Prediction of Ground Motion for Metro Station Construction Predicting Ground Motion for Metro Station Excavation

Emmanouella Makrymanolaki

June 15, 2025

Abstract

This report summarizes the data processing performed and provides recommendations for predicting ground motion during metro station excavation, leveraging advanced computational techniques. The methodology presented offers a robust approach for ensuring the safety and stability of metro station construction and can be adapted for similar applications. This paper presents a detailed methodology for predicting ground motion during the construction of a metro station using data acquired through ground excavation. The methodology includes data preprocessing, model selection, training, evaluation, and prediction using Neural Partial Differential Equations (PDEs). The technical details and governing equations are discussed in depth.

Introduction

The construction of metro stations involves significant ground excavation, which can lead to ground motion and potential risks to surrounding structures. Accurate prediction of ground motion is crucial for ensuring the safety and stability of the construction site and its vicinity. This paper outlines a robust approach using Neural PDEs to model and predict ground motion. I give a short summary of the work.

- *Data Description* The dataset comprises measurements of ground motion of the walls of the trench at various depths and times. Key variables include:
  - Time (t): The timestamp of each measurement.
  - Coordinates (x, y, z): The spatial coordinates of the measurement points on the trench walls.
  - Deformation (dX, dY, dZ): The corresponding deformation values in three dimensions.
- *Methodology*
  - *Data Loading and Preprocessing* The data is loaded from an Excel file into a pandas DataFrame. Preprocessing steps include handling missing values, normalizing the data, and splitting it into training and testing sets.
  - *Model Selection* A Neural Partial Differential Equation (PDE) model is selected for predicting ground motion. Neural PDEs are suitable for capturing the complex dynamics of ground motion in an elastic material.
  - *Governing Equations* The motion of an elastic material is governed by the Navier-Cauchy equations, which can be written as:
$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda + \mu)\nabla(\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{1}$$
    where $\mathbf{u}$ is the displacement vector, ρ is the material density, λ and μ are Lamé parameters, and $\mathbf{f}$ is the body force vector. For a linear elastic material, the stress-strain relationship is given by Hooke's law:
$$\sigma_{ij} = C_{ijkl}\epsilon_{kl}, \tag{2}$$
    where σ_{ij} is the stress tensor, C_{ijkl} is the stiffness tensor, and ϵ_{kl} is the strain tensor.
  - *A priori estimates* Combining energy estimates from the governing equations, standard as well as novel PDE techniques I derive localized estimates for the solutionm of the *Navier-Cauchy* system that I incorporate in the loss function
  - *Model Training* The Neural PDE model is trained using the training dataset. The training process involves optimizing the model parameters to minimize the prediction error. The loss function for the training is defined as the Mean Squared Error (MSE) between the predicted and actual deformation values.

  - *Model Evaluation*
    The model is evaluated using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on the testing dataset. These metrics provide insights into the model's accuracy and performance.

- *Results* The trained model is used to predict ground motion for new data points. The predictions are compared against actual values to assess the model's effectiveness. Visualizations such as plots and graphs are included to illustrate the results.

- *Discussion* The paper discusses the implications of the model's predictions and their significance for metro station construction. The use of Neural PDEs allows for capturing the complex dynamics of ground motion, providing a robust approach for ensuring safety and stability. Potential improvements and future work are also outlined, including the incorporation of additional data sources and advanced machine learning techniques.

- *Conclusion* Predicting ground motion using Neural PDEs offers a robust approach for ensuring the safety and stability of metro station construction. The methodology presented in this paper can be adapted and extended for similar applications in other construction projects. Future work will focus on enhancing the model's accuracy and exploring its applicability to different construction scenarios.

Data Extraction and Relevance

The dataset for predicting ground motion comprises measurements acquired through ground excavation, specifically from the walls of the trench at various depths and times. The provided data, originating from the "Xaidari St._AGS files.xlsx" file (specifically "Sheet1.csv"), contains critical information for understanding ground behavior.

The following columns were extracted and are fundamental for analyzing and predicting the time evolution of ground deformation:

- Column E (`PROJ_CONT`): Corresponds to Time (t), the timestamp of each measurement.

- Columns G (`PROJ_MEMO`), H (`PROJ_DATE`), I (`PROJ_CID`): These represent the spatial coordinates (x, y, z) of the measurement points on the trench walls.

- Columns J (`PROJ_PROD`), K (`PROJ_RECV`), L (`?PROJ_ISNO`): These capture the corresponding *deformation values* (dX, dY, dZ) in three dimensions at the measured points.

Accurate prediction of ground motion is crucial for ensuring the stability and safety of the metro station excavation site and surrounding structures.

Predicting Ground Motion using Neural Partial Differential Equations (PDEs)

To predict the time evolution of deformation, particularly considering the material properties of the ground and the principles of elasticity, *Neural Partial Differential Equations (Neural PDEs)*, especially *Physics-Informed Neural Networks (PINNs)*, are a highly recommended approach. Neural PDEs are suitable for capturing the complex dynamics of ground motion in an elastic material.

What are Physics-Informed Neural Networks (PINNs)

PINNs are a type of artificial neural network that integrates both data-driven learning and the governing physical laws (expressed as PDEs) directly into their training process. Unlike traditional neural networks that rely solely on data, PINNs incorporate the equations of physics directly into their loss function.

This unique characteristic allows them to:

- *Respect Physical Laws*: Ensure that predictions adhere to known scientific principles, such as the equations of elasticity relevant to ground mechanics.

- *Handle Sparse Data*: Perform well even with limited observational data by leveraging the strong constraints imposed by the physical equations.

- *Solve Forward and Inverse Problems*: Not only predict outcomes (forward problem) but also infer unknown parameters or initial conditions (inverse problem).

For predicting ground motion in an excavation context, PINNs can model the complex stress-strain relationships within the ground material over time, providing insights into potential displacements and deformations. PINNs have been successfully applied to problems in elasticity, including tissue elasticity reconstruction, plane elasticity, and complex solid mechanics scenarios, and are used in microstructure-sensitive deformation modeling and rate- and temperature-dependent plasticity.

Technical Details and Governing Equations

The motion of an elastic material, such as the ground during excavation, is fundamentally governed by the *Navier-Cauchy equations*. For a linear elastic material, the *stress-strain relationship is given by Hooke's law*. These physical laws are embedded directly into the learning process of PINNs.

The comprehensive set of mathematical equations that describe elastic motion under stress includes:

- *Strain-Displacement Relations (Kinematics)*: Strain (ϵ_{ij}) describes the deformation of a material. For small deformations, the infinitesimal strain tensor is related to the displacement field $\mathbf{u} = (u_x, u_y, u_z)$:

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

- *Stress-Strain Relations (Constitutive Equations - Hooke's Law)*: For an isotropic linear elastic material, the stress (σ_{ij}) is related to the strain (ϵ_{ij}) by Hooke's Law, involving two material constants: Young's Modulus (E) and Poisson's Ratio (ν), or alternatively, Lamé's constants (λ and μ). In terms of Lamé's constants:

$$\sigma_{ij} = \lambda\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij}$$

, where δ_{ij} is the Kronecker delta, and ϵ_{kk} is the volumetric strain. Lamé's constants can be expressed as:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)} = G \quad \text{(shear modulus)}$$

- *Equations of Motion (Equilibrium Equations)*: These equations relate the stresses to external body forces (f_i) and inertial forces (due to density ρ and acceleration $\ddot{u}_i$). For dynamic equilibrium:

$$\frac{\partial \sigma_{ij}}{\partial x_j} + f_i = \rho\frac{\partial^2 u_i}{\partial t^2}$$

- *Navier-Cauchy Equations (Equations of Elasticity in terms of Displacement)*: By substituting the strain-displacement relations into the stress-strain relations, and then those into the equilibrium equations, the Navier-Cauchy equations are obtained, expressed solely in terms of displacement components:

$$\rho\frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) + \mu\nabla^2\mathbf{u} + \mathbf{f}$$

  These equations, along with appropriate boundary conditions (e.g., fixed displacements, applied tractions, or contact conditions), form the basis for predicting the time evolution of ground deformation. When using PINNs, these equations are typically incorporated into the loss function, allowing the neural network to learn a solution that satisfies these physical laws.

Equations of Motion for the Bulk Material

For an elastic material like the ground during metro station excavation, the motion of the bulk (the material itself, away from specific boundaries) is governed by the *Navier-Cauchy equations*. These are partial differential equations (PDEs) that describe the dynamic equilibrium of the elastic body.

The Navier-Cauchy equations are expressed in terms of the displacement vector $\mathbf{u}$ and material properties. The general vector form is given as:

$$\mu\nabla^2\mathbf{u} = (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) + \mathbf{f} = \rho\frac{\partial^2\mathbf{u}}{\partial t^2}$$

Where:

- $\mathbf{u}$ is the *displacement vector* (representing u_x, u_y, u_z in Cartesian coordinates).
- ρ is the *material density*.
- λ and μ are *Lamé parameters*, which are material constants describing the elastic properties of the material. These can also be expressed in terms of Young's Modulus (E) and Poisson's Ratio (ν).
- $\mathbf{f}$ is the *body force vector* here gravity.

Equations of Motion for the Moving Boundary of the Hole: the bank of the ecscavation

The s *equations of motion of the boundary* as a separate set of PDEs are usually accounted for through *appropriate boundary conditions* applied to the bulk Navier-Cauchy equations.

For a *moving hole*, the key is to specify the conditions *at the interface* between the elastic material and the void of the hole, and how these conditions evolve over time as the hole moves. These are typically dynamic boundary conditions, which can be of two main types:

- *Kinematic (Displacement) Boundary Conditions*: These specify the displacement or velocity of the material points on the surface of the hole. If the hole's boundary has a known, prescribed motion, then the displacement $\mathbf{u}$ of the material particles on that boundary must follow that motion over time. For example, if a drilling tool is excavating a moving hole, the surface of the material in contact with the tool would be forced to displace according to the tool's movement.
- *Traction (Force) Boundary Conditions*: These specify the forces (stresses) acting on the surface of the hole. For an open, unpressurized void, the normal and shear stresses on the boundary would typically be zero (a "free surface"). If the hole is created by a tool exerting pressure or if it's filled with a fluid, a specific traction vector (stress) might be applied to the boundary. These tractions would also be time-dependent as the hole moves and its interaction with the surrounding material changes.

Open-Source Libraries used for Implementation

Implementing a Neural PDE or PINN model requires specialized libraries that facilitate the integration of neural networks with differential equations. Key open-source resources include:

- `DeepXDE`: A comprehensive Python library for scientific machine learning and physics-informed learning. It supports various algorithms, including PINNs, and can solve forward and inverse ordinary/partial differential equations, compatible with `TensorFlow, PyTorch, JAX` backends. It's highly suitable for defining custom PDE constraints and integrating data.
- `jinns`: A Python library specifically for physics-informed neural networks built on the JAX ecosystem, providing a versatile framework for prototyping real-world problems.
- `NeuralOperator`: A PyTorch-based Python library focusing on operator learning, which generalizes neural networks to map between function spaces, beneficial for solving families of PDEs efficiently.
- `neurodiffeq`: A Python package built on PyTorch for solving differential equations with neural networks, offering flexibility in defining network architectures, optimizers, and sampling strategies.
- `NeuralPDE.jl`: A Julia package providing automatic Physics-Informed Neural Networks (PINNs) for solving various differential equations, including PDEs, and can mix xDE solving with data fitting.

Implementation Considerations and Complexity

While powerful, implementing a neural PDE model for ground motion prediction is a complex endeavor that requires expertise in several areas. The general methodology involves:

- *Data Loading and Preprocessing*: The data is loaded, and preprocessing steps include handling missing values, normalizing the data, and splitting it into training and testing sets. The time, 3D coordinates, and deformation data will be used to train the network and evaluate the data-driven component of the loss function.
- *Neural Network Design*: Carefully selecting the appropriate architecture, including the number of layers, neurons, and activation functions, as well as regularization techniques, for the specific problem.
- *Formulating the Physics-Informed Loss Function*: Accurately translating the governing equations of elasticity (e.g., Navier-Cauchy equations) and relevant boundary conditions into a differentiable loss function that the neural network can minimize. This is where domain-specific knowledge of continuum mechanics and geotechnical engineering becomes crucial. The loss function for training is typically defined as the Mean Squared Error (MSE) between the predicted and actual deformation values. The sources state that when using PINNs, the governing equations of elasticity, such as the Navier-Cauchy equations, are*typically incorporated into the loss function.* This involves accurately translating these equations into a differentiable loss component. The overall *loss function for training* is defined as the Mean Squared Error (MSE)

between *the predicted and actual deformation values* (the data-driven component). To this, I add terms that penalize violations of the physical laws. For example, *a priori* estimate is related to the expected behavior of the solution (e.g., smoothness, boundedness, or adherence to certain boundary conditions that aren't explicitly data points), I formulate these as additional MSE or L_1/L_2 penalties.

Combined Loss Function Structure :

- *Data Loss*: $MSE(NN_{\text{output}}(data\_points), actual\_deformation_data)$
- *Physics Loss*: $MSE(PDE_{\text{residuals}}(NN_{\text{output}}(collocation\_points)), 0)$
- *Boundary Loss* Another component of the loss function that enforces the specified boundary conditions (either by penalizing deviations from prescribed displacements or by penalizing non-zero tractions) on the points defining the moving boundary of the hole.
- *A Priori Estimate Loss*: This is an additional term that penalizes deviations from *a priori* estimate. For instance, for an initial guess $u_{\text{approx}}(x, y, z, t)$ that the solution $u(x, y, z, t)$ is close to, I add a term like

$$MSE(NN_{\text{output}}(collocation_points), u_{\text{approx}}(collocation\_points))$$

  or a regularizer based on its properties. This term would essentially act as a soft constraint guiding the network towards your prior belief.

- *Model Training*: Optimizing the neural network parameters by minimizing the combined physics-informed and data-driven loss.
- *Custom Code*: I write *custom code* to define the specific PDE constraints and integrate in time, 3D coordinates, and deformation data from the `XaidariSt._AGS files.xlsx" (specifically Sheet1.csv) into the network's training process.`
- *Model Evaluation*: `Evaluating the trained model using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on the testing dataset to provide insights into the model's accuracy and performance.`
- *Computational Resources*: `Training complex PINN models is computationally intensive, often requiring access to GPUs or specialized hardware.`

These libraries provide the necessary foundational tools, but the successful development and deployment of such a model will require a strong foundation in both deep learning and the underlying physics of material deformation. Specifically these libraries provide frameworks where I define:

- The neural network architecture (number of layers, neurons, activation functions).
- The *domain of the problem*: the 3D space of the trench and the time dimension).
- *Collocation points*: Points where the PDE loss is enforced.
- *Boundary conditions*: These are crucial for defining the problem and are incorporated into the loss function.
- *The specific PDE constraints*: I write custom code using the library's API to represent the Navier-Cauchy equations and any additional *a priori* estimate terms you want to incorporate.

Future work will focus on enhancing the model's accuracy and exploring its applicability to different construction scenarios.

In the context of Neural PDE solvers, particularly *Physics-Informed Neural Networks (PINNs)*, these governing equations for the bulk and the boundary conditions as well as resulting *a priori* estimates for the solution are *incorporated directly into the loss function* that the neural network optimizes during training. This means that the neural network is trained not only to fit any available data (e.g., deformation measurements) but also to satisfy the physical laws (Navier-Cauchy equations) and the specified boundary conditions at collocation points within the domain and on its boundaries. This ensures that the model's predictions adhere to known scientific principles, even with sparse data. For the *moving hole* within this elastic material, such as the bank of the excavation, the behavior is handled through *time-dependent boundary conditions**applied to the Navier-Cauchy equations These conditions can be kinematic (prescribed displacement) or traction (prescribed stress/force). In a PINN framework, these boundary conditions are *incorporated into the loss function* alongside the PDE residuals and data-driven loss components.

A Lagrangian functional (or Action functional) for a dynamic system is typically defined as the time integral of the Lagrangian, $L = T - V$, where T is the kinetic energy and V is the potential energy. The principle of least action states that the actual path of a system is one that extremizes this functional. The equations of motion (like the Navier-Cauchy equations) can be derived from this principle.

The Lagrangian of an elastic body

For the elastic material described in the sources, the components of a theoretical Lagrangian would involve:

- *Kinetic Energy (T):* This would be an integral over the volume of the material, involving the material density and the square of the velocity of the material particles. The sources define ρ as the material density and $\frac{\partial^2 \mathbf{u}}{\partial t^2}$ as acceleration, which implies $\frac{\partial \mathbf{u}}{\partial t}$ is velocity.

$$T = \int_V \frac{1}{2} \rho \left(\frac{\partial \mathbf{u}}{\partial t} \right)^2 dV$$

- *Potential Energy (V):* This would comprise two main parts for an elastic material under gravity:

- *Elastic Strain Energy:* This part accounts for the energy stored in the material due to deformation. The sources explicitly state the stress-strain relationship (Hooke's Law) and the strain-displacement relations, which are fundamental for defining this energy. The Lamé parameters (λ and μ) are also introduced as material constants describing elasticity. The elastic potential energy density is typically given by

$$W = \frac{1}{2} \sigma_{ij} \epsilon_{ij}$$

. Integrated over the volume,

$$V_{\text{elastic}} = \int_V W dV$$

Using Hooke's law and strain-displacement relations, this can be expressed solely in terms of displacement $\mathbf{u}$ and material properties λ, μ. *Gravitational Potential Energy* This accounts for the energy due to body forces, such as gravity. The sources mention $\mathbf{f}$ as the body force vector. If $\mathbf{f}$ is derived from a potential ϕ, then

$$V_{gravity} = - \int_V \mathbf{f} \cdot \mathbf{u} dV$$

(or more generally $\int_V \rho \phi dV$ if $\mathbf{f} = -\rho \nabla \phi$).

- *Boundary Motion Term:* Since the *boundary motion term* for a hole whose boundary is *free to move due to gravity and internal motion stresses* then in a variational principle, conditions on the boundary (like a free surface where tractions are zero, or a surface where external forces are applied) manifest naturally as boundary terms in the variation of the action integral. In the context of a Lagrangian, a free surface would imply that no external work is done on that surface, so no explicit work term due to external tractions would appear for the hole's boundary in the boundary integral of the action. The motion of the boundary itself would be implicitly governed by the material's response (via the bulk equations) to the change in geometry and applied boundary conditions at the moving interface.

The Navier-Cauchy equations are the *Euler-Lagrange equations* that one would obtain by taking the variation of the described Lagrangian functional and setting it to zero. This means the Navier-Cauchy equations represent the dynamic equilibrium of the elastic body under the influence of elastic forces, body forces, and inertial forces, and they are derived from the principle of least action. In the context of Physics-Informed Neural Networks (PINNs), as described in the sources, the approach **directly incorporates the governing PDEs (Navier-Cauchy equations) and the boundary conditions into the loss function**. Instead of formulating and varying a Lagrangian functional, the PINN is trained to minimize the residual of the Navier-Cauchy equations across the domain and to satisfy the specified boundary conditions on the surfaces (including the moving boundary of the hole). This method effectively solves the same problem without explicitly requiring the user to derive or work with the variational principle.

Time dependent boundary conditions and motion of the bank

Governing Equations and Material Behavior

The ground material, including the walls of the trench, is modeled as an *elastic material* As it is explained thoroughly its motion and deformation are governed by the *Navier-Cauchy equations* These equations, along with *appropriate boundary conditions*, form the basis for predicting the time evolution of ground deformation.

The Nature of the Trench Walls' Motion

The scenario involves *ground excavation for a metro station*, which leads to *ground motion. The dataset comprises measurements of ground motion of the walls of the trench at various depths and times, including spatial coordinates and deformation values over time. This confirms that the boundary (trench walls) is indeed moving and deforming over time. The problem focuses on predicting the time evolution of deformation. The excavation process itself implies that the geometry of the trench, and thus its boundary, is changing over time.*

Boundary Conditions and Time-Dependency:

In continuum mechanics, boundary conditions are essential for solving the governing PDEs. They specify the behavior of the material at the boundaries of the domain. I indicated that the walls of the trench are free to move due to gravity and internal motion stresses. In the context of an excavation or a "hole" within an elastic material, this typically corresponds to a traction-free boundary condition . A traction-free boundary means that no external forces or stresses are applied on that surface; its movement is purely a response to the internal stresses and body forces within the material. Since the excavation is an ongoing process and the ground deformation is observed over "Time (t)", these boundary conditions are inherently time-dependent. This means either the shape of the boundary itself changes over time (as the excavation proceeds) or the conditions applied to a fixed boundary change over time. For an excavation, it's primarily the former – the boundary's geometry evolves.

Integration into Physics-Informed Neural Networks (PINNs):

The chosen methodology, Neural Partial Differential Equations (Neural PDEs), specifically Physics-Informed Neural Networks (PINNs), are designed to incorporate these physical laws and boundary conditions directly into their training process. A critical step in implementing a PINN model is Formulating the Physics-Informed Loss Function. This involves Incorporating the governing equations of elasticity and boundary conditions into the loss function, alongside any data-driven loss. This ensures that the neural network learns a solution for the displacement field ($\mathbf{u}$) that not only fits the measured deformation data but also adheres to the underlying physical laws and the specified behavior at the boundaries. The PINN automatically learns the solution that satisfies these conditions, including their time evolution. The data itself provides the observed deformation of the trench walls over time, which implicitly defines the outcome of these time-dependent boundary conditions.

Deformed boundary and analque of Laplace-Young

An analogue of the Young-Laplace equation exists for the bounding surface of an elastic material that is deformed, but it is more complex than the classical equation used for fluid interfaces. The direct application of the Young-Laplace equation, which relates pressure difference to surface tension and curvature for fluids, is not valid for solids due to their ability to resist shear.

For elastic materials, the concept shifts from a simple scalar surface tension to tensorial surface stress and surface elasticity. Specifically:

- *Surface Stress vs. Surface Tension: In fluids, surface tension is a scalar quantity representing the energy per unit area of the interface. In solids, surface energy (also energy per unit area) still exists, but when a solid surface is *deformed*, additional elastic energy is stored. This leads to the concept of surface stress, which is the reversible work per unit area needed to elastically stretch a pre-existing surface. Unlike fluid surface tension, surface stress is a tensorial quantity, meaning it has directional components and can resist shear.*
- *Generalized Equations: Researchers have developed generalized forms that extend the Young-Laplace concept to solids by incorporating surface stress. These more elaborate equations account for:*
  - *Bulk stresses: The stresses within the main volume of the material.*
  - *Surface membrane stresses: Stresses acting within the thin surface layer itself, which deform the surface.*
  - *Surface transverse stresses: Stresses perpendicular to the surface within the surface layer, related to out-of-plane deformations.*
  - *These surface stress contributions depend on both the curvature of the surface and the elastic strain within the surface layer. They can also induce residual stresses in the bulk material.*

- *Shuttleworth Relation: A fundamental relationship in surface elasticity is the Shuttleworth relation, which connects surface stress ($\boldsymbol{\tau}$) to surface energy (γ) and surface strain ($\boldsymbol{\epsilon}_s$):*

$$\boldsymbol{\tau} = \gamma \mathbf{I} + \frac{\partial \gamma}{\partial \boldsymbol{\epsilon}_s}$$

  where $\mathbf{I}$ is the identity tensor. This equation highlights that surface stress is not simply equal to surface energy, but also includes a term related to how surface energy changes with surface strain.

- *Significance at Small Scales: These surface effects (surface stress and surface elasticity) become increasingly dominant at smaller length scales, such as in nanomaterials (nanoparticles, thin films, nanowires), because the ratio of surface area to volume increases significantly. At these scales, the contributions from the surface energy and surface stress can significantly influence the overall mechanical behavior, stability, and deformation of the material.*

In summary, while there isn't a direct one-to-one "analogue" that simply replaces fluid surface tension with a scalar for solids, the field of surface elasticity provides a more comprehensive framework using concepts like surface stress and surface energy density, which effectively generalize the underlying physical principles of surface-driven phenomena to elastic materials. These generalized models account for the solid's ability to sustain shear and elastic deformation within the surface itself.

Further elaboration *: Surface stress is a fundamental concept in solid mechanics, particularly relevant for understanding the mechanical behavior of materials at surfaces and interfaces, especially at the nanoscale. It's crucial to distinguish it from surface tension, which applies primarily to liquids.*

Definition: Surface stress ($\boldsymbol{\tau}$) is defined as the reversible work per unit area required to elastically deform a pre-existing surface of a solid material. It represents the excess force per unit length acting within the surface layer of a solid.

Key Characteristics and Distinctions from Surface Tension (Liquids) *Unlike the scalar surface tension of a liquid, surface stress in solids is a tensor. This means it has components in different directions (normal and shear components) within the surface plane. This tensorial nature arises because solids can resist shear deformation, allowing for directional forces within their surfaces.*

*Origin of Stress Surface tension in liquids arises primarily from the imbalance of intermolecular forces at the surface, where molecules experience a net inward pull. When a liquid surface area changes, molecules can easily rearrange to maintain equilibrium, and there's no stored elastic energy associated with the surface's stretching. Surface stress in solids also originates from the different bonding environments of atoms at the surface compared to the bulk. However, when a solid surface is *elastically deformed* (stretched or compressed), the atoms at the surface are displaced from their equilibrium positions, leading to the storage of elastic strain energy within that surface layer. This stored energy contributes to the surface stress.*

The Shuttleworth relation

$$\boldsymbol{\tau} = \gamma \mathbf{I} + \frac{\partial \gamma}{\partial \boldsymbol{\epsilon}_s}$$

*surface stress and surface energy (the excess free energy associated with creating a new surface area) are generally *not* the same for solids. The first term, $\gamma\mathbf{I}$, represents the contribution from the surface free energy itself. The second term, $\frac{\partial \gamma}{\partial \boldsymbol{\epsilon}_s}$, accounts for the change in surface free energy as the surface is elastically strained. This derivative term is typically non-zero for solids, making surface stress a distinct quantity from surface energy.*

Surface stress can significantly influence the mechanical properties and behavior of solid materials, especially at small scales (e.g., in thin films, nanoparticles, or nanowires) where the surface-to-volume ratio is high. Examples include:

- *Size-dependent elasticity: The effective elastic modulus of nanomaterials can be affected by surface stress.*
- *Phase transformations: Surface stress can influence phase transitions and crystal growth.*
- *Actuation and bending: Gradients in surface stress can cause bending or other deformations in structures.*
- *Residual stresses: Surface stresses can induce internal (residual) stresses within the bulk of a material, even in the absence of external loads.*

In essence, surface stress is a measure of the force required to stretch or compress a solid surface, reflecting the elastic response of the surface itself to deformation. It is a more comprehensive concept than surface tension, accounting for the unique mechanical properties of solid surfaces.

After all these I introduce the standard equation for the motion of a surface by its Gauss-curvature and add term adjusted via Neural PDes. Neural PDEs can address modeling the excavation bank motion and incorporate adjustable terms.

YActually Neural PDEs, particularly Physics-Informed Neural Networks (PINNs), are designed precisely to adjust terms in a PDE that are incorporated into the modeling process. This is a core mechanism by which they operate for problems like ground deformation during metro station excavation.

Here's how this adjustment works within the framework presented in the sources:

- *Embedding Physical Laws: PINNs embed the knowledge of physical laws directly into the learning process. For modeling ground motion in an elastic material, the relevant physical laws are primarily the Navier-Cauchy equations. These equations describe the motion of an elastic material based on its displacement vector, material density, Lamé parameters, and body force. They are derived from fundamental principles including strain-displacement relations, Hooke's Law (stress-strain relations), and equilibrium equations.*
- *Formulating the Physics-Informed Loss Function: When modeling with PINNs, the governing physical equations (such as the Navier-Cauchy equations for ground elasticity) are transformed into a differentiable component of the loss function. This component represents how well the neural network's predictions satisfy these physical laws. Alongside this, there is typically a data-driven loss component, which measures the Mean Squared Error (MSE) between the model's predictions and the actual observed deformation values from the dataset.*
- *Optimization and "Adjustment": During the training process, the neural network's parameters are optimized to minimize this combined physics-informed and data-driven loss function. By minimizing the physics-informed part of the loss, the neural network is compelled to find a solution (i.e., its internal "terms" or parameters) that inherently adheres to the known scientific principles of elasticity, even if the observational data is sparse or noisy. Essentially, the neural network adjusts its internal weights and biases such that its output (the predicted displacement field) not only fits the given deformation data but also satisfies the mathematical constraints imposed by the physical PDEs.*

Motivated form the elastic energy in the lagrangian I introduce a model equation for the motion of the bank of the following the form

$$\frac{\partial \underline{X}}{\partial t} = K_+ H_\Sigma^2 + B(\underline{X})$$

where $\underline{X}$ is a point on the surface of the bank Σ, H_Σ, K_Σ are mean and Gauss curvature of the bank, λ, B the terms adjusted with Neural PDEs.

The following observations are in line

- *Mean and Gauss Curvature and Strain Field: The core of the model is governed by the Navier-Cauchy equations. These equations describe the motion of an elastic material solely in terms of the displacement vector ($\mathbf{u}$), material density (ρ), Lamé parameters (λ, μ), and a body force vector ($\mathbf{f}$). The derivation of the Navier-Cauchy equations involves:*
  - *Strain-Displacement Relations How deformation (strain ϵ_{ij}) is related to the displacement field $\mathbf{u}$.*
  - *Stress-Strain Relations (Hooke's Law): How internal forces (stress σ_{ij}) relate to strain via material parameters.*
  - *Equations of Motion (Equilibrium Equations): How stresses, body forces, and inertial forces balance.*

  The literature does not introduce or incorporate explicit terms related to mean curvature or Gauss curvature within the fundamental governing equations (Navier-Cauchy equations), or within the stress-strain relations. The model, as described, focuses on the volumetric (bulk) response of the elastic ground material, where the surface motion is a manifestation of the bulk's deformation constrained by boundary conditions. While these geometric concepts are vital in disciplines like differential geometry or thin-shell theory, they are not part of the described linear elastic bulk continuum model.
- *Adjustable Local Functions and Overall Additional Term:*
  - *Adjustable Local Functions: The model uses PINNs. A key aspect of PINNs is that the neural network itself acts as a highly flexible and adjustable function that learns to represent the continuous displacement field ($\mathbf{u}$) across the entire spatial and temporal domain. The network's parameters are adjusted during training to minimize the loss function, which includes adherence to the physical laws and available data. This allows the model to capture complex and even "local" variations in the displacement field as long as they are consistent with the underlying physics.*

- *Overall Additional Term: The Navier-Cauchy equations already include a body force vector ($\mathbf{f}$). This term represents external forces acting on the material's volume, such as gravity or other distributed loads. Therefore the "overall additional term" refers to external forces or influences that act on the bulk, this existing term can account for them.*
- *I introduce entirely new physical terms (e.g., related to curvature) into the governing partial differential equation itself, that would represent an extension of the underlying physical model beyond what is currently described in the sources. The literature define the governing equations strictly as the Navier-Cauchy equations for an elastic material. Modifying the fundamental PDE would mean modeling a material with different constitutive behavior than a simple linear elastic solid.*

In summary, the model inherently uses adjustable functions (the neural network) to represent the displacement field, from which consistent strain and stress fields are derived. It also includes a term for body forces ($\mathbf{f}$). However, the described physical framework of linear elasticity, as presented in the sources, does not include explicit terms based on mean or Gauss curvature in its governing equations for predicting ground motion. The consistency of the results stems from the PINN's ability to enforce the Navier-Cauchy equations, which link displacement, strain, and stress via material parameters, throughout the learning process. PINNs actually treat the surface of the material as part of the bulk. The model I propose suggests how the bank moves after the motion of the surface. Afterall the sataset consists of measurements on the surface.

A very important point about the nature of the dataset, that it is a common scenario when applying Physics-Informed Neural Networks (PINNs). The dataset consists solely of measurements of ground motion of the walls of the trench at various depths and times, including spatial coordinates (x, y, z) and corresponding deformation values (dX, dY, dZ). This means I have information only about the deformation occurring at the surface (specifically, the trench walls), and not from the interior of the ground material.

However, this does not contradict the fact that the PINN model describes the motion of the bulk [my previous response]. Instead, the surface data plays a crucial role in defining the problem for the bulk:

- *Bulk Governing Equations Remain Fundamental: The core of the model is still based on the understanding that the motion of the elastic ground material is governed by the Navier-Cauchy equations, which describe the displacement vector ($\mathbf{u}$) for every point within the material's volume . These equations are derived from fundamental principles of continuum mechanics, including strain-displacement relations, stress-strain relations (Hooke's Law), and equations of motion that apply throughout the entire body.*
- *Surface Data as Boundary Conditions: The measurements you have from the trench walls (the surface) serve as critical boundary conditions for the problem. While the Navier-Cauchy equations describe the physics within the material, boundary conditions describe the interactions or constraints at its edges. In the case at hand, the measured deformations at the trench walls dictate how the bulk material is behaving at those specific boundaries.*
- *PINNs Excel with Sparse Data: Physics-Informed Neural Networks are uniquely suited for scenarios with sparse or noisy data, precisely like having only surface measurements for a bulk phenomenon. Here's how it works within the PINN framework:*
  - *Physics-Informed Loss: The PINN is trained to satisfy the Navier-Cauchy equations throughout the entire domain (both surface and interior) by incorporating these equations directly into its loss function. This physics-informed component of the loss ensures that the model's predictions adhere to known scientific principles everywhere, even where there's no data.*
  - *Data-Driven Loss: Simultaneously, the PINN is also trained to match your available surface deformation data. This data-driven component of the loss ensures that the model's predictions align with the observed reality at the measurement points (the trench walls).*
- *Learning the Continuous Field: By combining these two loss components, the PINN learns a continuous displacement field ($\mathbf{u}$) across the entire spatial and temporal domain of the ground material. Even though your data is only from the surface, the physical laws (Navier-Cauchy equations) provide the necessary constraints for the PINN to infer the behavior of the *interior* of the material. The surface measurements guide and constrain this inference at the boundaries, ensuring the overall solution is physically consistent and matches observations.*

In essence, the surface is part of the bulk, and its observed motion provides the crucial clues or anchors that, when combined with the governing physical laws of the elastic material, allow the PINN to predict the motion and deformation of the entire ground mass.

Challenges

Neural Partial Differential Equations (PDEs), particularly Physics-Informed Neural Networks (PINNs), are well-suited to address several specific challenges in ground deformation, especially within the context of metro station construction and excavation.

Here are the specific challenges Neural PDEs can effectively address:

- *Capturing Complex Dynamics of Ground Motion: The construction of metro stations involves significant ground excavation, which can lead to complex ground motion. Neural PDEs are specifically chosen and are **suitable for capturing the complex dynamics of ground motion in an elastic material**. They provide a **robust approach for ensuring safety and stability**.*
- *Ensuring Safety and Stability of Construction Sites: Accurate prediction of ground motion is **crucial for ensuring the safety and stability of the construction site and its vicinity**, as ground excavation poses potential risks to surrounding structures. Neural PDEs offer a robust method to achieve this.*
- *Handling Sparse or Noisy Data: Unlike traditional neural networks that rely solely on extensive data, PINNs can find solutions to partial differential equations even with sparse or noisy data. This is a significant advantage in real-world scenarios where complete and clean datasets might be unavailable.*
- *Respecting Physical Laws: PINNs embed the knowledge of physical laws (like the equations of elasticity) directly into the learning process. This ensures that the model's predictions adhere to known scientific principles, such as the equations of elasticity relevant to ground mechanics. This helps in modeling the complex stress-strain relationships within the ground material over time.*
- *Predicting Deformed Textures and Their Evolution: PINNs can be used in material deformation modeling, including problems involving large deformation and material nonlinearities, and can help predict deformed textures and their evolution over time.*
- *Solving Forward and Inverse Problems: PINNs can not only predict outcomes (forward problems) but also infer unknown parameters or initial conditions (inverse problems).*

  By integrating both data-driven learning and governing physical laws, Neural PDEs offer a powerful tool for analyzing and predicting the time evolution of ground deformation, enhancing safety and understanding in complex construction projects.

Consolidation

In contrast to Terzaghi's one-dimensional (1D) consolidation theory, which assumes vertical drainage and compression only, real-world geotechnical problems often involve two-dimensional (2D) or three-dimensional (3D) consolidation. This means that both pore water flow and soil deformation can occur in multiple directions simultaneously.

**Challenges in Multi-Dimensional Consolidation*

Extending consolidation theory to 2D or 3D introduces several complexities:

- *Multi-Directional Pore Water Flow: In 2D or 3D scenarios, excess pore water can dissipate not only vertically but also horizontally (e.g., towards vertical drains, along permeable layers, or out through exposed boundaries). This requires considering flow gradients in multiple spatial directions.*
- *Complex Stress States: The stress distribution within the soil mass is no longer simple and one-dimensional. Stress changes can occur in all directions, influencing the pore water pressure generation and dissipation more intricately.*
- *Coupled Hydro-Mechanical Behavior: The deformation of the solid soil skeleton and the flow of pore water are inherently coupled. Changes in pore water pressure affect effective stresses and deformation, while deformation can, in turn, influence permeability and drainage paths. This coupling is more pronounced and harder to simplify in multiple dimensions.*
- *Anisotropy: Soil properties, particularly permeability (k) and coefficient of consolidation (c_v), often exhibit anisotropy, meaning they differ in horizontal (k_h, c_h) and vertical (k_v, c_v) directions. This is common in naturally deposited clays.*

Governing Equation for Multi-Dimensional Consolidation

The generalized governing partial differential equation for the dissipation of excess pore water pressure (Δu) in 3D consolidation, assuming isotropic permeability and compressibility (for simplicity, though anisotropy is often included in practice), is:

$$\frac{\partial(\Delta u)}{\partial t} = \frac{k}{m_v \gamma_w}\left(\frac{\partial^2(\Delta u)}{\partial x^2} + \frac{\partial^2(\Delta u)}{\partial y^2} + \frac{\partial^2(\Delta u)}{\partial z^2}\right)$$

This can also be written using the coefficient of consolidation in different directions:

$$\frac{\partial(\Delta u)}{\partial t} = c_h\left(\frac{\partial^2(\Delta u)}{\partial x^2} + \frac{\partial^2(\Delta u)}{\partial y^2}\right) + c_v\frac{\partial^2(\Delta u)}{\partial z^2}$$

Where:

- $c_h = \frac{k_h}{m_v \gamma_w}$ *is the coefficient of consolidation in the horizontal direction.*
- $c_v = \frac{k_v}{m_v \gamma_w}$ *is the coefficient of consolidation in the vertical direction.*
- *The terms represent the diffusion of excess pore pressure in the x, y, and z directions, respectively.*

Solutions and Practical Approaches

Due to the complexities, analytical (closed-form) solutions for 2D and 3D consolidation are scarce and limited to highly simplified geometries and boundary conditions. Therefore, numerical methods are predominantly used in geotechnical practice for multi-dimensional consolidation analysis:

- *Finite Difference Method (FDM): This method discretizes the soil domain and time into a grid, approximating the partial derivatives with finite differences. It's conceptually simpler but can be challenging for irregular geometries.*
- *Finite Element Method (FEM): This is the most powerful and widely used numerical technique for multi-dimensional consolidation. The soil mass is divided into a mesh of small elements, and the governing differential equations (for both fluid flow and solid deformation) are solved for each element, considering the interactions between them. FEM software packages are standard tools in geotechnical engineering.*
- *Coupled Analyses: Modern FEM analyses often perform fully coupled hydro-mechanical analyses, where the pore fluid flow equations are solved simultaneously with the solid mechanics (deformation) equations. This allows for a realistic simulation of how pore water pressure changes affect effective stresses and soil deformation, and vice versa.*
- *Boundary and Initial Conditions: For numerical solutions, defining appropriate boundary conditions (e.g., permeable/impermeable boundaries, applied loads, initial pore pressures) is critical for accurately representing the field problem.*

Practical Applications

Multi-dimensional consolidation analysis is essential for many geotechnical projects, including:

- *Embankments on Soft Clays: Predicting settlement and stability of embankments that cause lateral as well as vertical stresses.*
- *Foundations for Buildings/Structures: Especially for large mat foundations or rafts where the load distribution and drainage patterns are multi-dimensional.*
- *Deep Excavations: Analyzing ground movements and pore pressure changes around excavations, which can influence surrounding structures.*
- *Ground Improvement with Vertical Drains: Assessing the accelerated consolidation due to the installation of vertical drains, which promote horizontal drainage.*

In essence, while Terzaghi's 1D theory provides fundamental insights, 2D and 3D consolidation analyses, typically performed using sophisticated numerical methods like FEM, are indispensable for accurate design and prediction in complex real-world geotechnical scenarios.

Appendix

Incorporating a priori estimates for the solution of the deformation equation could speed up the training of a Neural PDE solver. While the sources don't explicitly state a priori estimates speed up, they emphasize that Physics-Informed Neural Networks (PINNs) embed the knowledge of physical laws directly into the learning process, which allows them to find solutions even with sparse or noisy data and ensures predictions adhere to known scientific principles. By providing a priori estimate, I am essentially guiding the neural network with additional physical insight or a good initial guess, which can make the optimization landscape smoother and lead to faster convergence.

I incorporate such knowledge into the Python code, drawing on the principles outlined in the sources:

How to Incorporate *a priori* Estimates in Python Code (Based on PINN Principles)

The primary mechanism for incorporating physical knowledge, including a priori estimates (if they can be formulated as constraints or regularizers), into a Neural PDE solver like PINNs is through the loss function.

In summary, a priori estimates can be formulated as differentiable constraints or regularizers, they are added as components to the overall loss function. The provided open-source libraries offer the tools to define these complex loss functions and neural network architectures in Python.

Loss Function with Supremum and Infimum Constraints

Let $f(x)$ be a function defined over a cluster of points C. We aim to enforce constraints on the supremum and infimum of $f(x)$ over C.

Constraints

We define the constraints as follows:

- $\sup_{x \in C} f(x) \leq U$, *where U is the upper bound.*
- $\inf_{x \in C} f(x) \geq L$, *where L is the lower bound.*

Loss Function

The loss function $L(f)$ is designed to penalize violations of these constraints. It is given by:

$$L(f) = L_{original} + \lambda \cdot \max\left(0, \sup_{x \in C} f(x) - U\right) + \lambda \cdot \max\left(0, L - \inf_{x \in C} f(x)\right)$$

where:

- $L_{original}$ *is the original loss function (e.g., mean squared error).*
- λ *is a hyperparameter controlling the strength of the penalty.*

Explanation

- *The term* $\max(0, \sup_{x \in C} f(x) - U)$ *penalizes the loss function if the supremum of $f(x)$ over C exceeds the upper bound U.*
- *The term* $\max(0, L - \inf_{x \in C} f(x))$ *penalizes the loss function if the infimum of $f(x)$ over C is below the lower bound L.*

Example in Python

Here is a conceptual example in Python:

```
import numpy as np

# Example function f(x)
def f(x):
    return x**2

# Cluster of points C
```

```
C = np.linspace(-10, 10, 100)

# Upper and lower bounds
U = 50
L = 10

# Original loss function (dummy example)
def L_original(f, C):
    return np.mean(f(C))

# Penalty terms
sup_f_C = np.max(f(C))
inf_f_C = np.min(f(C))

penalty_sup = max(0, sup_f_C - U)
penalty_inf = max(0, L - inf_f_C)

# Combined loss function
lambda_ = 0.1
L = L_original(f, C) + lambda_ * (penalty_sup + penalty_inf)

print("Original Loss:", L_original(f, C))
print("Penalty for Sup Constraint:", penalty_sup)
print("Penalty for Inf Constraint:", penalty_inf)
print("Combined Loss:", L)
```

Interior Energy estimates

The Navier-Cauchy euqations lead to the definition of elastic energy of vsarious orders for the energy contained with a given material region Ω. The usual energy takes the form

$$E_1(t,\Omega) = \frac{1}{2}\int_\Omega \rho u_t^2 + (\lambda+\mu)(\nabla\cdot u)^2 + \mu|\nabla u|^2$$

while higher order energies are defined for polyindex $j=(j_1,j_2,j_3), |j| = j_1+j_2+j_3 > 1 \psi_j = \partial^{|j|-1}u$ as

$$E_j(t,\Omega) = \frac{1}{2}\int_\Omega \rho \psi_{j,t}^2 + (\lambda+\mu)(\nabla\cdot \psi_j)^2 + \mu|\nabla \psi_j|^2$$

with euclidean tensor norms :

$$|\psi_j| = \sum_{(j_1,j_2j_3)} |\partial_j u|^2$$

These satisfy inequalities of the form:

$$E_j(t) \leq c_j(\Omega)\int_0^t E_j(\tau)d\tau + f(t)$$

These prrovide estimates of the from:

$$E(t) \leq C(\Omega)E(0) + F(t), \qquad F(t) = \int_0^t e^{c(\Omega)(t-t)} f(\tau)d\tau$$

Through weighted Sobolev inequalities (resulting from a combination of Sobolev with Hardy inequalities) and Nash-DeGiorgi-Moser iteration I arrive at the usual $L_\infty, -L_2, L_2 - \inf$ and harnack inequalities

Boundary Energy estimates

Using the model that I propose and techoniques similar to those n the bulk combined with geometric PDE I obtain estimates for the surface stress and strain of the preceding form.

Use of the estimates

I incorprate these etimates in the Neural PDE as I outlined above.