Report - APL405

Deep Learning in Strain Gradient Elasticity

Deep bansal – 2020ME10927
HarshVardhan Singh – 2020ME10934
Vanshika Chauhan – 2020ME10980
Aaryan Ahuja- 2020ME10898
Parth jaiswal – 2020ME10959

April 19, 2022

Nomenclature

A Area of cross-section (m^2)

E Young's modulus (MPa)

ν Poisson's Ratio

b Width of Beam

bh Higher order bending parameter for given material(epoxy)

h Height of Beam

L Length of Beam

P Point Load applied at the free end

Contents

1 Introduction

Exact analytical solutions for the behaviour of mechanical elements may not be achieved in many cases due to complications present in micro/nano-scale systems, such as the presence of complex forces such as electrostatic, Casimir, Van Der Waals, and capillary forces, complex geometry, or other issues such as the existence of squeeze film damping.
According to the experimental findings, classical continuum mechanics is unable to adequately represent the mechanical behaviour of micro/nano scale structures and capture the size-dependency exhibited in such systems. As a result, non-classical continuum theories such as the non-local theory, strain gradient theory, and couple stress theory have arisen, developed, updated, and used to explore the mechanical behaviour of micro-scale structures in recent years.
Some material parameters called the length scale parameters are considered in non-classical continuum theories like strain gradient and couple stress theories, in addition to the two classical parameters, i.e. elastic modulus and Poisson ratio, which allows these theories to capture the size-dependency observed in micro/nano-scale structures.
As previously said, establishing non-traditional structural elements aids researchers and designers in dealing with challenges in micro/nano-scale systems that have two primary characteristics:

- The numerical technique is required because the analytical solution to the problem cannot be obtained or requires too many
- In order to solve the problem, the size-effect must be considered.

The first is most likely owing to loading or geometrical complications, while the second is related to the micro/nano-dimensions system's being equivalent to its material length scale characteristics. Based on the strain gradient theory, non-classical Euler–Bernoulli beam elements are created to build a non-classical finite element technique capable of capturing size-dependence and correctly simulating micro/nano-scale systems.
In this report we will be comparing strain gradient theory,modified couple stressed theory and classical theory by applying neural network architecture.

1.1 Example

One of the example we are considering is a Microcantilever- beam with a point load applied at the free end of the beam with breadth b, height h and length L.

The governing equation(Euler Bernoulli equation) is:-

$$D1\frac{\partial^4 w}{\partial x^4} - D2\frac{\partial^6 w}{\partial x^6} + \rho A\frac{\partial^2 w}{\partial x^2} = f(x,t) \quad (1)$$

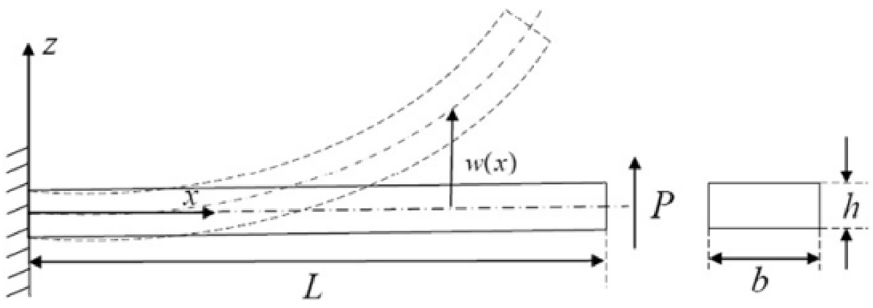


Figure 1: A Micro-cantilever with uniform rectangular cross-section subjected to a concentrated force at its free end.

Boundary conditions for cantilever beam

$$w(0) = 0 \quad \frac{\delta w}{\delta x}|_{x=0} = 0 \quad \frac{\delta^2 w}{\delta x^2}|_{x=0} = 0$$

$$V(L) = P \quad M(L) = 0 \quad Q(L) = 0$$

2 Problem statement

2.1 Static Problem

Define the problem with governing differential equations, boundary conditions and initial conditions. For classification problem, explain the data set and its pre-processing (if any). To develop a PINN solver for Euler Bernoulli Beam.

$$D1\frac{\partial^4 w}{\partial x^4} - D2\frac{\partial^6 w}{\partial x^6} = f(x,t) \tag{2}$$

at x = 0 , L

$$(V(x,t) - \hat{V}) = 0 \;\; OR \;\; (w(x,t) - \hat{w}) = 0 \tag{3}$$

$$(M(x,t) - \hat{M}) = 0 \;\; OR \;\; (\theta(x,t) - \hat{\theta}) = 0 \tag{4}$$

$$(Q(x,t) - \hat{Q}) = 0 \;\; OR \;\; (\kappa(x,t) - \hat{\kappa}) = 0 \tag{5}$$

where :

- w = displacement of beam under applied load
- f(x,t) = distributed lateral load exerted to the beam
- V (x,t) = resultant transverse force acting on the beam sections, work conjugate to w
- M(x,t) = the resultant moment on a section caused by the classical non-classical stress components
- Q(x,t) = work conjugate to y and the higher-order resultant on a section caused by higher-order stresses, work conjugate to k

w , and stand for the possible prescribed deflection, slope and curvature of the beam at the end sections respectively.

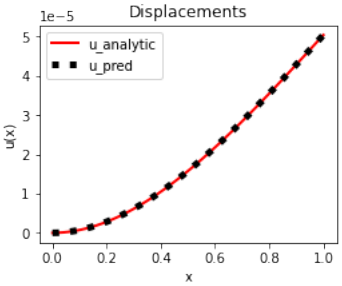
3 Methodology

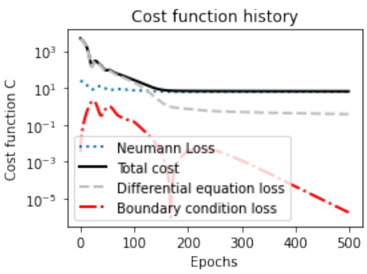
3.1 Approach 1

We started by applying the governing differential equation and boundary conditions to the static model for the cantilever beam with point load applied at the free end of the beam.

- A neural network architecture was created to understand and replicate the given problem statement and hence, solve it.
- The neural network consisted of 5 layers and 32 nodes.Boundary conditions as specified in Example1.1 were used to calculate boundary losses.
- Initially, the use of the sigmoid activation function was used which resulted in an obstacle. this led us to use other activation functions like tanh and ReLU.
- Various PyTorch optimizers were used such as Adam, LBFGS, Adadelta and SGD.

This model was working for the classical beam theory where higher order parameters were taken to be zero. This model gave the correct results for the classical beam theory.





3.2 Approach 2

After facing the problem of vanishing gradient,we scaled up the the differential equation and boundary condition losses.

- The problem was still persistent as the gradient was still vanishing.
- Proceeding further, we changed the scale for differential equation as well as the boundary conditions into micro meter and nano meter by scaling all the other factors to the same units.
- We, then inserted these scaled up values into our neural network to find further results by running them for 20,000 epochs and tried it for 7 hidden layers.
- This helped us in clearly identifying the first order differentials but still doesn't solve the problem of vanishing of the second order derivatives.

This Approach gave a better plot for the predicted displacement but still the gradients were vanishing and the graph obtained was not consistent. It was giving different graphs at different epochs. Running it for 20,000 epochs took more than an hour.

3.3 Approach 3

Our next approach was non dimensionalizing all the the equations and parameters which were used in our problem.

- This was done in order to remove all the ambiguity present in equations.
- The values of deflections were now between 0 and 1 and thus the problem of vanishing gradients was solved.
- This was done so that the value of all the terms has the value of same order.

:

```
1
Epoch: 0/3999          Differential equation loss = 8.762e-01         Boundary condition loss = 6.688e-02
oss = 9.431e-01
Epoch: 100/3999        Differential equation loss = 4.217e-06         Boundary condition loss = 6.253e-06
oss = 1.047e-05
Epoch: 200/3999        Differential equation loss = 1.168e-06         Boundary condition loss = 1.313e-06
oss = 2.481e-06
Epoch: 300/3999        Differential equation loss = 2.572e-07         Boundary condition loss = 1.626e-07
oss = 4.198e-07
Epoch: 400/3999        Differential equation loss = 4.889e-08         Boundary condition loss = 1.195e-08
oss = 6.084e-08
Epoch: 500/3999        Differential equation loss = 9.275e-09         Boundary condition loss = 1.525e-09
oss = 1.080e-08
Epoch: 600/3999        Differential equation loss = 2.542e-09         Boundary condition loss = 5.154e-10
oss = 3.057e-09
Epoch: 700/3999        Differential equation loss = 1.469e-09         Boundary condition loss = 1.375e-10
oss = 1.607e-09
Epoch: 800/3999        Differential equation loss = 1.307e-09         Boundary condition loss = 2.645e-11
oss = 1.333e-09
Epoch: 900/3999        Differential equation loss = 1.283e-09         Boundary condition loss = 3.937e-12
oss = 1.287e-09
Epoch: 1000/3999               Differential equation loss = 1.280e-09         Boundary condition loss = 4.930e-13
Total loss = 1.280e-09
Epoch: 1100/3999               Differential equation loss = 1.279e-09         Boundary condition loss = 5.784e-14
Total loss = 1.279e-09
Epoch: 1200/3999               Differential equation loss = 1.279e-09         Boundary condition loss = 9.316e-15
Total loss = 1.279e-09
Epoch: 1300/3999               Differential equation loss = 1.278e-09         Boundary condition loss = 8.544e-15
Total loss = 1.278e-09
Epoch: 1400/3999               Differential equation loss = 1.278e-09         Boundary condition loss = 1.416e-14
Total loss = 1.278e-09
Epoch: 1500/3999               Differential equation loss = 1.278e-09         Boundary condition loss = 1.943e-14
Total loss = 1.278e-09
Epoch: 1600/3999               Differential equation loss = 1.277e-09         Boundary condition loss = 1.818e-14
Total loss = 1.277e-09
```

4 Results and discussion

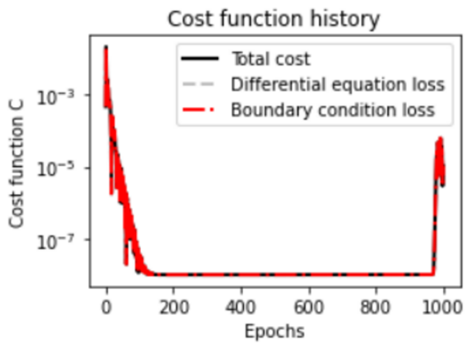
4.1 Results of approach 1

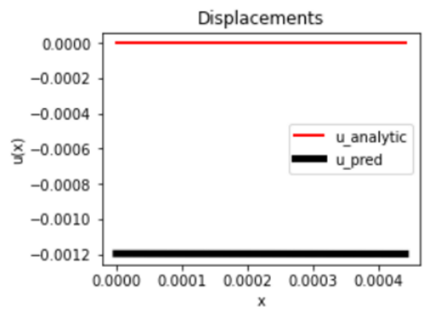
:

- After running it for several times we found out that the differential equation loss was not changing even though the boundary condition loss was changing.
- This was due to the fact that the gradient was vanishing which resulted in constant differential equation loss.
- On going into greater depths we came to the conclusion that derivative of 'u' wrt to 'x' which were of order higher than 1 were turning out to be zero.:

We obtained the following graphs for displacement and cost function.

x



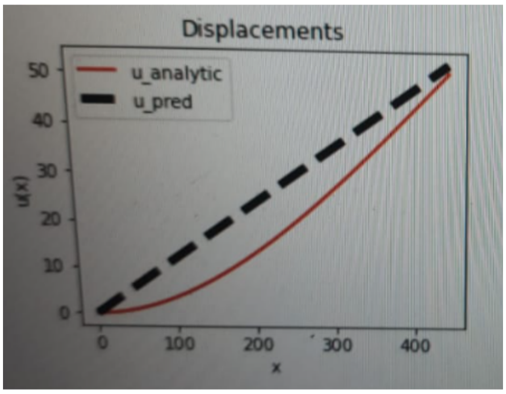


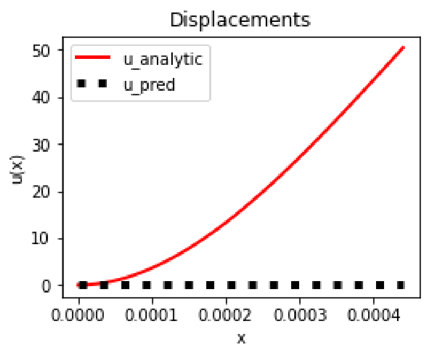
4.2 Results of approach 2

:

- Still the problem was same.Differential equation loss was not changing apart from first order derivative.
- It was also ambiguous as it gave different results when the code was run for different number of epochs.
- Also after a few iterations the value of differential equation loss shooted up which resulted the differential equation loss as 'nan'.:

We obtained the following graphs for displacement where the first figure was obtained when epochs were 20,000 with 7 hidden layers and the second one consist of epochs 4000 with 3 hidden layers.



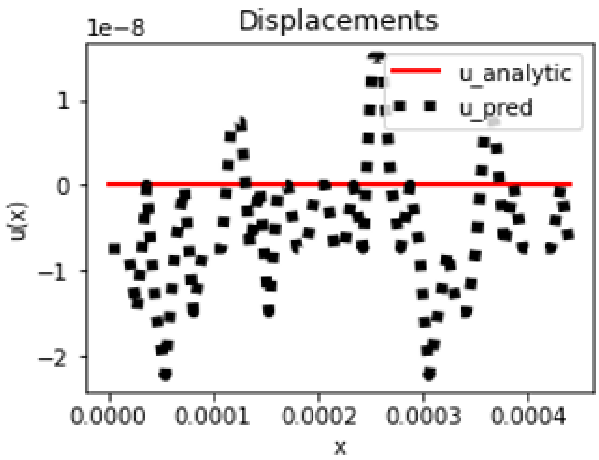


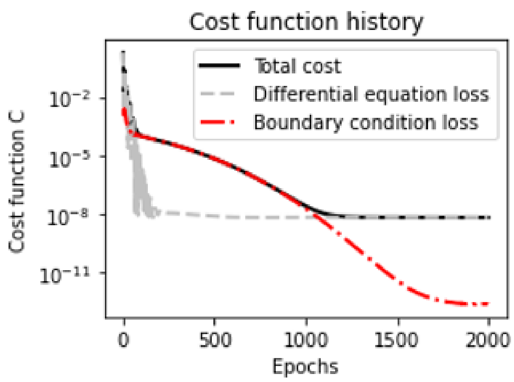
4.3 Results of approach 3

:

- With this approach the resultant differential equation was now decreasing with number of epochs.
- But the graph was quit erratic but was consistent with number of epochs.
  :

We obtained the following graphs for displacement and cost function history.





5 Conclusion

:

- Activation functions like ReLU and sigmoid give problem of vanishing gradients.This was due to the fact that gradients were coming out to be negative but due to positive range of ReLU and sigmoid it matched the grad with zero resulting in no change of differential equation loss.
- Hence for our problem,Tanh is a better suited function as it didn't led to vanishing of gradients due to its range which goes from -1 to 1.
- In all our approaches, closest graph to deflection was given on scaling the parameters to micrometer scale but the graph was not consistent.
- Consistency, although irregular in nature is shown in case of non-dimensionalisation.
- If we were to continue this project, our first approach would be to try MATLAB for the project because we tried everything in python.

References

:

- Kahrobaiyan, M. H., M. Asghari, and M. T. Ahmadian. "Strain gradient beam element." Finite Elements in Analysis and Design 68 (2013): 63-75.

- Jafari, Akbar, and Meysam Ezzati. "Investigating the non-classical boundary conditions relevant to strain gradient theories." Physica E: Low-dimensional Systems and Nanostructures 86 (2017): 88-102.

- Akgöz, Bekir, and Ömer Civalek. "Analysis of micro-sized beams for various boundary conditions based on the strain gradient elasticity theory." Archive of Applied Mechanics 82.3 (2012): 423-443.

- Zhao, Junfeng, et al. "Nonlinear microbeam model based on strain gradient theory." Applied Mathematical Modelling 36.6 (2012): 2674-2686.

- Moheimani, Reza, and Hamid Dalir. "Static and dynamic solutions of functionally graded micro/nanobeams under external loads using nonlocal theory." Vibration 3.2 (2020): 51-69.