XI International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2025
A. Larese, S. Perotto and G. Rozza (Eds)

REDISCOVERING THE MULLINS EFFECT WITH PHYSICS AUGMENTED NEURAL NETWORKS

MARTIN ZLATIĆ*, **MARKO ČANAĐIJA**†

*Faculty of Engineering (RITEH)
University of Rijeka
Vukovarska 58, 51000 Rijeka, Croatia
e-mail: martin.zlatic@riteh.uniri.hr

†Faculty of Engineering (RITEH)
University of Rijeka
Vukovarska 58, 51000 Rijeka, Croatia
e-mail: marko.canadija@riteh.uniri.hr

Key words: Isotropic Damage, Hyperelasticity, Neural Network Model

Abstract. Neural networks (NNs) can be used to describe relationships without explicitly knowing them, but by approximating them through a number of superpositioned functions. In this way, a wider range of unknown behaviours can be modelled with a single approach. In this paper, we present the possibility of incorporating previously assumed relationships, i.e. analytical models, into an NN and then comparing them with a more general NN to model the same relationship without making any prior assumptions. This is demonstrated by modelling the Mullins effect, a simple isotropic damage model for hyperelastic materials where incompressibility is a priori assumed. The data was artificially generated by using the incompressible Ogden model as the underlying hyperelastic behaviour. A framework for modelling simple isotropic damage is shown that provides a high degree of flexibility while accounting for certain physical properties such as objectivity, normalisation of energy and thermodynamic consistency. This work follows other works that incorporate physical constraints in NNs, an approach referred to as *Physics Augmented Neural Networks* (PANNs). The NN model is only trained on simple deformation modes such as uniaxial, equibiaxial and planar tension and captures the complete 3D behaviour. Several numerical examples in Abaqus are presented to show the accuracy of the model and the simplicity of the numerical implementation in a widely used finite element software.

1 INTRODUCTION

Neural networks are powerful regression tools and are present in the area of material modelling for their ability to approximate a wide range of functions [1]. They can be used in different ways for modelling hyperelastic materials, for example they can establish the stress-strain relation [2] or learn the strain energy function [3] and calculate the tangent moduli using automatic differentiation. In this paper the approach with predicting the strain energy is taken, but with certain modifications that ensure the model respects certain restrictions. The type of neural network that it used in this paper is a *Physics Augmented Neural Network* (PANN), and a general guide on implementing the restrictions is given by [6]. The idea is that certain restrictions are built into the network architecture rather than just being enforced through the loss function during training. These conditions include the *normalisation of energy and stress*, *polyconvexity* and *objectivity*, and in the case of compressible behaviour additionally include the *growth condition*. Some examples of the application of the PANN architecture are given in [7] where the approach was used on parametrized hyperelasticity and in [8] it was applied on coupled magneto-elastic behaviour. In the work of [9] a similar architecture called *Constitutive Artificial Neural Network* (CANN) has been demonstrated and proved capable of capturing different material behaviours. Another takeaway from [9] is the introduction of multiple different functions commonly used in material modelling as activation functions in NNs, amongst them is the linear exponential unit which was applied in [4] for modelling both the isotropic damage and hyperelastic response.

In this paper incompressible behaviour is assumed and the results of modelling Mullins-type damage of hyperelastic materials is presented.

2 DATA GENERATION AND NEURAL NETWORK ARCHITECTURE

The base data is generated using the incompressible Ogden model [5]:

$$\psi_{\text{Ogden}} = \sum_{p=1}^{3} \frac{\mu_p}{\alpha_p} \left(\lambda_1^{\alpha_p} + \lambda_2^{\alpha_p} + \lambda_3^{\alpha_p} - 3 \right), \tag{1}$$

where λ_a $(a = 1, 2, 3)$ are the principal stretches or the eigenvaules of the left $(\mathbf{V})$ or right $(\mathbf{U})$ stretch tensors, with α_p being dimensionless constants and μ_p the shear moduli. The damaged Mullins energy takes the form

$$\begin{gathered} \psi_{\text{Mullins}} = \left(1 - \zeta\left(\beta\right)\right) \psi_{\text{Ogden}}, \\ \zeta = \zeta_\infty \left[1 - \exp\left(-\frac{\beta}{\iota} \right) \right], \quad \beta = \max_{s \in [0,t]} \psi_{\text{Ogden}}\left(s\right), \end{gathered} \tag{2}$$

with ζ being the damage coefficient dependent on β which is the maximum historic value of the undamaged strain energy. There are also the additional material parameters ι and ζ_∞ which is the maximum attainable value of damage. The training was done on the uniaxial, equibiaxial, and planar tension cases allowing the NN ot be trained under the assumption of plane stress conditions.

The values of the material parameters used to generate the data are given in Table 1. The damaged behaviour for the uniaxial case is shown in Fig. 1.

Table 1: Values of the Ogden material and damage parameters.

p	α_p [-]	μ_p [MPa]	ζ_∞
1	1.3	0.63	0.8
2	5	0.0012	ι
3	-0.01	-2	1

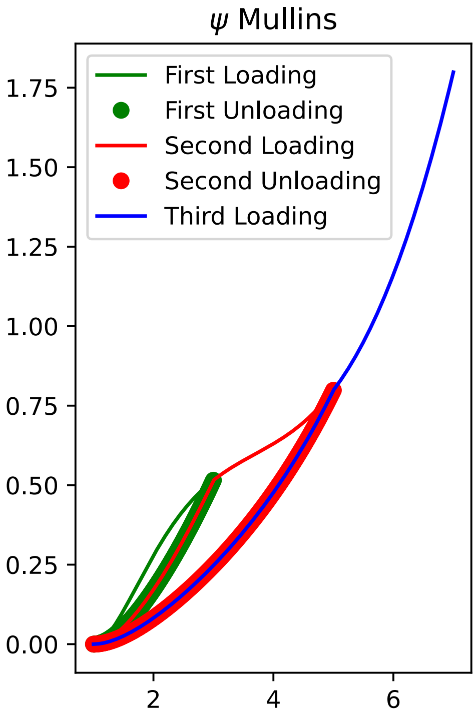

Figure 1: Uniaxial behaviour with 2 unloading parts. The damaged Mullins behaviour is shown and the dependency on the maximum historic value of the strain energy is shown.

As stated in Section 1 there are multiple ways to model material behaviour using NNs. In this work the choice the NN is used as an invariant based model given that it satisfies objectivity by design and also it facilitates the incorporation of other restrictions.

For the activation function, the linear exponential activation function was chosen from [9] since faster convergence was observed when experimenting during training. The activation function takes the form

$$h(x) = e^{\alpha x} - 1, \tag{3}$$

where α is a trainable parameter. The base neural network for modelling the response of the undamaged strain energy ψ_0 is a shallow feed-forward neural network, i.e. it consists of one hidden layer. The general architecture of the NN is shown in Fig. 2. The notation for the individual weights is $w_{ij}^{[l]}$ where i corresponds to the neuron in the previous layer, j to the neuron in the following layer and $[l]$ is the number of the following layer. The input layer is labeled as $l = 0$, the hidden layer as $l = 1$ and the output layer as $l = 2$.

Objectivity is satisfied by using the invariants of the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ as inputs to the neural network. The invariants can be described as following functions of $\mathbf{C}$:

$$I_1(\mathbf{C}) = \mathrm{tr}\,(\mathbf{C}) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \tag{4}$$

$$I_2(\mathbf{C}) = \frac{1}{2}\left(\mathrm{tr}(\mathbf{C})^2 - \mathrm{tr}(\mathbf{C}^2)\right) = \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2, \tag{5}$$

$$I_3(\mathbf{C}) = \det(\mathbf{C}) = \lambda_1^2\lambda_2^2\lambda_3^2, \tag{6}$$

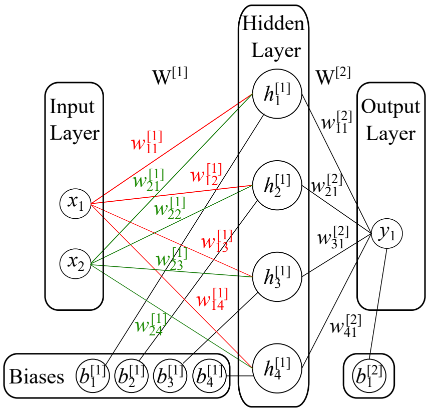

Figure 2: General architecture of the used feed-forward neural network.

however the third invariant I_3 will not be used since for the incompressible case it is always equal to 1. A useful property of the invariants is that for the undeformed case where $\mathbf{F} = \mathbf{C} = \mathbf{I}$, the invariants $I_1 = I_2 = 3$.

The inputs x_1 and x_2 in Fig. 2 are replaced with I_1 and I_2, and the output is replaced with the undamaged strain energy ψ_0.

Normalisation of energy is guaranteed by using the activation function from Eq. (3) and modifying the input so that in place of x_1 and x_2 the values are $(I_1 - 3)$ and $(I_2 - 3)$. If the biases are excluded then in the undeformed case $\mathbf{F} = \mathbf{C} = \mathbf{I}$, then the activation function takes the value

$$h(I_1 - 3) = h(0) = e^{\alpha \cdot 0} - 1 = 1 - 1 = 0, \tag{7}$$

guaranteeing that for no deformation there is no energy.

Normalisation of stress is satisfied by using an incompressible formulation. The stress tensor (in this example the 2$^{\text{nd}}$ Piola-Kirchhoff tensor) is defined as

$$\mathbf{S} = 2\frac{\partial\psi(I_1, I_2)_{\text{NN}}}{\partial C} - p\mathbf{C}^{-1} = 2\left[\left(\frac{\partial\psi_{\text{NN}}}{\partial I_1} + I_1\frac{\partial\psi_{\text{NN}}}{\partial I_2}\right)\mathbf{I} - \frac{\partial\psi_{\text{NN}}}{\partial I_2}\mathbf{C}\right] - p\mathbf{C}^{-1}, \tag{8}$$

and from the spectral decomposition of $\mathbf{S}$ if plane stress conditions are assumed so that $S_3 = 0$ the Lagrange multiplier p that enforce incompressibility can be obtained as

$$p = 2\lambda_3^2\left[\frac{\partial\psi\text{NN}}{\partial I_1} + (\lambda_1^2 + \lambda_2^2)\frac{\partial\psi_{\text{NN}}}{\partial I_2}\right], \tag{9}$$

ensuring that for the undeformed state $\mathbf{S}(\mathbf{C} = \mathbf{I}) = \mathbf{0}$.

Polyconvexity requires that the strain energy is convex with respect to $\mathbf{F}, \operatorname{cof}(\mathbf{F})$ and $J(= \det \mathbf{F})$ so that the existence of a minimum exists. The invariants can be rewritten as $I_1 = ||\mathbf{F}||^2$ and $I_2 = ||\operatorname{cof}(\mathbf{F})||^2$, where $||\cdot||^2$ denotes the inner product of the tensor with itself.

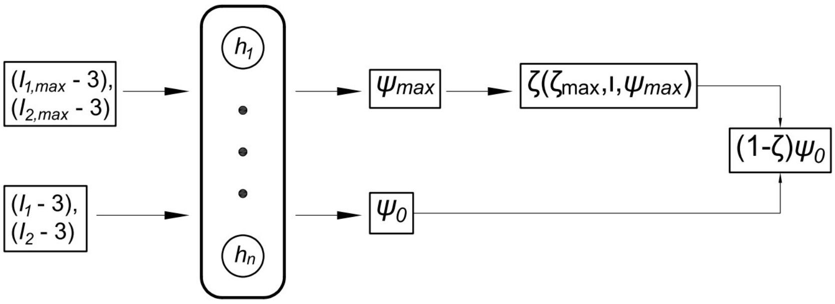

Figure 3: Expanded neural network architecture accounting for the Mullins effect.

Looking at Fig. 2 the output undamaged strain energy can be written as

$$\psi_0 = \sum_{i=1}^{4} w_{i1}^{[2]} h_i^{[1]}, \quad h_i = \exp\left[\alpha_i w_{1i}^{[1]} (\underbrace{I_1}_{||\mathbf{F}||^2} - 3) + w_{2i}^{[1]} (\underbrace{I_2}_{||\mathrm{cof}(\mathbf{F})||^2} - 3)\right] - 1. \tag{10}$$

If polyconvexity is to be guaranteed, that would require that within the composition of functions all functions should be convex and non-decreasing. The exponential function is convex, however to ensure it is non-decreasing all the inputs must be non-negative. Additionally, the weights in the output $w_{ij}^{[2]}$ also need to be constrained to be non-negative to guarantee polyconvexity.

2.1 Accounting for the Mullins Effect

The base network was expanded in order to account for the Mullins effect. In Eq. $(2)_1$ the Mullins energy consists of the undamaged energy, the maximum historical value of the undamaged energy and the damage coefficient. In Fig. 3 it is shown how the neural network framework is expanded to accommodate all of these changes. The base neural network with inputs $(I_1-3), (I_2-3)$ and output ψ_0 is kept and all the weights are reused but with the historical variables $(I_{1,\mathrm{max}} - 3), (I_{2,\mathrm{max}} - 3)$ to obtain the output which is the maximum historical value of the undamaged strain energy $\psi_{0,\mathrm{max}}$. Afterwards, $\psi_{0,\mathrm{max}}$ is passed on to another layer that calculates the damage coefficient ζ. This can be treated as an activation function or its own individual subnetwork with one input ($\psi_{0,\mathrm{max}}$) and one output (ζ). Finally, all of these values give the Mullins energy predicted by the neural network ψ_{NN}.

The final NN uses 5 neurons in the hidden layer for predicting ψ_0 and $\psi_{0,\mathrm{max}}$, and a custom activation function in the form of Eq. $(2)_2$ with ι and ζ_∞ replaced with the trainable parameters ι_{NN} and ζ_{NN}, with β being replaced with $\psi_{0,\mathrm{max}}$.

Using auto-differentiation the 2nd Piola-Kirchhoff stress can be easily defined as previously in Eq. (8) since the partial derivatives of the energy w.r.t. the invariants are easily obtainable. Thus, the loss function contains the stress instead of the energy which is originally predicted by the NN. This is significant for 2 reasons. Firstly, it means that **thermodynamic consistency** is satisfied which requires the stress definition follows from the strain energy. Secondly, it means the neural network is trained on its derivatives which renders a better NN than training directly on the predicted quantity, see [10] for a comparison.

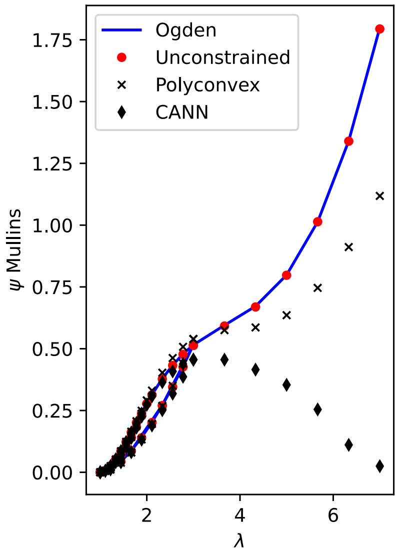

Figure 4: Predictions of various neural networks. The unconstrained neural network does not have the restrictions in place to fulfil polyconvexity, the polyconvex network is described in Section 2, and the CANN architecture from [9] is also added for comparison.

3 NUMERICAL EXAMPLES

The results for the uniaxial case are shown in Fig.4. The Ogden model is a polyconvex model, however neither the polyconvex NN nor the CANN can successfully capture it. Relaxing the restrictions of the polyconvex network gives a NN model that can correctly capture the Ogden model. It should be noted that only the polyconvexity condition is relaxed, the other restrictions are still fulfilled. Further results are presented using the best, i.e. unconstrained, neural network model.

Given that only two unloading cycles were given during training, another cyclic loading/unloading test is performed to investigate did the NN model capture the Mullins effect fully or only a part of it. The results are shown in Fig. 5 and confirm that the NN model fully captured the Mullins effect.

3.1 Solid Rubber Disc

An example with a more complex geometry, boundary conditions and loading is taken from the Abaqus examples for the Mullins effect [11]. The example is a solid rubber disc, a simplification of a tyre. It is first pressed into the ground, achieved by contact with an analytical surface, and then rotated by 360 degrees. The geometry is given in Fig. 6, the displacement and rotation are prescribed at the disc centre.

The magnitudes of the reaction force and moment are given in Fig. 7 and show good agreement between the NN model and the reference Ogden model showing that although the NN was trained on plane stress data it can be used to model a wider range of conditions including full 3D behaviour with a mixed fully incompressible finite element formulation. Both the error for the reaction force and moment are around 1%.

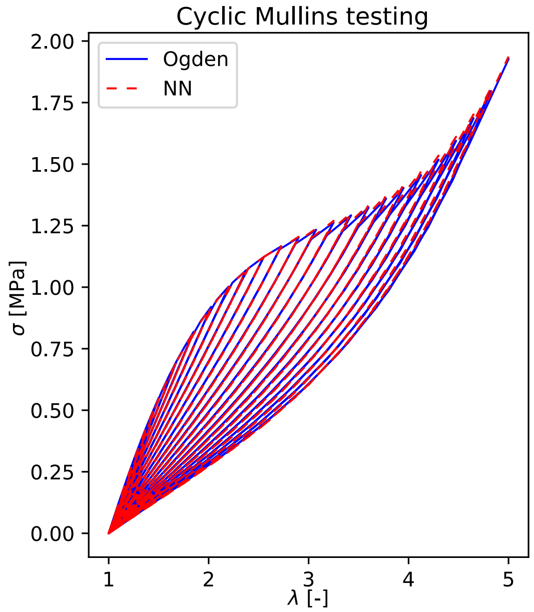

Figure 5: Cyclic testing, 20 loading/unloading cycles.

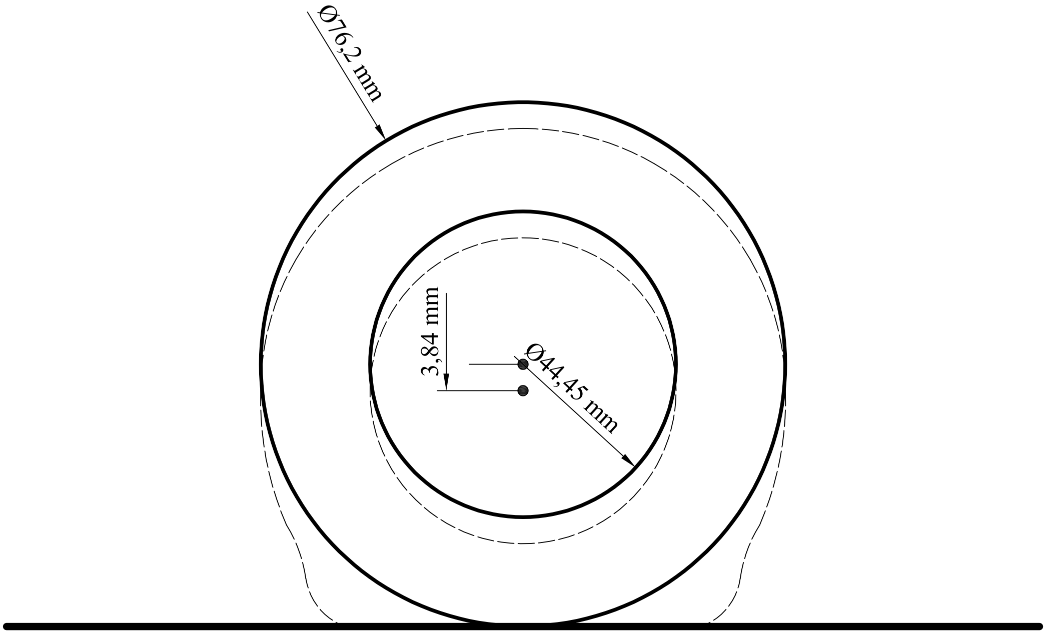

Figure 6: Dimensions of the solid rubber disc, dashed line shows the approximate deformed configuration when the disc is subject to loading. The displacement and rotational load is assigned at the centre dot.

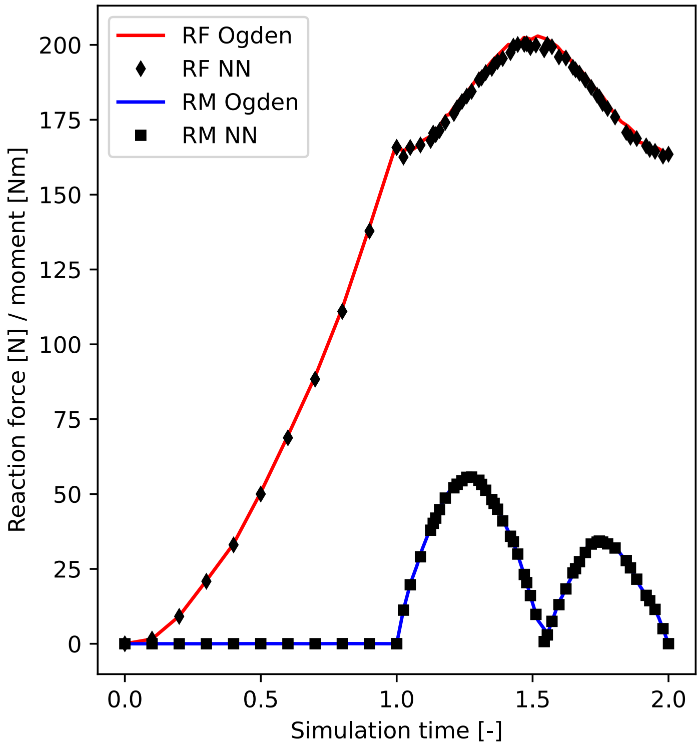

Figure 7: Reaction force and moment magnitudes for the solid rubber disc numerical example. Measured at the point where the displacement and rotation were prescribed.

4 CONCLUSION

In this work it has been shown that *Physics Augmented Neural Networks* can serve as a basis for modelling isotropic damage in a way that allows for full flexibility of the NN design with the ability to change the base undamaged models as required to achieve the best material model. The path forwards may include adding additional branches to account for other physical effects or to include more complicated damage models.

REFERENCES

[1] Hornik, K., Stinchcombe, M. and White, H., Multilayer feedforward networks are universal approximators. *Neural Networks*, **2**, 359–366, (1989.)

[2] Weber, P., Geiger, J., Wagner, W. Constrained neural network training and its application to hyperelastic material modeling. *Computational Mechanics* **68**, 1179–1204., (2021.)

[3] Shen, Y., Chandrashekhara, K., Breig, W. F., and Oliver, L. R., Neural network based constitutive model for rubber material. *Rubber Chemistry and Technology* **77**, 257–277., (2004.)

[4] Zlatić, M. and Čanađija, M., Recovering Mullins damage hyperelastic behaviour with physics augmented neural networks. *Journal of the Mechanics and Physics of Solids*, **193**, 105839, (2024.)

[5] Ogden, R., Large deformation isotropic elasticity – on the correlation of theory and experiment for incompressible rubberlike solids. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, **326**, 565–584, (1972.)

[6] Linden, L., Klein, D. K., Kalina, K. A., Brummund, J., Weeger, O., and Kästner, M., Neural networks meet hyperelasticity: A guide to enforcing physics. *Journal of the Mechanics and Physics of Solids*, **179**, 105363, (2023.)

[7] Klein, D. K., Roth, F. J., Valizadeh, I., Weeger, O., Parametrized polyconvex hyperelasticity with physics-augmented neural networks. *Data-Centric Engineering*, **4**, 25, (2023.)

[8] Kalina, K. A., Gebhart, P., Brummund, J., Linden, L., Sun, W., and Kästner, M., Neural network-based multiscale modeling of finite strain magneto-elasticity with relaxed convexity criteria. *Computer Methods in Applied Mechanics and Engineering*, **421**, 116739, (2024.)

[9] Linka, K. and Kuhl, E., A new family of constitutive artificial neural networks towardsautomated model discovery. *Computer Methods in Applied Mechanics and Engineering*, **403**, 115731, (2023.)

[10] Zlatić, M., Rocha, F., Stainier, L., Čanađija, M., Data-driven methods for computational mechanics: A fair comparison between neural networks based and model-free approaches. *Computer Methods in Applied Mechanics and Engineering*, **431**, 117289, (2024.)

[11] Dassault Systèmes Simulia Corp., Abaqus Analysis User's Manual, Version 6.14. *Dassault Systèmes Simulia Corp*, (2014.)