Generalized Physics-Informed Machine Learning for Numerically Solved Transient Physical Systems

Rishith Ellath Meethal,[1 2] **Leela Sai Prabhat Reddy Kondamadugula,**[1 2]
Mohamed Khalil,[1] **Birgit Obst,**[1] **Roland Wüchner** [2]

[1]Siemens AG, T RDA SDT MSO-DE, Otto-Hahn-Ring 6, 81739 Munich, Germany
[2]Chair of structural analysis, Technical University of Munich, 80333 Munich, Germany
[1]{rishith.ellath_meethal, leela.kondamadugula, khalil.mohamed, birgit.obst}@siemens.com,
[2]wuechner@tum.de

Abstract

We introduce a generalized physics-informed machine learning workflow to accurately predict the behavior of a transient physical system with enhanced physics conformity. A *physics-guided machine learning* (PGML) model is developed to achieve this goal. Our model consists of two main parts for a given transient system: (1) a physics-based numerical model which solves the system using conventional numerical methods and returns the stiffness matrix and force vector at each time step; (2) a neural network (NN) based machine learning (ML) surrogate model which predicts the solution of the system using a custom physics-guided loss function constructed from system matrix and force vector. The proposed workflow results in a physics-aware Machine Learning (ML) model. Such a trained model can be used to avoid the prohibitively expensive step of running a transient system simulation at the desired resolutions in space and time. We demonstrate and test the model on single-degree-of-freedom (SDOF) and multiple-degree-of-freedom (MDOF) system's examples from structural dynamics. Our results show that the method predicts the simulation results accurately. The proposed workflow can be directly adapted to any other physics and numerical method as it is not tailored towards a specific physics or a numerical method.

1 Introduction

The partial differential equations (PDE) which govern transient physical systems are solved using numerical methods such as Finite Element Method (FEM) (Zienkiewicz et al. 2000) and Finite Difference Method (FDM) (Forsythe and Wasow 1960). However, it is computationally intensive to run simulations for these models for long time durations at desired resolutions in space and time. This cost scales linearly when the required number of simulations is large. Over the recent years, a common technique to reduce this cost has been to use Machine Learning (ML) models as a surrogate for the solution in engineering and natural science disciplines (Reich and Barai 1999; Kutz 2017; Tarca et al. 2007). Nonetheless, there are two major constraints faced by ML models:

1. **Lack of training data.** Reliable ML models require large amounts of data which needs enormous computing resources. It also takes painstakingly long to generate because we need to run simulations for long duration at desired spatial and time resolutions.
2. **Lack of physics conformity.** Due to their *black box* nature, standalone ML-based models are often incapable of producing physics-conforming results and lead to poor generalization.

These two shortcomings have made researchers explore the possibilities of integrating knowledge of physical laws into ML models. (von Rueden et al. 2019) introduces the umbrella term *informed machine learning* and surveys different approaches on the explicit integration of prior knowledge into machine learning pipelines. He explains that one of the approaches to achieve informed machine learning is by incorporating physical laws (in the form of PDEs) as custom loss terms. More recently, (Willard et al. 2020) provides an overview of approaches which integrate traditional physics-based modeling techniques with ML. The authors categorize these approaches into five classes; (i) Physics-guided loss function, (ii) Physics-guided initialization, (iii) Physics-guided design of architecture, (iv) Residual modeling, (v) Hybrid physics-ML models. Our proposed model falls under (i) Physics-guided loss function. ML models based on physics-guided loss function provide synergistic integration of prior physics knowledge into ML pipelines resulting in reduced "black-box" nature and improved data-efficiency of the models. We demonstrate our method with the help of SDOF and a MDOF systems from the *Structural dynamics* domain. The transient response of SDOF and MDOF systems to external excitation is a classical topic in *Structural Dynamics* due to its application in many engineering systems.

Related Work (Wu and Jahanshahi 2019) predict the transient response of SDOF and MDOF systems using multi-level perceptron and convolutional neural networks (CNN). Their ML model forecasts displacement in SDOF systems while taking velocity, acceleration and excitation as inputs. In another work, (Stinis 2019) elucidate integration techniques to enforce the constraints from physical system in

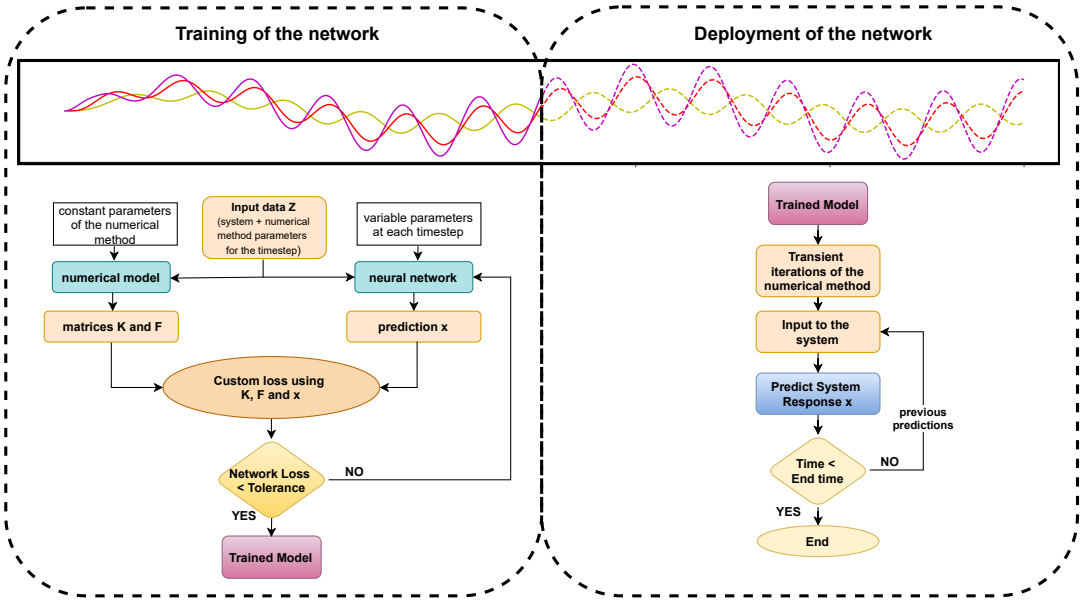


Figure 1: Overview of generalized physics-informed machine learning workflow for transient physical systems. This workflow eliminates modification of a ML model's training loss function or its architecture prior to the start of model training when presented with a new transient system.

supervised, semi-supervised and reinforcement learning for predicting flow-map of a dynamic system. Their model predicted the flow-map of a system iteratively using the present state. The error correcting terms and extra physics based constraints resulted in striking improvement during prediction of the Lorenz System.

Recently, (Zhang, Liu, and Sun 2020) apply a multi-LSTM neural network which maps the excitation force to the response of the system. They couple custom model architecture and loss functions to represent the underlying physics resulting in a model which outperforms conventional data driven LSTM models in terms of robustness and accuracy. Latterly, (Wang and Wu 2020) devise and present a *Knowledge-Enhanced Deep Learning* (KEDL) algorithm which trains a neural network (NN) to predict response of a system for a specific excitation. The authors used both input-output data and prior knowledge in the form of equations into the NN's training loss function.

Our Contribution It is important to note that almost all of the previous work explicitly define the governing PDE(s) of the given physical system as the NN's training loss function or, constitute a ML model architecture specific to the given physical system. A major drawback of such approach is that it is not generalizable. When a new transient system is provided, one needs to modify the ML model's training loss function or its architecture prior to the start of model training. *To overcome this problem, we introduce a generalized physics-informed machine learning workflow for transient problems. The proposed approach is highly generalizable, makes use of physics-guided loss function which results in physics-conforming and data-efficient ML model.* The proposed method can be used with any numerical method which results in a matrix form of equation system as in 1 .

$$Kx^h = F \tag{1}$$

Where x^h is the unknown of the problem and K, F are the matrices results from the numerical method used, discretization scheme used and boundary and initial conditions applied.
We analyze the predictive ability of our approach on time-series data of a linear SDOF and MDOF systems from structural dynamics.

2 Methodology

Consider a transient physical system characterized by a partial differential equation (PDE) defined on a domain Ω given by:

$$\mathcal{L}(x) = 0 \qquad on\ \Omega, \tag{2}$$

$$x = x_d \qquad on\ \Gamma_D, \tag{3}$$

$$\frac{\partial x}{\partial n} = g \qquad on\ \Gamma_N, \tag{4}$$

where x_d and g are the Dirichlet and Neumann boundary conditions given by equations 3 and 4 respectively. The solution to equation 2 can be computed using various methods such as FEM and FDM. In this contribution, we restrict the discussion to Galerkin-based FEM (Thomée 1984). A finite element formulation of Equation 2 on a domain with n nodes with given boundary conditions will result in the system of Equations as in 5 for each time step. Here, we assume all the necessary conditions on the test and trial spaces are fulfilled.

$$\underbrace{\begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,n} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n,1} & k_{n,2} & \cdots & k_{n,n} \end{pmatrix}}_{K(x^h)} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}}_{x^h} = \underbrace{\begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{pmatrix}}_{F}, \quad (5)$$

where $K(x_h)$ is the non-linear stiffness matrix, x_h is the discrete system response, and F is the Force vector. The elements of the stiffness matrix and force vector depends upon the time integration, numerical method and space discretization used.
Application of FDM or FVM also results in such system equations represented by matrices. The proposed method can also be applied with other numerical methods too.

2.1 Generalized physics-informed ML workflow

We propose a generalized physics-informed ML workflow to devise physics conforming ML models. Our workflow consists of the following steps (see Figure 1): Input Z to the system are the physical system parameters and numerical method specific parameters of the problem. A subset of this which varies for each timestep are the input to the neural network.

1. A conventional physics-based FEM model applies some numerical method to solve the PDE and outputs; response of the system x_h, stiffness matrix $K(x_h)$ and force vector F at discrete intervals of time.
2. The matrices from the physics-based FEM model are used in the custom loss function of the NN-based ML model. The ML model is a surrogate for the solution to the system described in equation 2. It takes last three timestep response as input and predicts the response at the present timestep. Last three timestep responses are taken as input since most of common time-integration schemes uses last two or three timesteps. The model is trained with the help of a physics-guided loss-function.
3. Upon completion of training, deploy the trained model to avoid the computationally expensive step of using conventional time-integration schemes and forward linear solvers to run simulations of transient physical systems. In the following, we tested the accuracy of the trained model on the untrained region instead of recursive prediction.

2.2 Physics-guided loss function

In conventional training setting, the actual response of the system (x) is compared with the one predicted by ML model. This comparison is done using an loss (error) function and the model tries to minimize the output of the loss function during training. Mean Squared Error (MSE) is one of the most common choice. As mentioned in Section 1, this approach has a major drawback – the resulting model is a "blackbox".

Our approach addresses the latter by making use of a custom physics-guided loss function in the training process instead of a conventional MSE loss function. This custom loss function operates on the stiffness $K(x^h)$ and force F matrices produced by the physics-based FEM model. It is given by:

$$Loss = \sum_{t=1}^{T} \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n} K_{i,j} x_j \right) - F_i \right)^2 \quad (6)$$

where T indicates the number of time steps used for training the model, n represents the number of unknowns which describe the discrete system response (x^h). The loss term represents the residual R of the equation in numerical methods. The prediction error gets magnified by K and results in "NaN" for some physical problems where K is high. This is avoided by scaling the loss term with F_{norm},the L2 norm of the force matrices used for the training. It also brings all the rows of the $\sum_{i=1}^{n} \left(\sum_{j=1}^{n} K_{i,j} x_j \right) - F_i$ in same scale and avoid the optimizer in concentrating on one row of the prediction array. Thus, the final loss function is given by:

$$Loss = \sum_{t=1}^{T} \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \left(\frac{K_{i,j} \times x_j}{F_{norm}} \right) \right) - \left(\frac{F_i}{F_{norm}} \right) \right)^2 \quad (7)$$

3 Experiments and Results

This section discusses the results of the proposed method in predicting the transient simulation results for SDOF and MDOF systems in structural dynamics. In both the examples, the model takes last three time-step values as input to predict the present time-step value. Currently the method is tested on the untrained data, not for a recursive prediction. Even though we focus on these two problems, the method can be directly used with any transient simulation solved using any numerical methods.

3.1 Single-degree-of-freedom system

A simple vibration system can be represented by a single mass connected to a spring and a damper. Such systems are called SDOF system and governed by the second order differential equation of single variable given by:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = f(t) \quad (8)$$

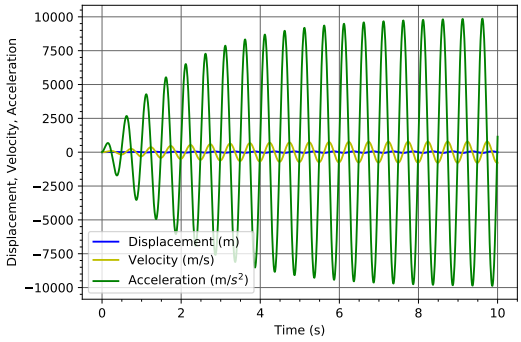
where m is the mass, c the damping constant, k the stiffness and $f(t)$ the excitation force. The response of the system can be represented as $[x, \dot{x}, \ddot{x}]$, where x is the displacement, $\dot{x}$ is the velocity and $\ddot{x}$ is the acceleration of the system.

m	10 kg
c	10 Ns/m
k	1580 N/m
$f(t)$	$1000sin(4\pi t)N$

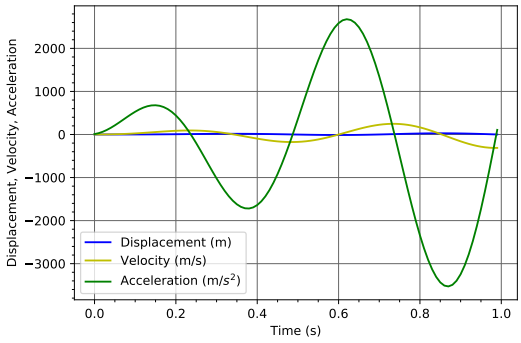
Table 1: System parameters

We use a physics-based solver to solve the equation and compute response of the system for a given excitation force

$f(t)$ with the help of (Newmark 1959) time integration scheme. The time integration used a time-step of $\Delta t = 0.01$ and beta of $\beta = -0.3$. A timestep of size 0.01 is used. Table 1 list down the system parameters used for carrying out the experiments. There are three unknowns which characterize the system described in Equation 8 – displacement x, velocity $\dot{x}$ and acceleration $\ddot{x}$. The response of the system for the first 10 seconds is given in Figure 2a and the response in a shorter time window is shown in Figure 2b.



(a) Response

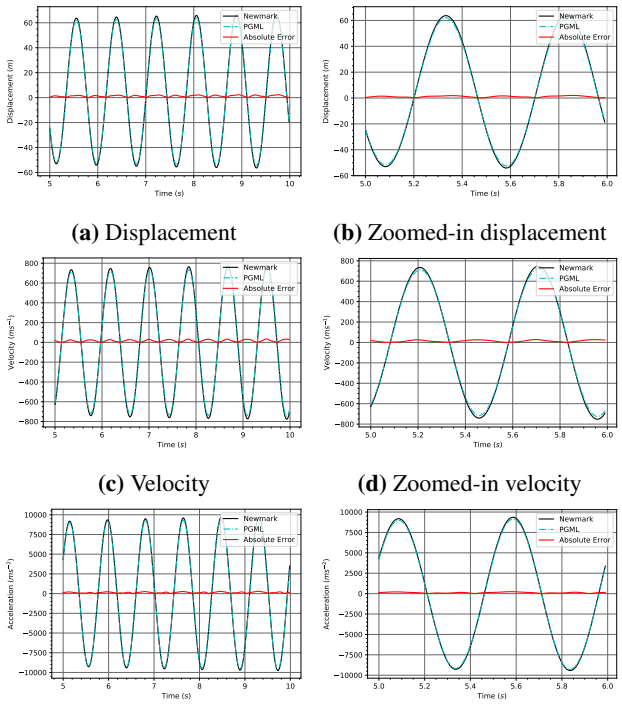


(b) Response in shorter time window

Figure 2: Response of SDOF system under external excitation

We devise a physics-guided ML model (PGML) based on neural network and uses a custom physics-guided loss function during training. The model's architecture is composed of a simple Long-short-term-memory (LSTM) network with 60 hidden units. We train this model on the system response of first 500 time-steps and the model is tested on the next 500 time-steps. The proposed neural network model takes system response from the previous three time-steps as input and predicts the response at the subsequent time-step. Adamax optimizer with a learning rate of $1e^{-4}$ is used for the training. A dropout value of 0.2 , $\beta_1 = 0.9$ and $\beta_2 = 0.99$ are used in the network.

Figure 3 shows the displacement of the system predicted using the trained network. It also shows the *reference solution* calculated using the physics-based FEM model with the help of Newmark time-integration scheme. A good accuracy is maintained between the predicted (PGML) and numerical



(a) Displacement **(b)** Zoomed-in displacement

(c) Velocity **(d)** Zoomed-in velocity

(e) Acceleration **(f)** Zoomed-in acceleration

Figure 3: Plots of predicted displacement, velocity and acceleration

method computed (Newmark) solutions. The plot of absolute values of errors between Newmark and PGML are also shown in Figure 3. The prediction of velocity and acceleration also follows the same behavior as that of displacement. It is to be noted that the proposed method is able to predict all three variables accurately even when their magnitude were in different scales.

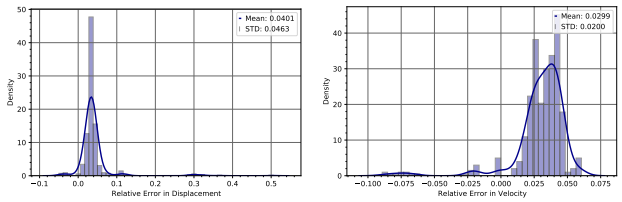
The error distribution in predicting displacement, velocity and acceleration are given in Figure 4. The relative error is plotted against the number of occurrence. It can be seen that error in all three variables are concentrated near zero. The relative error have a mean of 0.0401, 0.0299, 0.0033 for displacement, velocity and acceleration respectively. It is found that relative prediction error in acceleration have a high standard deviation of 0.1534. Whereas, standard deviation for relative error is 0.0463 and 0.0200 for displacement and velocity respectively.

3.2 Multiple degree of freedom system

The equation of motion of a multi degree of freedom system (MDOF) in structural dynamics is given by

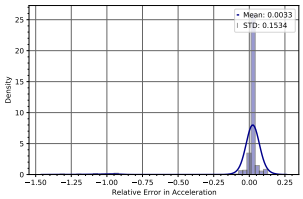
$$\mathbf{M}\frac{d^2\mathbf{X}}{dt^2} + \mathbf{C}\frac{d\mathbf{X}}{dt} + \mathbf{K}\mathbf{X} = \mathbf{F}(t) \tag{9}$$

where, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are the global mass, damping and stiffness matrices and $\mathbf{F}$ is the external force on the system. Here $\mathbf{X}$ represents the collection all degrees of freedoms of the system. We use a 20 DOF system to demonstrate the



(a) Error distribution in displacement prediction

(b) Error distribution in velocity prediction



(c) Error distribution in acceleration prediction

Figure 4: Distributions of relative error in prediction for SDOF system

model. Each DOF in this system is either the $x-$direction or the $y-$direction displacement of one of the 10 masses in the system. The external force applied is $\mathbf{F}(t) = sin(1.25 \times 2\pi t)N$ on all masses. The displacement of the masses of the system for the first 10 seconds is given in Figure 5. It is calculated using FEM with generalized alpha time integration scheme. We used a timestep of $\Delta t = 0.001$. Only selected DOFs are plotted in the figure.

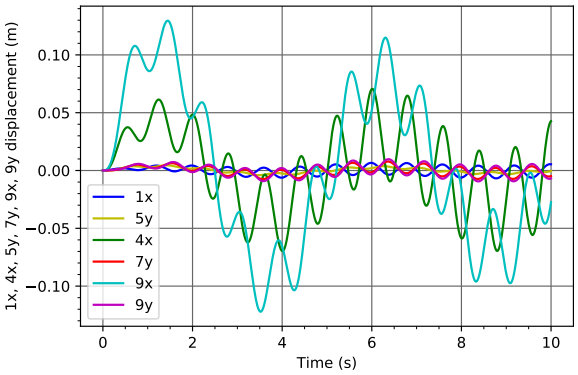
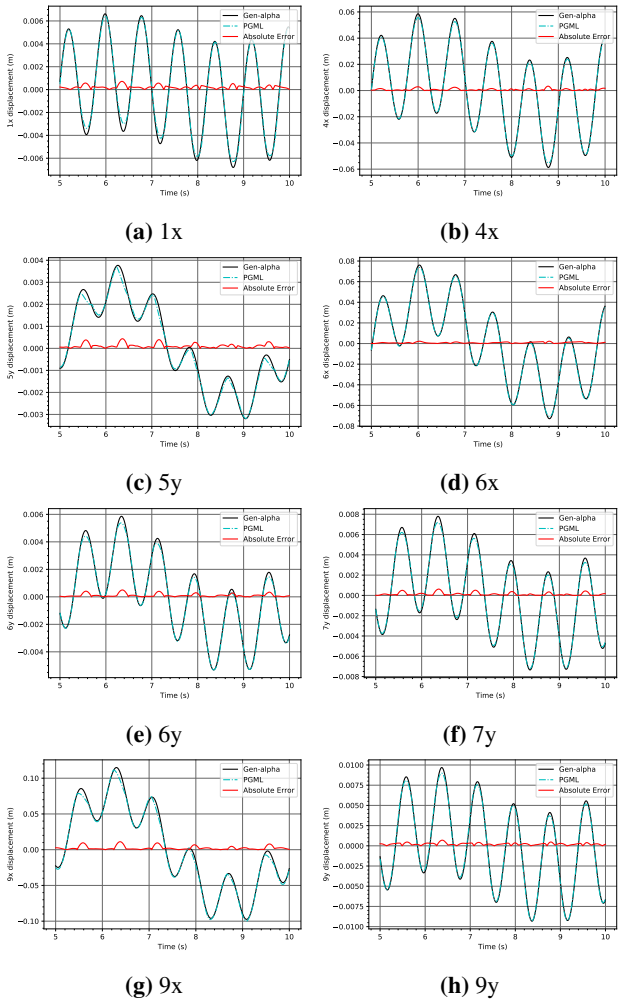


Figure 5: Response of MDOF system under external excitation

Each curve in the Figure 5 represents one degree of freedom, such as the first mass displacement in the $x-$direction $1x$ and the first mass displacement in the $y-$direction $1y$. Each DOF behave differently according the properties of the system and applied force. It makes the MDOF system more complex in comparison to the SDOF system.

The PGML model used consisted of a three layer LSTM network with 200 hidden units. The training used first 5000 time-steps for training the network. The trained model is used to predict the solutions of the remaining part of the simulation. We used Adamax optimizer with a learning rate of $1e^{-4}$, dropout value of 0.3 , $\beta_1 = 0.9$ and $\beta_2 = 0.99$.



(a) 1x **(b)** 4x

(c) 5y **(d)** 6x

(e) 6y **(f)** 7y

(g) 9x **(h)** 9y

Figure 6: Plots of predicted displacement for the selected DOFs

Figure 6 shows the prediction using the trained network for the next 5000 time steps (5-10 seconds). It is compared against the actual solution calculated using FEM with Generalized Alpha time integration scheme. Only selected DOFs from the 20 DOFs are plotted. The results show a good agreement between the model prediction and the FEM solution. The model maintained the prediction accuracy for all the twenty DOFs. Even though the displacements of various DOFs differ in scale and pattern across time, the model is able to capture these variations and make accurate prediction.

The error between the prediction and the FEM solution are also plotted in Figure 6. The prediction error is close to zero for all the DOFs. But, it is observed that the prediction error is high close to the crest and trough of the response for some of the DOFs. We think that the high gradient of

displacement at crests and troughs is causing this behavior. This can be solved by taking excitation force also as an input parameter as this is what triggering the change in displacement.

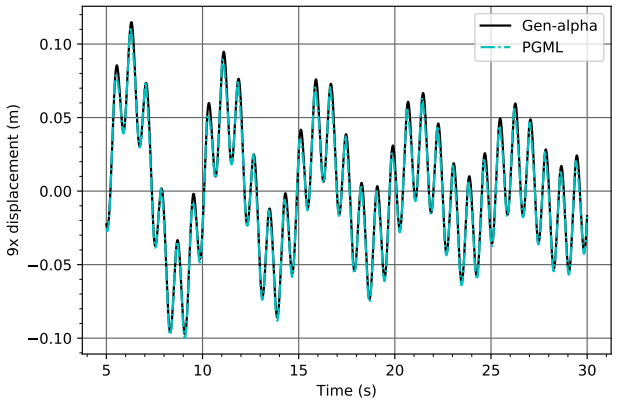


Figure 7: 9x displacement for longer time

MDOF prediction for longer duration The trained network is used to predict the solution of the simulation for longer duration (5-30 seconds). Figure 7 shows the prediction of displacement for 9x. The model maintains the accuracy even for varying amplitude and slope of the displacement curve. The same is observed with other DOFs too.

4 Conclusion

We introduced a generalized physics-guided machine learning workflow to train a neural network for transient simulations. The PGML model developed to achieve this goal takes system matrix and force vector constructed from the numerical method for the training. Since the loss function used directly reflects the residual from the numerical methods, the trained model is physics conforming in comparison to a conventional "black box" neural network. The proposed model can be easily adapted to different physics and numerical methods. The prediction capacity of the proposed model is demonstrated with the help of two examples from structural dynamics, SDOF and MDOF systems. The results point towards a promising algorithm for training neural networks for transient simulations. Such a trained model can be deployed in a real system and the signals from real system can be compared with prediction for anomalies.

5 Future Work

Transient simulations for real applications typically consist of large DOF systems. Such systems also involve complex input forces, which vary in time. The proposed algorithm needs to be tested with such complex scenarios for its robustness. In such a case, more parameters such as force and system parameters might be needed at the input side of the neural network. Another drawback of the current implementation is that there is no mechanism to correct the error that accumulates over a long series of recursive predictions. Presently, recursive prediction using the same model diverges due to the accumulation of the error. Algorithms and architectures tailored for recursive prediction also to be tested with our model in the future work.

The proposed algorithm can only be used for predicting the simulation for later timesteps once trained with the simulation done so far. The next step includes training a model that can produce a generalized model, performing a complete simulation given the initial conditions and system parameters.

References

Forsythe, G. E.; and Wasow, W. R. 1960. Finite Difference Methods. *Partial Differential* .

Kutz, J. N. 2017. Deep learning in fluid dynamics. *Journal of Fluid Mechanics* 814: 1–4.

Newmark, N. M. 1959. A method of computation for structural dynamics. *Journal of the engineering mechanics division* 85(3): 67–94.

Reich, Y.; and Barai, S. 1999. Evaluating machine learning models for engineering problems. *Artificial Intelligence in Engineering* 13(3): 257–272.

Stinis, P. 2019. Enforcing constraints for time series prediction in supervised, unsupervised and reinforcement learning. *arXiv preprint arXiv:1905.07501* .

Tarca, A. L.; Carey, V. J.; Chen, X.-w.; Romero, R.; and Drăghici, S. 2007. Machine learning and its applications to biology. *PLoS Comput Biol* 3(6): e116.

Thomée, V. 1984. *Galerkin finite element methods for parabolic problems*, volume 1054. Springer.

von Rueden, L.; Mayer, S.; Beckh, K.; Georgiev, B.; Giesselbach, S.; Heese, R.; Kirsch, B.; Pfrommer, J.; Pick, A.; Ramamurthy, R.; et al. 2019. Informed Machine Learning–A Taxonomy and Survey of Integrating Knowledge into Learning Systems. *arXiv preprint arXiv:1903.12394* .

Wang, H.; and Wu, T. 2020. Knowledge-Enhanced Deep Learning for Wind-Induced Nonlinear Structural Dynamic Analysis. *Journal of Structural Engineering* 146(11): 04020235.

Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; and Kumar, V. 2020. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919* .

Wu, R.-T.; and Jahanshahi, M. R. 2019. Deep convolutional neural network for structural dynamic response estimation and system identification. *Journal of Engineering Mechanics* 145(1): 04018125.

Zhang, R.; Liu, Y.; and Sun, H. 2020. Physics-Informed Multi-LSTM Networks for Metamodeling of Nonlinear Structures. *arXiv preprint arXiv:2002.10253* .

Zienkiewicz, O. C.; Taylor, R. L.; Taylor, R. L.; and Taylor, R. L. 2000. *The finite element method: solid mechanics*, volume 2. Butterworth-heinemann.