The new boundary conditions are:

1. Left: $\frac{\partial u}{\partial x} = 0$
2. Right: $u = 0$,
3. Bottom: $\frac{\partial u}{\partial z} = 0$
4. Top: $u = 0$
5. Initial condition: $u = 1$

The resulting effective stresses can be calculated by subtracting $u(t)$ from $\Delta\sigma_{zz}$ and finally calculating strain and displacement from the resulting effective stresses.

3.2.2 PINN implementation coupled

The PINN implementation for the coupled problem can equivalently to the uncoupled problem be split into three parts.

The consolidation phase and all boundary conditions in all phases except the following are unchanged from the uncoupled problem. The changes are

1. Initial: $\varepsilon_v = \varepsilon_{zz} = \frac{\Delta p}{K_u}$

in addition to adding the full storage equation to the loss function. The model was trained with a fixed-stress split in accordance with 1, because of its less constrained stability region.

The non-dimensional parameters can be defined as (Haghighat et al., 2022):

$$\begin{aligned} x^* &= \frac{x}{\alpha_l}, \quad z^* = \frac{z}{\alpha_l}, \quad t^* = \frac{t}{\alpha_t}, \quad \sigma_{xx}^* = \frac{\sigma_{xx}}{\alpha_\sigma}, \quad \sigma_{zz}^* = \frac{\sigma_{zz}}{\alpha_\sigma} \\ v^* &= \frac{v}{\alpha_d}, \quad w^* = \frac{w}{\alpha_d}, \quad u^* = \frac{u}{\alpha_\sigma}, \quad \varepsilon_{xx}^* = \frac{\varepsilon_{xx}}{\alpha_\varepsilon}, \quad \varepsilon_{zz}^* = \frac{\varepsilon_{zz}}{\alpha_\varepsilon} \end{aligned} \tag{82}$$

Were l is length, t is time, σ is stress, and ε is strain. Because of the potentially big relative magnitude of displacement and domain, d for displacement is used instead of l. These nondimensionalized parameters further define

$$p^* = \frac{\sigma_{xx}^* + \sigma_{xx}^* + \sigma_{xx}^*}{3} = \alpha_\sigma p \tag{83}$$

$$\varepsilon_v^* = \varepsilon_{xx}^* + \varepsilon_{zz}^* = \alpha_\varepsilon \varepsilon_v \tag{84}$$

Insert the dimensionless parameters in the stress-split storage equation combining Algorithm 1 and Equation (37)

$$\frac{\alpha_\sigma}{\alpha_t}\left(\frac{\alpha^2}{K} + \frac{n}{K_w}\right)\frac{\partial u^*}{\partial t^*} + \frac{\alpha_\sigma}{\alpha_t}\frac{\alpha}{K}\frac{\partial p^*}{\partial t^*} - \frac{k_x}{\gamma_w}\frac{\alpha_\sigma}{\alpha_l^2}\frac{\partial^2 u^*}{\partial x^{*2}} - \frac{k_z}{\gamma_w}\frac{\alpha_\sigma}{\alpha_l^2}\frac{\partial^2 u^*}{\partial z^{*2}} = 0 \tag{85}$$

The equation can be simplified considerably by first canceling out α_σ. Defining α_t as

$$\alpha_t = \left(\frac{\alpha^2}{K} + \frac{n}{K_w}\right)\frac{\gamma_w}{k}\alpha_l^{*2} \tag{86}$$

is valid for isotropic flow, i.e. $k_z = k_x$, and reduces the equation to

$$\begin{aligned} \frac{k}{\gamma_w \alpha_l^2}\frac{\partial u^*}{\partial t^*} + \frac{k}{\gamma_w \alpha_l^2}\frac{\alpha}{\alpha^2 + \frac{nK}{K_w}}\frac{\partial p^*}{\partial t^*} - \frac{k}{\gamma_w \alpha_l^2}\frac{\partial^2 u^*}{\partial x^{*2}} - \frac{k}{\gamma_w \alpha_l^2}\frac{\partial^2 u^*}{\partial z^{*2}} &= 0 \\ \frac{\partial u^*}{\partial t^*} + \frac{\alpha}{\alpha^2 + \frac{nK}{K_w}}\frac{\partial p^*}{\partial t^*} - \nabla^{*2} u^* &= 0 \end{aligned} \tag{87}$$

The equilibrium equations

$$
\begin{aligned}
K\left(1+\frac{1-2\nu}{2(1+\nu)}\right)\frac{\alpha_\varepsilon}{\alpha_l}\frac{\partial\varepsilon_v^*}{\partial x^*}+K\frac{3(1-2\nu)}{2(1+\nu)}\frac{\alpha_d}{\alpha_l^2}\nabla^{*2}v^*-\alpha\frac{\alpha_\sigma}{\alpha_l}\frac{\partial u^*}{\partial x^*}=0\\
K\left(1+\frac{1-2\nu}{2(1+\nu)}\right)\frac{\alpha_\varepsilon}{\alpha_l}\frac{\partial\varepsilon_v^*}{\partial z^*}+K\frac{3(1-2\nu)}{2(1+\nu)}\frac{\alpha_d}{\alpha_l^2}\nabla^{*2}w^*-\alpha\frac{\alpha_\sigma}{\alpha_l}\frac{\partial u^*}{\partial z^*}=0
\end{aligned}
\tag{88}
$$

Defining the normalized strain and displacement so the bulk stiffness cancels from the equilibrium equations

$$
\alpha_\varepsilon=\frac{\alpha_d}{\alpha_l},\alpha_d=\frac{a_\sigma}{K}
\tag{89}
$$

Inserting the values from the problem definition yields the final equations

$$
\begin{aligned}
1.125\frac{\partial\varepsilon_v^*}{\partial x^*}+0.376\nabla^{*2}v^*-\frac{\partial u^*}{\partial x^*}=0\\
1.125\frac{\partial\varepsilon_v^*}{\partial z^*}+0.376\nabla^{*2}w^*-\frac{\partial u^*}{\partial z^*}=0
\end{aligned}
\tag{90}
$$

4 Results and discussion

The results and discussion chapter provides a comprehensive presentation and analysis of the findings obtained from the research. The discussion will be localized to the individual results. A more general discussion of the findings in terms of the research questions can be found in the General Discussion Chapter (5).

The results are presented in the following order:

4.1 One-dimensional Terzhagi consolidation

The results from the model predictions are categorized into three subsections, forward deterministic, forward probabilistic, inverse probabilistic, and inverse probabilistic with noisy measurements.

4.1.1 Forward problem deterministic

The model prediction for the forward problem is visualized in a contour plot in Figure 15.

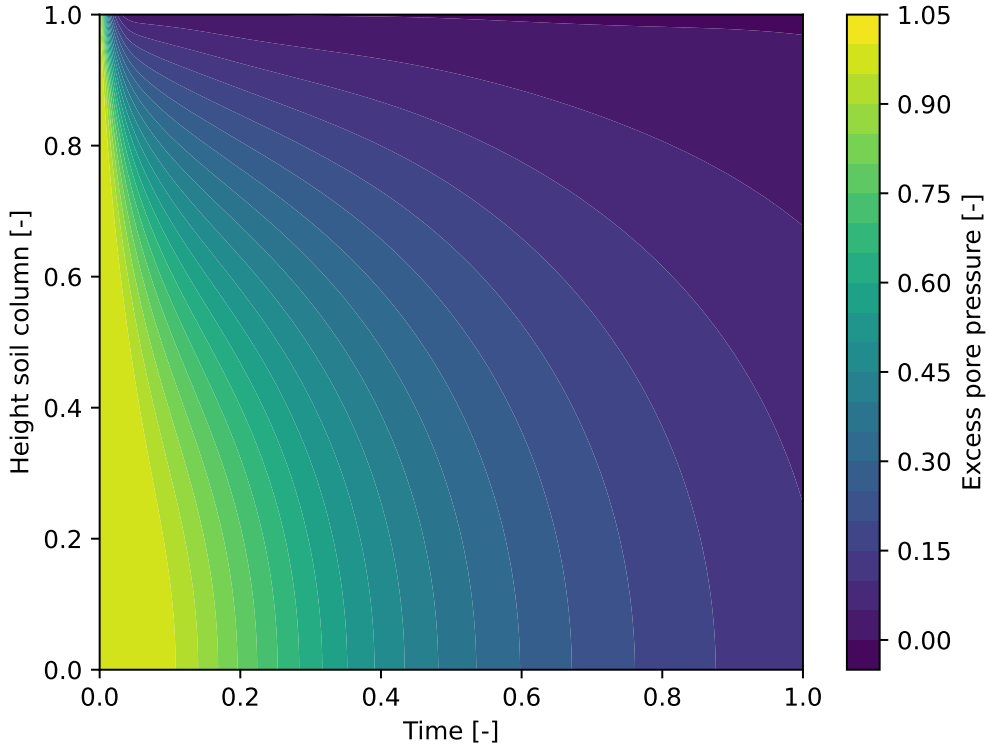
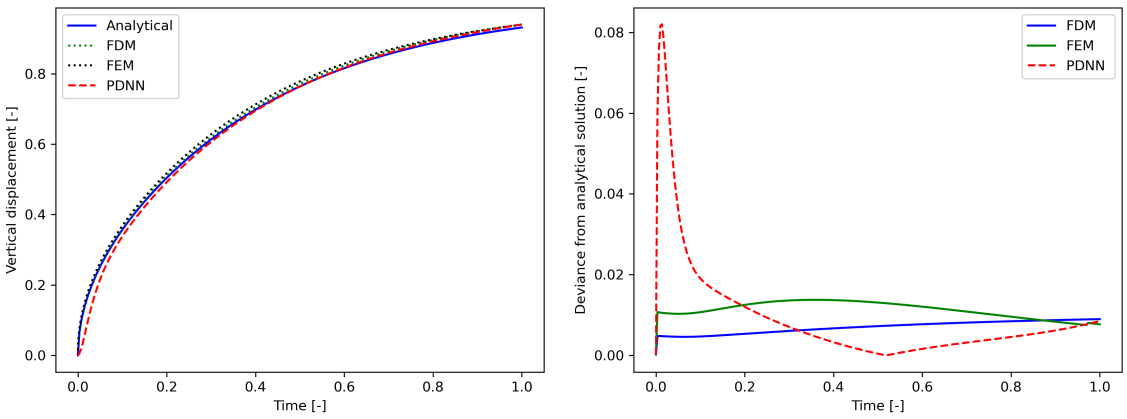


Figure 15: A visualization of the model prediction of excess pore pressure over time. Fluid pressure is considered positive.

The prediction in Figure 15 shows that the model is able to capture the physics of the problem, even without data. As shown by the values on the color scale, the output ranges between $u \approx 1$ and $u \approx 0$, with some slight overshooting and undershooting.

In Figure 16, the resulting change in vertical effective stress due to pore pressure dissipation is calculated and integrated, giving the resulting vertical displacement of the surface as in Equation (76).

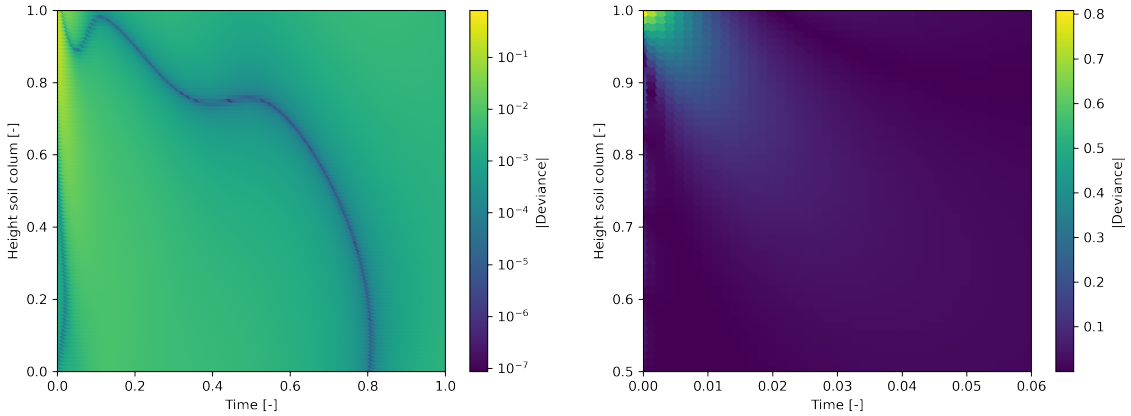


(a) Vertical deformation over time

(b) Deviance from the analytical solution

Figure 16: A comparison of the numerical method's performance in calculating the vertical displacement of the top of the soil column. Displacement in the direction of gravity is considered positive.

Compared with the traditional numerical methods, the PINN performs quite similarly with some exceptions. The deviance from the analytical solution is especially large for the first timesteps, which is also where the analytical solution is the most inaccurate. Unlike the numerical solutions, the analytical solution for excess pore pressure at $t = 0$ is not exactly one. This is the reason for the initial deviance between the numerical methods and the analytical solution. The PINN prediction undershoots the initial conditions by a small margin. This could be due to the point $u(0, 1)$ highlighted in Figure 17b.



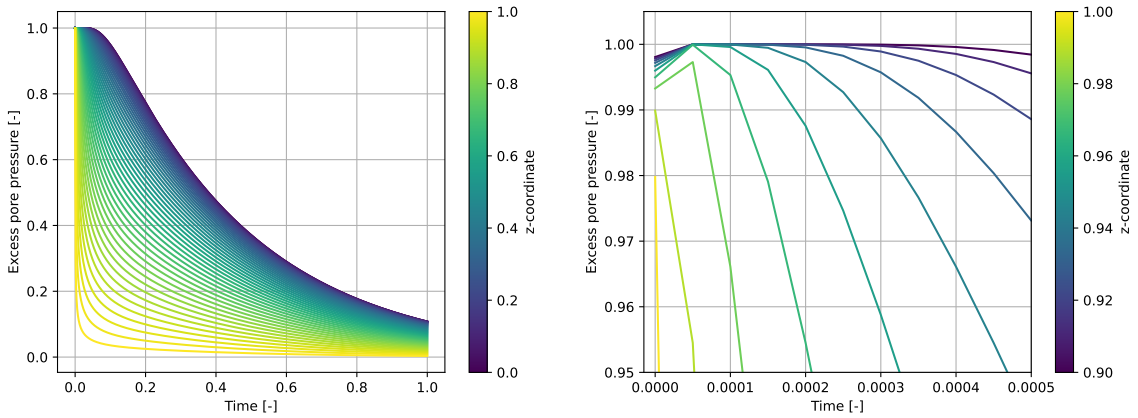
(a) Entire domain

(b) Clip from upper left corner

Figure 17: Deviance plots comparing the PINN prediction with the analytical solution. The color levels in the plot for the entire domain are in log-scale.

This point is the intersection between two opposing boundary conditions $u(0, z) = 1$ and $u(t > 0, 1)$. This can be enforced in traditional numerical methods, but the algorithm optimizing the gradients of Equation (59) probably learns $z = 1 \implies u = 0$ and $t = 0 \implies u = 1$, which results in an estimate somewhere in between. Since the time discretization was finer than space, the model's bias leans towards the boundary condition and not the initial condition because it was trained on more points on the boundary than the initial conditions. Altering the discretization to a more uniform grid, and possibly feeding the network with more points at $(0, 1)$ during training could lower the error in that area. This could on the other hand affect the network with biases that contradicts the governing equation. The continuous strip of $error \approx 0$ in Figure 17a is quite interesting, and shows the borders of where the model overshoots and undershoots the analytical solution.

Figure 18 shows an alternative plot of the analytical solution to the forward problem, which puts more emphasis on the numerical values of the excess pore pressure u.

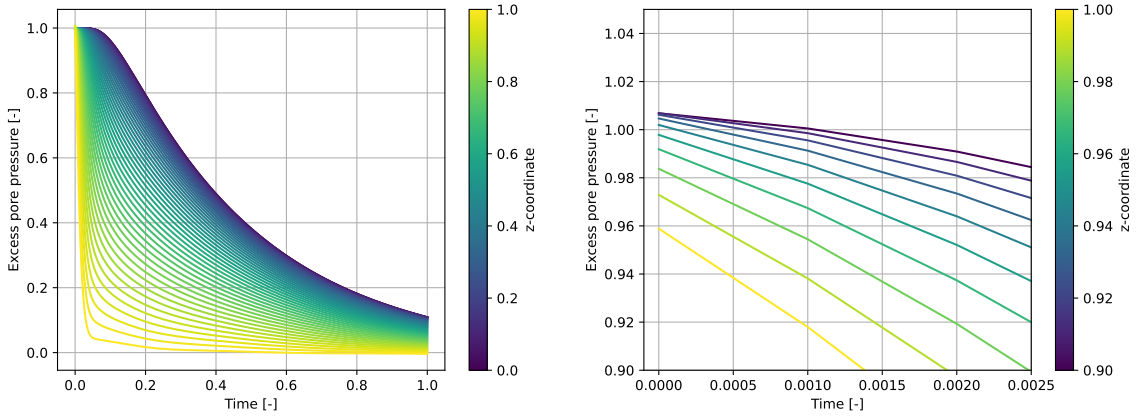


(a) Full range of time and height

(b) Focused plot of the non-linearity

Figure 18: Analytical solution of excess pore pressure over time for different z-values. Fluid pressure is considered positive.

By comparing the upper left corner, 18b and 19b, the problem area of the PINN prediction is clear. The nonlinearity is not learned, which clearly affects the dissipation rate for early time steps. The problems seem to be mostly localized, and the error does not seem to propagate substantially. Figure 18b also shows the reason for the initial deviance between the numerical methods and the analytical solution. The initial excess pore pressure should be $u(t = 0, z) = 1$, but $u \neq 1$ at the upper boundary due to the finite Fourier series approximation. The numerical methods treat this as a hard constraint and it will therefore be enforced regardless of mesh size, element type, grid size, or time step if the time integration is of forward type.



(a) Full range of time and height

(b) Focused plot of the non-linearity

Figure 19: Model prediction of excess pore pressure over time for different z-values. Fluid pressure is considered positive.

The model's capability to interpolate from the training grid to the different testing grids is showcased in Figure 20. The accuracy is not affected by the grid at all, which is further emphasized in Figure 21.

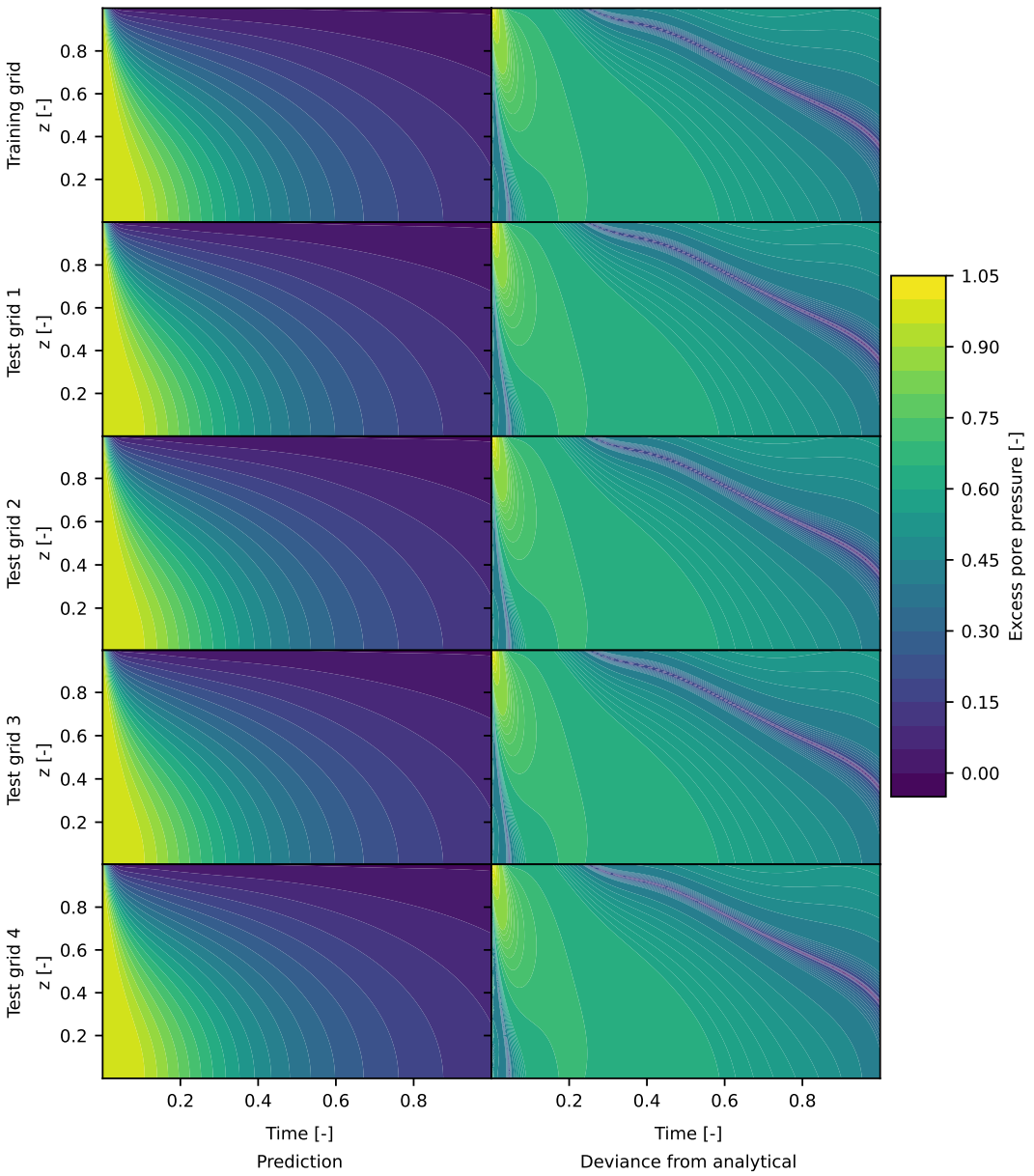


Figure 20: Contour plots of the model prediction for the different test grids. The contour levels in the deviance plots are plotted with a logarithmic scale.

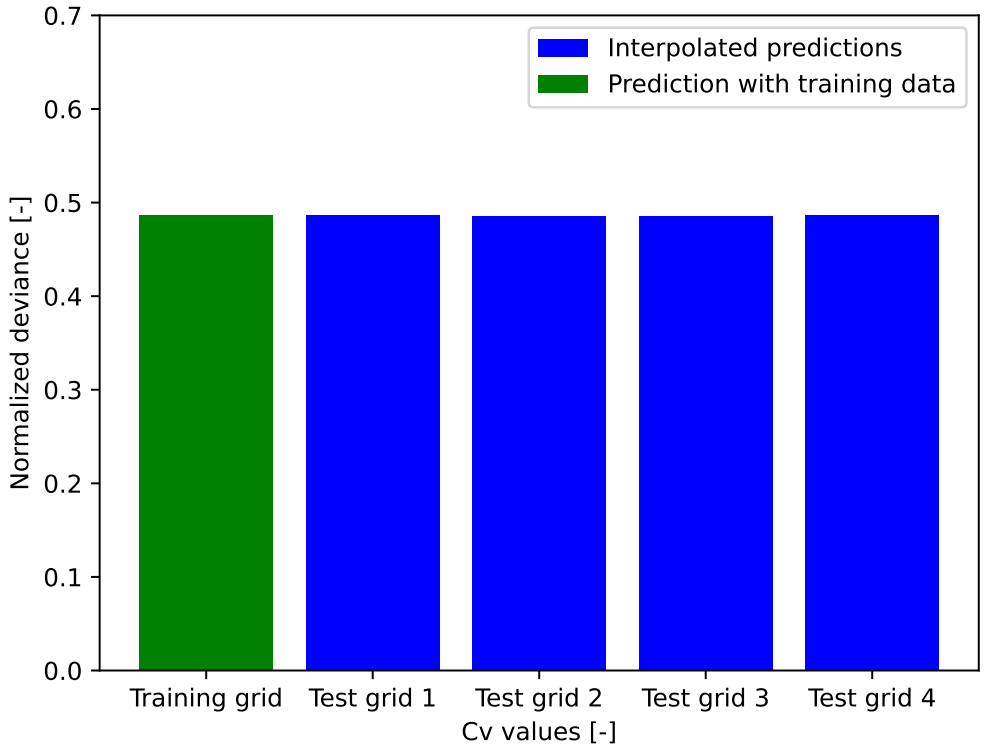


Figure 21: Comparison of the model's performance for the different testing grids. The deviances are normalized based on the highest measured deviance of the interpolation tests, appearing in Figure 23.

When it comes to the interpolation of the C_v values, the model shows more variation in the results. The results are visualized in Figure 22 and Figure 23.

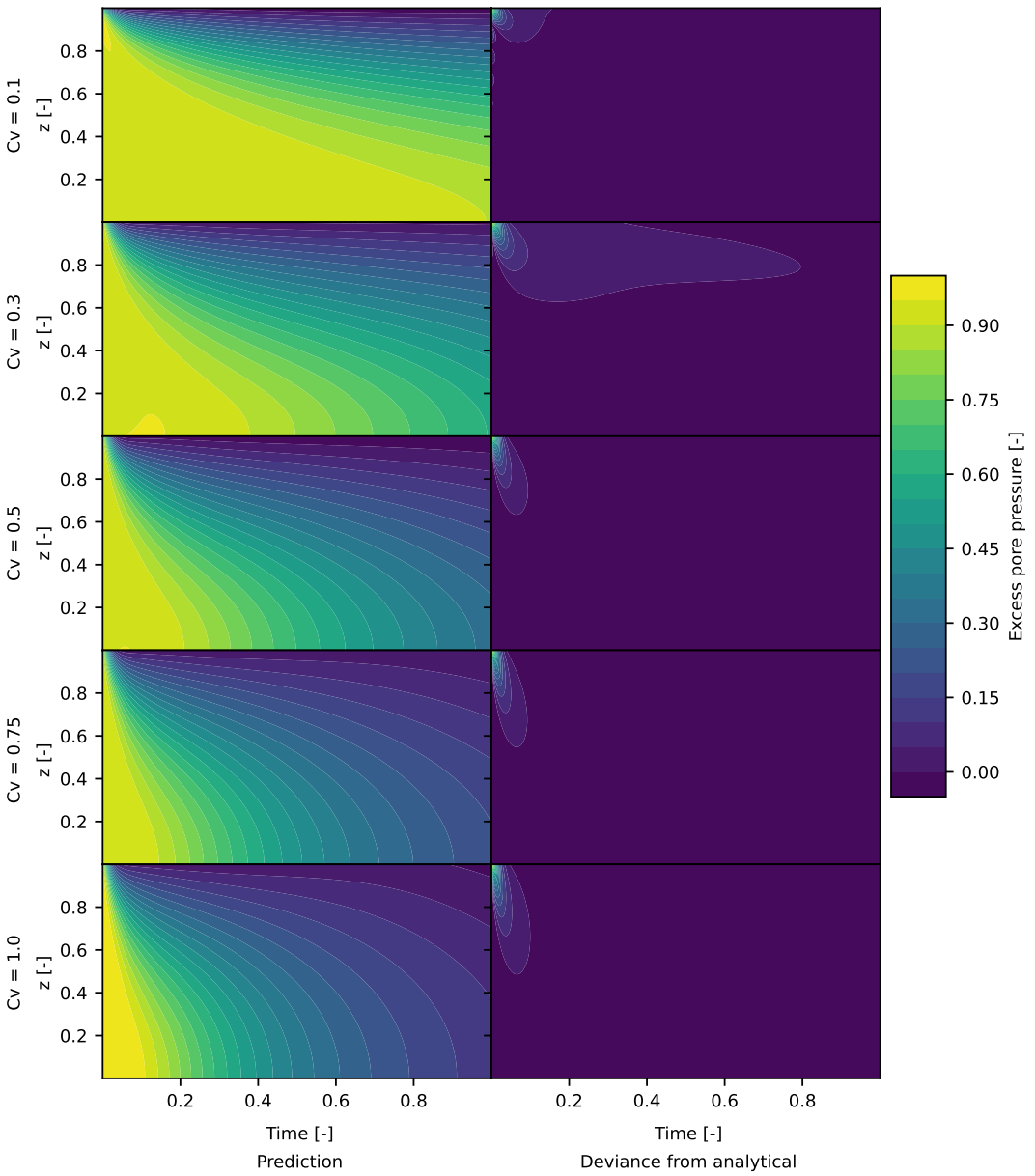


Figure 22: Plot of interpolated and trained C_v values. Each row of subplots is associated with a C_v value, where $C_v = 0.3$ and $C_v = 0.75$ are interpolated predictions.

The trend from the predictions with C_v values from the training data is that the deviance from the analytical solution increases with increasing C_v values. Figure 22 emphasizes the earlier observation that the majority of deviance is due to the irregularity at the upper boundary for small values of t.

Integrating the error for the entire grid, shown in Figure 23, shows that the interpolated C_v values between 0.1 and 0.5 give a considerably less accurate result than the interpolated C_v values between 0.5 and 1.0. This opposes the general trend from the training data, that the deviance increases with increasing C_v values.

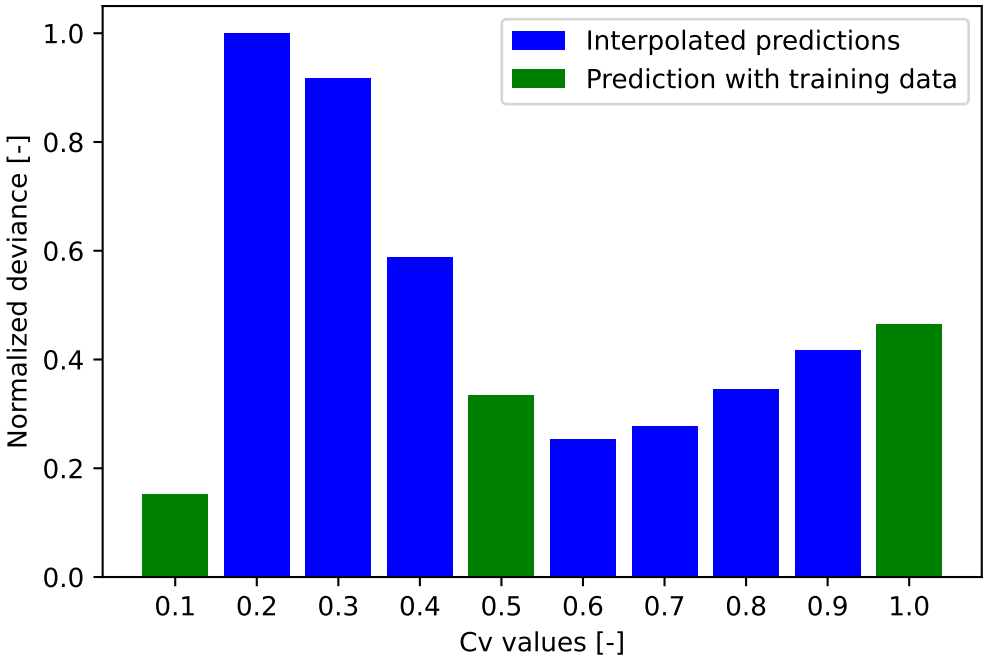


Figure 23: Comparison of the model performance for the interpolated C_v values. The deviances are normalized based on the deviance for $C_v = 0.2$.

The results for the extrapolated C_v values, presented in Figure 24, show much of the same as the interpolated values. The values close to the training data show little to no deviance from the training predictions. However, the predictions become unstable for $C_v \approx 1.7$ where the contour lines start to show non-physical behavior, deviating from the regular shape of the PDE. It is also worth noticing how the predicted pressures for $C_v >= 1.7$ in Figure 24 start to contradict the boundary conditions. Both non-zero pressure values at the upper boundary and non-zero gradients at the lower boundary can be observed.

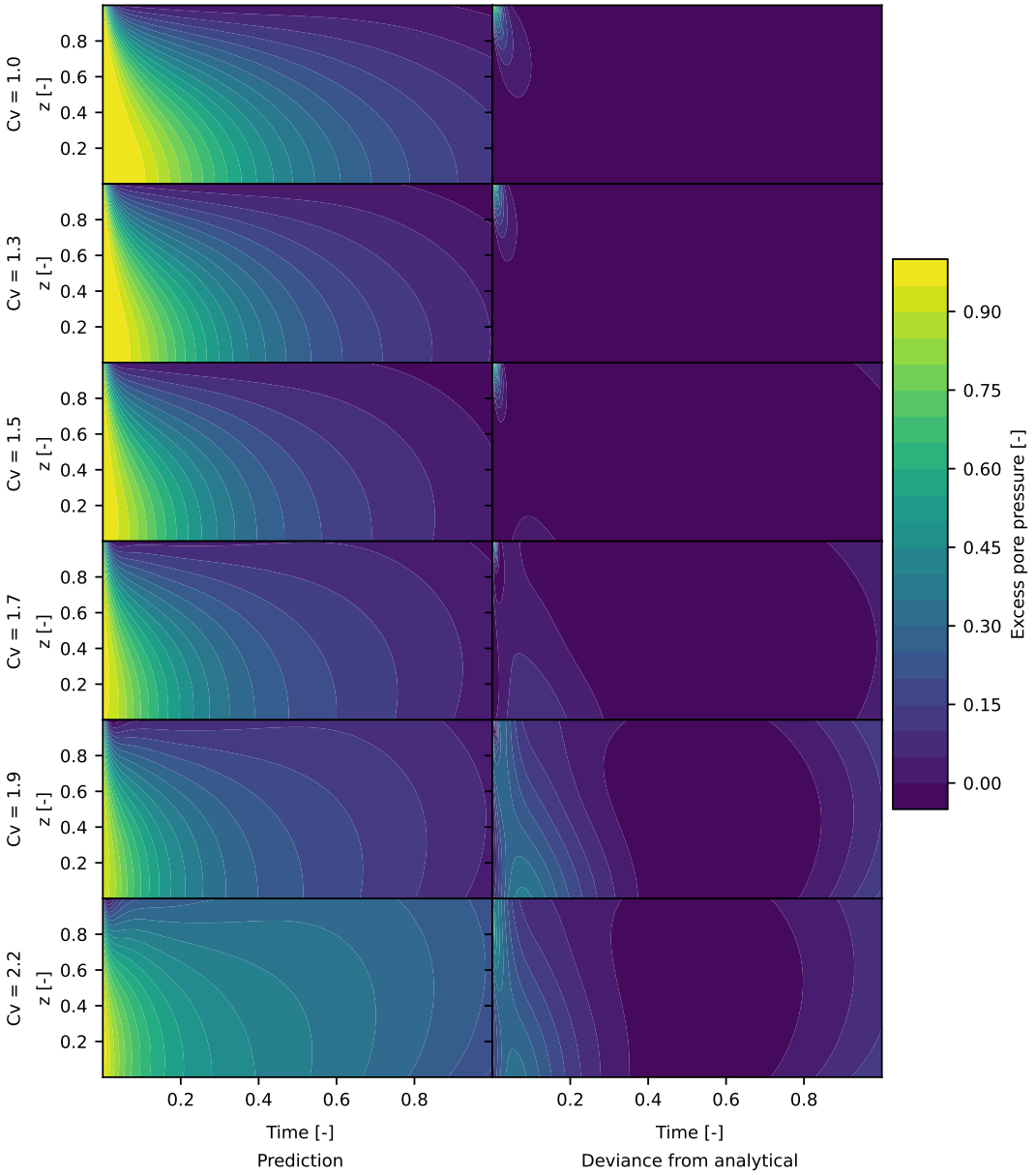


Figure 24: Model prediction for the extrapolated C_v values

Integrating the error shows some interesting results for the extrapolated C_v values, illustrated in Figure 25. The deviance is smallest for $C_v = 1.5$ which is a 50% increase from the highest training value. In fact, this is the second lowest deviance after $C_v = 0.1$.

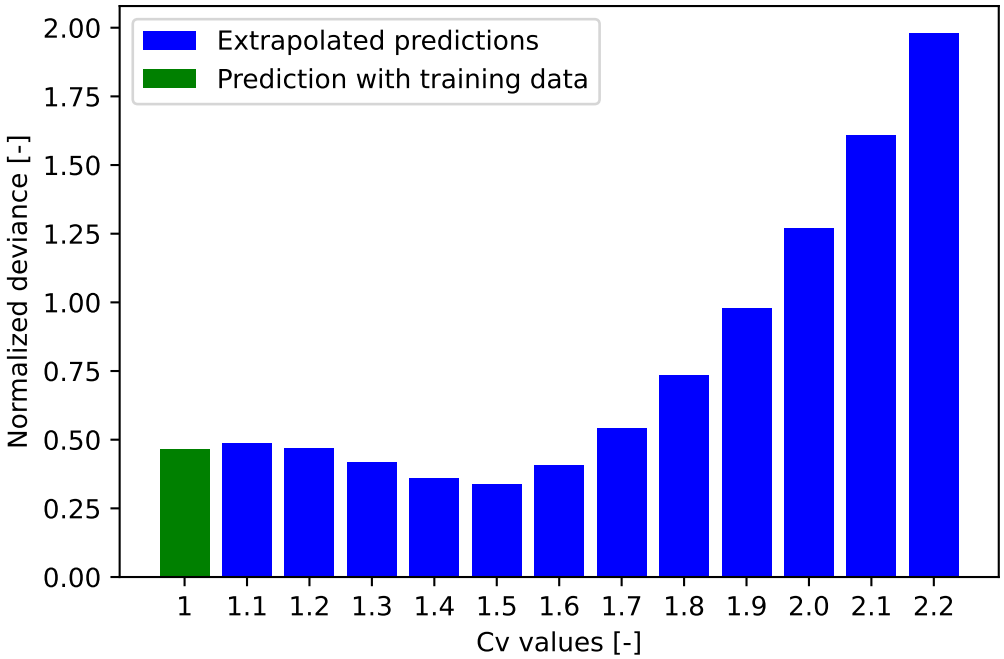


Figure 25: Comparison of the model performance for the extrapolated C_v values. The deviances are normalized based on the highest measured deviance of the interpolation tests, appearing in Figure 23.

4.1.2 Forward probabilistic problem

The result of the model with dropout implemented in both the training phase and in the prediction is presented in Figure 26. Both the mean prediction and the 95% confidence interval were estimated by sample mean and sample standard deviation with Monte Carlo simulations.

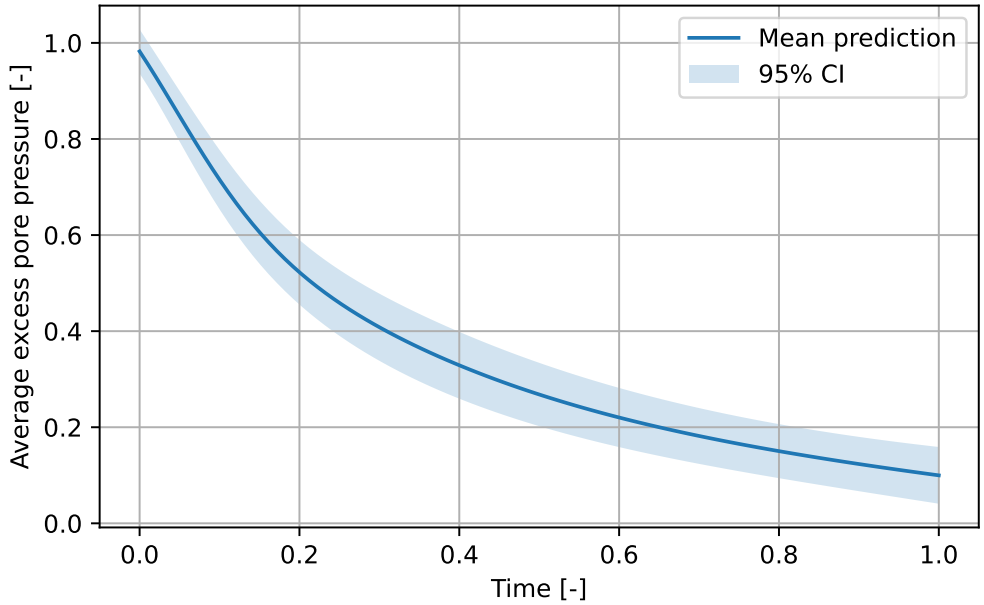


Figure 26: Visualization of the modeling uncertainty using dropout. The pore pressure values are the average over the height of the soil column. Fluid pressure is considered positive.

The wide band in Figure 26 that represents the 95% confidence interval indicates that the model is relatively sensitive to small changes in architecture structure. However, the trend of dissipation is continuous also in the border of the error band, which could indicate that the physical behavior of dissipation is kept through the differing model compilations.

To be able to efficiently compute the statistical moments of the target distribution, the underlying probability distribution of the function needs to be determined. The distribution can be estimated by running Monte Carlo simulations with a wide variety of C_v values at certain time steps. The resulting vertical displacements can then be gathered and presented in histograms, as illustrated in Figure 27.

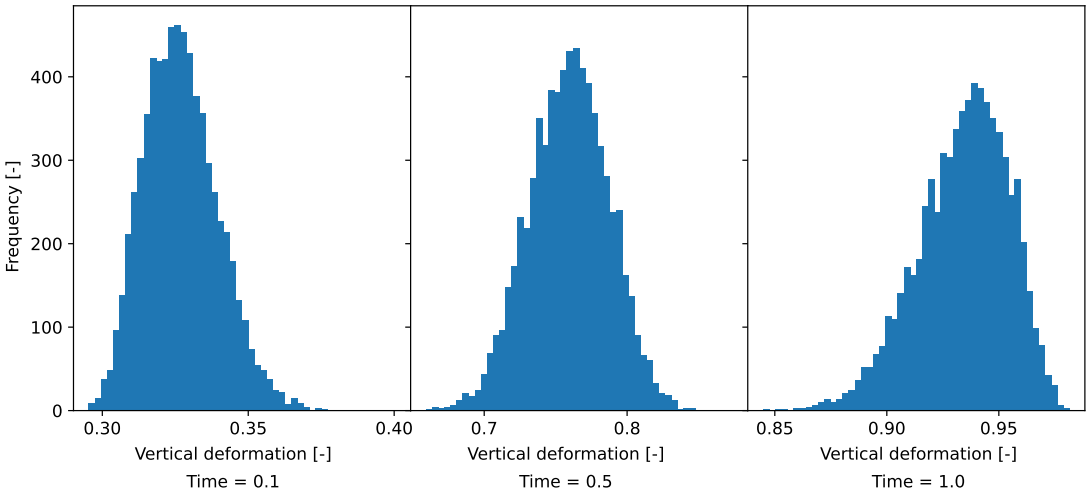
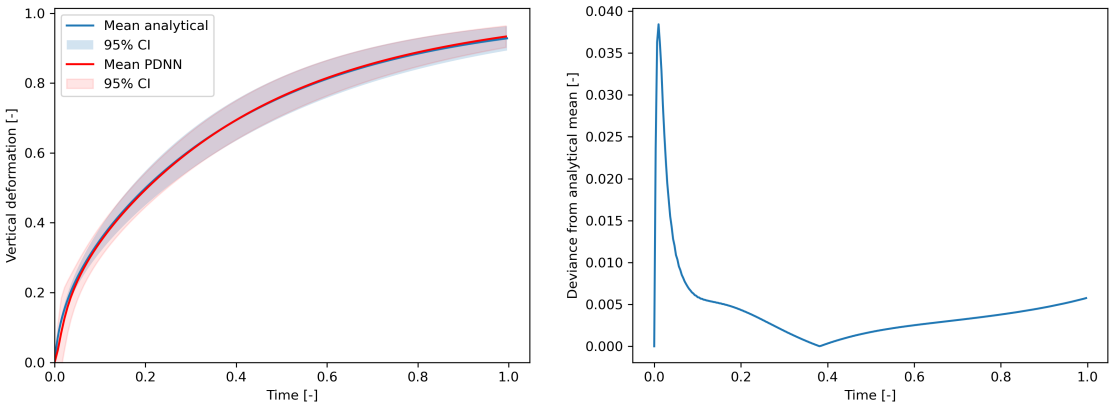


Figure 27: Histogram of calculated vertical displacements with C_v values.

The histograms show that assuming the target distribution to be normally distributed is fairly accurate for the tested timesteps, which implies that the likelihood function can be estimated with a Gaussian function in Equation (60) and that the relation 95th percentile $\approx \mu \pm 1.96\sigma$ is valid.

The results of the forward problem with C_v as a random variable are presented in Figure 28. The predictions of the different consolidation equation compilations coincide well with the analytical solutions, and the mean prediction is actually closer to the deterministic analytical solution than the prediction of the deterministic model, by comparison of Figure 28b and Figure 16b.



(a) Vertical deformation of column top with time

(b) Deviance from the analytical solution

Figure 28: Results from the prediction of vertical displacement over time with uncertain C_v values. Displacement in the direction of gravity is considered positive.

The deviance is approximately reduced by 50% by adding Gaussian white noise and calculating the mean prediction. However, this adds a computational cost that in most cases outweighs the slight increase in accuracy, since the main merit of PINNs as forward solvers is computational efficiency.

Treating C_v as a random variable, and using it as an input to the model, opens the possibility of creating a parameterized model, visualized in Figure 29. Because the nondimensionalized PDE from (65) is the driving force of the model, scaling the model output to fit any of the input parameters is trivial and computationally efficient.

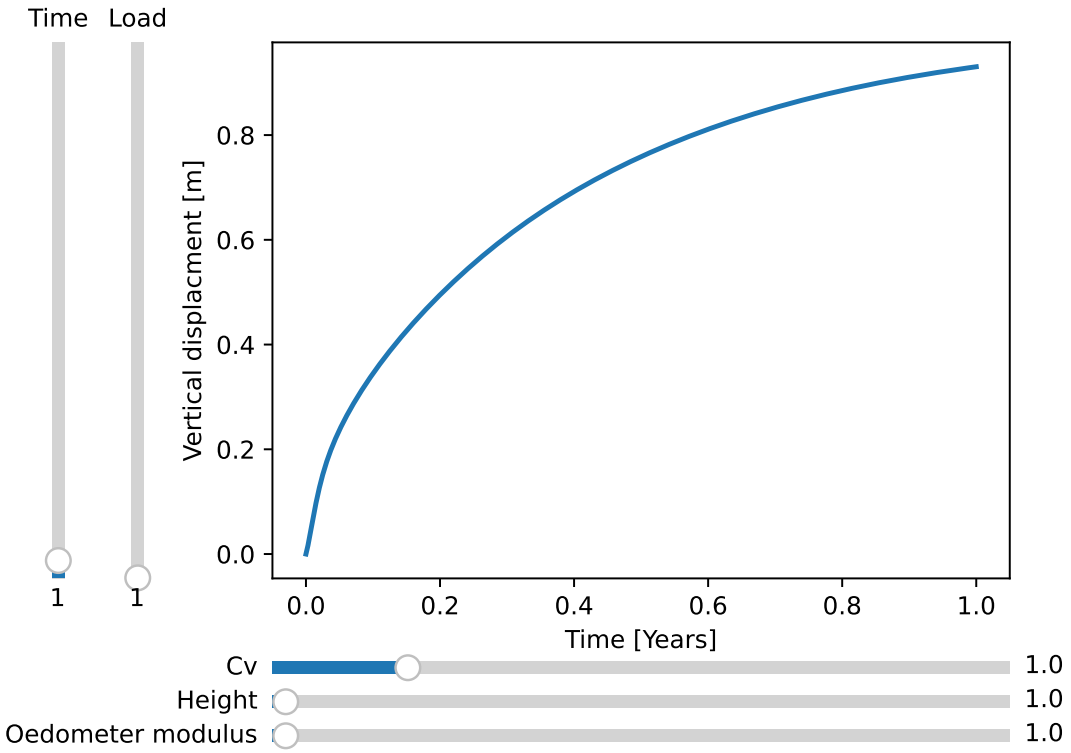


Figure 29: Illustration of a PINN-driven parameterized consolidation model.

Having an efficient parameterized model for one-dimensional consolidation is not especially practical, but increasing the number of dimensions or adding the number of inputs will not increase the computation time drastically since PINN predictions, in reality, are nothing more than matrix operations. So for forward problems, the main benefit of PINNs is the highly efficient computation of predictions after the training. It can therefore be used as first-order approximations for parameterized models which require responsive calculations more than accuracy. FEM, and in the cases where FDM is practical, outperforms PINN in accuracy, interpretability, and reproducibility. So without data, PINNs are less suitable for numerical modeling than their counterparts, especially for more advanced geometries.

4.1.3 Inverse problem with uncertain hydraulic conductivity

For the inverse problem with one random variable, the hydraulic conductivity k_z was treated as unknown. Since C_v is proportional to k_z, the C_v value was used as the random variable in the model.

Figure 30 shows the histogram of the randomly sampled C_v values from the assumed prior distribution $C_v \sim \mathcal{N}(\mu = 1, \sigma = 0.3)$.

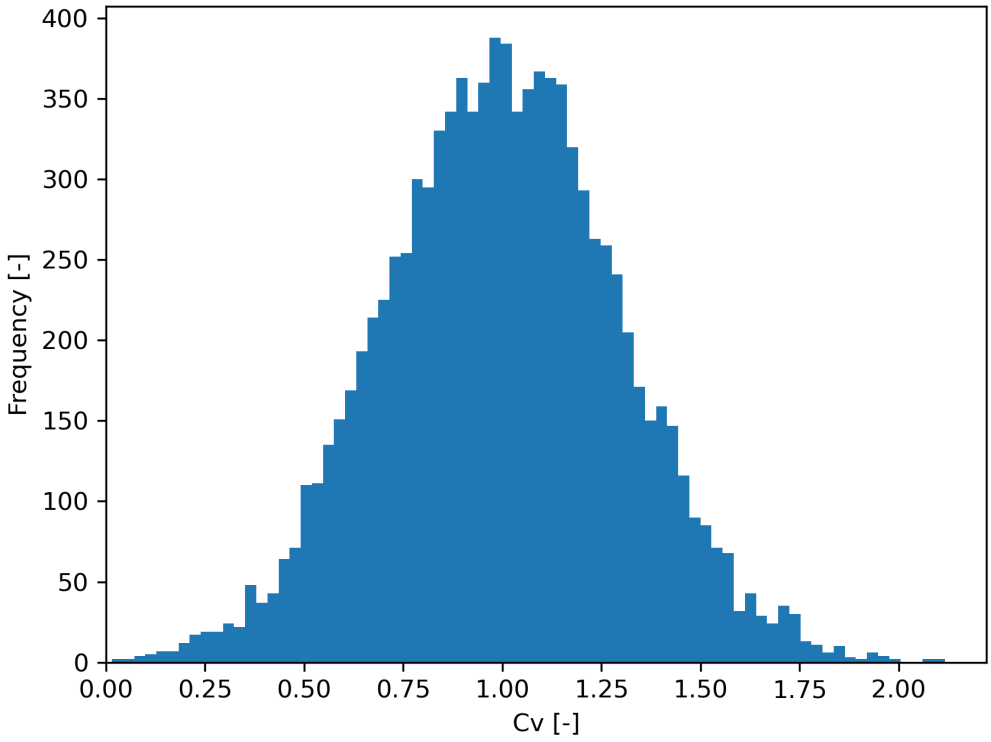
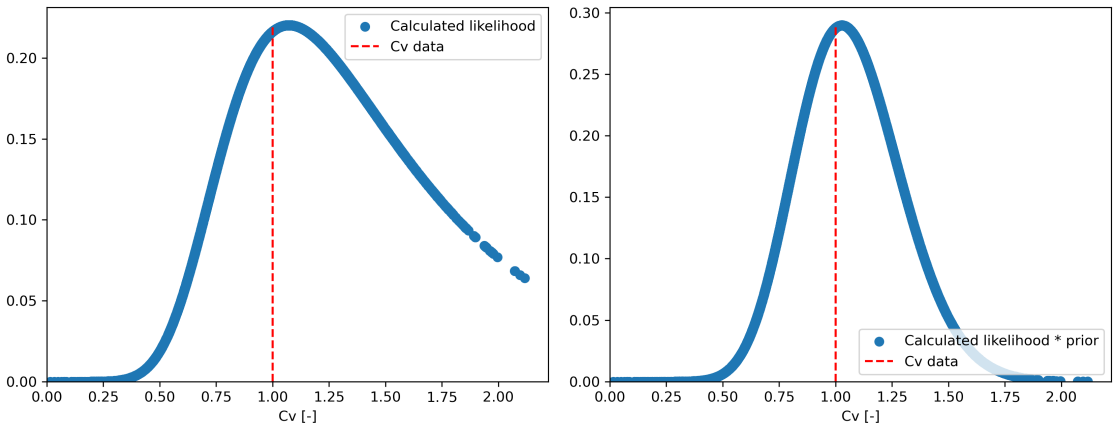


Figure 30: Histogram of C_v values used for training and prior for Bayesian updating.

The results from the MLE without utilizing Markov Chain Monte Carlo (MCMC) inference are shown in Figure 31. The likelihood $P(Data|C_v)$ shows that a C_v just over 1.0 minimizes the deviance between the measurements and the model prediction, thus maximizing $P(Measurements = Data)$. By utilizing Bayes theorem and multiplying with the prior, the distribution gets more concentrated around $\mu = 1$.



(a) Likelihood

(b) Likelihood x Prior

Figure 31: Likelihood functions of the C_v value given the entire data set.

Due to the relatively large range of possible C_v values, calculating the likelihood of all

the possible C_v values was not a computationally efficient way of estimating the likelihood function. The likelihood functions presented in Figure 32 are estimated with the Metropolis-Hastings Algorithm 2, which was faster and equally precise in estimating the likelihood functions.

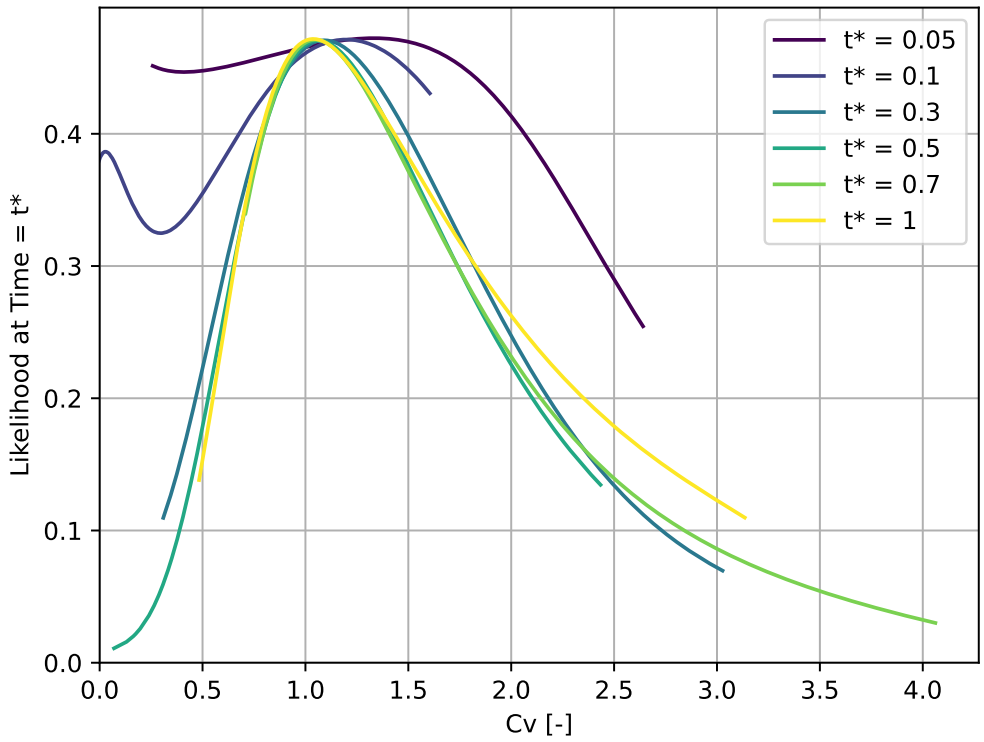
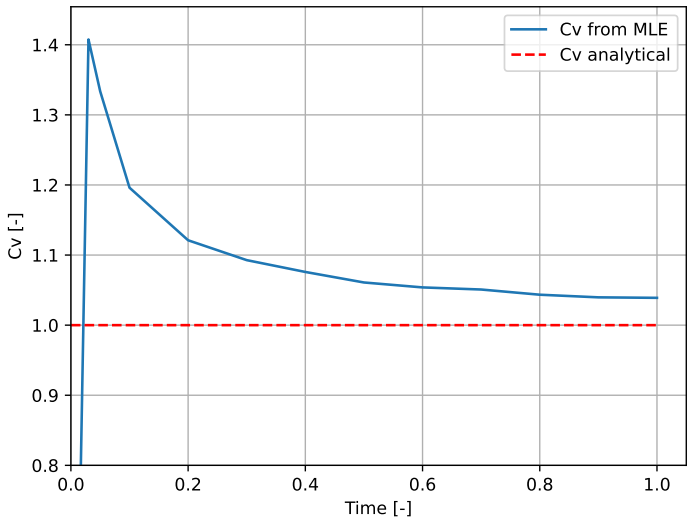


Figure 32: Estimated likelihood functions of the C_v value at different time steps by utilizing an MCMC algorithm.

C_v estimates during the consolidation process are shown as likelihood functions in Figure 32. The mean of the likelihood function estimates converges to the C_v value from the analytical solution with time, as expected. The relative variance in likelihood is not affected as much, due to using the same expected variance of the likelihood function at all time steps. The likelihood functions in Figure 32 are obviously not normalized to probability density functions (PDF), but they do indicate something about the relative size of the standard deviation. This is especially apparent when comparing the shape of the distribution for $t^* = 0.05$ and $t^* = 1$. A narrower shape indicates a more concentrated likelihood, much like a PDF, which again indicates that the model is more confident of its prediction for $t^* = 1$ compared with $t^* = 0.05$. This could be further emphasized by changing the expected variance in the likelihood estimation function, but in reality, this will require prior knowledge of how the model uncertainty changes at the different timesteps.

The likelihood from Figure 32 can be reduced to a scalar value that represents the MLE. The C_v values with the highest likelihood for the selected time steps are plotted and tabulated in Figure 33.



(a) Convergence plot

Time [-]	C_v [-]
0.01	0.475
0.03	1.408
0.05	1.334
0.1	1.196
0.2	1.121
0.3	1.093
0.4	1.076
0.5	1.061
0.6	1.054
0.7	1.051
0.8	1.043
0.9	1.040
1.0	1.039

(b) Table of Cv values

Figure 33: Convergence plot of C_v-Time of prediction, with underlying values tabulated.

After an apparent initialization at $C_v \approx 0.475$, the estimation mostly overshoots the analytical value. The prediction for the smallest time step is greatly affected by the numerical challenges discussed for the forward problem, so discarding the earliest time step can therefore provide a more interesting insight into the relationship between the measurements and the model predictions. Disregarding $t^* = 0.05$ provides a clear trend that the MLE converges from above, which is a recurring theme. There are clear similarities between the convergence plot of the MLE and the dissipation rate of the excess pore pressure illustrated in Figure 26, which may be an important correlation to consider when assessing the model precision.

The updated predictions of vertical displacement with time based on C_v values from the MLE are visualized in Figure 34. This form of Bayesian updating with new evidence provides a robust way of incorporating measurements to update model predictions and can be used for other numerical methods as well. The evolution of predictions of vertical displacement with time follows the same trend as the MLE, where the earliest time step undershoots the analytical solution and the rest overshoots. The later the prediction, the closer to the analytical solution. The last few predictions where $C_v \to 1$ is only slightly more deviant from the analytical solution than the deterministic model.

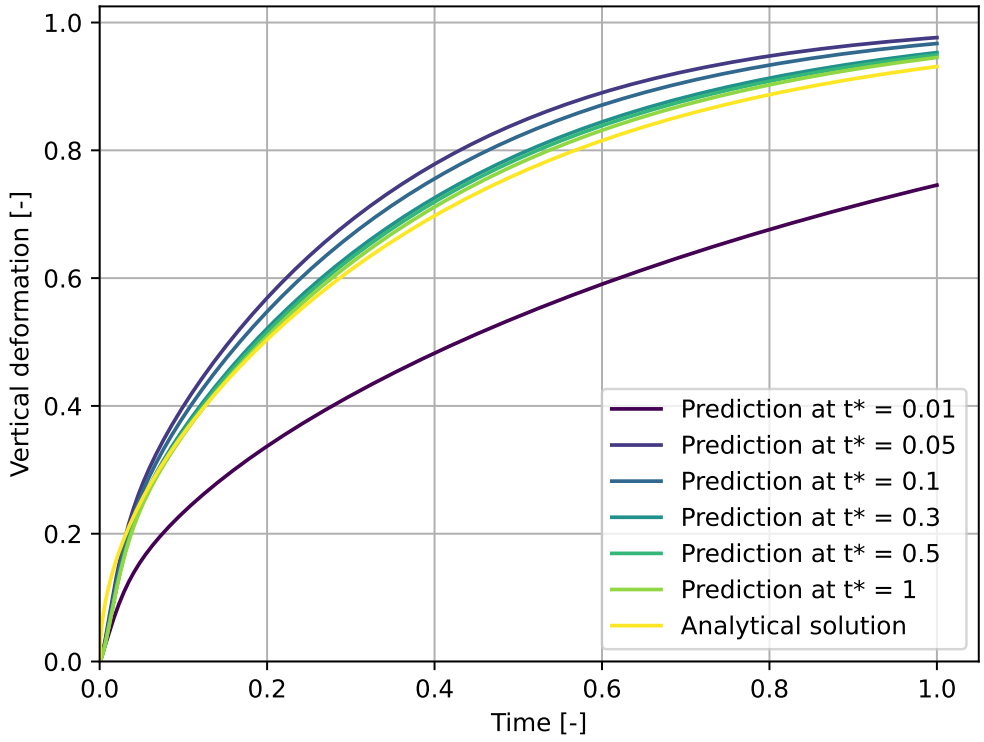


Figure 34: Plot of predictions of vertical deformation of the column top at different time steps. Displacement in the direction of gravity is considered positive.

The prediction for the final settlement can be isolated from the values from Figure 34 by focusing on the prediction for $Time = 1$. The deviances of the predictions compared with the analytical solution are presented in Figure 35. The convergence rate of the decline in error in final settlement prediction is similar to that of the C_v estimation. Starting from a value well above 7% at $t^* = 0.03$, the deviance quickly converges to roughly 3.7% at $t^* = 0.1$. The deviance seems to fully converge and stabilize at 1.5% for what essentially is a prediction of the current state.

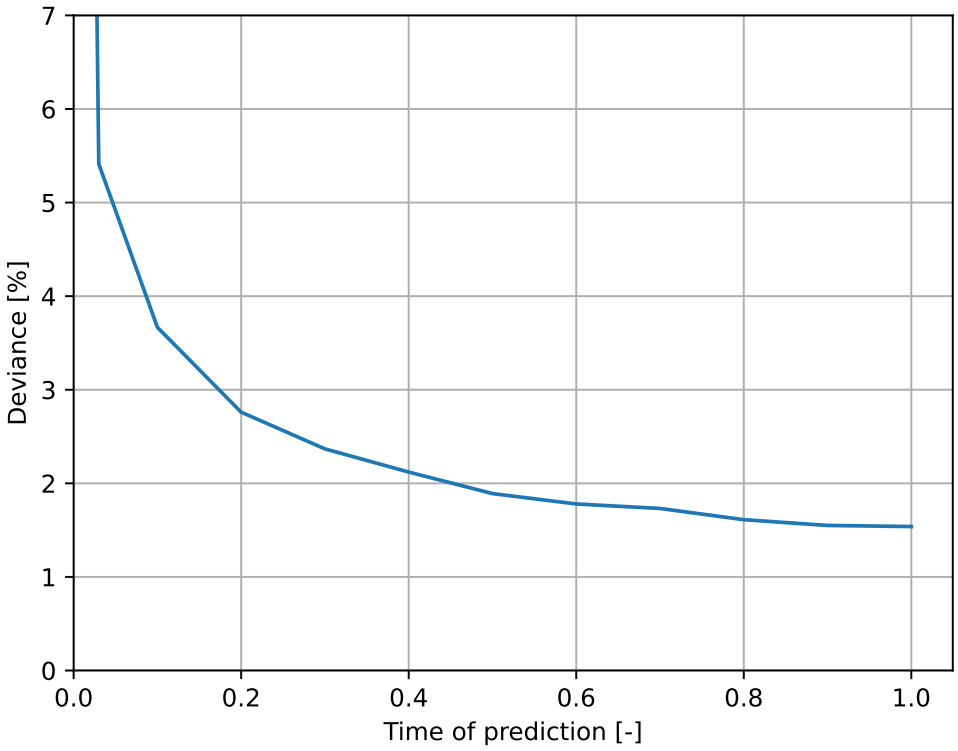


Figure 35: Plot of the deviance between analytical solution and predictions of final settlement at different time steps. The deviance is relative to the absolute value of δ_v from the analytical solution.

The results from the alternative option, where the MLE calculation is performed inside the PINN model as per Algorithm (3), are presented in Figure 36. The results show that the PINN optimization of the parameter follows the same overshooting trend as MLE in postprocessing. The accuracy is quite similar in terms of MLE for $t^* = 1$, but the range of tested C_v values is less diverse in the inbuilt option and the accuracy is of course dependent on the training stage. The benefit of this method is that the C_v parameter is treated as a parameter and not additional input. So in addition to the MLE calculation, the parameter is being optimized in the same manner as the weights and biases of the model. The returned scalar is then the optimized value of C_v based on the provided measurements.

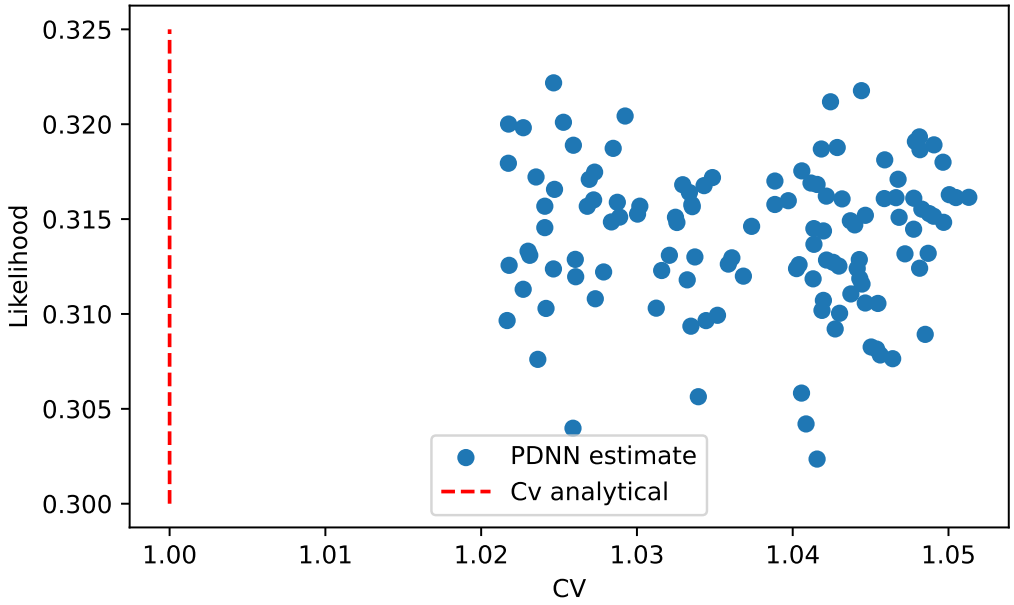
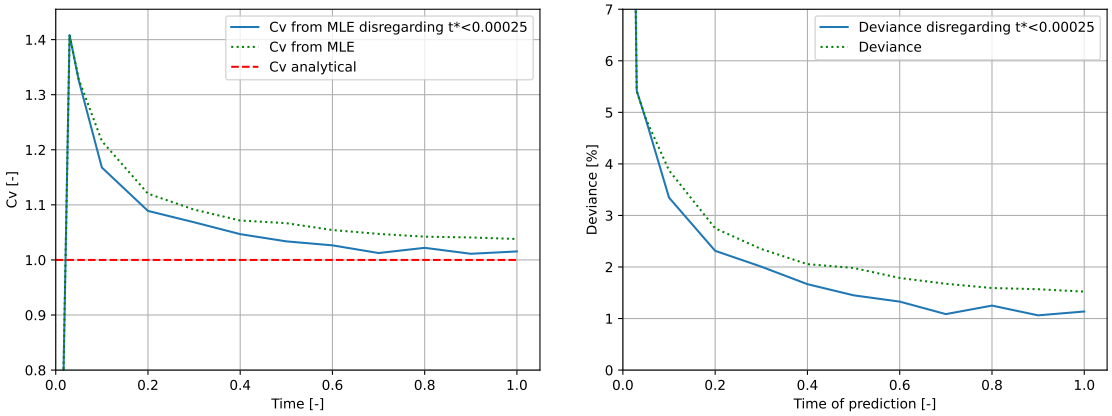


Figure 36: Plot of the likelihood estimates during training.

By disregarding the earliest time steps, the deviance between the MLE and the analytical C_v value decreases. Figure 37 compares the estimates with and without the time steps before $t^* = 0.00025$. The results from the computation of the deterministic forward problem showed that the deviation between the model predictions and the analytical solution was relatively high for the early time steps. This error seems to propagate to the MLE calculations for later time steps. The fact that the early time steps affect the prediction at $t^* = 1$ to the extent that the deviance is 50% larger, emphasizes the importance of extensive knowledge of model behavior.



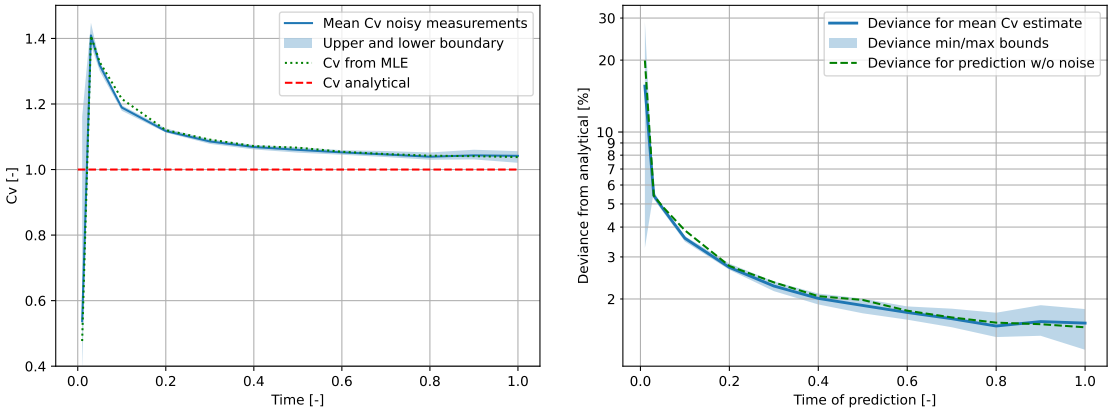
(a) MLE at different timesteps

(b) Deviances from analytical solution

Figure 37: A comparison of the original MLE calculation and the MLE calculation disregarding early time steps.

4.1.4 Predictions with noisy measurements

Adding Gaussian white noise to the piezometer data affects the MLE results moderately overall, with some exceptions, as illustrated in Figure 38.



(a) MLE at different timesteps

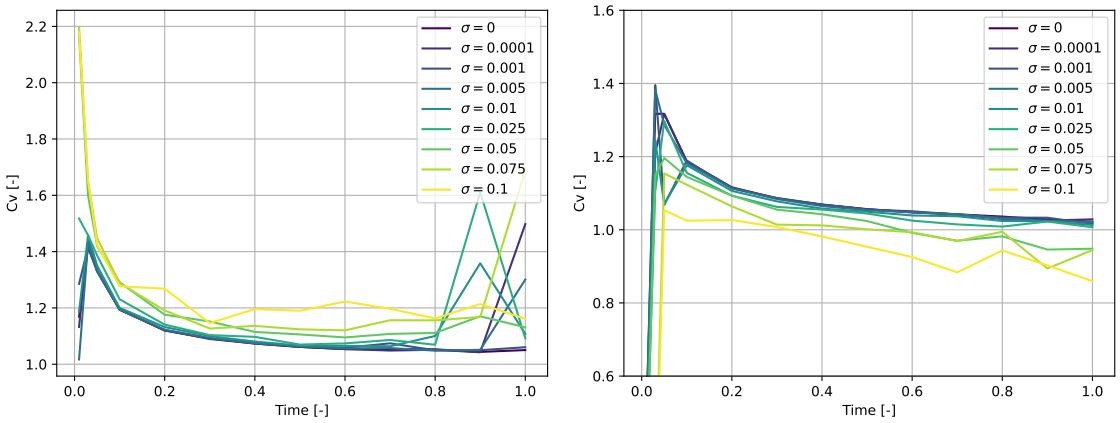
(b) Deviance final settlement prediction, log-scale

Figure 38: A comparison of the analytical solution and model predictions with and without noise.

The predictions are especially sensitive to noisy data for the early and late time steps. Since the time derivative of u is small for small values of t^*, the rate of change in measurements is almost exclusively decided by noise and not change in pore pressure. This leads to inaccurate and imprecise C_v estimates for $t^* < 0.05T$. The precision increases with time, and the upper and lower bounds are practically equal for $0.05T < t^* < 0.3T$ which is where the time derivative of u is largest. The precision decreases for $t^* > 0.3T$, where the time derivative of u decreases again.

The mean MLE with noisy measurements shows the same tendency as the probabilistic forward problem. The accuracy of the mean prediction seems to be unaffected by Gaussian noise, as long as the mean is calculated for a sufficiently large number of samples. This coincides with the expected outcome of predictions with white Gaussian noise from a statistical point of view.

Figure 39 shows the extremal results after Monte Carlo simulations of MLE for different levels of noise. The results are consistent with the previous findings, but the plot of the maxima estimates highlights the problem with measurements at early and late time steps. The sensitivity to noise is heightened by a considerable amount when the rate of change in u is low. For the timesteps in the middle, however, only the maxima for the highest noise level show unstable behavior.



(a) Max C_v estimates

(b) Min C_v estimates

Figure 39: Figures displaying the extremal MLE plots for the unknown C_v value obtained through Monte Carlo simulations at various timesteps and noise levels.

It is also interesting that the lowest MLE results for the lower noise levels are higher than the analytical C_v. This provides further evidence as to how strong the model bias is. Another interesting observation is that the lowest MLE predictions do not suffer from the same instability for the latest timesteps. The difference between the noisiest prediction and the prediction without noise is almost linear after $t^* = 0.1$.

The mean prediction at different timesteps after the Monte Carlo simulations for different noise levels is shown in Figure 40.

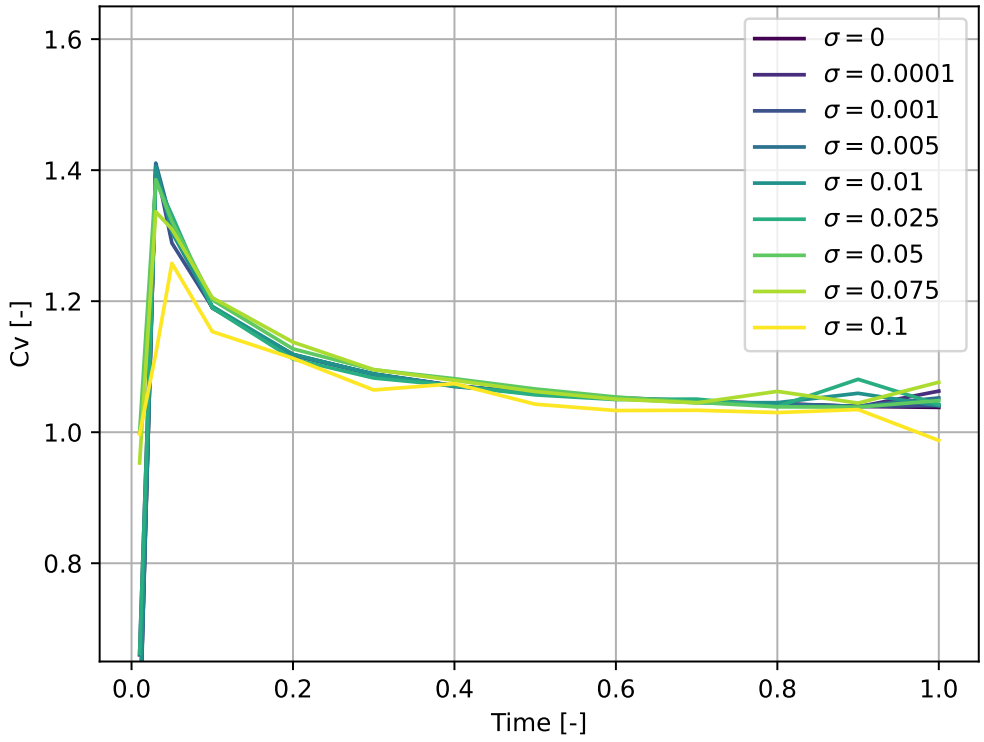


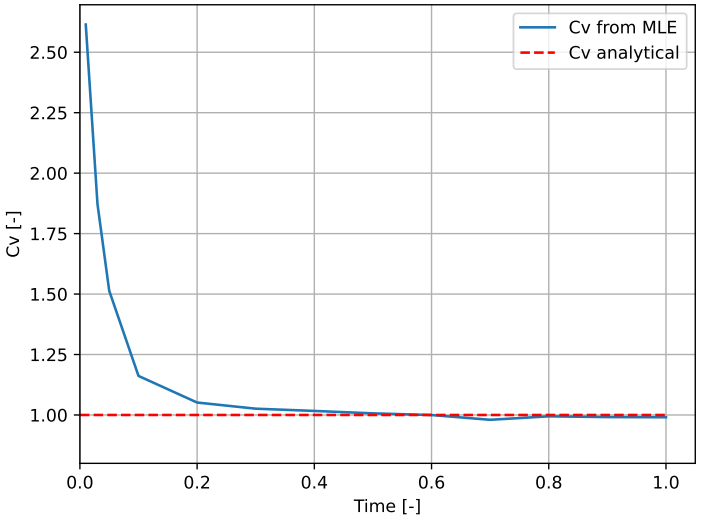
Figure 40: Average of the MLE at different timesteps with varying noise levels.

The overall tendency shows again that the noise levels effectively cancel out if the number of Monte Carlo simulations is sufficiently large. However, there are some interesting new observations to be made at the last timestep. The bias is clearly towards higher C_v estimates at the last timestep for all noise levels except $\sigma = 0.1$ which dips notably downwards.

4.1.5 Displacement measurements

If the piezometer measurements are replaced with displacement measurements, the accuracy increases but so does also the computation time. Instead of evaluating u for one value of z, u has to be evaluated and integrated for the entire soil column each time step.

The line plot and table in Figure 41 show the results from the MLE with displacement measurements.



(a) Convergence plot

Time [-]	C_v [-]
0.01	2.615
0.03	1.872
0.05	1.514
0.1	1.161
0.2	1.051
0.3	1.026
0.4	1.017
0.5	1.007
0.6	1.000
0.7	0.980
0.8	0.994
0.9	0.991
1.0	0.991

(b) Table of Cv values

Figure 41: Convergence plot of C_v-Time of prediction, for displacement measurements.

The results show that the MLE with displacement measurements converges much faster and more accurately toward the analytical solution than the equivalent algorithm with piezometer measurements. After just $t^* = 0.3$ the prediction beats the accuracy of the piezometer at $t^* = 1$.

Figure 42 shows that the deviance in the predictions of final settlement is effectively halved with displacement measurements. Both Figure 41 and 42 show a turning point at $t^* = 0.7$, which may be connected to the intersection point of the PINN predictions and the analytical solution as seen in Figure 28 and 16. The fact that this intersection point changes from $t^* = 0.4$ to $t^* = 0.5$ and lastly $t^* = 0.7$ for different models is unfortunate regarding model precision.

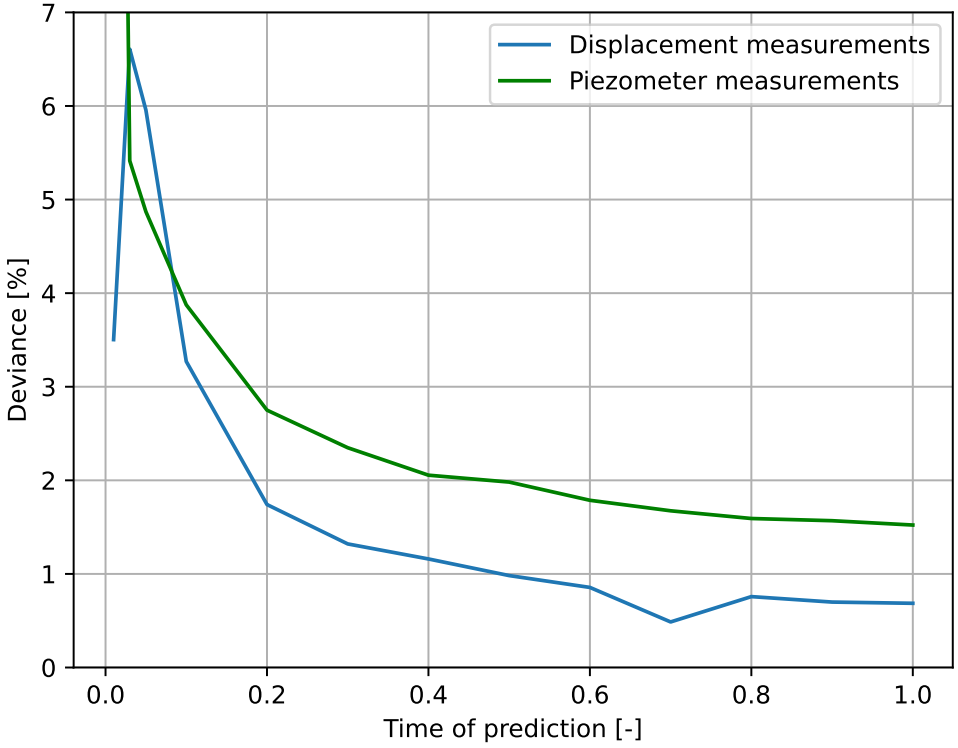
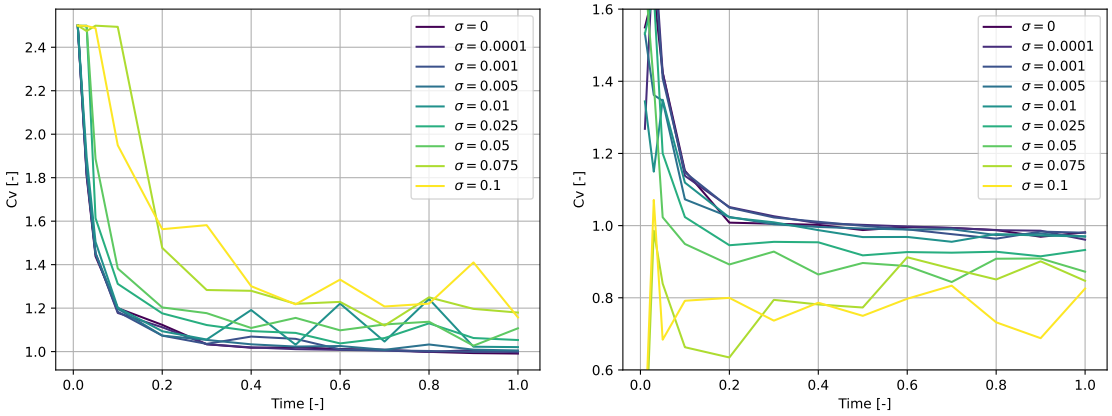


Figure 42: Comparison of the deviation in final settlement prediction for different measurement types at different timesteps.

Adding noise to the displacement measurements yields similar results compared with the piezometer measurements, however, due to the computational demand of integrating the strains, there are fewer evaluation points in the MLE per Monte Carlo simulation. Therefore, these results show more erratic behavior, showcased in Figure 43. The erratic behavior implies that the number of evaluation points in the MLE algorithm should be increased to ensure accurate predictions.



(a) Max C_v estimates

(b) Min C_v estimates

Figure 43: MLE of the C_v parameter over time with different levels of noise added to the displacement measurements.

Despite the sparse number of MLE evaluations, the mean estimate of the Monte Carlo simulations, visualized in Figure 44, yields similar results compared with the clean measurements. Interestingly, the mean predictions seem to converge to a slightly lower C_v value than shown in Figure 41 for $t^* = 1$.

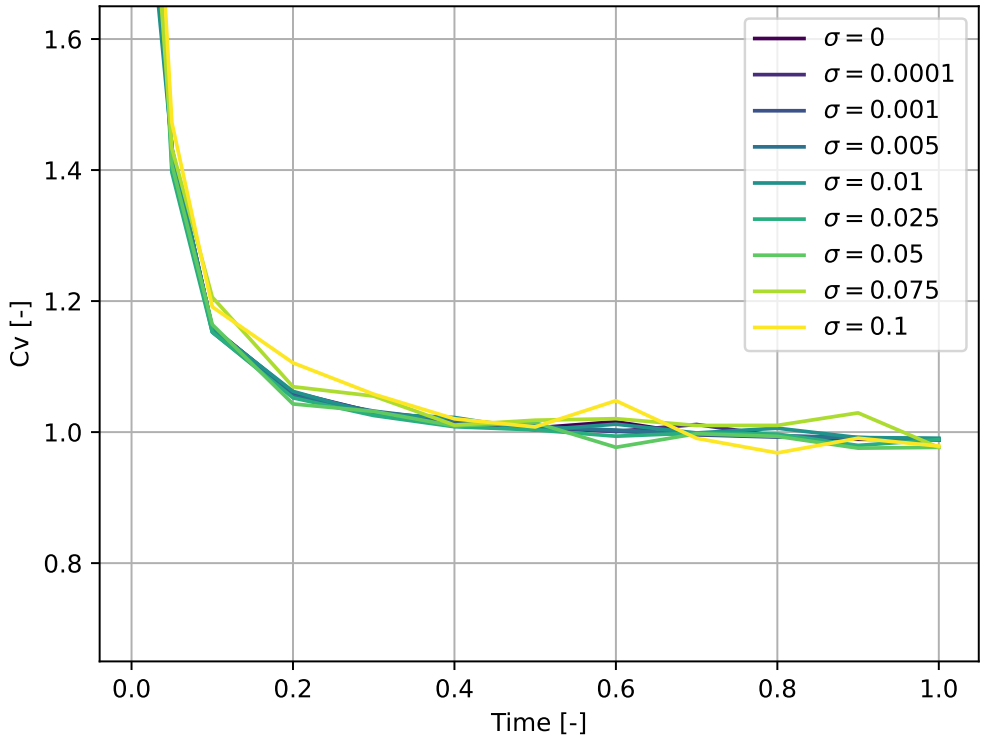


Figure 44: Average of the MLE estimates at different timesteps with different levels of added noise to the displacement measurements.

4.1.6 Two random variables

By utilizing both types of measurements, MLE can provide estimates for both the permeability k_z and the compressibility E_{oed}.

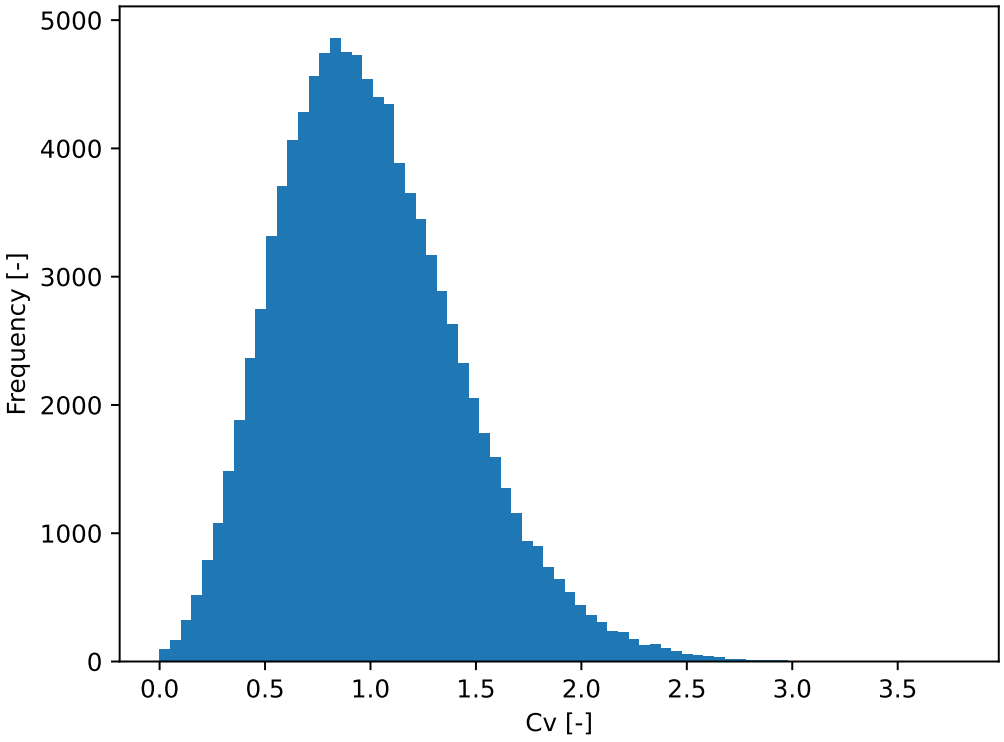


Figure 45: Histogram of C_v values based on prior for E_{oed} and k_z

The calculation of C_v based on random sampling of the two normally distributed random variables yields a slightly skewed histogram of C_v values. This is mainly due to the fact that negative values for k_z and E_oed is impossible. Assuming C_v to be normally distributed in the computations will therefore yield a small modeling error.

The only thing changing in the likelihood calculation of C_v with two random variables is that the standard deviation of the prior distribution $\frac{E_{oed}k_z}{\gamma_w}$ now is larger, which is evident in Figure 46. Nonetheless, the end result of the MLE for the entire range of prior C_v values is similarly accurate as for one random variable.

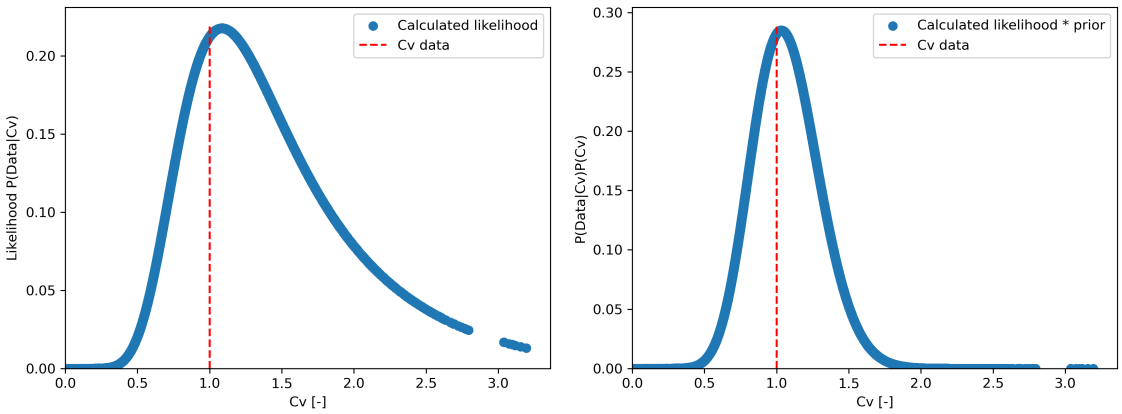


Figure 46: Likelihood functions of the C_v value given the entire data set.

The results of the MLE at different time steps for both of the random variables are shown in Figure 47. The estimate for E_oed has fully converged after $t^* = 0.05$, which is considerably faster than the estimates for C_v with just piezometer or displacement measurements. The estimates of the permeability k_z, on the other hand, exhibit more similar behavior compared with the previous results.

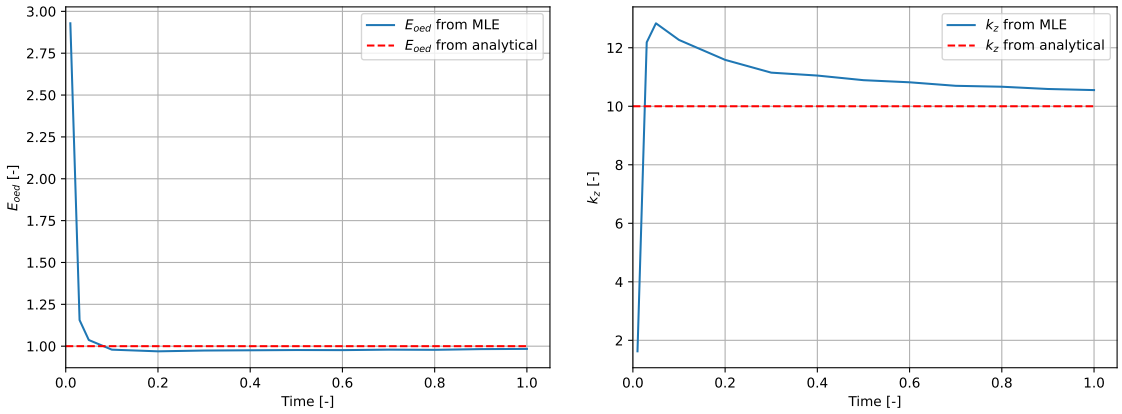


Figure 47: MLE of the two random variables at different time steps

In the end, the deviance of the predictions with both measurement types converges much faster, as illustrated in Figure 48. The final predictions are also closer to the analytical solution. So in addition to the possibility of treating both the compressibility E_oed and permeability k_z as random variables, the accuracy increases for almost all time steps. The exception is the earliest time steps, where none of the results were close to the analytical solution.

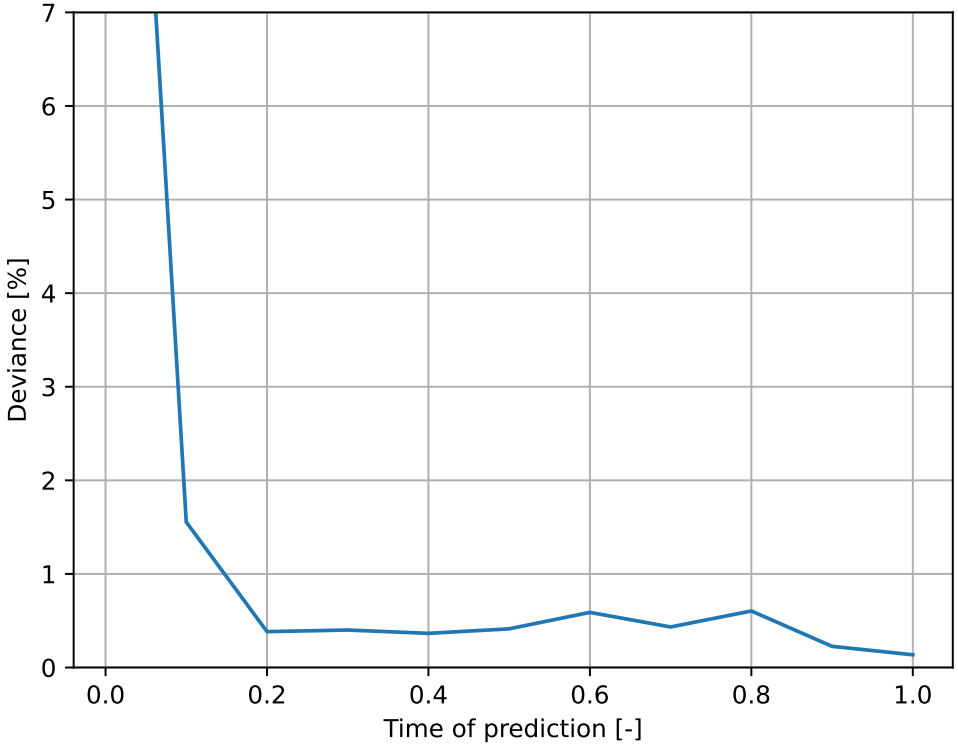


Figure 48: Deviance in final settlement prediction over time with two random variables.

4.1.7 Computational efficiency

Table 4 shows an overview for comparison of computation time for the different methods used to obtain the results. Some of the more computationally heavy calculations are sensitive to fluctuations in available processor power, so these are not exact measurements and should therefore be interpreted as approximate values.

Table 4: Computational efficiency 1D.

Task	Parameters	Time
Training time forward	3000 epochs, 100.000 samples, 2 inputs	$\approx 12000s$
Training time inverse	3000 epochs, 100.000 samples, 3 inputs	$\approx 12000s$
Prediction time forward	100.000 samples	$0.14s$
Prediction time inverse	100.000 samples	$0.14s$
Calculation time Fourier series	100.000 samples, n=10	$3.04s$
Calculation time FEM	100 linear elements, 20000 time steps	$1.58s$
Calculation time FDM	200.000 samples	$2.23s$
Calculation time MLE(u)	MCMC(n=300), $\mathcal{L}(n = 10)$	$\frac{6s}{timestep}$
Calculation time MLE(δ_v)	MCMC(n=300), $\mathcal{L}(n = 10)$	$\frac{60s}{timestep}$

The results show that when trained, predictions with PINNs are very computationally effective compared with the other methods. The uncertainty quantification with noisy measurements took $12736.4s$ with piezometer measurements and $131587.3s$ with displacement measurements. Extrapolating the times listed in Table 4 with these numbers shows the importance of computational efficiency in these types of problems.

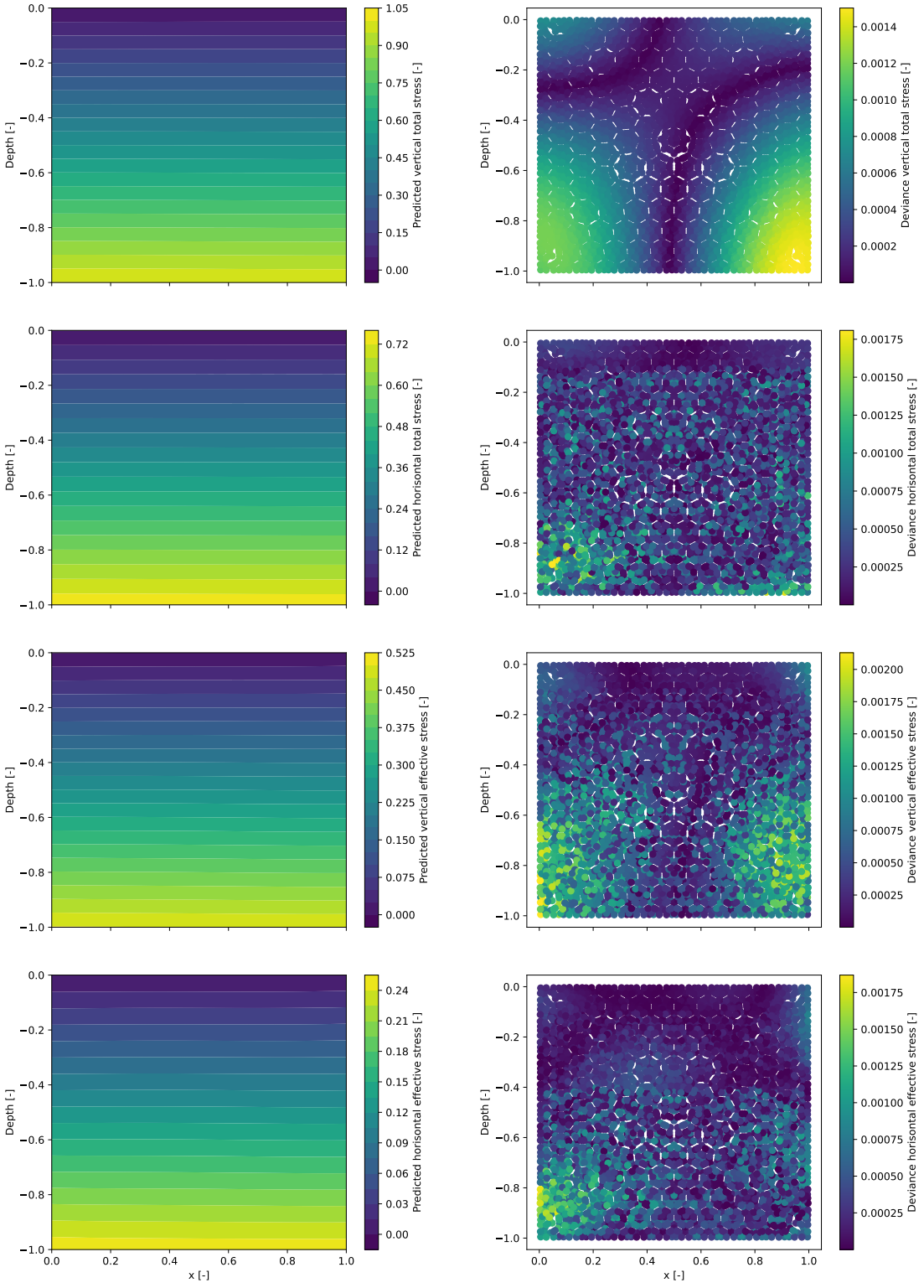
4.2 Two-dimensional uncoupled flow

The results from the 2D analysis of a line load on a homogenous soil layer are presented and compared with results from traditional numerical methods. The predictions from the PINN model and the FDM solution assume negligible horizontal strains, whereas the Plaxis model does not. The performance of the PINN model in regards to accuracy for solving 2D diffusion problems should therefore mainly rely on comparison with the FDM approximation.

The calculation and therefore also presentation of results consists of three different phases, the initial phase, loading phase, and consolidation phase. They are presented in chronological order. The results from the first two phases consist of the model prediction and the deviance from the Plaxis analysis. All the relevant stresses are plotted and compared, whereas strain and displacement are dropped for the first two phases since they are all zero due to the problem definition. Each plot consists of two horizontally aligned subplots, prediction and deviance. So each row represents a stress or pressure component and the columns represent prediction or deviance.

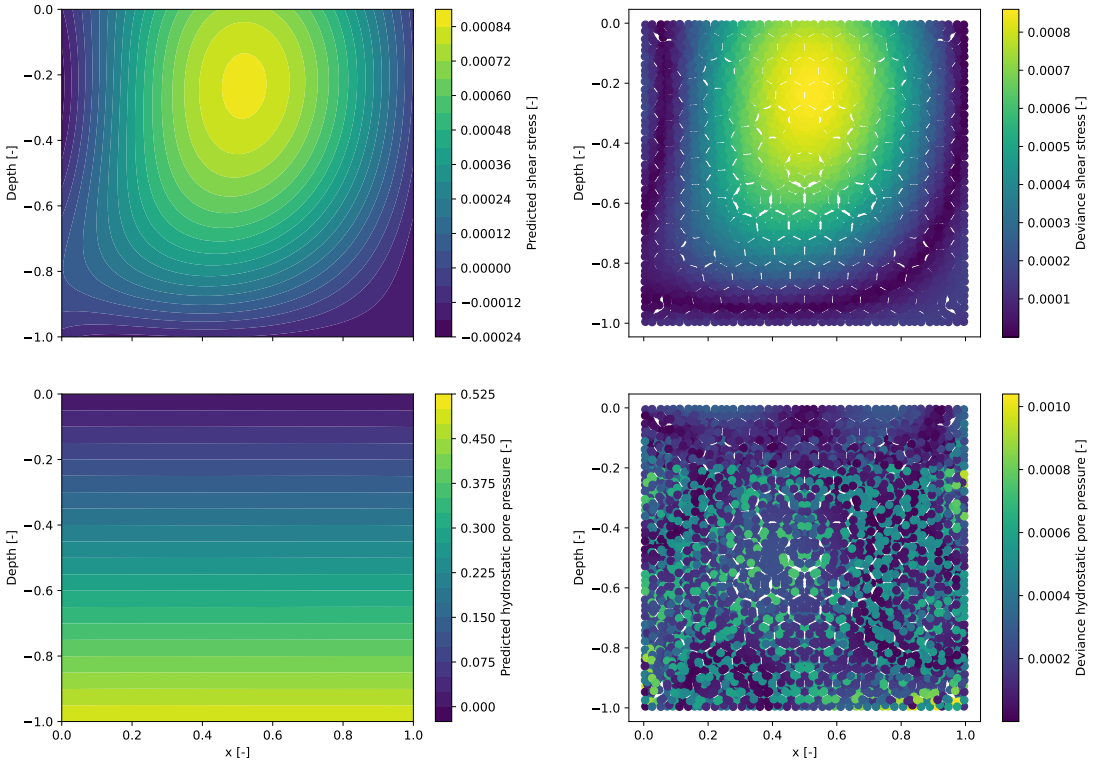
The presentation of the consolidation phase differs considerably from the previous phases. The FDM approximation of 2D diffusion is included, and the spatial distribution of stress, pore pressure, and displacement is plotted for three different time steps. For the uncoupled problem, the vertical displacement is of primary interest, so only the results regarding vertical stress, displacement, and excess pore pressure are presented. Each plot represents a functional, where each row of subplots represents a time step and each column represents the numerical method used.

4.2.1 Initial phase



(a) Model prediction

(b) Deviance from Plaxis results



(c) Model prediction

(d) Deviance from Plaxis results

Figure 49: Results from the in-situ stress initialization and comparison with Plaxis results. Each row is a stress or pressure component identified with the label of the color scale. The sign convention is that both compression and pore pressure is positive.

The results from the stress initialization of the soil layer in the initial phase are presented in Figure 49. The model prediction of the initial stress distribution is in general accurate, both in terms of absolute values and physical behavior.

Some interesting results can be found by taking a closer look at the error plots. The model prediction is performed on the stress points from Plaxis, which differs from the training grid. However, the results from the 1D problem, showcased in Figure 21, showed that this kind of interpolation results in little to no error.

The relative deviance from the Plaxis results is the smallest for vertical total stress and pore pressure. This could result from the fact that those two properties are uniquely defined by one equation each, Equation (78) for vertical stress, and the linear equation $p_w = \gamma_w z$ for hydrostatic pore pressure. The prediction of the vertical effective stress is influenced by the prediction of both pore pressure and total stress. The prediction is therefore prone to inherit the error of both predictions, which may cancel each other out or accumulate. A closer inspection of the deviances related to the three mentioned stresses and pressures shows that the deviance is highest in the lower corners. This indicates that the deviance of pore pressure and vertical total stress accumulates, and not cancels in the prediction of vertical effective stress. The deviance is in this case relatively small, but since the initial stresses affect further calculations and results, the accumulated error may propagate and cause considerable uncertainty in the final prediction.

The horizontal effective stress is defined by its relation to three functionals. Vertical effec-

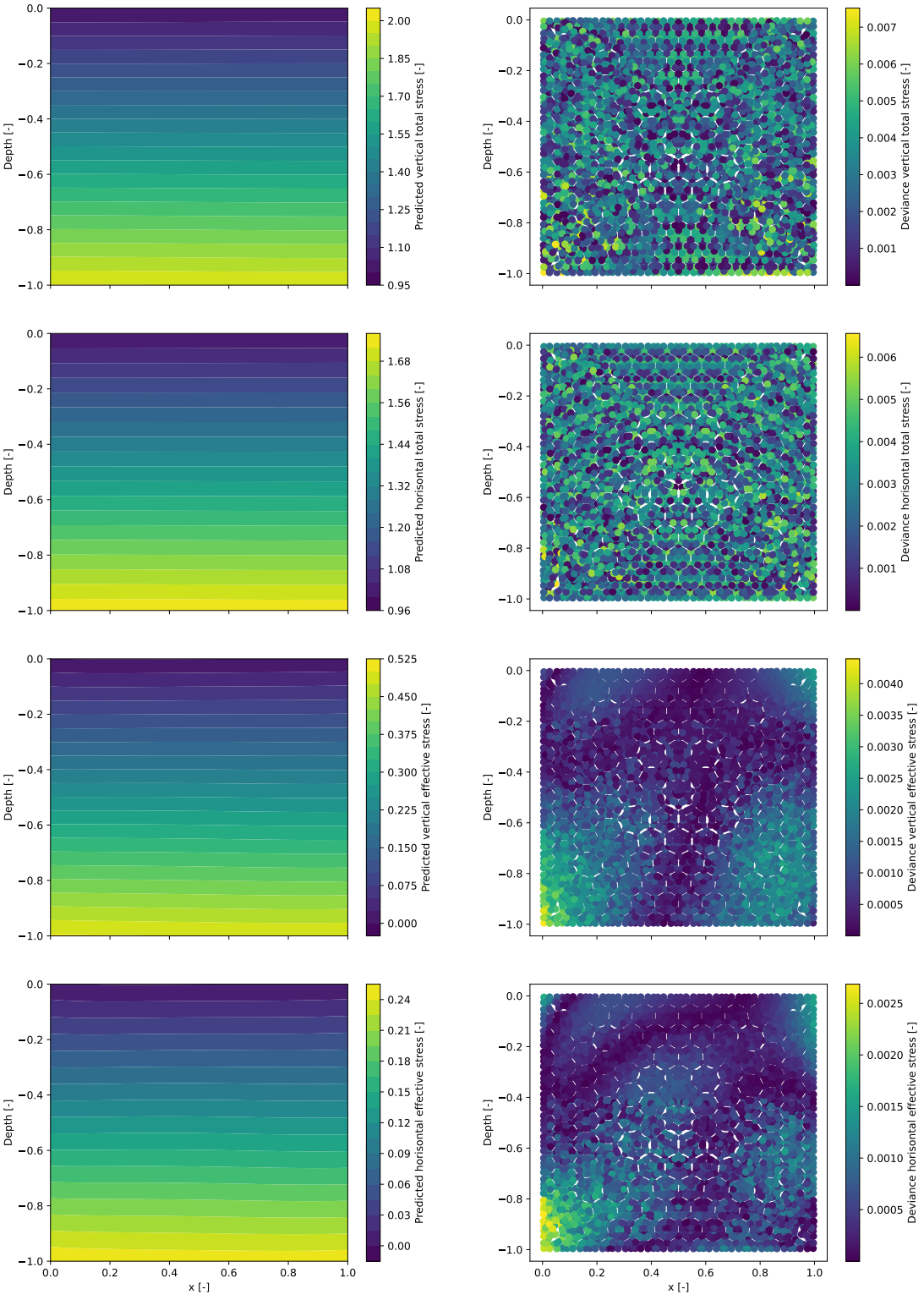
tive stress through the constitutive relation of effective stresses at rest and the difference between horizontal total stress and pore pressure. Since the stress initialization is defined by the gravitational body force in the vertical direction, it is natural to think of the causality or computational order as:

$$(\sigma_{zz}, \sigma_{xz}, u) \rightarrow (\sigma'_{zz}) \rightarrow (\sigma'_{xx}) \rightarrow (\sigma_{xx}) \tag{91}$$

However, the optimization algorithm in ANNs is not sequential, so all the dependencies should be multidirectional in the batch optimization. It is therefore interesting to see that the vertical total stress and shear stress seem to be so strongly influenced by the PDE, as is evidenced in the smoothness of the deviance plots. The horizontal total stress, which is also directly involved in a PDE, does not have a smooth deviance plot. It is likely that part of the explanation is that the contribution of the PDE in the gradient optimization is relatively large, thus overshadowing the other constraints. The inclusion of a body force in the vertical equilibrium equation can have skewed the bias in favor of optimizing this equilibrium equation before everything else. This can in many cases be solved by extending the training to reach a more optimal solution. Another possibility is to use adaptive weights, which can scale down the gradients of the overly sensitive components in the loss function.

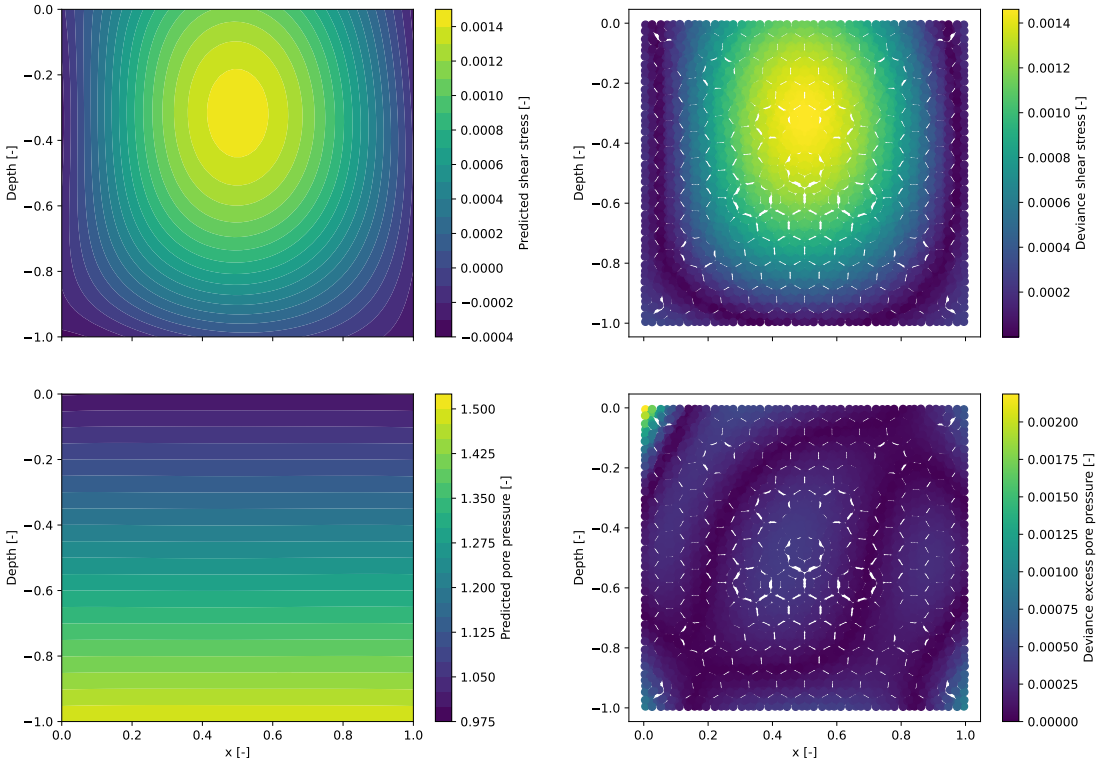
In general, the largest relative deviances seem to be in the corners, and all the deviance plots have symmetrical tendencies.

4.2.2 Phase 2 - undrained loading



(a) Model prediction

(b) Deviance from Plaxis results



(c) Model prediction

(d) Deviance from Plaxis results

Figure 50: Results from the undrained loading and comparison with Plaxis results. Each row is a stress or pressure component identified with the label of the color scale. The sign convention is that both compression and pore pressure is positive.

The results from the undrained loading of the soil layer are presented in Figure 50. The model prediction of the soil's response to the undrained loading is in general accurate, both in terms of absolute values and physical behavior.

The tendencies are the same as for the initial phase, which is natural since the only change is the introduction of a surface load and increased pore pressure. The values in the deviance plots increase a little more than the relative change in stress values, but the accuracy is still high. The irregularities in the corners are still prominent, especially for the pore pressure and the effective stresses.

4.2.3 Consolidation phase

The consolidation phase introduces a new dimension, so the presentation of the results will be structured differently. The relevant outputs for the vertical displacement are presented in their own plots. Each plot is structured in a row-column format, where the rows correspond to specific timesteps and the columns correspond to the computational method used. The contour plots are presented to show the physical behavior of the PINN prediction in comparison with the numerical methods, so the axis system of the plots has been removed for a more compact presentation.

The results from the prediction of the dissipation of excess pore pressure during the consolidation phase are presented in Figure 51. The model predictions are accurate for all three timesteps, both compared with FDM and Plaxis. Even though the Plaxis results are for coupled flow-deformation, the pore pressure levels are virtually identical. However, this is highly dependent on the magnitude of the deformation of the soil.

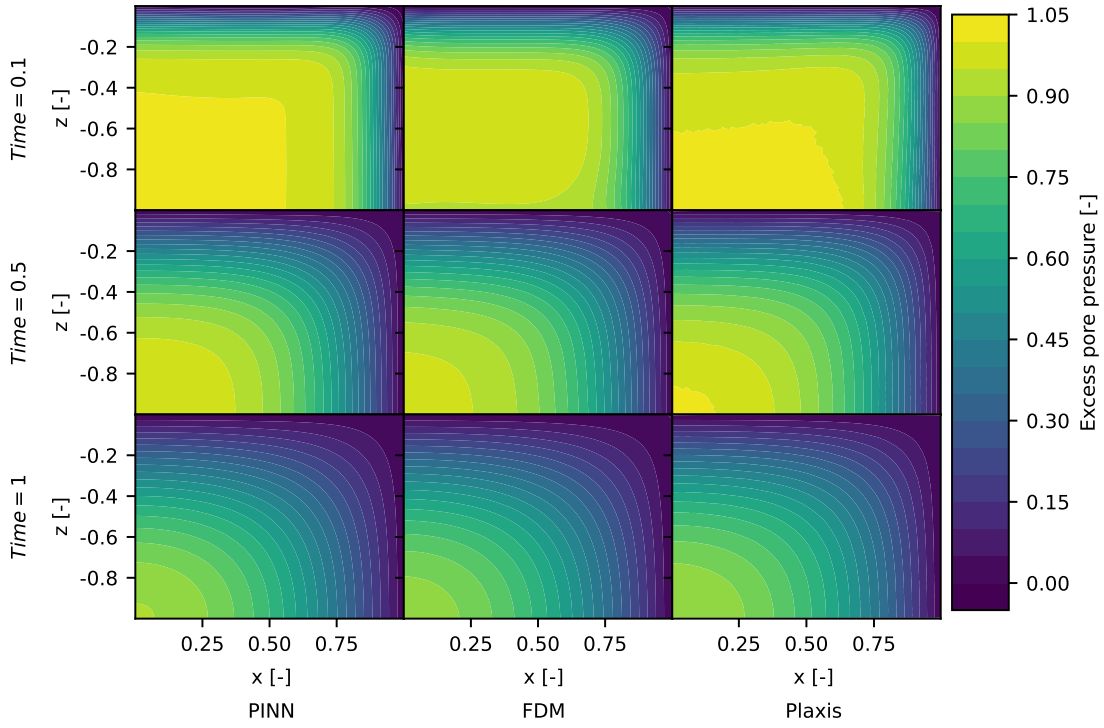


Figure 51: Plot of the excess pore pressure at different timesteps. The colorbar is representative of the pressure levels in all the subplots. Fluid pressure is considered positive.

The predictions of the excess pore pressures can be used to derive the effective stresses directly, because of the assumption of negligible horizontal strain and constant vertical total stress. The same equation for vertical strain as for the 1D problem can then be used to calculate the vertical strains in the entire 2D domain. The resulting vertical displacement can then be found by integrating the strains, starting from the bottom of the soil layer.

The resulting vertical displacement of the grid is illustrated in Figure 52 for the three timesteps. The resulting displacements of the PINN prediction and the FDM calculations are naturally much larger than the Plaxis results, due to the assumption of negligible

redistribution of stress. The conformity between the results obtained from the PINN model and the FDM calculations is naturally equivalent to that of the pore pressure.

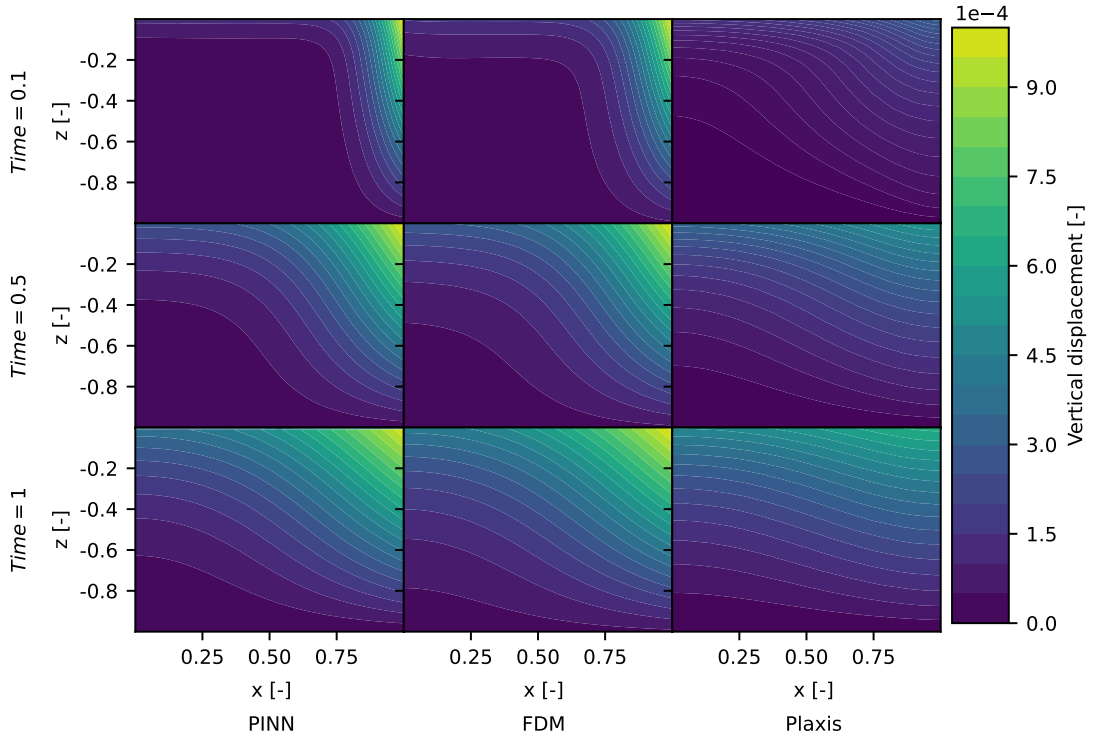


Figure 52: Plot of the vertical displacement at different time steps. The colorbar is representative of the pressure levels in all the subplots. Displacements in the direction of gravity are considered positive.

To compare the differences between coupled and uncoupled consolidation analysis, the predicted vertical effective stress is presented in Figure 53, along with the results from Plaxis. The deviance between the coupled and uncoupled problem was small for the pore pressure, but there are substantial differences in the calculated stresses. Along the right boundary, where the model is horizontally confined and drained, the deviance in vertical effective stress is almost 33%. This is due to the assumption of constant vertical stress, which is a fair assumption for the situation in the middle of the loaded area but is evidently less accurate at the edges of the strip load. The redistribution of total stress causes higher horizontal stresses and strains in this area, defying the modeling assumptions of both the uncoupled storage equation and the one-dimensional strain.

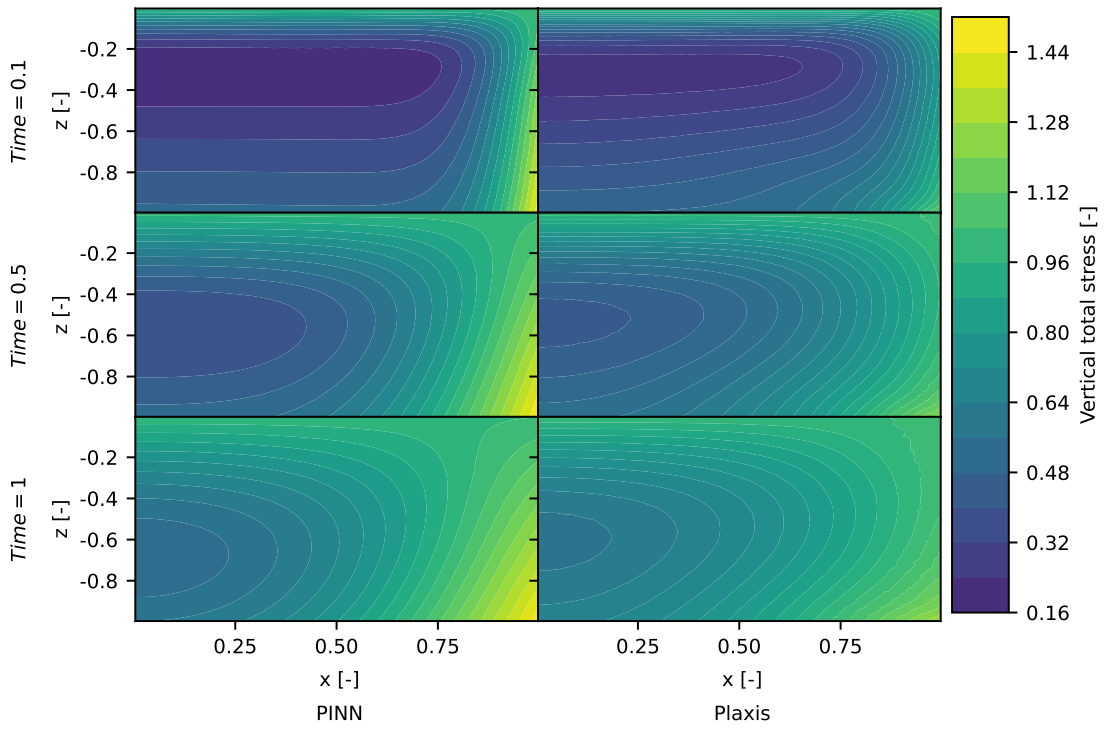


Figure 53: Plot of the vertical effective stress at different timesteps. The colorbar is representative of the pressure levels in all the subplots. Compression forces are considered positive.

The deviance in excess pore pressure and vertical effective stress between uncoupled and coupled consolidation is emphasized in 54. The deviance in the vertical stress is an order of magnitude larger compared with the pore pressure. The stress deviance is considerably smaller for $x = 0$ which is under the middle of the load when considering the full problem.

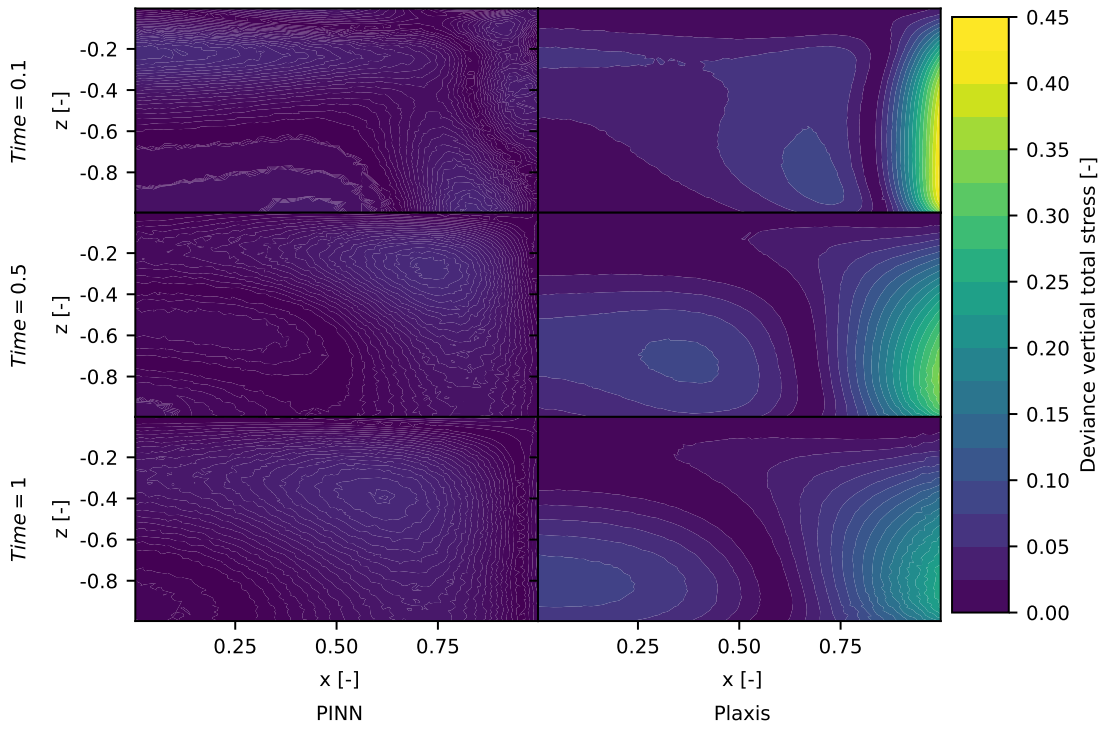


Figure 54: Plot of the deviance between the uncoupled and coupled calculation of excess pore pressure and vertical effective stress. The colorbar is representative of the deviance levels in all the subplots.

4.3 Coupled flow

The results from the coupled flow computation are structured in the same manner as for the consolidation phase of the uncoupled flow computation. The FDM results are, however, replaced by fully coupled flow-deformation computation with Plaxis. The initial stresses from the stress initialization in the uncoupled problem are reused for the computation of the new stress distribution. The contour plots are presented to show the physical behavior of the PINN prediction, whereas the line plots are included to emphasize the numerical values.

The results from the prediction of excess pore pressure for fully coupled flow show greater deviance than for the uncoupled problem. Figure 55 shows that the PINN model predicts a faster dissipation compared with both consolidation analysis and fully coupled flow-deformation in Plaxis. The behavior of the model prediction is closer to the results from the fully coupled soil model. The bottom left plot in Figure 55 shows that the model begins to defy the physical laws it has been trained on as time goes by. This is likely due to the fact that the absolute values of the gradients of both the governing equations and boundary conditions converge to zero. The model may therefore neglect to change its weights due to the small contribution to the global error, thus risking non-physical behavior. The contour lines show that only the lower and left boundary conditions are upheld.

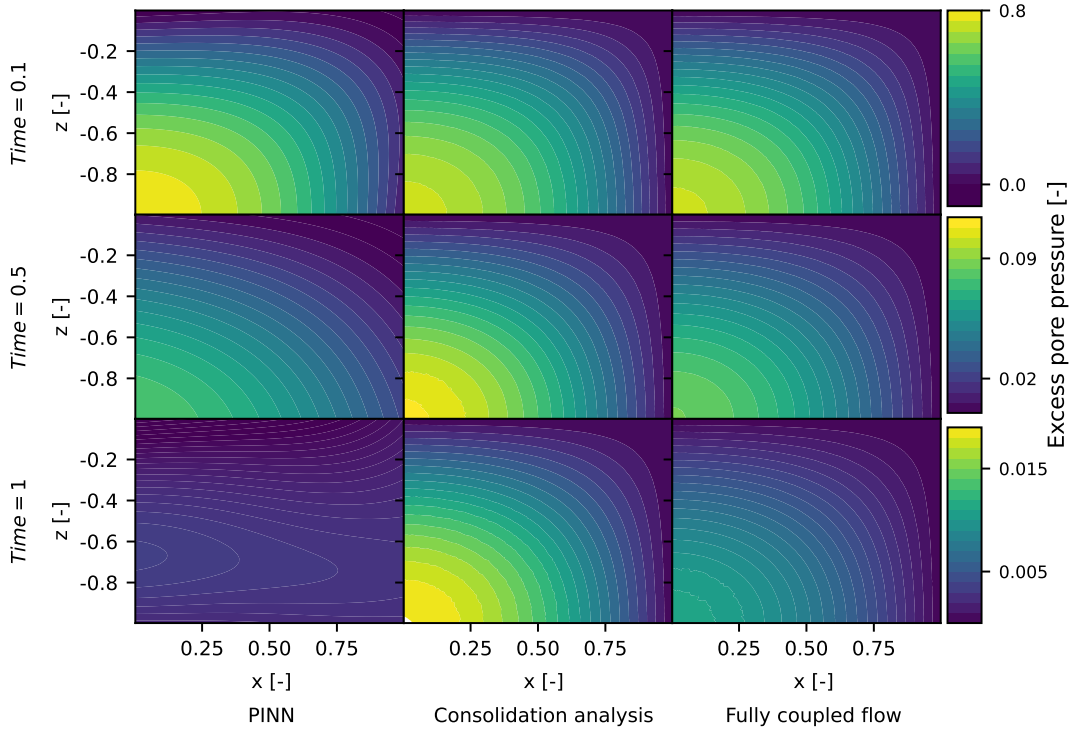


Figure 55: Plots of the excess pore pressure at different time steps with PINN and comparable numerical analysis with Plaxis. The colorbars are representative of the pressure levels for the respective timesteps. Fluid pressure is considered positive.

The plots of the deviances are shown in Figure 56. The deviance plots show that the PINN prediction is better than indicated by the slightly deceiving Figure 55. The deviance of the PINN model compared with the fully coupled flow model shows similarities to the deviance between consolidation analysis and fully coupled flow. Especially the plots of the last timestep show that the result from the fully coupled flow is approximately the average of the PINN prediction and the consolidation analysis.

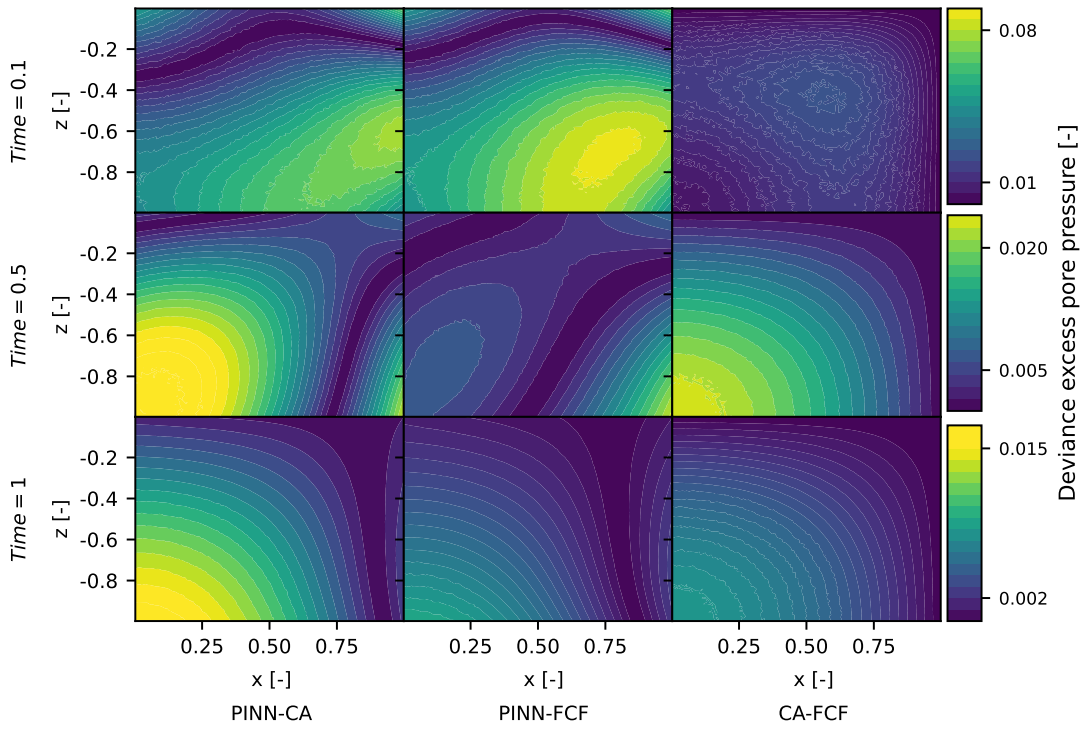


Figure 56: Plots of the deviances of the excess pore pressure calculations at different time steps between PINN and the numerical analysis with Plaxis. Abbreviations: CA = Consolidation Analysis, FCF = Fully coupled flow. The colorbars are representative of the deviance for the respective timesteps.

As can be seen in Figure 57, the model prediction is not so wrong as indicated in Figure 55. The choice of timesteps for the illustration was determined critically to emphasize the struggle areas of the PINN model, especially where the model starts to defy the physical constraints. The visualization in Figure 57 shows that the model is able to predict the initial increase in excess pore pressure at the middle of the lower boundary furthest away from the draining boundaries, which is called the Mandel-Cryer effect. However, the values are a little lower than the ones from the numerical solutions from Plaxis. The behavior over time is accurate compared with the fully coupled flow analysis, both for point $(0, -1)$ and $(0.5, -0.5)$. It is interesting that the biggest deviance is for the initial condition at the bottom, where the value for u is implicitly given by the relation $\Delta u = B\Delta p$. The model undershoots the initial pore pressure level at $(0, -1)$ and overshoots at $(0.5, -0.5)$, when they should be identical given the initial condition. Numerical instabilities at the corners, previously addressed during the initial phase of stress initialization, might explain the substantial deviance observed between the predictions of the two points. The corners may suffer from instabilities due to overlapping boundary conditions and being the extremal points of the PDE domain.

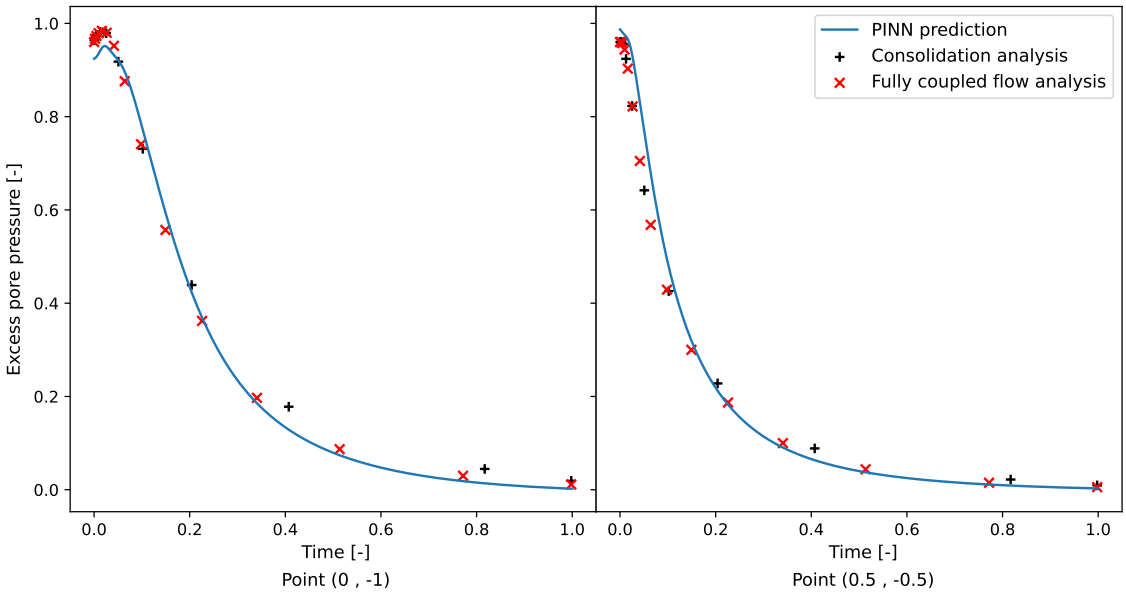


Figure 57: Plots of the excess pore pressure over time at two different spatial coordinates. Fluid pressure is considered as positive.

Figure 58 shows the predicted vertical total stress at $Time = 0.1$ and $Time = 0.5$. The last time step was omitted due to the small changes in total stress after $Time = 0.5$. The results show that the PINN model accurately predicts the redistribution of stress caused by the pore pressure dissipation. Similarly to the predictions of the excess pore pressure, the PINN model predicts a slightly faster dissipation and following stress redistribution than the FEM calculations with Plaxis. This is more prominent in the pore pressure prediction visualized in Figure 55, but can also be observed in the contour lines in Figure 58.

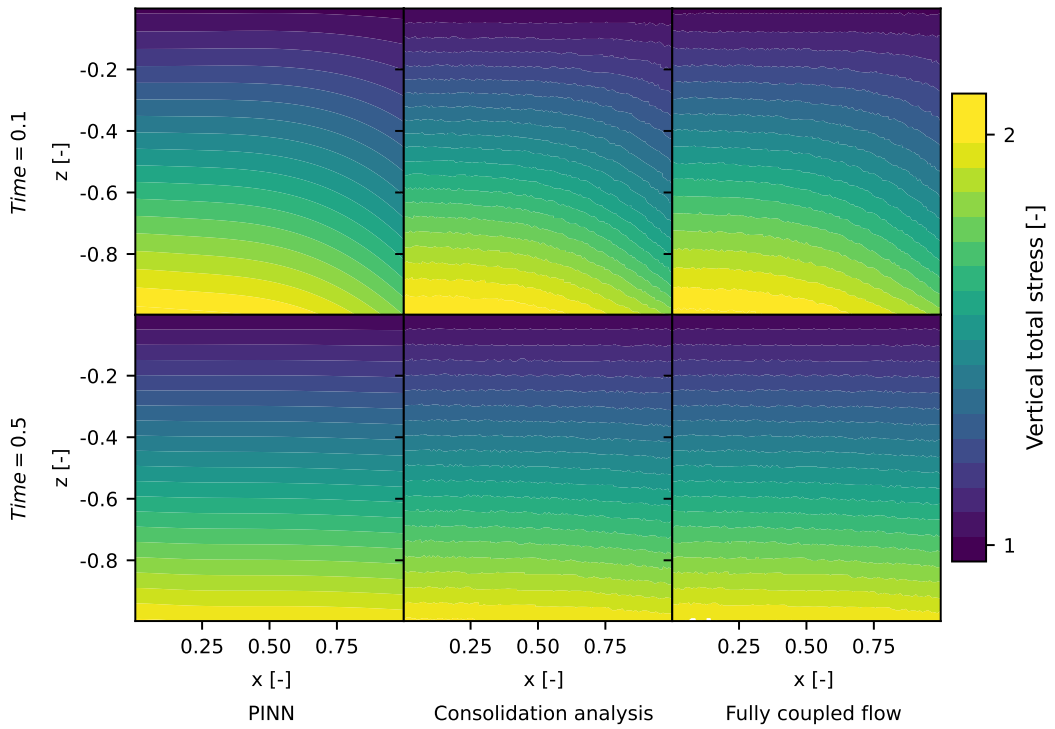


Figure 58: Plots of the vertical total stress at different time steps with PINN and comparable numerical analysis with Plaxis. Compression is considered positive.

The comparisons made in the chapter on uncoupled consolidation showed the importance of accurately estimating the redistribution of stress to model the soil behavior in the whole domain. The effect of redistribution can be further emphasized by subtracting the initial stress in the soil to isolate the change in stress from the external loading. Figure 59 shows the isolated change in stress due to external loading at the same time steps as the total stress in Figure 58. At $Time = 0.1$, there is a clear distinction in stress level between the middle of the soil layer and the lateral boundaries. The vertical total stress level in the middle is greater than the introduced surface load, whereas the vertical total stress level on the sides is lower.

Comparing the predictions of the PINN model with the Plaxis results show that the model predicts a slightly more pronounced difference between the middle and the boundaries, as shown in Figure 59. However, the physical behavior is captured in the same manner as the FEM solutions. The plots for $Time = 0.5$ show that the redistribution of stress has virtually converged to a steady state. A comparison with the calculated excess pore pressure levels for the same timestep, illustrated in Figure 55, shows that the convergence is expected since most of the excess pore pressure has dissipated by then, implying that the majority of the settlement has occurred.

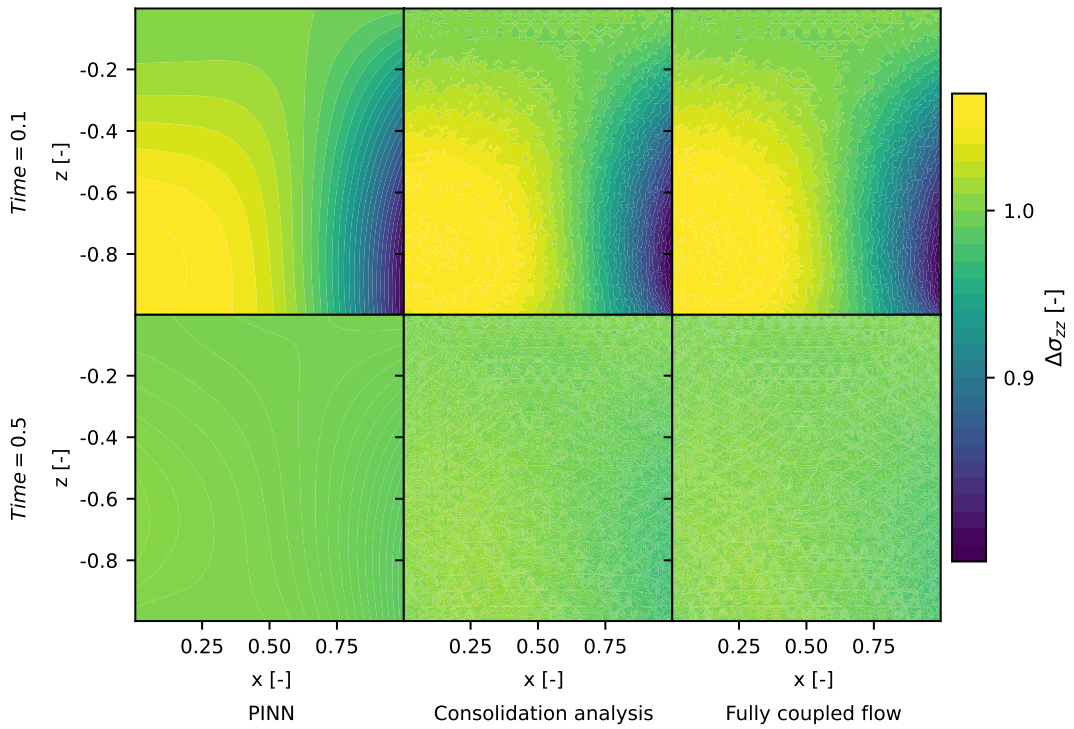


Figure 59: Change in vertical total stress due to external loading at two different timesteps. Compression is considered positive.

Figure 60 illustrates the results obtained from the model prediction of the vertical displacements. The accuracy of the displacement predictions appears to be comparable to that of the vertical stress. Notably, at $Time = 0.1$, the model predicts a slightly less prominent displacement at the drained boundary compared to Plaxis, which opposes the results from the simplified uncoupled model prediction. The change in prediction or calculation between $Time = 0.5$ and $Time = 1$ is marginal, confirming the indications from the results for excess pore pressure and total stress that the consolidation process has mostly converged by $Time = 0.5$.

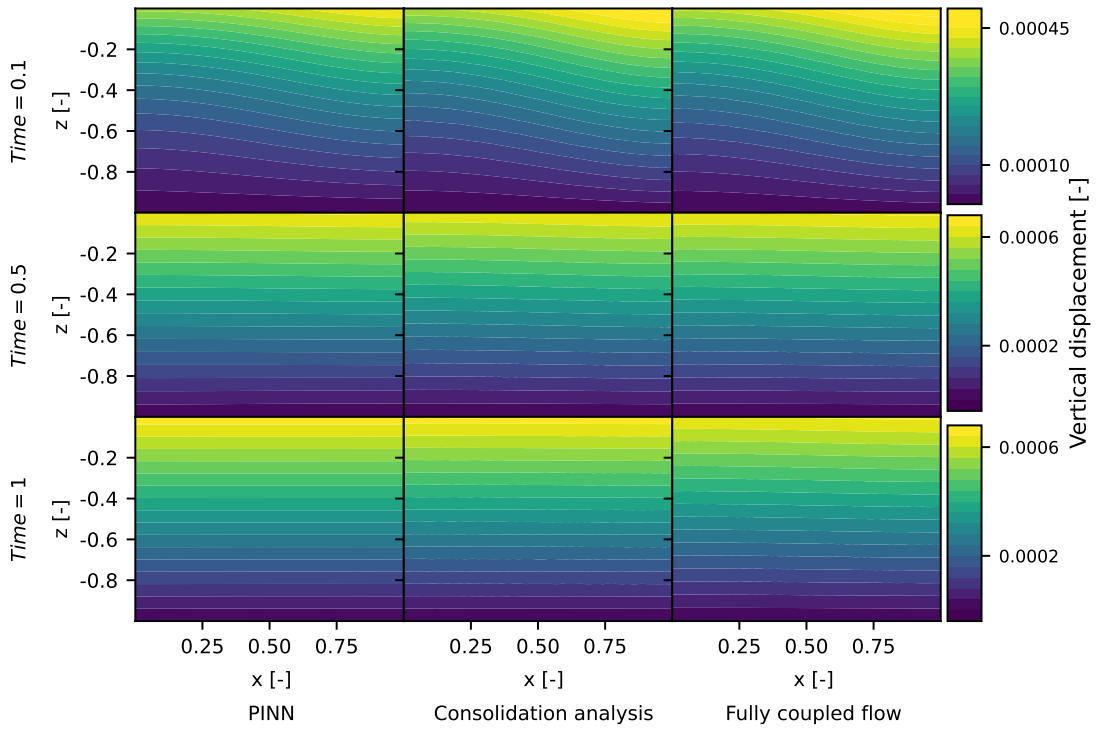


Figure 60: Plots of the vertical displacement at different time steps with PINN and comparable numerical analysis with Plaxis. Displacement in the direction of gravity is considered positive.

The deviances of the vertical displacement calculations are presented in Figure 61. Compared with the Plaxis calculations, the PINN model is the least accurate at $Time = 0.1$, where the model underestimates the vertical displacement. For $Time = 0.5$, the PINN prediction and the fully coupled flow calculation are almost equal, similar to the results from the excess pore pressure. At the last timestep, the deviance is the smallest between the consolidation analysis and the PINN prediction. By assessing the total figure, the subfigures showing the least deviance form a diagonal in the matrix representation. This implies that the behavior of the PINN model varies with time, both in regard to accuracy and physical interpretation.

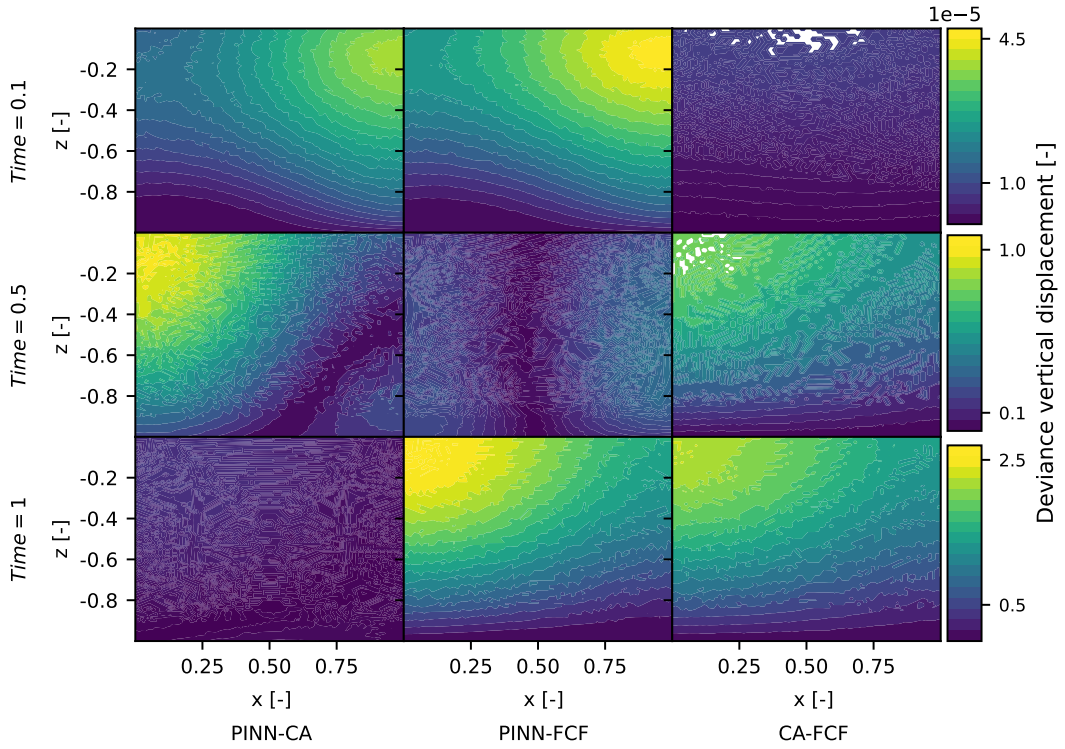


Figure 61: Plots of the deviances of the vertical displacement calculations at different time steps between PINN and the numerical analysis with Plaxis. Abbreviations: CA = Consolidation Analysis, FCF = Fully coupled flow. The colorbars are representative of the deviance for the respective timesteps.

Concentrating on the boundary points of the surface and the middle point of the domain enables a continuous illustration of displacement over time. Figure 62 shows that the predictions of the model show the same behavior as the Plaxis calculations, but are to some degree delayed in time. The rate of change in the model predictions is lower than for the Plaxis results for all three points. This trend was also apparent in the results of the vertical displacement in the 1D problem. The slightly higher deviance for point (1,0) compared with (0,0) is caused by this phenomenon since the rate of change for $t < 0.01$ is considerably higher at the drained boundary. The model is also able to encapture this effect, but the difference is less pronounced. The behavior of the convergence to the fully consolidated state is similar across all three methods.

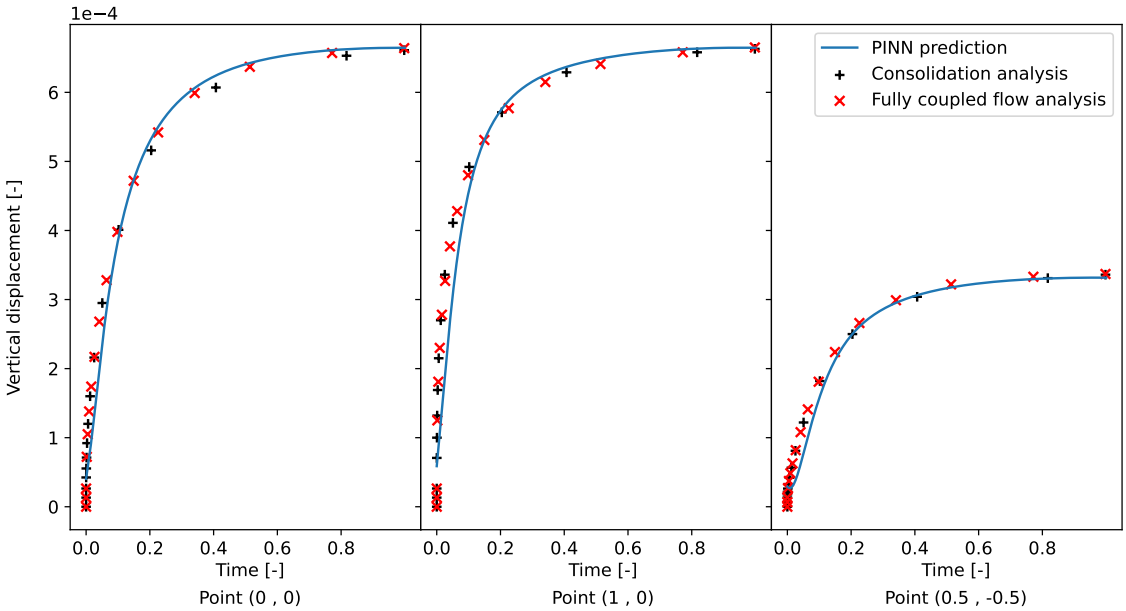


Figure 62: Vertical displacement at different time steps

The predictions of the horizontal displacements are presented in Figure 63 for three different timesteps. The horizontal displacements are small compared with the vertical displacement which greatly affects the accuracy of the model. The shape of the contour plot is reasonably accurate for $Time = 0.1$ and to some degree also $T = 0.5$, but the correct physical behavior steadily diminishes with the absolute values of the displacements. The physical constraint of no displacement along the bottom of the soil layer is contradicted for all three timesteps. For $Time = 1$ the contour lines reveal that the prediction contradicts the provided physical constraints at the lateral boundaries as well.

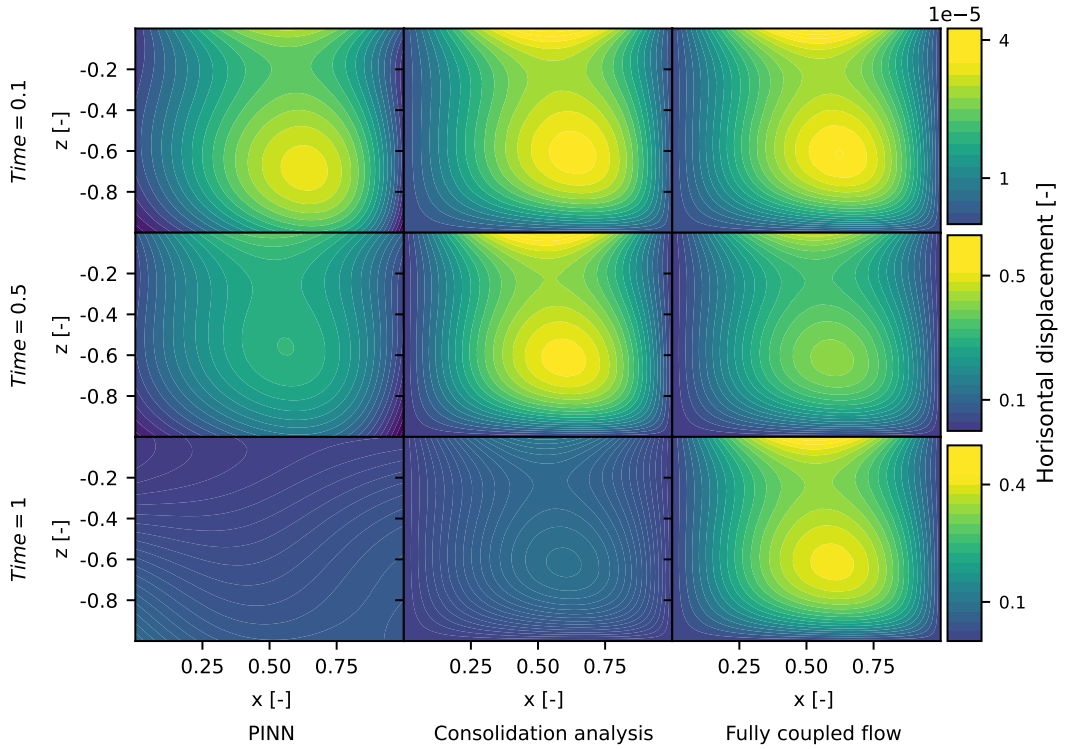


Figure 63: Vertical displacement at different time steps

Similarly to Figure 62 for the vertical displacement, Figure 64 shows the horizontal displacement over time at three selected spatial coordinates. The coordinates are at the boundary (0,0) where the horizontal displacement is supposed to be zero, the midpoint of the halved surface (0.5,0) where the horizontal displacement is relatively large, and the midpoint of the halved domain (0.5,-0.5) where there is a certain degree of horizontal displacement. The results show that the model predicts an initial horizontal displacement at the horizontally confined boundary. A similarly large, but mirrored, displacement is found at the opposite boundary, with the same horizontal confinement, for point (1,0). The prediction stabilizes at around zero displacement after roughly $Time > 0.1$. The prediction for point (0.5,0) shows that the relative error in the prediction of horizontal displacement is considerably larger than the relative error in the prediction of vertical displacement. The predicted maximum horizontal displacement is roughly 60% of the calculated maximum horizontal displacement from the Plaxis results. A possible cause of this deviance is the relatively small magnitude of horizontal displacement compared with vertical displacement and pore pressure. As previously discussed in the theory chapter and for the stress initialization phase, the optimization algorithm prioritizes to update the weights that contribute the most to the total loss of the prediction. So for finite training time, the model may be stuck on searching for a minima related to the biggest contributors to the loss, and barely learn the small contributors, like horizontal displacement in this case. A possible solution to this problem is the use of user-defined weighting of the loss functions, or alternatively adaptive weights. If the horizontal displacement is of primary interest, the loss function related to v_x may be weighted with a scalar to artificially increase the value of the loss related to the functional, thus directing the model to prioritize optimization of the weights in relation to horizontal displacement.

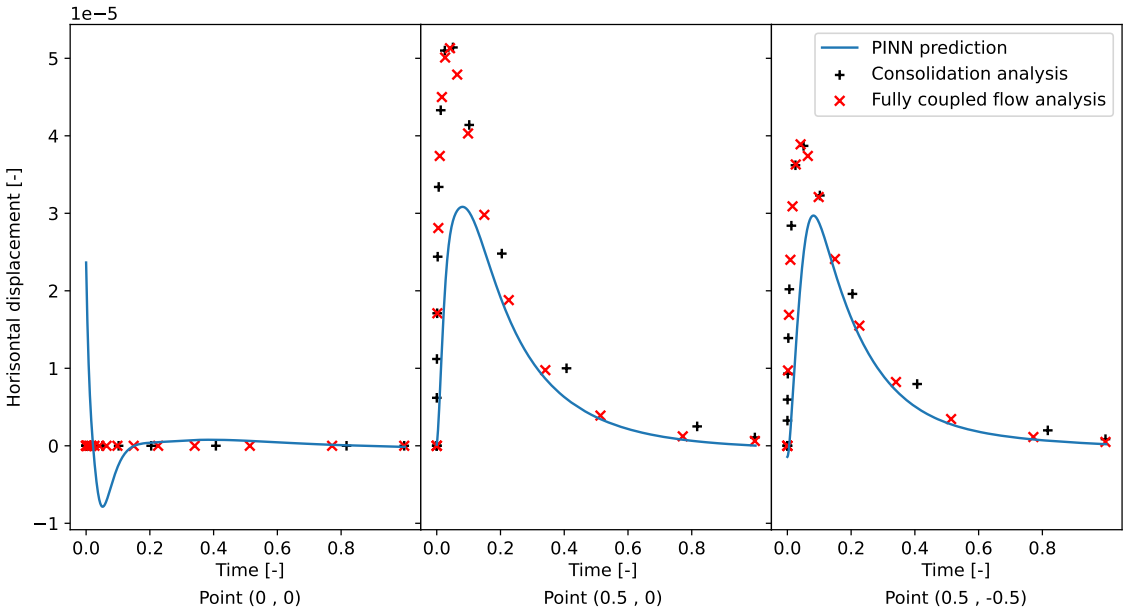


Figure 64: Vertical displacement at different time steps

4.3.1 Computational efficiency 2D

Table 4 shows an overview for comparison of computation time for the different methods used to obtain the results for the 2D problems. The structure is the same as for the 1D problem, as well as the limitation of the accuracy of the measurements due to fluctuations in available processing power.

Table 5: Computational efficiency 2D.

Task	Parameters	Time
Training time stress initialization	500 epochs, 2500 samples, 2 inputs	$63.66s$
Training time undrained loading	500 epochs, 2500 samples, 2 inputs	$156.20s$
Training time uncoupled consolidation	1000 epochs, 125000 samples, 3 inputs	$\approx 9000s$
Training time coupled consolidation	2000 epochs, 125000 samples, 3 inputs	$\approx 20.000s$
Prediction time stress initialization	2500 samples, 6 outputs	$0.024s$
Prediction time undrained loading	2500 samples, 6 outputs	$0.024s$
Prediction time coupled consolidation	12000 samples, 6 outputs	$1.02s$
Calculation time stress initialization	288 elements, 2401 nodes, Default iteration parameters	0.0
Calculation time undrained loading	288 elements, 2401 nodes, Default iteration parameters	$1.53s$
Calculation time consolidation analysis	288 elements, 2401 nodes, Default iteration parameters	$4.15s$
Calculation time fully coupled flow	288 elements, 2401 nodes, Default iteration parameters	$3.91s$
Calculation time FDM uncoupled	Discretization: $50x50$ in space, 20.000 in time	$533.88s$

The results for the computational efficiency of the 2D problems show much of the same as for the 1D case. However, the introduction of Plaxis to the list showcases the efficiency of their numerical solvers. With default iteration parameters, the calculations are done for relatively large time steps, so in contrast with PINN and FDM, the Plaxis calculation only stores the results for a few points in time. This numerical efficiency is very beneficial for forward solving, but for inverse problems where measurements are received frequently over time, this is less relevant to some degree.

5 General discussion

The general discussion will try to relate the obtained results from the result and discussion chapter to the research questions and limitations presented in the introduction chapter. A more focused discussion of individual results can be found in the results and discussion chapter. The interpretability of the model is discussed parallel with the accuracy due to their close relation.

5.1 Accuracy

The first research question is related to the accuracy of the model, and how it compares with the alternative traditional methods. The results from the Terzhagis problem show that the PINN is able to accurately predict the physical behavior of a 1D diffusion problem with mixed boundary conditions. Compared with the analytical solution, see Figure 16, the accuracy can be deemed as on par and fully competitive with traditional methods for $time >> 0$. For early time steps, the accuracy of the PINN is negatively affected by the overlapping boundary and initial conditions, in contrast to the numerical methods. However, the approximation of the analytical solution to the problem is also fairly inaccurate for small-time steps, making the comparison with the numerical methods slightly more relevant in the early stages. In a comparable study of PINN performance on a 1D consolidation problem by Bekele (2020), the results showed a similar pattern of elevated error for early timesteps. However, the primary objective of this study was not uncertainty quantification, so the deviance for the early timesteps was not highlighted in the same manner.

For the case of the forward problem, an argument can be made that the early stages of a consolidation problem are less important than the long-term prediction. For the prediction of final settlement and time of consolidation, the model is as accurate as numerical methods. Which is worth noting for the important overarching question of applicability. An interesting and important observation is that the accuracy increases considerably when infusing Gaussian white noise into the model, running Monte Carlo simulations, and estimating the mean simulation. A 50% decrease in deviance makes the mean prediction with Gaussian noise more accurate than the numerical approximations for $time >> 0$. The fact that it reduces the total deviance is one thing, but the deviance reduction for the first time steps is quite interesting since the initial pore pressure level is kept constant. This can be caused by the fact that PINNs are solved as an optimization problem. In contrast to the numerical methods, where the calculation flow is in one direction, the predictions in PINNs during training affects the whole training domain. So new predictions for later timesteps may propagate backward in time and update the prediction for early timesteps. This calculation flow is important to keep in mind when dealing with uncertainty, because errors may propagate through time and space both forward and backward.

For the inverse problem, the accuracy for the early timesteps becomes much more relevant. The results from the parameter optimization and MLE show that the parameter estimation for early timesteps is greatly affected by the lacking accuracy, thus leading to poor predictions of future behavior. However, by utilizing previous knowledge of the problem's nature and model behavior, the global error due to lack of accuracy for the early timesteps can be reduced. For instance, altering the optimization algorithm to prioritize updating the unknown parameters based on measurements from domain points where the data and the prediction are in agreement. The results also show that increased accuracy can be

obtained by considering alternative or even multiple measurement types. The accuracy of MLE with displacement measurements outperformed the equivalent algorithm with piezometer measurements, and combining the two made inferences of both E_{oed} and k_z possible simultaneously due to conditional independency given C_v, where the conditional independence is based on the problem and modeling assumptions.

The model prediction of the 2D diffusion problem, where the storage equation is uncoupled by assuming negligible horizontal strains and constant vertical total stress, is similarly accurate as for the 1D problem. The results from the predictions of the initial stress state show that the PINN model is able to learn the basic equilibrium equations from continuum mechanics, again completely independent of data. The deviations from the Plaxis model are the largest for the unknown physical quantities that are not directly tied to the PDE, which may be due to inherent bias in the optimization algorithm. For soil mechanics purposes, the boundary conditions, initial conditions, and constitutive relations between the physical quantities are equally important as the PDE. So for PINNs to be applicable as numerical solvers in problems related to soil mechanics, a comprehensive approach needs to be established to ensure an even holistic treatment of the PDE and all the other physical constraints. Utilizing adaptive weights, as in the research from Haghighat et al. (2022), is a possible solution that evens out the magnitudes of the gradients in a neural network. However, this may slow down the training process considerably due to the additional numerical operation for each epoch. Another approach is to utilize previous domain knowledge by feeding the model with more training data from the boundaries. However, this approach essentially introduces a new bias to counterbalance the previous bias, which may be unfortunate for the ability of the model to generalize its knowledge for use in other domains. A similar, less computationally expensive solution is to simply utilize a user-defined scaling of the loss functions

$$\mathcal{L} = \alpha_1 \mathcal{L}_{bc} + \alpha_2 \mathcal{L}_{ic} + \alpha_3 \mathcal{L}_{pde} + \alpha_4 \mathcal{L}_{data} \tag{92}$$

This was done for the 1D consolidation problem, where the loss term related to the initial condition was amplified in an attempt to increase the accuracy for the early timesteps. The deviance of a biased model was slightly lower than an unbiased model compared with the analytical solution. The differences between the models with different loss configurations diminished for longer training time, and a perfectly trained model should be independent of the scaling. However, the global minimum is seldom achieved, so in realistic scenarios, the model behavior may be slightly different.

The advantage of user-defined scaling is the ability to prioritize certain aspects of the problem that are of special interest or highlight irregularities or nonlinearities. For instance, if the horizontal displacements in the 2D soil layer are of great interest, the loss connected to the constraints of the horizontal displacement may be scaled by a factor reflecting the difference in magnitude between vertical and horizontal stress. As shown in the results for the coupled 2D problem, the model prediction of vertical displacement is substantially more accurate than the prediction of horizontal displacement. This may be due to the fact that the vertical displacements are an order of magnitude larger than the horizontal displacements, thus prioritized in the weight and bias optimization. Amplifying the losses connected to the lateral confinements and the equilibrium equation for the x-direction could therefore benefit the accuracy of the prediction of horizontal displacement. However, this may affect the accuracy of the prediction of the other physical quantities, and should therefore be done critically and only for instances where the small quantity is of primary interest. Usually, the largest quantities of the stresses, strains, and displacement are of primary interest, like the vertical displacement of the surface in consolidation analysis.

Due to the normalization of the quantities in the nondimensionalization, the issue with model bias is mainly apparent between directional components of different magnitudes within the same tensor.

The overall results from the coupled consolidation problem showed that the PINN model was able to predict the behavior of coupled PDEs, even without data. The model was able to learn the general tendencies of all the relevant stresses and displacements, some more accurately than others. However, compared with advanced commercial numerical solvers, the accuracy is considerably lower and the behavior is less stable. The problem was not solved with an optimized model, so significant improvement in both accuracy and efficiency may be obtainable by tweaking the model's architecture and hyperparameters. This is on the other hand also one of the drawbacks of the methodology. The accuracy is dependent on what is to some extent random choices of parameters and network size. A parallel may be drawn to FEM and the choice of element type and mesh size, which both the accuracy and convergence rate depend on. Similarly, deeper neural networks are usually associated with higher accuracy but demand greater computational power to train.

The total uncertainty of a model is the sum of the Aleatoric and the Epistemic uncertainty. The results in this thesis show that PINN models contribute to increased Epistemic uncertainty, compared with commercial FEM, due to its inferior accuracy in predicting the correct physical behavior. So for deterministic problems, uncertainty quantification with PINNs will probably lead to a higher total uncertainty due to this inherent error. Nonetheless, the results from the predictions on the independent test grids show that a well-trained PINN model is able to predict independently of the training grid. So for deterministic tasks like parametric design where interpolation and extrapolation independent of grid or mesh is beneficial, PINNs can prove valuable even for deterministic tasks.

5.2 Efficiency

The results from the evaluation of computational efficiency showed that the PINN model is considerably faster than the other numerical methods. Even for the low-dimensional problems explored in this project, the differences are considerable. For the 1D probabilistic forward problem, where the C_v parameter was treated as a random variable, estimating the mean and standard deviation of the predictions involves Monte Carlo simulations over the sample space. Monte Carlo simulations in general involve solving variants of a problem repeatedly in an iterative process storing the outcomes for analysis of the statistical moments of the target distribution. So even for relatively simple problems, like the 1D consolidation problem with one random variable, the computation time may be significant if the sample space of the random variable is large. The sample space is usually dependent on a prior belief of the underlying distribution, so according to Bayesian statistics, a stronger prior belief will reduce the sample space and consequently reduce the computational cost. However, this requires prior knowledge of the underlying distribution which is not always obtainable due to inherent randomness or practical reasons.

By extrapolating the time per prediction with the number of necessary predictions to cover the sample space, the computational time would have been $0.14s * 100000 \approx 3.89h$ with the PINN model, $1.58s * 100000 \approx 43.89h$ for FEM, $2.23s * 100000 \approx 61.94h$ for FDM and $1.58s * 100000 \approx 84.44h$ with the approximation of the Fourier series. These differences showcase the efficiency of matrix operations in Python compared with nested loops. FEM and PINN are drastically faster than FDM and Fourier series approximation due to this

fact. The reason PINN is faster than FEM is the time integration scheme necessary for calculating transient problems with FEM. For stationary problems, FEM will be equally fast, thus diminishing the main advantage of PINNs for forward problems. It is worth noting that these times are extrapolated for a larger-than-necessary solution space, and that there exist efficient sampling techniques, like MCMC used in the inverse problem, to reduce the number of necessary calculations.

Summarized, the results show great promise for probabilistic problems, where the contribution of Aleatoric uncertainty to the total uncertainty is significant. The PINN model proves its effectiveness in exploring the target distribution of the stochastic PDE more efficiently, enabling better mapping of the aleatoric uncertainty compared with FEM for the same calculation time.

Another computationally demanding task is parameter estimation in the inverse problem. The results show that the prediction time is unaffected by an additional input parameter, which is highly beneficial for the efficiency of the MLE for use in multi-dimensional parameter spaces. However, when the dimensions of the parameter space increase, the necessary computations of MLE increase exponentially

$$\begin{aligned} F_{X1}(X_1) &= \int_{X1} f_{X1}(x_1)dx_1 \\ F_{X_1,X_2}(X_1,X_2) &= \int_{X_1,X_2} f_{X_1,X_2}(x_1,x_2)dx_1dx_2 \\ &\vdots \end{aligned} \tag{93}$$

By fixing X_1 and evaluating for all X_2, and vice versa, the MLE of two independent random parameters can be solved with brute force MCMC. Alternatively, the optimization problem can be solved by gradient descent in the PINN algorithm. The latter is considerably more computationally efficient and effectively facilitates for computations with multiple random variables, which until recently demanded substantial computational power for most problems.

Due to the findings in regard to computational efficiency, a possible beneficial framework may be to use PINNs for a preliminary assessment of the solution space. The PINN model's predictions can be used to receive an initial first-order approximation of the problem and possibly be able to reduce the parameter space by dimensionality reduction or updating the prior belief of the underlying parameter distributions. As seen from the results from the MLE and parameter optimization in the inverse problem, the PINN model is able to effectively update the prior distribution of the unknown parameter, even at early time steps. The model is then able to utilize the updated belief to make new predictions, all in a matter of seconds.

6 Concluding remarks

In conclusion, the results of this work show that PINNs offer a promising alternative to computationally expensive simulations of probabilistic numerical models. Although the accuracy was shown to be inferior compared with FEM, the computational efficiency and generality of the PINN approach offer efficient integration with data and measurements ideal for uncertainty quantification and parameter estimation in inverse problems. Although not explicitly proven in this project due to limitations in processing power, combining the proven efficiency from the parameter estimation and the acceptable accuracy from the coupled flow-deformation problem, indicates that PINN models are capable and efficient in performing high-dimensional optimization tasks related to poroelasticity and soil mechanics in general. While this methodology may be redundant for forward deterministic modeling, the prospect seems promising for multi-dimensional parameter estimation and simulation for statistical analysis of stochastic PDEs.

7 Recommendations for further work

There are several ways to explore the capabilities and applicability of PINNs further, both in relation to poromechanics and soil mechanics in general.

- Geometry: The work in this project has only dealt with idealized geometries, which limits the applicability to a narrow range of problems. To prove the PINN model's ability to learn the physical behavior of an arbitrary problem domain, performance tests for problems with more than one soil layer, varying depths to bedrock, or inclined surfaces could be investigated.
- Advanced soil models: Linear stress-strain relationship has been assumed in this project work, which is unrealistic for many problems in soil mechanics. Therefore, the PINN model's ability to learn the stress-strain relation and unloading-reloading characteristics of elastoplastic soil models is of interest for applicability in advanced soil modeling. Both in problems with soil-fluid interaction, soil-structure interaction or just focusing on soil behavior for an arbitrary loading.
- Periodic loading: The PINN model's ability to predict the behavior of soils subjected to cyclic loading could be investigated. This could be done in combination with elastoplastic soil models, to see if the model is able to predict long-term elastoplastic deformation patterns due to for instance cyclic loading.
- Increased dimensionality: PINNs capability and efficiency in solving high-dimensional optimization problems for design optimization and uncertainty quantification could be investigated, note that this will most likely demand extensive computational power.

Bibliography

Bekele, Y. (2020). Physics-informed deep learning for one-dimensional consolidation. *Journal of Rock Mechanics and Geotechnical Engineering*, 13.

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer.

Box, G. E. P. and Tiao, G. C. (2011). *Bayesian Inference in Statistical Analysis*. John Wiley & Sons. Google-Books-ID: T8Askeyk1k4C.

Brownlee, J. (2018). What is the difference between a batch and an epoch in a neural network. *Machine Learning Mastery*, 20.

Cheng, A. (2016). *Poroelasticity*, volume 27.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335. Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/00031305.1995.10476177.

Churchill, R. V. (1971). *Operational Mathematics*. McGraw-Hill College, New York, 3rd edition edition.

CRYER, C. W. (1963). A COMPARISON OF THE THREE-DIMENSIONAL CONSOLIDATION THEORIES OF BIOT AND TERZAGHI. *The Quarterly Journal of Mechanics and Applied Mathematics*, 16(4):401–412.

Engelbrecht, A. (2007). *Computational Intelligence: An Introduction, Second Edition*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Haga, J. M. (2022). A general overview of ai, and its applicability in hybrid modelling for geotechnical problems.

Haghighat, E., Amini, D., and Juanes, R. (2022). Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training. *Computer Methods in Applied Mechanics and Engineering*, 397:115141. arXiv:2110.03049 [cs].

Haghighat, E. and Juanes, R. (2021). Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 373:113552.

Irgens, F. (2008). *Continuum Mechanics*. Springer, Berlin, Heidelberg.

Kim, J., Tchelepi, H. A., and Juanes, R. (2011). Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1591–1606.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kurz, S., De Gersem, H., Galetzka, A., Klaedtke, A., Liebsch, M., Loukrezis, D., Russenschuck, S., and Schmidt, M. (2022). Hybrid modeling: towards the next level of scientific computing in engineering. *Journal of Mathematics in Industry*, 12(1):8.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100.

Nordal, S. (2020). Geotechnical Engineering Advanced Course.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations.

Rau, G., Post, V., Shanafield, M., Krekeler, T., Banks, E., and Blum, P. (2019). *Error in hydraulic head and gradient time-series measurements: a quantitative appraisal*. Journal Abbreviation: Hydrology and Earth System Sciences Discussions Publication Title: Hydrology and Earth System Sciences Discussions.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Pearson, Upper Saddle River, 3rd edition edition.

Skempton, A. W. (1984). *Selected Papers on Soil Mechanics*. Thomas Telford. Google-Books-ID: 60EYMuPunzsC.

Smith, R. C. (2013). *Uncertainty Quantification: Theory, Implementation, and Applications*. SIAM. Google-Books-ID: 4c1GAgAAQBAJ.

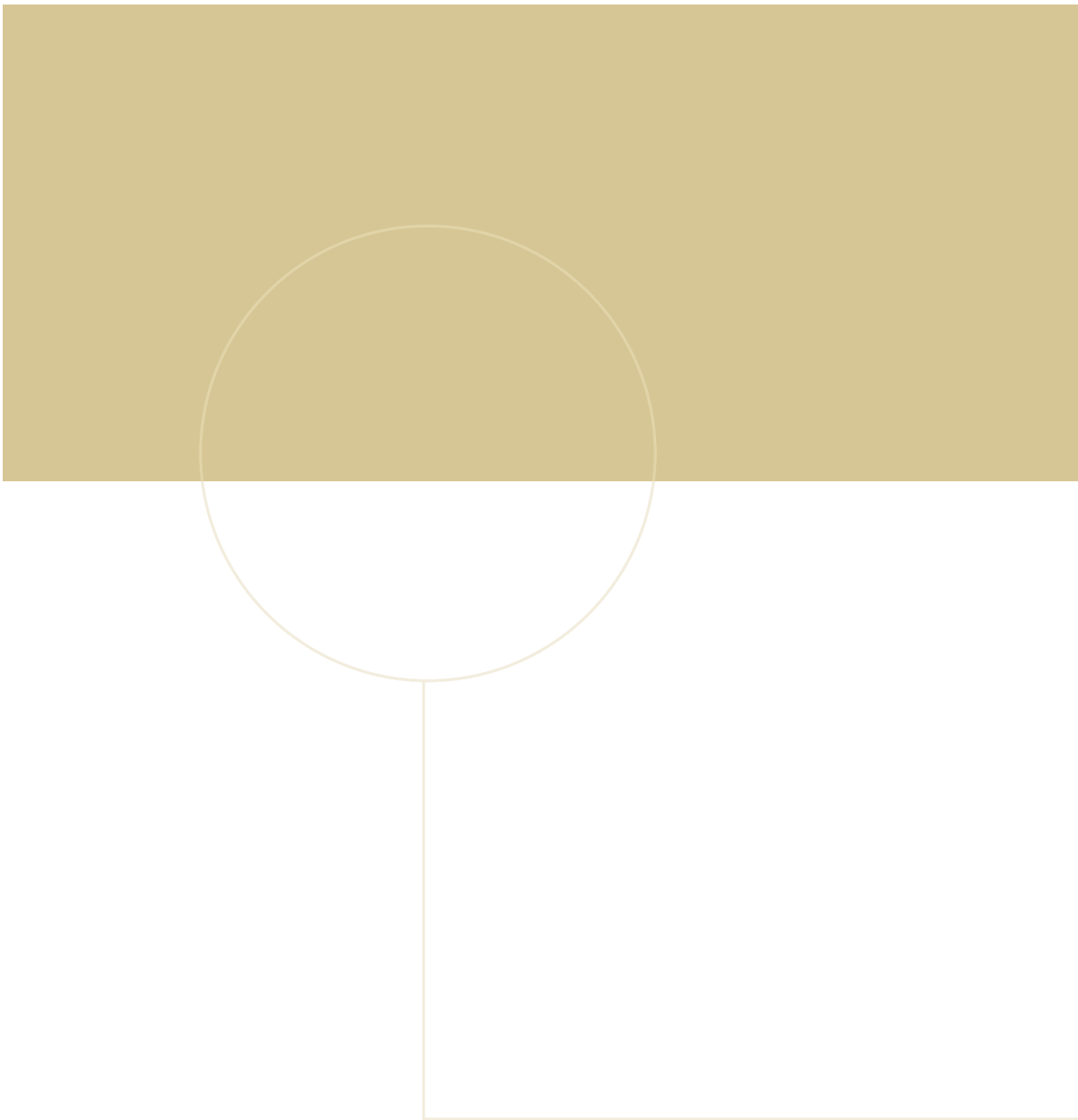
Verruijt, A. (2016). *Theory and problems of poroelasticity*. Delft University of Technology.

Whitaker, S. (1986). Flow in porous media I: A theoretical derivation of Darcy's law. *Transport in Porous Media*, 1(1):3–25.

Wu, B., Hennigh, O., Kautz, J., Choudhry, S., and Byeon, W. (2022). Physics Informed RNN-DCT Networks for Time-Dependent Partial Differential Equations. arXiv:2202.12358 [physics].

Xu, Y. and Goodacre, R. (2018). On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of analysis and testing*, 2(3):249–262.

Yang, L., Meng, X., and Karniadakis, G. E. (2021). B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913.



NTNU

Norwegian University of
Science and Technology