

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/388263534>

Hyperparameter Optimization for Causal Marching Physics Informed Neural Network for Hyperelasticity

Conference Paper · October 2024

DOI: 10.23967/eccomas.2024.157

CITATION

1

READS

22

3 authors, including:



Vikrant Pratap

Ollscoil na Gaillimhe – University of Galway

6 PUBLICATIONS 6 CITATIONS

SEE PROFILE



Bharat Bhushan Tripathi

Ollscoil na Gaillimhe – University of Galway

32 PUBLICATIONS 125 CITATIONS

SEE PROFILE

HYPERPARAMETER OPTIMIZATION FOR CAUSAL MARCHING PHYSICS INFORMED NEURAL NETWORK FOR HYPERELASTICITY

Vikrant Pratap¹, Michael D. Gilchrist² and Bharat B. Tripathi¹

¹ School of Mathematical & Statistical Sciences,
University of Galway, Galway, Ireland.
e-mail: bharat.tripathi@universityofgalway.ie, www.universityofgalway.ie/tripathilab/

² School of Mechanical & Materials Engineering,
University College Dublin, Dublin, Ireland.

Key words: Computational Mechanics, Physics Informed Neural Networks, Hyperelasticity

Summary. This study presents an approach for hyperparameter optimization in the Causal-Marching Physics-Informed Neural Networks (CMPINNs) framework, specifically designed to model hyperelasticity. Physics-Informed Neural Networks (PINNs) are powerful tools for solving governing partial differential equations (PDEs) in physical systems. The CMPINNs model proposed in this work enhances the PINN framework by minimizing the residuals of the governing PDEs while enforcing the boundary conditions for the nonlinear mechanical responses of hyperelasticity. We study the accuracy of using CMPINNs to solve the Neo-Hookean hyperelastic model using soft and hard constrained boundary conditions. Additionally, the study presented a hyperparameter optimization for CMPINNs to identify the best suitable set of hyperparameters for deformation like biaxial compression. This optimization process ensures that the CMPINN effectively captures the complex stress-strain relationships in hyperelastic materials under deformation. This research advances the development of robust, physics-informed computational models for hyperelastic materials, reducing reliance on labelled or synthetic data.

1 Introduction

Traumatic brain injury (TBI) is a significant global health and economic issue, often caused by external forces from falls, assaults, and road accidents, with contact sports being a leading cause among youth [1]. In 2019, there were 27.16 million new TBI cases globally, a significant rise from 18.85 million in 1990. An estimated 64 to 74 million people worldwide suffer from TBI. Europe reports 235 to 262 cases per 100,000 people yearly, including 19,000 in Ireland [1].

The Sport Concussion Assessment Tool 5th Edition is widely used for evaluating sports-related concussions in individuals over 13 years old. Due to the high incidence of TBI, various computational models have been developed to estimate the brain's mechanical response to different loads, relying heavily on hyperelastic material models [2]. Hyperelastic materials are modelled using the first Piola-Kirchhoff stress tensor (\mathbf{P}) derived from strain-energy density functions [3]. Brain tissue, modelled as an incompressible hyperelastic solid under quasi-static loading, utilises finite deformation theory, which is nonlinear and increases computational demands.

Partial differential equations are among the complex challenges in scientific computing, addressed through the Finite Difference Method and spectral methods. These techniques rely on polynomials and basic functions set on a finite grid but struggle with the Curse of Dimensionality. Mesh-based methods can introduce discretisation errors due to insufficient grid resolution. Recent advancements in artificial intelligence have introduced Deep Learning techniques for modelling partial differential equations [4]. While Neural Networks have been known since the 1990s for their flexibility in solving PDEs, Deep Neural Networks (DNNs) have become particularly popular for predicting solutions to physics-based problems. However, DNNs require extensive training datasets and involve challenging hyperparameter optimization. To overcome these limitations, Physics-Informed Neural Networks (PINNs) have been developed as specialized DNNs. PINNs embed the underlying physics along with initial and boundary conditions directly into the loss function, using automatic differentiation to compute partial derivatives to space and time, thus offering a mesh-free solution to the curse of dimensionality [5].

PINNs use irregularly sampled points instead of a grid for spatiotemporal stepping and employ the nonlinear representation of neural networks (NNs) to approximate PDE solutions, unlike the linear piecewise polynomials in traditional methods [6]. PINNs focus on optimizing NN weights and biases rather than grid point values, incorporating the PDE into the loss function rather than as an algebraic matrix system. This allows PINNs to use gradient optimizers as error minimizers. One advantage of PINNs is that they do not require precomputed solutions for training; instead, they use collocation points as the training set. However, training PINNs can be difficult for multiphysics problems requiring significant experimentation to identify the apt model for the given problem.

This work introduces a causal-marching physics-based machine learning model designed to capture the nonlinear response of the Neo-Hookean hyperelastic model. It aims to simulate brain tissue response under deformation modelled using the Neo-Hookean hyperelastic model undergoing sequential training. The model retains previous loss minimizations and focuses on sequentially updated loading steps using a self-adaptive loss coefficient. The generalized model can capture hyperelastic material responses under various loading cases. This research compares the soft and hard constraints for implementing BCs in a homogeneous domain. The article is structured as follows: Section 2 provides a brief overview of homogeneous deformation, hyperelastic materials and governing equations; Section 3 details the different BC constraints implemented using CMPINNs; Section 4 presents stress response and loss minimisation results; and Section 5 concludes the findings.

2 Theoretical Background

This research aims to predict the deformation of Neo-Hookean hyperelastic material under biaxial compression conditions using a machine-learning model without labelled data. It also demonstrate the performance comparison of CMPINNs with soft and hard constrained BCs. This section explores the theoretical background of biaxial compression alongside the Neo-Hookean hyperelastic material model.

2.1 Homogeneous Deformation

Homogeneous deformation occurs when the straight filaments of a material stay straight even after undergoing deformation. The function χ maps the body in reference (\mathbf{X}) and current

configuration (\mathbf{x}) as $\mathbf{x} = \chi(\mathbf{X}, t)$. The deformation gradient \mathbf{F} is a second-order tensor, that maps the gradient of the motion field $\chi(\mathbf{X}, t)$. The right (\mathbf{C}) and the left (\mathbf{B}) Cauchy Green deformation tensors are represented as $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ and $\mathbf{B} = \mathbf{F} \mathbf{F}^T$ respectively. The invariants of \mathbf{C} are crucial for maintaining objectivity and determining volumetric changes, where $\det(\mathbf{F}) = 1$ indicates incompressibility. We have considered a biaxial compression test case for our study as it exhibits higher non-linearity with increasing order of the hyperelastic model. The biaxial compression is characterised as,

$$x = \lambda X, \quad y = \lambda Y, \quad z = \frac{1}{\lambda^2} Z, \quad \forall \lambda \in \{0.5 \quad 1\}. \quad (1)$$

This deformation arises when a body is compressed along two mutually perpendicular axes. This results in a reduction in length in the direction of applied compression as shown in Fig. 1.

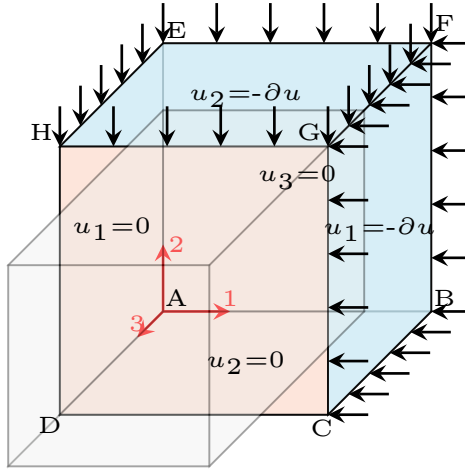


Figure 1: Boundary and applied loading conditions, along with the deformed configuration under biaxial compression.

2.2 Hyperelastic Material Models

Hyperelastic models describe materials undergoing large elastic deformations, typically applied to elastomers, rubbers, and soft tissues [7]. The strain energy function quantifies the energy stored based on the deformation, relating stress to strain. The strain energy density function for an incompressible polynomial model, known as the Rivlin model [8], is expressed as:

$$W = \sum_{i,j=0}^N C_{ij} (I_1 - 3)^i (I_2 - 3)^j. \quad (2)$$

Here, C_{ij} in (2) parameterises the material properties and is obtained through experimental studies.

The Neo-Hookean hyperelastic material model is a simplified isotropic model depending, on the first invariant of the right Cauchy-Green deformation tensor (\mathbf{C}). The neo-Hookean model uses the shear modulus (μ) to define the strain energy function. When $N = 1$ and

$C_{01} = C_{11} = 0$ in the Rivlin model, the expression simplifies to represent the Neo-Hookean model for incompressible material as:

$$W = C_{10} (I_1 - 3), \quad C_{10} = \frac{\mu}{2}. \quad (3)$$

2.3 Governing Equations

The mechanical response of a body to external forces is governed by partial differential equations representing equilibrium, boundary conditions, and constitutive relations. In this work, CMPINNs are used to solve these governing equations for hyperelastic deformations, optimising neural network parameters through non-dimensionalization to ensure uniformity among loss terms. This approach leverages the governing physics to predict material behaviour without relying on labelled data.

The governing differential equations and boundary conditions for a mechanical body undergoing quasi-static deformation from a stress-free reference configuration are as follows:

$$\nabla_{\mathbf{X}} \cdot \mathbf{P} + \mathbf{f}_B = \mathbf{0}, \quad \mathbf{X} \in \Omega, \quad (4)$$

$$\mathbf{u} = \mathbf{u}_b, \quad \mathbf{X} \in \Gamma_u, \quad (5)$$

$$\mathbf{P} \cdot \mathbf{N} = \mathbf{f}_t, \quad \mathbf{X} \in \Gamma_t, \quad (6)$$

where \mathbf{P} denotes the first Piola-Kirchhoff stress tensor, \mathbf{f}_B represents the body force, and $\nabla_{\mathbf{X}}$ is the gradient operator. The outward normal unit vector in the reference configuration is given by \mathbf{N} . The terms \mathbf{f}_t and \mathbf{u}_b correspond to the traction and displacement boundary conditions, respectively. Ω refers to the material domain, while Γ_u and Γ_t are the surfaces where displacement and traction boundary conditions are applied. The constitutive relation for a material undergoing hyperelastic deformation is expressed as follows:

$$\mathbf{P} = \frac{\partial W(\mathbf{F})}{\partial \mathbf{F}}. \quad (7)$$

3 Numerical Methods

We have developed a causal-marching physics-informed neural network for boundary value problems associated with the hyperelastic material model. The common input layer consists of four neurons, representing 3D geometrical coordinates and the applied load (deformation in this context). In contrast, the output layer comprises three neurons, tasked with predicting the three displacement components at any given spatial coordinate for each material domain. The tanh activation function is employed to capture the nonlinear response of the deformation, as the ReLU activation function struggles with the stiff nonlinearity in the biaxial compression. The predicted displacement is used to compute the predicted deformation gradient as $\hat{\mathbf{F}} = \mathbf{I} + \text{Grad}(\hat{\mathbf{u}})$. The $\hat{\mathbf{F}}$ is used to calculate the predicted first Piola-Kirchhoff stress tensor ($\hat{\mathbf{P}}$), which is a function of the deformation gradient as stated in (7). Additionally, the residuals of $\nabla_{\mathbf{X}} \cdot \hat{\mathbf{P}}$, serving as the governing equation, are obtained and termed as \mathcal{L}_D . The errors associated with enforcing the displacement and traction boundary conditions are referred to as \mathcal{L}_U and \mathcal{L}_T , respectively, as illustrated in Figs. 2 - 3. The error between the predicted $\hat{\mathbf{F}}$ and the actual \mathbf{F} is minimised to enforce the incompressibility condition using \mathcal{L}_F .

The process involves summing up all loss terms and minimizing them sequentially using both Adam and L-BFGS (limited Broyden Fletcher Goldfarb Shanno) optimizers to fine-tune the

network’s hyperparameters. Training continues iteratively until the model is fully optimized for the given loading condition. This CMPINNs framework is implemented in TensorFlow through the deep neural network, and the network parameters are optimised using Adam and L-BFGS optimisers [9, 10]. Adam, a first-order optimizer, quickly converges with adaptive learning rates, while L-BFGS, a second-order optimizer, refines the model by approximating the Hessian matrix of the objective function.

3.1 Soft Boundary Enforcement in CMPINNs:

Soft boundary enforcement integrates boundary conditions into the model by adding them as penalty terms within the loss function rather than enforcing them exactly through the neural network’s architecture or output transformation. This approach allows the network to prioritize fitting the physics-based constraints while still allowing some flexibility in how strictly these conditions are met. The boundary conditions are satisfied depending on the relative weighting of the penalty terms in the overall loss function.

Consider the governing equations for a body undergoing hyperelastic deformation defined in (4)-(6) over the domain Ω . The residual of (4) at any step of the training is given by,

$$\mathcal{L}_D = \sum_{i=1}^{N_e} \frac{1}{\lambda_i^D} D_i, \quad D_i = \left[\frac{1}{M_l} \sum_{j=1}^{M_l} |\nabla_{\mathbf{x}} \cdot \hat{\mathbf{P}}_j|^2 \right]_i. \quad (8)$$

Here λ_i^D is the trainable penalty coefficient with i ranging from 1 to $N_e = 3$, corresponding to three components of $\nabla_{\mathbf{x}} \cdot \hat{\mathbf{P}}$. D_i denotes the residuals of x, y and z components of the governing equation, with M_l internal collocation points. The total displacement boundary loss incurred in satisfying the displacement boundary condition at ‘ N_f ’ faces of the deformed body at any steps of training is given by,

$$\mathcal{L}_U = \sum_{i=1}^{N_f} \frac{1}{\lambda_i^U} U_i, \quad U_i = \left[\frac{1}{M_d} \sum_{j=1}^{M_d} |\hat{\mathbf{u}}_j - \mathbf{u}_j|^2 \right]_i. \quad (9)$$

λ_i^U denotes the penalty coefficient for the loss term in satisfying displacement boundary condition at i^{th} face of a body having $N_f = 6$ faces for a cubical domain. U_i represents the average loss in satisfying displacement boundary condition for all the M_d points of i^{th} face. The total loss incurred in satisfying the traction boundary condition is given as,

$$\mathcal{L}_T = \sum_{i=1}^{N_t} \frac{1}{\lambda_i^T} T_i, \quad T_i = \left[\frac{1}{M_t} \sum_{j=1}^{M_t} |\hat{\mathbf{P}}_j - \mathbf{P}|^2 \right]_i. \quad (10)$$

where λ_i^T represents the penalty coefficient of the loss term in satisfying the traction boundary condition at i^{th} boundary face where traction is prescribed and T_i yields an average loss in satisfying traction boundary conditions at M_t points of i^{th} boundary faces. N_t represents the number of faces where the traction is specified. The loss term penalizing the losses in satisfying the gradient of deformation (\mathbf{F}) at N_f faces besides the internal domain of the body at any steps

of the training is given as,

$$\mathcal{L}_F = \sum_{i=1}^{N_f} \frac{1}{\lambda_i^F} \mathbf{F}_i + \frac{1}{\lambda_{N_f+1}^F} \mathbf{F}_d, \quad \mathbf{F}_i = \left[\frac{1}{M_s} \sum_{j=1}^{M_s} |\hat{\mathbf{F}}_j - \mathbf{F}_j|^2 \right]_i, \quad \mathbf{F}_d = \frac{1}{M_f} \sum_{j=1}^{M_f} |\hat{\mathbf{F}}_j - \mathbf{F}_j|^2 \quad (11)$$

where, λ_i^F , where $i = 1, \dots, N_f$, and $\lambda_{N_f+1}^F$ are the penalty coefficients for the loss in satisfying the gradient of deformation at i^{th} surface of the body (\mathbf{F}_i) and inside the domain (\mathbf{F}_d), respectively. M_f represents all the points inside the domain, while M_s denotes the number of points on the surface of the body, where the loss in satisfying the gradient of the deformation field is penalized.

In soft boundary enforcement, the neural network's output $\hat{u}(\mathbf{x})$ is not inherently designed to satisfy the given boundary conditions. Instead, the boundary conditions are included as part of the loss function used to train the network. The total loss for the body undergoing hyperelastic deformation is given by

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_U + \mathcal{L}_T + \mathcal{L}_F \quad (12)$$

The schematic of causal-marching physics-informed neural networks incorporating hyperelasticity using soft constrained boundary conditions is illustrated in Fig. 2.

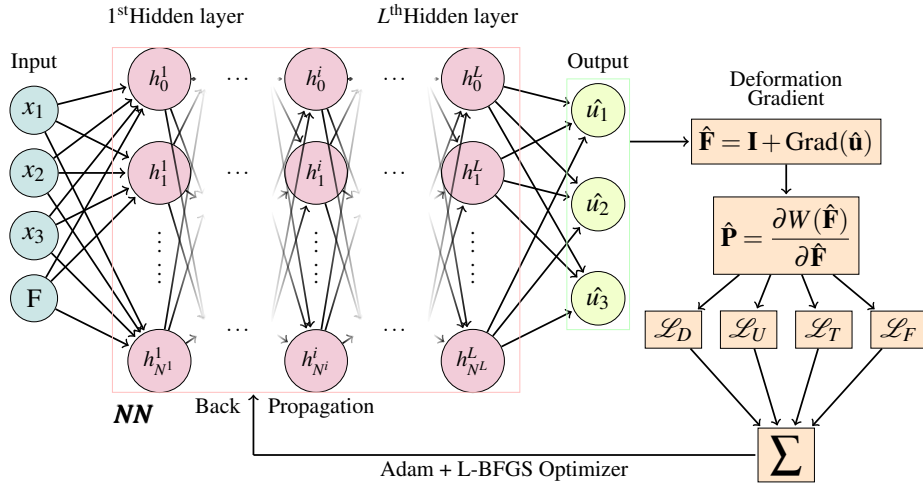


Figure 2: Detailed schematic of causal-marching physics-informed Neural Networks (CMPINNs) incorporating hyperelasticity using soft constrained BCs.

3.2 Hard Boundary Enforcement in CMPINNs:

Hard boundary enforcement ensures that the boundary conditions of a partial differential equation are strictly satisfied throughout the training process. This method integrates the boundary conditions directly into the architecture of the neural network or manipulates the output to inherently satisfy the boundary conditions by construction [11]. The output $u(\mathbf{x})$ of the causal-marching physics-informed neural networks incorporating hyperelasticity using hard

constrained boundary conditions as illustrated in Fig. 3:

$$u(\mathbf{x}) = \eta(\mathbf{x}) + \phi(\mathbf{x}) \cdot \hat{u}(\mathbf{x}) \quad (13)$$

where $\eta(\mathbf{x})$ is a particular solution that satisfies the boundary conditions by itself; $\phi(\mathbf{x})$ is a smooth function that is zero on the boundary $\partial\Omega$; $\hat{u}(\mathbf{x})$ is the output of the neural network that is unconstrained by the boundary conditions. This construction ensures that $u(\mathbf{x}) = \mathbf{u}_b$ on $\partial\Omega$, satisfying the boundary condition exactly by the design of $\phi(\mathbf{x})$ and $\eta(\mathbf{x})$. The total loss for the hard constrained BCs for the body undergoing hyperelastic deformation is given by

$$\mathcal{L} = \mathcal{L}_D + \mathcal{L}_T + \mathcal{L}_F \quad (14)$$

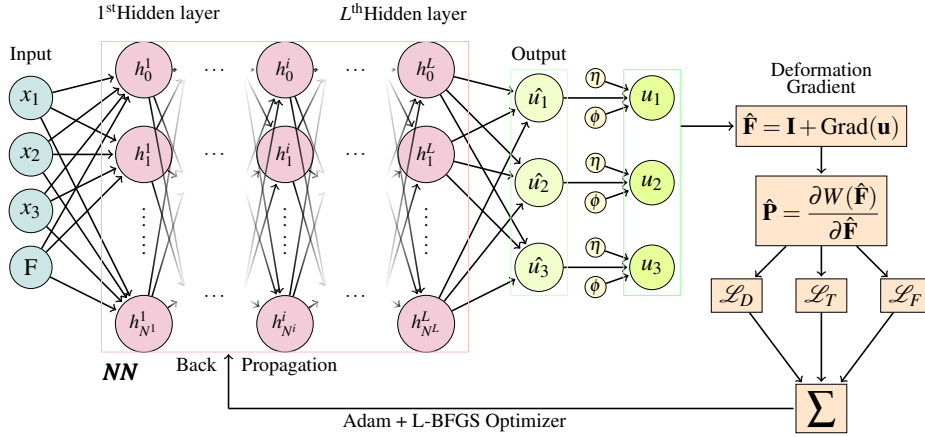


Figure 3: Detailed schematic of causal-marching physics-informed Neural Networks (CMPINNs) incorporating hyperelasticity using hard constrained BCs.

4 Results

A homogeneous, isotropic, incompressible unit cube with a Poisson's ratio, $\nu = 0.499$, material density, $\rho = 1000 \text{ kg/m}^3$ and shear modulus, $\mu = 1401 \text{ Pa}$ is considered to represent tissue-mimicking soft solids. The unit cube is subjected to biaxial compression loading to analyse the non-linear stress response of the material. The stress response is studied under hard and soft boundary constraints for the Neo-Hookean hyperelastic model. The hyperparameter tuning is also carried out to find out the best suitable parameters for both models.

4.1 Biaxial compression for Neo-Hookean model: soft and hard constrained BCs

The performance of CMPINNs for the Neo-Hookean hyperelastic model undergoing biaxial compression was analysed for soft and hard constrained BCs in Figs. 4 and 5. Fig. 4a and 5a compare CMPINN's performance to the analytical solution for the Neo-Hookean hyperelastic model with soft and hard constrained BCs respectively. The best set of hyperparameters for the soft and hard constrained BCs are mentioned in Table 2, demonstrating CMPINN's ability to capture the material's response despite nonlinearity. Additionally, Fig. 4b shows that with soft constrained boundary conditions, the Neo-Hookean model's relative error remains below

0.5% under higher compression above $\lambda = 0.6$. In contrast, Fig. 5b shows that with hard constrained boundary conditions, the relative error shoots above 1% under higher compression above $\lambda = 0.6$. Figs. 4c and 5c illustrate the Neo-Hookean model's overall loss and its loss components in both cases. The overall loss and loss components converge to the order 10^{-5} . The sudden spike in losses is observed when the optimiser switches from Adam to L-BFGS. The root mean square error (RMSE) for soft and hard constrained BCs Neo-Hookean model are 0.12% and 0.39% respectively. The hard constrained model poses a higher relative error as compared to the soft constrained model.

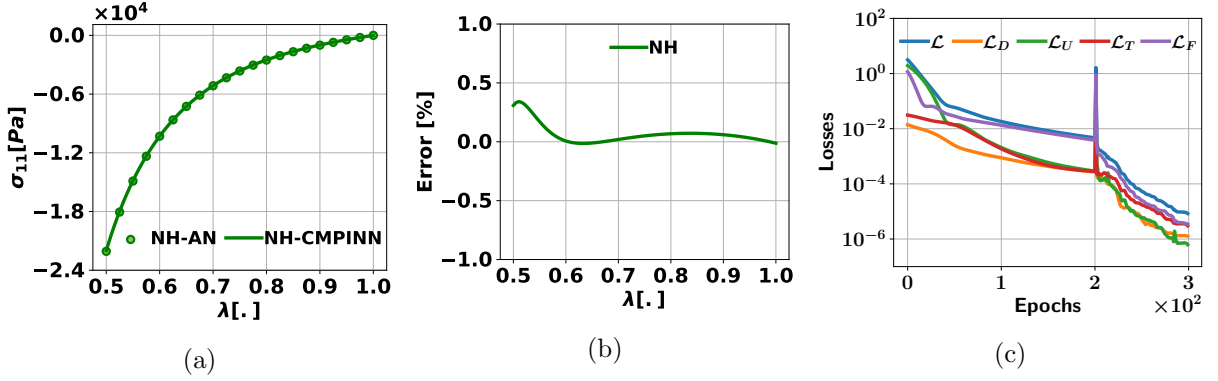


Figure 4: Comparison of CMPINN with soft constraints BC and analytical (AN) solution for the Neo-Hookean (NH) model undergoing biaxial compression: (a) Stress response, (b) Relative Error, (c) Total training Losses (\mathcal{L}) with Divergence Losses (\mathcal{L}_D), Displacement Boundary Condition Losses (\mathcal{L}_U), Traction Boundary Condition Losses (\mathcal{L}_T) and Deformation Losses (\mathcal{L}_F).

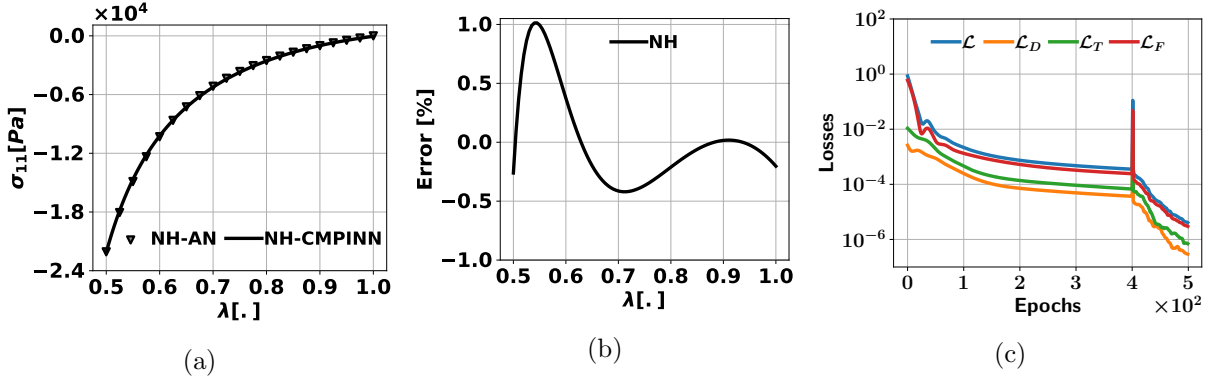


Figure 5: Comparison of CMPINN with hard constraints BC and analytical (AN) solution for the Neo-Hookean (NH) model undergoing biaxial compression: (a) Stress response, (b) Relative Error, (c) Total training Losses (\mathcal{L}) with Divergence Losses (\mathcal{L}_D), Traction Boundary Condition Losses (\mathcal{L}_T) and Deformation Losses (\mathcal{L}_F).

4.2 Hyperparameter tuning for hard and soft constrained BCs

Table 1: Search space for hyperparameter tuning

Parameter	Search space
Number of hidden layers	{4, 6, 8}
Number of neurons for each layer	{25, 50, 75}
Adam optimiser learning rate	{0.0001}
Adam training epochs	{100, 200, 300, 400}
L-BFGS training epochs	{200, 400, 600, 800}

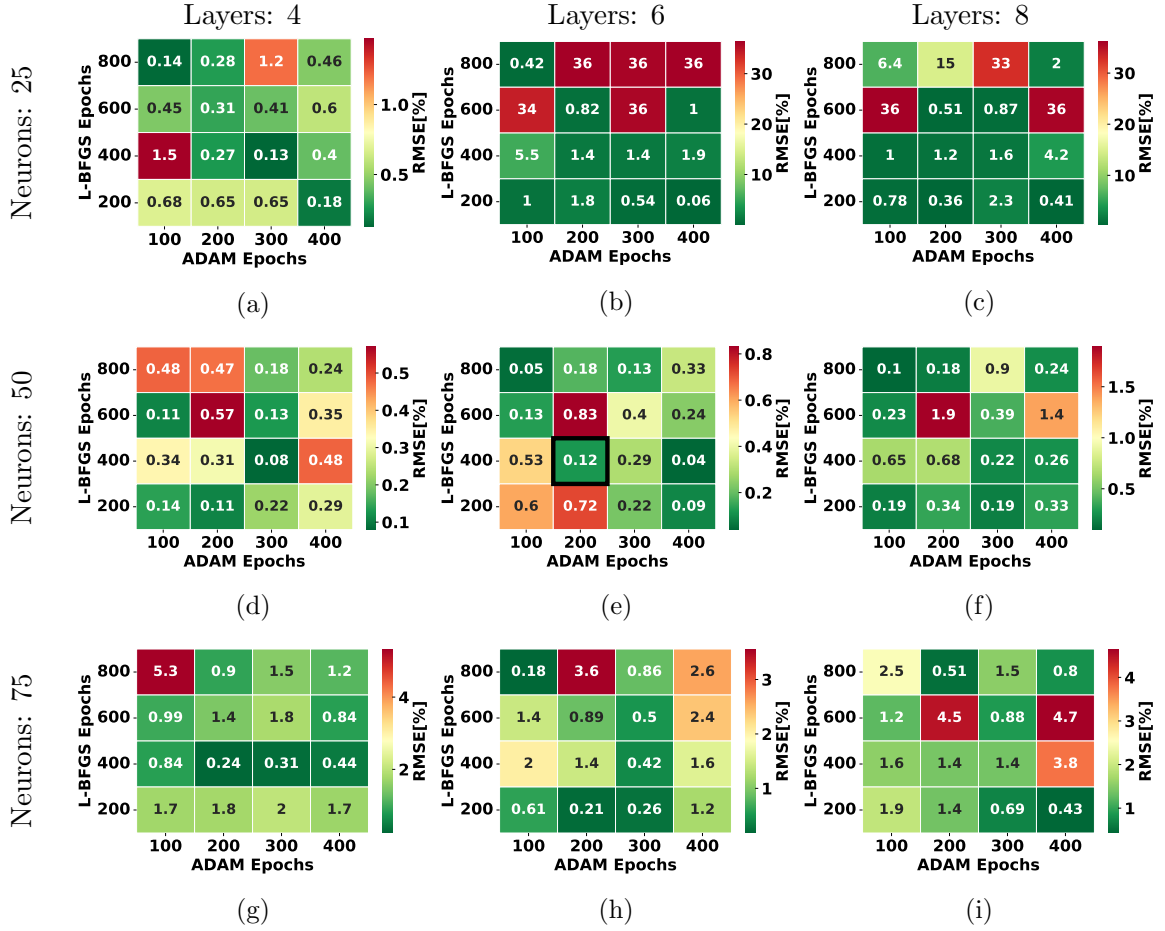


Figure 6: Hyperparameter Investigation for soft constrained BCs: Assessing the influence of Adam and L-BFGS optimizer epochs on a neural network configuration featuring (a) 4 hidden layers \times 25 neurons, (b) 6 hidden layers \times 25 neurons, (c) 8 hidden layers \times 25 neurons, (d) 4 hidden layers \times 50 neurons, (e) 6 hidden layers \times 50 neurons, (f) 8 hidden layers \times 50 neurons, (g) 4 hidden layers \times 75 neurons, (h) 6 hidden layers \times 75 neurons, (i) 8 hidden layers \times 75 neurons with tanh activation function, and an Adam learning rate set at 1×10^{-4} .



Figure 7: Hyperparameter Investigation for hard constrained BCs: Assessing the influence of Adam and L-BFGS optimizer epochs on a neural network configuration featuring (a) 4 hidden layers \times 25 neurons, (b) 6 hidden layers \times 25 neurons, (c) 8 hidden layers \times 25 neurons, (d) 4 hidden layers \times 50 neurons, (e) 6 hidden layers \times 50 neurons, (f) 8 hidden layers \times 50 neurons, (g) 4 hidden layers \times 75 neurons, (h) 6 hidden layers \times 75 neurons, (i) 8 hidden layers \times 75 neurons with tanh activation function, and an Adam learning rate set at 1×10^{-4} .

Grid search is a systematic hyperparameter optimization technique used to exhaustively explore a specified parameter space for a machine learning model [12]. In this approach, a multidimensional grid is constructed where each dimension represents a different hyperparameter, such as learning rate, number of layers, number of neurons for each layer and training epochs. Each point in this grid corresponds to a unique combination of hyperparameter values. The model is trained and evaluated on each combination to obtain a robust estimate of model performance. The objective is to identify the hyperparameter configuration that minimises the RMSE for the predicted spatial and loading points. While grid search thoroughly explores the specified hyperparameter space, it is computationally expensive due to the exponential growth of combinations with each additional hyperparameter and value range considered. The search space for hyperparameter tuning for CMPINNs Neo-Hookean hyperelastic model with soft and

hard BCs constraints are summarised in Table 1

The Figures 6 - 7 illustrate the results of hyperparameter tuning across various configurations for soft and hard constrained CMPINNs for the Neo-Hookean hyperelastic model, ranging from shallow to deeper neural networks with different hyperparameter configurations. The configurations explore different numbers of layers, neurons per layer, optimizers, Adam training epochs and L-BFGS training epochs. Shallow networks even with higher Adam and L-BFGS training epochs often fail to minimise the losses effectively, resulting in higher RMSE in both soft and hard constrained BCs as shown in first row of Figs. 6 - 7. This is likely due to the insufficient capacity to learn non-linearity. In contrast, deeper networks with more layers and neurons perform significantly better in both soft and hard constrained BCs as shown in third row of Figs. 6 - 7.

Notably, the configuration with 6 hidden layers, 50 neurons per hidden layer, tanh activation function with an Adam learning rate of 1×10^{-4} , and 200 training epochs using the Adam optimizer, as well as 400 epochs using the L-BFGS optimizer, achieved the best balance between performance and computational cost as shown in Fig. 6e for the soft constrained BCs with the parameters summarised in Table 2. The configuration with 4 layers, 75 neurons per layer, tanh activation function with an Adam learning rate of 1×10^{-4} , and 400 training epochs using the Adam optimizer, as well as 400 epochs using the L-BFGS optimizer achieved the lowest RMSE for hard constrained BCs as shown in Fig. 7g and the hyperparameters are summarised in Table 2.

Table 2: Best found hyperparameters for soft and hard constrained BCs

Parameter	Soft constrained	Hard Constrained
Number of hidden layers	6	4
Number of neurons for each layer	50	75
Adam optimiser learning rate	0.0001	0.0001
Adam training epochs	200	400
L-BFGS training epochs	400	400

5 Conclusions

In this work, we focused on developing and applying boundary conditions for a causal-marching physics-informed neural network for the Neo-Hookean hyperelastic model. Our investigation assessed the performance of CMPINNs in handling boundary conditions through both soft and hard enforcement approaches. The findings reveal that utilising soft constraints for boundary conditions enables accurate modelling of the non-linear behaviour of the Neo-Hookean hyperelastic model. Additionally, we investigated the optimal combination of hyperparameters for CMPINNs when implementing soft and hard constrained boundary conditions. In soft constrained BCs, the network has more flexibility during training because the BCs are not rigidly enforced from the start. The network is free to explore a larger solution space and gradually learns to satisfy the governing equations along with the boundary conditions. In contrast, hard constrained BCs restrict the solution space, ensuring that the boundary conditions are exactly and strictly satisfied throughout the entire training process. The hard constrained BCs guarantee physical consistency and may lead to more robust solutions, especially in scenarios where strict adherence to the boundary conditions is critical. Although soft constrained boundary

conditions performed better in this particular test case, hard constrained boundary conditions are theoretically superior and more robust.

Acknowledgement

This work is done under the project, “DigiBrain”, funded by the EU Commission Recovery and Resilience Facility under the Science Foundation Ireland Future Digital Challenge Grant Number 22/NCF/FD/11007.

REFERENCES

- [1] Wouter Peeters, Ruben van den Brande, Suzanne Polinder, Alexandra Brazinova, Ewout W Steyerberg, Hester F Lingsma, and Andrew IR Maas. Epidemiology of traumatic brain injury in europe. *Acta neurochirurgica*, 157:1683–1696, 2015.
- [2] Timothy James Horgan and Michael D Gilchrist. The creation of three-dimensional finite element models for simulating head impact biomechanics. *International Journal of Crashworthiness*, 8(4):353–366, 2003.
- [3] Raymond W Ogden. *Non-linear elastic deformations*. Dover Publications, New York, first edition, 1997.
- [4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, Cambridge, first edition, 2016.
- [5] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.
- [6] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.
- [7] O. Morrison, M. Destrade, and B. B. Tripathi. An atlas of the heterogeneous viscoelastic brain with local power-law attenuation synthesised using Prony-series. *Acta Biomaterialia*, 2023.
- [8] Melvin Mooney. A theory of large elastic deformation. *Journal of applied physics*, 11(9):582–592, 1940.
- [9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [10] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 8 1989.
- [11] Luning Sun, Han Gao, Shaowu Pan, and Jian Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.
- [12] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.