Data-driven and Physics-informed Muscle Model Surrogates for Cardiac Cycle Simulations

Bogdan Milicevic, Milos Ivanovic, Boban Stojanovic, Miljan Milosevic, Vladimir Simic, Milos Kojic and Nenad Filipovic, Member, IEEE

***Abstract*— Health professionals can utilize biomechanical simulations of left ventricle to assess different possible situations and hypothetical scenarios. Understanding of the molecular mechanisms behind muscle contraction has resulted in the development of Huxley-like muscle models. Unlike Hill-type muscle models, Huxley-type muscle models can be used to simulate non-uniform and unstable contractions. However, Huxley models demand considerably more computational resources than Hill models, which limits their practical use in large-scale simulations. To address this, we have developed a data-driven and physics-informed surrogate models that mimic the Huxley muscle model, while requiring significantly less processing power. We collected data from various numerical simulations and trained deep neural networks to replace Huxley's muscle model. Data-driven surrogate model was an order of magnitude faster than the original model, while being quite accurate. Our surrogate models were integrated into a finite element solver and used to simulate a complete cardiac cycle, which would be much harder to do with original Huxley's model.**

***Index Terms*—finite element analysis, surrogate modeling, physics-informed neural networks, recurrent neural networks, Huxley's muscle model**

I. Introduction

The functionality of the left ventricle relies on factors such as ventricular shape, the passive mechanical properties of the myocardium, the arrangement of muscle fibers, and the force produced by the fibers. To understand muscle behavior, researchers typically use a continuum mechanics method to express active and passive stress components at each point in the myocardium as time-dependent functions of local strain. Biophysical processes are modeled at various spatial and temporal scales to analyze muscle performance through computational analysis. In multi-scale simulations, the finite element method is utilized to model continuum muscle mechanics, while muscle material characteristics at the microscopic level are defined using Huxley's muscle contraction model [1]. During transient finite element simulations, Huxley's model is used to calculate stress and stiffness, based on muscle activation, stretch, and other material parameters and properties [1]. However, these finite element simulations can be time-consuming, particularly in the microscale calculations. To reduce computational requirements, the authors developed a computationally efficient surrogate model to replace the original Huxley muscle model.

II. The Methods

Huxley considered the dynamics of the filaments within muscle and the probability of establishing connections (cross-bridges) of myosin heads to actin filaments inside sarcomeres [2]. The $n(x,t)$ function describes the rate of connections between myosin heads and actin filaments, as a function of position of nearest available actin binding site relative to equilibrium position of myosin head x:

$$\frac{\partial n(x,t)}{\partial t} - v\frac{\partial n(x,t)}{\partial x} = [1 - n(x,t)]f(x,a) - n(x,t)g(x), \forall x \in \Omega \qquad (1)$$

where $f(x,a)$ and $g(x)$ represent the attachment and detachment rates of cross-bridges respectively, v is the velocity of filaments sliding, positive in the direction of contraction, and a is muscle activation given as a function of time. The muscle activation can be derived from the calcium concentration function, which we prescribed in our numerical experiments. The partial differential equation (1) can be solved using the method of characteristics with initial condition $n(x,0) = 0$. Once the $n(x,t)$ values are acquired we

Bogdan Milicevic - Faculty of Engineering, University of Kragujevac, Sestre Janjic 6, 3400 Kragujevac, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: bogdan.milicevic@uni.kg.ac.rs), ORCID ID (https://orcid.org/0000-0002-0315-8263)

Milos Ivanovic - Faculty of Science, University of Kragujevac, Radoja Domanovica 12, 3400 Kragujevac, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: mivanovic@kg.ac.rs), ORCID ID (https://orcid.org/0000-0002-8974-2267)

Boban Stojanovic - Faculty of Science, University of Kragujevac, Radoja Domanovica 12, 3400 Kragujevac, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: boban.stojanovic@pmf.kg.ac.rs), ORCID ID (https://orcid.org/0000-0003-2901-0061)

Miljan Milosevic - Belgrade Metropolitan University, Tadeusa Koscuska 63, 11000 Belgrade, Serbia; Institute for Information Technologies, University of Kragujevac, Jovana Cvijica BB, 34000 Kragujevac, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: miljan.m@kg.ac.rs), ORCID ID (https://orcid.org/0000-0003-3789-2404)

Vladimir Simic - Institute for Information Technologies, University of Kragujevac, Jovana Cvijica BB, 34000 Kragujevac, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: vladimir.simic.991@gmail.com), ORCID ID (https://orcid.org/0000-0001-7842-8902)

Milos Kojic - The Methodist Hospital Research Institute, The Department of Nanomedicine, TX 77030 Houston; The Serbian Academy of Sciences and Arts, Kneza Mihaila 35, 11000 Belgrade, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: mkojic42@gmail.com), ORCID ID (https://orcid.org/0000-0003-2199-5847)

Nenad Filipovic - Faculty of Engineering, University of Kragujevac, Sestre Janjic 6, 3400 Kragujevac, Serbia and Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovica 6, 34000 Kragujevac, Serbia (e-mail: fica@kg.ac.rs), ORCID ID (https://orcid.org/0000-0001-9964-5615)

979-8-3503-0711-5/23/$31.00 ©2023 IEEE

can calculate generated force F within the muscle fiber and also stiffness K using the equations:

$$F(t) = k\sum_{-\infty}^{\infty} n(x,t)x\ dx \quad (2)$$

$$K(t) = k\sum_{-\infty}^{\infty} n(x,t)\ dx \quad (3)$$

where k is the stiffness of cross-bridges. Stress and stress derivative can be calculated as:

$$\sigma_m = F\frac{\sigma_{iso}}{F_{iso}} \quad (4)$$

$$\frac{\partial\sigma_m}{\partial e} = \lambda L_0 K\frac{\sigma_{iso}}{F_{iso}} \quad (5)$$

where F_{iso} is maximal force achieved during isometric conditions, σ_{iso} maximal stress achieved during isometric conditions, $L_0\ the$ initial length of sarcomere and λ is stretch. Calculated stresses and stress derivatives can be further used at the macro-level during finite element analysis.

To obtain the necessary data, the authors conducted numerous muscle contraction simulations on smaller 2D finite element models using different prescribed muscle activation functions, displacements and forces. The data collected from these simulations were used to develop surrogate models based on artificial neural networks. In our paper we present two different approaches based on (1) neural networks for time-series analysis and (2) physics-informed neural networks. In our first, data-driven approach, the neural network takes inputs in the form of time series. This input time series includes activation at current and previous time steps, stretch at current and previous time steps, stress at previous time steps, and instantaneous stiffness at previous time steps. The model outputs muscle stress and instantaneous stiffness at the current time step of the simulation. Given that the inputs are time series, we opted for recurrent and convolutional networks, which are well-suited for this prediction task. We trained multiple deep neural networks in order to find one which achieves satisfactory precision and generalization.

Our second approach is based on physics-informed neural networks (PINNs). PINNS are a novel class of machine learning models that can solve supervised learning problems based on the general nonlinear partial differential equations that describe physical principles [3]. PINNs are a powerful universal functions approximators and are able to incorporate physical laws as prior knowledge [3]. A residual network is added to a deep learning network, called a surrogate, to compute the residual value, which represents the difference between the computed solution and the true solution of the differential equations [4]. PINNs use automated differentiation to compute differential operators on graphs, and their basic formulation doesn't require labeled data, results from other simulations, or experimental data [4]. Only the residual function calculation is necessary for PINNs, but it is possible to train them with simulation or experimental data in a supervised manner. PINNs are a gridless technique that can use any point in the domain as input without requiring a mesh [4]. After training, the PINN network can predict the solutions on simulation grids with various resolutions and can be used for time-dependent problems without accounting for earlier time steps. PINN has been applied to solve equations such as the Burgers' equation, the Navier–Stokes equation, and the Schrodinger equation [5]. In our study, we utilized PINN to solve Huxley's muscle equation and obtain the distribution of attached myosin heads to the actin-binding sites. To implement PINN and incorporate the equation (1), we used SciANN [6], a high-level artificial neural networks API, written in Python using Keras and TensorFlow backends. SciANN is designed to abstract neural network construction for scientific computations and solution and discovery of partial differential equations (PDE) using the physics-informed neural networks [6]. Our physics-informed neural network, receives current and previous stretch, muscle activation, time, and x, and based on these values predicts the n value from equation (1). During training the residual of Huxley's muscle differential equation, residual of initial condition and error between true and predicted n values are minimized. Based on predicted n values, stresses and stress derivatives are calculated according to formulas (4) and (5).

III. Main Results

One of the most successful neural networks for our prediction task was GRU neural network with 992,770 weights. This neural network consists of 128 neurons in 1st, 4th and 5th hidden layer and 256 neurons in 2nd and 3rd hidden layer. It was constructed using Keras and Tensorflow. Correlation coefficients between original and predicted values are shown in Table I. We showed average correlation coefficients obtained in numerical experiments used to train the neural network and in numerical experiments used to test the network.

TABLE I
Correlation coefficients between original and predicted values obtained with GRU

GRU - Training		
	Correlation coefficient (stress)	**Correlation coefficient (stress derivative)**
Average value:	0.99999918	0.9999918
Standard deviation:	9.6×10^{-6}	1.7×10^{-5}
GRU - Test		
	Correlation coefficient (stress)	**Correlation coefficient (stress derivative)**
Average value:	0.99999998	0.99999984
Standard deviation:	6×10^{-8}	2.4×10^{-7}

Using the SciANN framework we constructed a physics-informed neural network with 8 layers, each containing 20 neurons with a hyperbolic tangent activation function. The network is trained by minimizing the difference between actual and predicted values and also by minimizing the residuals derived from equation (1) and its initial conditions. We used Adam optimizer with a learning rate of 5×10^{-5} and batch size of 16384, during 7000 epochs. We also used the neural tangent kernel (NTK) method to get the adaptive weights, balancing the difference between the number of points, used to minimize the residual of PDE, and the number of points used to minimize the residual of the initial condition. Correlation coefficients between original and predicted values are shown in Table II. We showed average correlation coefficients obtained in numerical experiments used to train the neural network and in numerical experiments used to test the network.

TABLE II
CORRELATION COEFFICIENTS BETWEEN ORIGINAL AND PREDICTED VALUES OBTAINED WITH PINN

PINN - Training		
	Correlation coefficient (stress)	**Correlation coefficient (stress derivative)**
Average value:	0.9837	0.9857
Standard deviation:	0.0209	0.0117
PINN - Test		
	Correlation coefficient (stress)	**Correlation coefficient (stress derivative)**
Average value:	0.9283	0.9702
Standard deviation:	0.0421	0.0109

One of the main goals of our work is to speed up the multi-scale calculations. While it can take around 4500 [s] to execute finite element simulation with one simple finite element and original Huxley's model, it only takes around 110 [s] to execute the same simulation with GRU instead of Huxley's model.

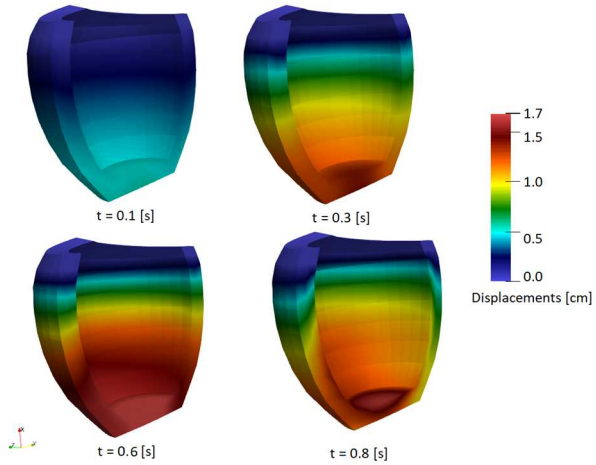

Fig. 1. Left ventricle displacements at start of the diastole (t=0.1[s], t=0.3[s]), and at the start and middle of the systole (t=0.6[s], t=0.8[s])

We also simulated cardiac cycle of the left ventricle using our surrogate model at micro-level. Our model consists of both fluid and solid components, but we will only show the results obtained in solid since muscles are only present in the left ventricle wall. We start our simulation at the beginning of the diastole and prescribed inlet velocity of 100 [mm/s] at the mitral valve and zero velocity is prescribed at aortic valve. During the diastole left ventricle expands, then the muscles are activated via calcium concentration function, and under the influence of the muscle contraction the blood flows out of the ventricle. During systole velocity at mitral valve is zero, and aortic valve is opened. In Figure 1 we showed displacement fields, in solid wall of the model, at start of the diastole and at the start and middle of the systole. Since muscles are activated during systole, larger displacements can be seen at the bottom of the left ventricle, during this phase of the cardiac cycle. Similar results are achieved with PINN and GRU surrogates, so we only show results with GRU in Figure 1.

IV. CONCLUSIONS

In our work we presented surrogate muscle models which can replace original Huxley's muscle model. Based on high correlation coefficients and achieved speed-up it can be concluded that our model can indeed be used in multi-scale simulations, which can potentially help the clinicians efficiently analyse left ventricle mechanical response.

ACKNOWLEDGMENT

This research was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 952603 (http://sgabu.eu/). This article reflects only the author's view. The Commission is not responsible for any use that may be made of the information it contains. Research was also supported by the SILICOFCM project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777204. This article reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information the article contains. The research was also funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia, contract numbers [451-03-68/2022-14/200107 (Faculty of Engineering, University of Kragujevac) and 451-03-68/2022-14/200378 (Institute for Information Technologies Kragujevac, University of Kragujevac)].

REFERENCES

[1] B. Stojanovic, M. Svicevic, A. Kaplarevic-Malisic, R. J. Gilbert, S. M. Mijailovich, "Multi-scale striated muscle contraction model linking sarcomere length-dependent cross-bridge kinetics to macroscopic deformation," in *Journal of Computational Science*, Elsevier, Volume 39, 2020.

[2] A. F. Huxley, "Muscle structure and theories of contraction," Prog. Biophys. Biophys. Chem, vol. 7, pp. 255–318, 1957.

[3] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics Informed Deep Learning (Part I): Data-Driven Solutions of Nonlinear Partial Differential Equations. New York City, NY: arXiv preprint arXiv:1711.10561. 2017.

[4] S. Markidis, The Old and the New: Can Physics-Informed Deep-Learning Replace Traditional Linear Solvers, Frontiers in Big Data, Vol. 4, ISSN 2624-909X 2021. https://doi.org/10.3389/fdata.2021.669097

[5] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 686–707. 2019.

[6] E. Haghighat and R. Juanes, Sciann: A Keras Wrapper for Scientific Computations and Physics-Informed Deep Learning Using Artificial Neural Networks. New York City, NY: arXiv preprint arXiv:2005.08803. 2020.