CONSTITUTIVE RELATION IDENTIFICATION USING PINN

APL405 : COURSE PROJECT

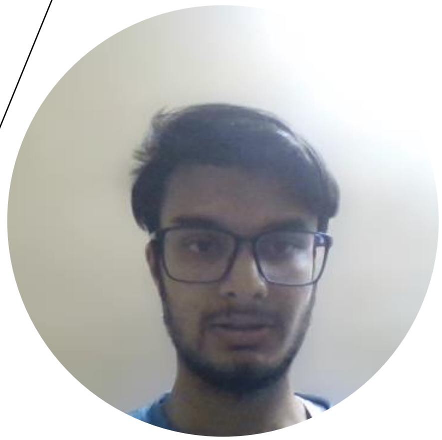
Aditya Agrawal (2021AM10198)

Swapnil Kashyap (2021AM10782)

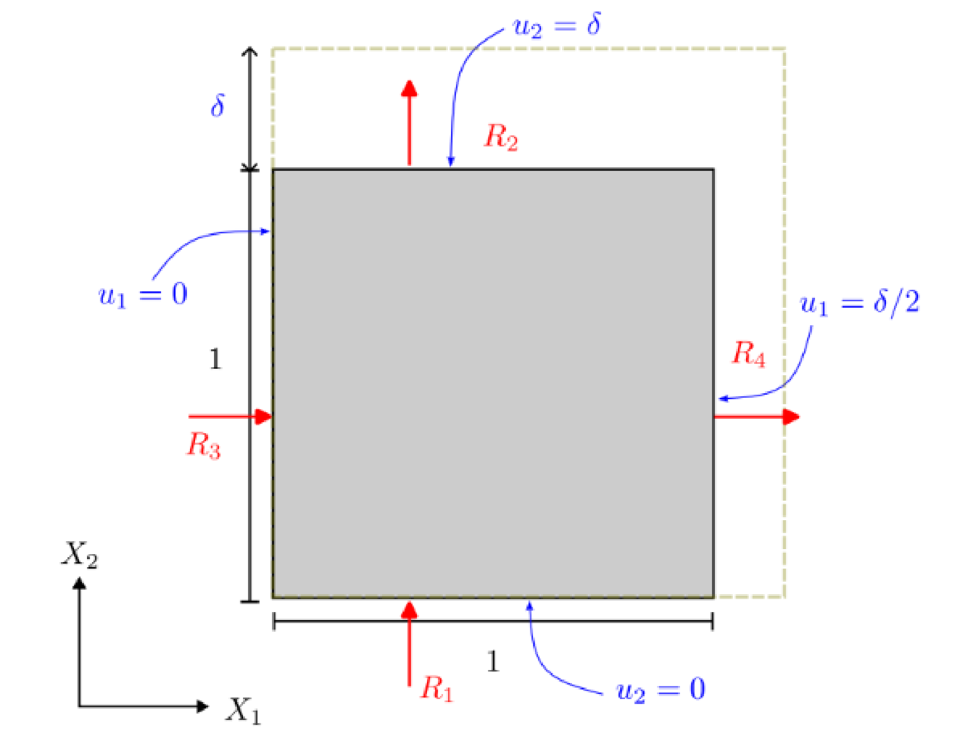
Prof. Rajdip Nayek

OBJECTIVE

- **Use Physics Informed Neural Networks(PINNs) to study the dependence of the strain energy density on the invariants of the Cauchy Green Deformation tensor in hyperelasticity.**



DEFINING THE BOUNDARY VALUE PROBLEM(BVP)



Governing Equations:

$$
\begin{aligned}
\underline{\nabla}_{\underline{X}} \cdot \underline{\underline{P}}(\underline{X}) &= \underline{0} \quad \text{in } \Omega, \\
\underline{u}(\underline{X}) &= \underline{\tilde{u}} \quad \text{on } \partial\Omega_u, \\
\underline{t}(\underline{X}) = \underline{\underline{P}}(\underline{X})\underline{n}(\underline{X}) &= \underline{0} \quad \text{on } \partial\Omega_t,
\end{aligned}
$$

P(X) : Piola-Kirchhoff Stress Tenso
u(X) : Displacement Field
ũ : Displacement B.C.
t(X) : Traction B.C.
Ω : Domain (Square Plate)
$\partial\Omega_u$: Displacement Bound
$\partial\Omega_t$: Traction Boundary

CONSTITUTIVE MODEL

$$\underline{\underline{F}}(\underline{X}) = \underline{\underline{I}} + \nabla_{\underline{X}} \underline{u}(\underline{X})$$

$$\underline{\underline{C}}(\underline{X}) = \underline{\underline{F}}(\underline{X})^{\mathrm{T}} \underline{\underline{F}}(\underline{X})$$

$$W(\underline{\underline{C}}(\underline{X})) = W\left(I_1(\underline{\underline{C}}), I_2(\underline{\underline{C}}), I_3(\underline{\underline{C}})\right)$$

$$I_1(\underline{\underline{C}}(\underline{X})) = \mathrm{tr}(\underline{\underline{C}}) + 1,$$

$$I_2(\underline{\underline{C}}(\underline{X})) = \frac{1}{2}\left[\left(\mathrm{tr}(\underline{\underline{C}}) + 1\right)^2 - \left(\mathrm{tr}(\underline{\underline{C}}^2) + 1\right)\right]$$

$$I_3(\underline{\underline{C}}(\underline{X})) = \det(\underline{\underline{C}})$$

$$W\left(I_1, I_2, I_3\right) = \underline{Q}^{\mathrm{T}}\left(I_1, I_2, I_3\right) \underline{\beta}$$

F(X) : Deformation Gradient

C(X) : Cauchy Green Deformation Tensor

W(C(X)) : Strain Energy Density Function

I_1, I_2, I_3 : Invariants of C(X)

Q(X) : Dictionary of potential non-linear functions of I_1,I_2 and I_3

β : Unknown Feature parameters of the functions i

CONSTITUTIVE MODEL:

$$\underline{Q}(I_1, I_2, I_3) = \underbrace{\left[(\bar{I}_1 - 3)^i(\bar{I}_2 - 3)^{j-i} \ : \ j \in \{1, \dots, N_a\}, i \in \{0, \dots, j\}\right]^{\mathrm{T}}}_{\text{Generalized Mooney-Rivlin features}}$$

$$\oplus \underbrace{\left[(J-1)^{2b} \ : \ b \in 1, \dots, N_b\right]^{\mathrm{T}}}_{\text{Volumetric deformation features}} \oplus \underbrace{\left[\log(\bar{I}_2/3)\right]}_{\text{Logarithmic feature}}$$

$$J = \det(\underline{\underline{F}}) = I_3^{1/2}, \ \bar{I}_1 = J^{-2/3} I_1, \ \bar{I}_2 = J^{-4/3} I_2,$$

$$\underline{\underline{P}}(\underline{X}) = \frac{\partial W(\underline{X})}{\partial \underline{\underline{F}}(\underline{X})} = \frac{\partial \underline{Q}^{\mathrm{T}}(I_1, I_2, I_3)\underline{\beta}}{\partial \underline{\underline{F}}} = \sum_{k=1}^{3} \frac{\partial \underline{Q}^{\mathrm{T}}(I_1, I_2, I_3)\underline{\beta}}{\partial I_k} \ \frac{\partial I_k(\underline{\underline{C}})}{\partial \underline{\underline{F}}}$$

$$\underline{r}(\underline{X}) = \underline{\underline{P}}(\underline{X})\underline{n}(\underline{X}), \quad \underline{X} \in \partial\Omega_u$$

$$\underline{R} = \int_{\underline{X} \in \partial\Omega_u} \underline{r}(\underline{X}) d(\partial\Omega_u)$$

r(X) : Reaction Force = Internal Traction

R : Net reaction force on boundary

PHYSICS INFORMED NEURAL NETWORK

Optimization Problem :

$$\phi^* = \arg\min_{\phi} \mathcal{L}(\phi).$$ **$\Phi^* = \{W^*, \beta^*\}$**

Where,

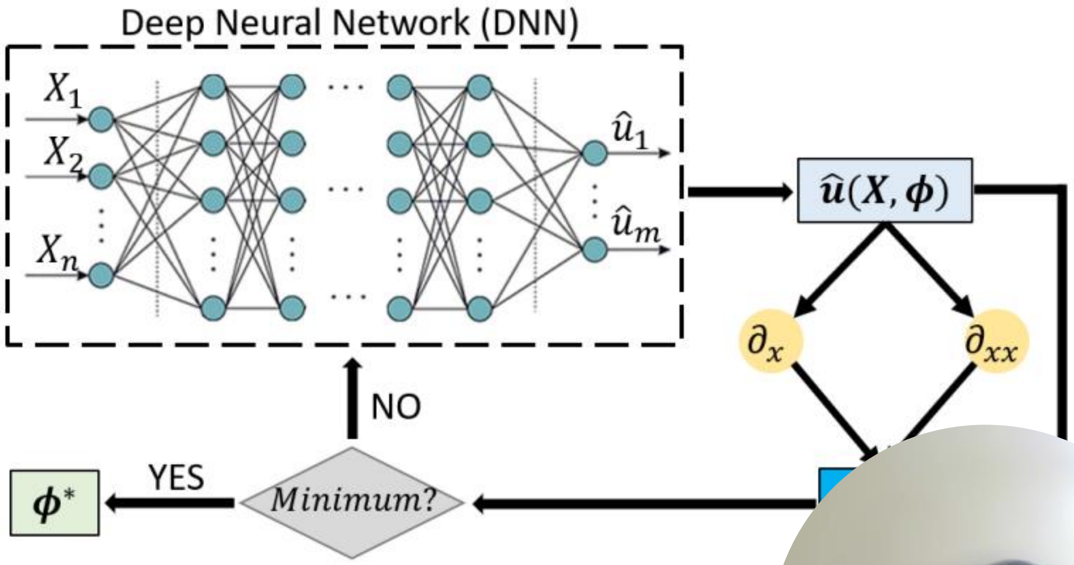
$$\mathcal{L} = MSE_G + \lambda_u\, MSE_u + \lambda_t\, MSE_t + \lambda_e\, MSE_e$$

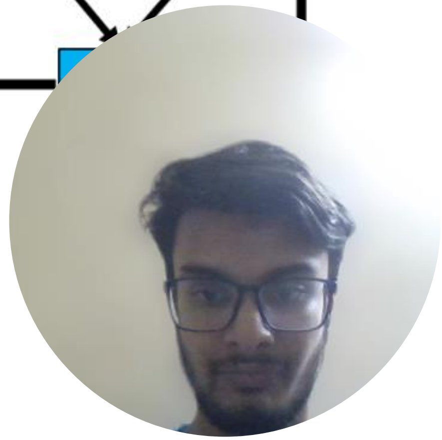
$$MSE_G = \frac{1}{N_G}\sum_{j=1}^{N_G} \| (\partial_t + \mathcal{N})\, \hat{\boldsymbol{u}}(t, \boldsymbol{X}; \phi) \|^2, \quad (t_j, \boldsymbol{X}_j) \in [0, T] \times \Omega$$

$$MSE_u = \frac{1}{N_u}\sum_{j=1}^{N_u} \|\hat{\boldsymbol{u}}(t, \boldsymbol{X}; \phi) - \overline{\boldsymbol{u}}\|^2, \quad (t_j, \boldsymbol{X}_j) \in [0, T] \times \Gamma_u$$

$$MSE_t = \frac{1}{N_t}\sum_{j=1}^{N_t} \|\hat{\boldsymbol{t}}(t, \boldsymbol{X}; \phi) - \overline{\boldsymbol{t}}\|^2, \quad (t_j, \boldsymbol{X}_j) \in [0, T] \times \Gamma_t$$

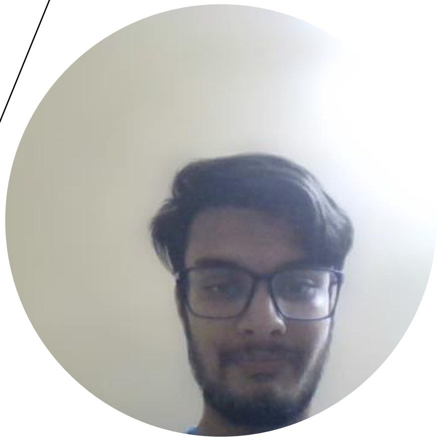
$$MSE_e = 1/N_e \sum \|\hat{u}(t,X;\Phi) - u(t,X)\|^2 \quad (t,X) \in [0,T] \times \Omega$$





ALGORITHM USED

Input: Physical domain, BCs, and DNN
  Material parameters β
  Sample points $\boldsymbol{X}_{int}$ from Ω
  Sample points $\boldsymbol{X}_u$ from Γ_u
  Sample points $\boldsymbol{X}_t$ from Γ_t
  Neural network architecture
  Neural network hyperparameters
  Optimizer (Adam followed by L-BFGS)
Initialization: Initial weights and biases of the DNN
Output: Optimized weights and biases of the DNN
while *Not minimized* **do**
  Obtain $\hat{\boldsymbol{u}}$ from the DNN
  Compute $\boldsymbol{\nabla}_{\boldsymbol{X}}\hat{\boldsymbol{u}}$ using automatic differentiation
  Compute $\boldsymbol{F} = \boldsymbol{I} + \boldsymbol{\nabla}_{\boldsymbol{X}}\hat{\boldsymbol{u}}$
  Compute $J = \det(\boldsymbol{F})$, $\boldsymbol{C}$, I_1, and $\boldsymbol{P}$
  if $\boldsymbol{X}_{int}$ **then**
    Compute $\boldsymbol{\nabla}_{\boldsymbol{X}} \cdot \boldsymbol{P}$ using automatic differentiation
    Calculate MSE_G
  else if $\boldsymbol{X}_t$ **then**
    Compute $\boldsymbol{t} = \boldsymbol{P} \cdot \boldsymbol{N}$
    Calculate MSE_t
  else
    Calculate MSE_u
  Calculate loss function
  Update the weights and biases
end



RESULTS

δ = 30cm; Shape of NN: 2x50x10x2; Epochs: 3

```
The beta parameters of the model are,
Index: 36, Beta: -0.03032270073890686
Index: 2, Beta: -0.02136821858584881
Index: 37, Beta: -0.02109895646572113
Index: 43, Beta: -0.020868590101599693
Index: 5, Beta: -0.019864879548549652
Index: 1, Beta: -0.01938806287944317
Index: 4, Beta: -0.019118856638669968
Index: 3, Beta: -0.0182194262742962
Index: 9, Beta: -0.014078034088015556
Index: 8, Beta: -0.013385855592787266
Index: 7, Beta: -0.012614504434168339
Index: 6, Beta: -0.011757155880331993
Index: 14, Beta: -0.006524367723613977
Index: 13, Beta: -0.005536359734833240
Index: 12, Beta: -0.004533857572823763
Index: 11, Beta: -0.003545812796801328
Index: 10, Beta: -0.002608978422358632
Index: 38, Beta: -0.001790193375200033
Index: 39, Beta: -0.000452453416073694
Index: 25, Beta: -0.000432354747317731
Index: 26, Beta: -0.000312241812935099
Index: 24, Beta: -0.00025708184693939984
Index: 15, Beta: 0.00019133489695377648
Index: 17, Beta: -0.0001383695926051587
...
Index: 31, Beta: 1.0179139280808158e-05
Index: 42, Beta: 9.8017535492545e-06
Index: 40, Beta: -4.8810870794113725e-06
Index: 30, Beta: -4.519060894381255e-07
```

```
===========Epoch================> 2
Batch: 0, Loss: 0.11956954747438431
Batch: 1, Loss: 0.12356039881706238
Batch: 2, Loss: 0.11865314096212387
Batch: 3, Loss: 0.12199420481920242
Batch: 4, Loss: 0.08449181914329529
Batch: 5, Loss: 0.08309056609869003
Batch: 6, Loss: 0.08225032687187195
Batch: 7, Loss: 0.08203616738319397
Batch: 8, Loss: 0.08343254774808884
Batch: 9, Loss: 0.08178643137216568
Batch: 10, Loss: 0.08209459483623505
Batch: 11, Loss: 0.08202366530895233
Batch: 12, Loss: 0.08208108693361282
Batch: 13, Loss: 0.08199996501207352
Batch: 14, Loss: 0.08186598867177963
Batch: 15, Loss: 0.08210473507642746
Batch: 16, Loss: 0.08133222162723541
Batch: 17, Loss: 0.0812196359038353
Batch: 18, Loss: 0.08141925930976868
Batch: 19, Loss: 0.08071540296077728
Batch: 20, Loss: 0.08099354058504105
Batch: 21, Loss: 0.08069579303264618
Batch: 22, Loss: 0.08042990416288376
Batch: 23, Loss: 0.08109515905380249
Batch: 24, Loss: 0.08166274428367615
Training Time:  1297.6093933582306
```

δ = 10cm; Shape of NN: 2x50x10x2; Epochs: 3

```
The beta parameters of the model are,
Index: 36, Beta: -0.030564846470952034
Index: 37, Beta: -0.02375234104692936
Index: 2, Beta: -0.021825416013598442
Index: 43, Beta: -0.021552903577685356
Index: 5, Beta: -0.020494718104600906
Index: 4, Beta: -0.019802140071988106
Index: 1, Beta: -0.01978423446416855
Index: 3, Beta: -0.018952779471874237
Index: 9, Beta: -0.016391858458518982
Index: 8, Beta: -0.01585765741765499
Index: 7, Beta: -0.015241487883031368
Index: 6, Beta: -0.014529050327837467
Index: 14, Beta: -0.009953312575817108
Index: 13, Beta: -0.00916887167841196
Index: 12, Beta: -0.008285663090646267
Index: 11, Beta: -0.007318933960050344
Index: 10, Beta: -0.006290603894740343
Index: 38, Beta: -0.005618561524897814
Index: 20, Beta: -0.001009539584629237
Index: 25, Beta: 0.00063336017774418
Index: 39, Beta: 0.0006062289467081428
Index: 24, Beta: 0.0005178358405828476
Index: 26, Beta: 0.0005088273901492357
Index: 19, Beta: -0.00045949884224683046
...
Index: 15, Beta: 1.3964428944746032e-05
Index: 41, Beta: -1.0297783774149138e-05
Index: 33, Beta: -6.92333560436964e-06
Index: 42, Beta: 2.956993739644531e-06
```

```
===============Epoch==========> 2
Batch: 0, Loss: 0.23695941269397736
Batch: 1, Loss: 0.2429947853088379
Batch: 2, Loss: 0.209515780210495
Batch: 3, Loss: 0.21302184462547302
Batch: 4, Loss: 0.08621767908334732
Batch: 5, Loss: 0.08333951979875565
Batch: 6, Loss: 0.0823962464928627
Batch: 7, Loss: 0.08243657648563385
Batch: 8, Loss: 0.08397434651851654
Batch: 9, Loss: 0.08241042494773865
Batch: 10, Loss: 0.08270702511072159
Batch: 11, Loss: 0.08274658024311066
Batch: 12, Loss: 0.0825682207942009
Batch: 13, Loss: 0.08292887359857559
Batch: 14, Loss: 0.08282697200775146
Batch: 15, Loss: 0.08333796262741089
Batch: 16, Loss: 0.0820881649851799
Batch: 17, Loss: 0.08219333738088608
Batch: 18, Loss: 0.08248143643140793
Batch: 19, Loss: 0.08116885274648666
Batch: 20, Loss: 0.08178413659334183
Batch: 21, Loss: 0.0813763290643692
Batch: 22, Loss: 0.08096446096897125
Batch: 23, Loss: 0.08232741057872772
Batch: 24, Loss: 0.08351408690214157
Training Time:  1725.0156118869781
```

δ = 50cm; Shape of NN: 2x50x10x2; Epochs: 3

```
The beta parameters of the model are,
Index: 36, Beta: -0.028982507064938545
Index: 2, Beta: -0.019829921424388885
Index: 43, Beta: -0.018234340474009514
Index: 1, Beta: -0.01777459867298603
Index: 5, Beta: -0.016955263912677765
Index: 4, Beta: -0.015920501202344894
Index: 3, Beta: -0.014717147685587406
Index: 37, Beta: -0.012779138050973415
Index: 9, Beta: -0.008368752896785736
Index: 8, Beta: -0.007433408405631781
Index: 7, Beta: -0.006407493259757757
Index: 6, Beta: -0.005308269057422876
Index: 20, Beta: 0.0007336471462622285
Index: 19, Beta: 0.0004912199219688773
Index: 16, Beta: -0.00043488582014106214
Index: 17, Beta: -0.0003187975380569696
Index: 15, Beta: -0.0002728163381107151
Index: 14, Beta: -0.00021215445303823
Index: 25, Beta: 0.00018640853522811085
Index: 26, Beta: 0.00016077274631243
Index: 10, Beta: -0.00014509752509184182
Index: 22, Beta: -0.0001206896995427087
Index: 23, Beta: -9.17998404474929e-05
Index: 11, Beta: -6.934937846381217e-05
...
Index: 31, Beta: 6.952626790734939e-06
Index: 41, Beta: -6.922108696016949e-06
Index: 40, Beta: 4.654953954741359e-06
Index: 12, Beta: -1.8227947293780744e-06
```

```
===============Epoch==========> 2
Batch: 0, Loss: 0.08524735271930695
Batch: 1, Loss: 0.08730721473693848
Batch: 2, Loss: 0.0854186937212944
Batch: 3, Loss: 0.0882968083024025
Batch: 4, Loss: 0.08352784067392349
Batch: 5, Loss: 0.08299772441387177
Batch: 6, Loss: 0.082362599670887
Batch: 7, Loss: 0.08196030557155609
Batch: 8, Loss: 0.0830799862742424
Batch: 9, Loss: 0.08173023164272308
Batch: 10, Loss: 0.08190938830375671
Batch: 11, Loss: 0.08175679296255112
Batch: 12, Loss: 0.08187545090913773
Batch: 13, Loss: 0.08159300684928894
Batch: 14, Loss: 0.08142910897731781
Batch: 15, Loss: 0.0814986303448677
Batch: 16, Loss: 0.08102010935544968
Batch: 17, Loss: 0.0808710977435112
Batch: 18, Loss: 0.08091156929731369
Batch: 19, Loss: 0.08058905601501465
Batch: 20, Loss: 0.08063548803329468
Batch: [illegible]38646936417
Ba[illegible]470871
[illegible]66196
[illegible]958
```

OBSERVATIONS AND ANALYSIS

- **We observe that in all the 3 cases, the β components with higher weightage are similar in magnitude but not the same. This proves that while PINNs produce consistent results when used in solving PDEs, they are not necessarily accurate.**
- **The features from the library that are significant are : 1,2,3,4,5,36,37, and 43.**
- **Also, the losses are similar i.e. 0.080 in all the cases where the starting loss was 0.335, 0.3346, 0.46950 respectively.**

CONCLUSION

- We observed the ability of Neural Networks to solve partial differential equations whose analytical solutions are too expensive to compute.
- The method used DCM(Deep Collocation Method) does not involve any sort of traditional meshing of the domain as in the case of other numerical methods such FEM and CFD.
- However, we see that the results are quite sensitive to the hyperparameters of the model.
- Cross-validation and testing is required to accurately obtain these hyperparameters for generalizing the model.

THANK YOU