Assignment in
Machine Learning in Computational Mechanics
FHLN40, 2024

Division of Solid Mechanics
Ralf Denzer

A maximum of 30 points can be achieved in this project assignment. To pass at least 15 points are required.

The aim of this assignment is to develop a Physics Informed Neural Network which solves the static equilibrium equations of a 2-dimensional linear elastic problem assuming plane strain conditions.

1 Manufactured solution (30p)

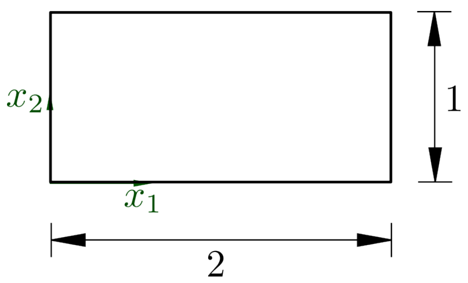

Figure 1: The rectangular domain Ω for the manufactured solution.

Given is the rectangular domain Ω as shown in Fig. 1. The bilinear manufactured displacement solution is given by

$$u_1(x_1, x_2) = 0.11x_1 + 0.12x_2 + 0.13x_1x_2 \tag{1}$$
$$u_2(x_1, x_2) = 0.21x_1 + 0.22x_2 + 0.23x_1x_2 \tag{2}$$

Assume plane strain conditions and an isotropic linear elastic material with Young's modulus $E = 1$ and Poisson's ratio $\nu = 0.25$.

1.1 Task 1.1

Determine the stress tensor $\boldsymbol{\sigma}$ and the body forces $\boldsymbol{b}$ from the manufactured displacement solution by hand.

1. Compute from the displacement vector $\boldsymbol{u}(x_1, x_2) = [u_1 \quad u_2]^T$ the displacement gradient $\boldsymbol{h} = \operatorname{grad} \boldsymbol{u}$
2. Compute the strain tensor $\boldsymbol{\varepsilon} = \frac{1}{2}(\boldsymbol{h} + \boldsymbol{h}^T)$
3. Compute the stress tensor $\boldsymbol{\sigma}$.
4. Compute the body forces $\boldsymbol{b}$ from the equilibrium conditions $\operatorname{div} \boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0}$.

Remark 1: If you compute the strain tensor as stated above, please keep in mind that its shear components $\varepsilon_{12} = \varepsilon_{21} = \frac{1}{2}\gamma_{12} = \frac{1}{2}\gamma_{21}$. With $\gamma_{12} = \gamma_{21}$ are the engineering shear strains.

Remark 2: You are free to use any symbolic software to compute the derivatives.

Remark 3: Instead of using a tensor notation you can also use the Voigt notation for the components of strain and stress tensor. The Voigt notation is a vector representation which is often used in the Finite Element Method.

1.2 Task 1.2

Develop a Neural Network which approximates the manufactured solution. The Neural Network has two input features x_1 and x_2 and the output are the displacements $u_1(x_1, x_2)$ and $u_2(x_1, x_2)$. Use the PyTorch library to develop the Neural Network and use tanh as an activation function. Train the Neural Network with the manufactured solution as training data and validate the Neural Network with validation data generated from the manufactured solution.

1.3 Task 1.3

Develop a Python program which computes for the given manufactured solution with the help of PyTorch `autograd` functions the displacement gradient, strain tensor, stress tensor and $\operatorname{div} \boldsymbol{\sigma}$. Compare the results with the analytical solution. Initially, resolve the domain with a small `torch.meshgrid`. E.g.,a `torch.meshgrid` of 3×2 points and afterwards increase the number of points when your code is working.

Remark: You may want to make plots of the different fields to verify your calculations and add them to your report.

1.4 Task 1.4

Develop a Physics Informed Neural Network which solves the static equilibrium equations. Take the body forces $\boldsymbol{b}$ from the previous task as load. Extract the Dirichlet boundary conditions from the manufactured solution. Define the cost function for this problem and train the Neural Network.

1.5 Remark

It is **not straightforward** to find good hyperparameters for training the Neural Network and/or a Neural Network (number of hidden layers / neurons) which can solve this problem. If you can not find a good solution, document your two best attempts and discuss the problems you encountered.

2 Block with Dirichlet and Neumann boundary conditions (*optional, 5 bonus points*)

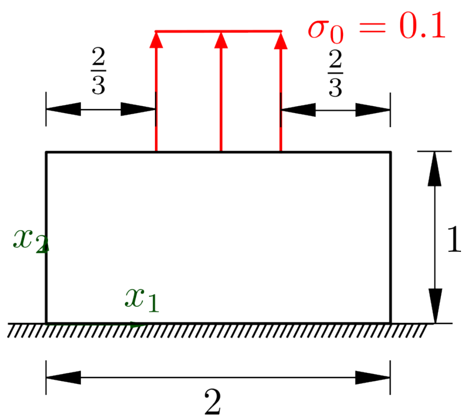

Figure 2: The rectangular domain Ω for the manufactured solution.

Aim of the second problem is to develop a Physics Informed Neural Network which solves the static equilibrium equations of 2-dimensional linear elastic problem assuming plane strain conditions. The domain is loaded with a distributed load $\sigma_0(x_1, x_2) = 0.1$ in x_2-direction as shown in Fig. 2. The bottom of the domain is fixed in x_1- and x_2-direction. Keep in mind that you need to apply the Neumann boundary conditions $\boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$ also on the load free boundary parts. Hereby $\boldsymbol{n}$ is the outward normal unit vector to the boundary of the domain. Use the same material parameters as in the previous problem.

Make use of your previous developed Neural Network to solve this problem. You need to extend your cost function in this problem by the Neumann boundary conditions.

2.1 Remark

It is **not straightforward** to find good hyperparameters for training the Neural Network and/or a Neural Network (number of hidden layers / neurons) which can solve this problem. If you can not find a good solution, document your two best attempts and discuss the problems you encountered.

3 Report

A fundamental part in all research is that it should be possible to **reproduce the results** obtained based on the report. In the present situation, this implies that the appended Python/PyTorch code should only be considered as supporting material. Moreover, note that one variable for grading the report is the structure and comments of the computer code, i.e. you should choose suitable names for variables etc. A common structure for the report is:

- **Title page**: Title of the report, authors including their person number, course name and code, date etc.
- **Introduction**: Description of the problem, geometry, boundary conditions etc. Keep this section short.
- **Procedure**: How the problems are solved. Note that you are encouraged to make references to textbooks, research papers etc. It is important to carefully present all calculations and assumptions in detail.

- **Results**: Present the results in illustrative figures and/or tables. The figures, labels etc. should have appropriate size. Note that the results should be commented such that the reader can not misunderstand the results (correct labels, figure texts etc.). The result section should also contain the hyperparameters you used for training the Neural Network and a description of the Neural Network itself. Like the number of layers/neurons, activation functions, the loss function, the optimizer, how you generated the training and validation data etc.
- **Discussion**: A discussion of the results. You might want to discuss sources of errors, accuracy and also problems with the Physics Informed Neural Network method in this section.
- **References and Tools**: State in a referenced list the literature and other (digital) tools you used during the project. Digital tools could also be ChatGPT and the like. Give also a list of the Python packages you used.
- **Computer Code**: The source code is part of the Appendix of your report. It should be easy to follow and all declared variables should have intuitive names, put adequate comments etc.

In general, the report should contain all the information that a reader can reproduce your results without any further information.

A well structured and detailed report is to be returned to the Division of Solid Mechanics (via the courses Canvas page) no later than **27 May 2024 at 13:00**. The reader of the report is assumed to have the same knowledge level as the authors. If the report contains theoretical errors, the report is returned in order to be corrected.

The assignment should be approved no later than **7 June 2024 at 13:00**. You should submit your report in PDF format to the Canvas Assignment Section of the FHLN40 course. In addition to your report you should also attach your Python-files and other files which are necessary to run your Python code. Compress these files in one zip-archive.

Remark: You are free to choose the **word processing software** for writing your report. Make sure,

1. that your word processing software can export your report as a pdf file.
2. that your word processing software provides an adequate formula editor[1].

Collaboration

The task should be solved in groups of *two*. Do not forget to write both names and person numbers on the title page of your report.

[1]As of March 2024, Word from Microsoft Office 365 running in a web browser offers not an adequate formula editor. E.g., you cannot write matrices with this formula editor.