Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2023) 836-844

ECCOMAS
Proceedia

UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12-14 June 2023

A PHYSICS-INFORMED NEURAL NETWORK SURROGATE MODEL FOR TIDAL SIMULATIONS

James Donnelly[1,2], **Alireza Daneshkhah**[2], **and Soroush Abolfathi**[1]

[1]School of Engineering,
University of Warwick
e-mail: {jamie.donnelly, soroush.abolfathi}@warwick.ac.uk

[2] Centre for Computational Science and Mathematical Modelling,
Coventry University
e-mail: ac5916@coventry.ac.uk

Abstract. *Notable advancements in computational power has facilitated the utilization of intricate numerical methods in flood modeling in recent years. Hydrodynamic modelling approaches to predict flood inundation are robust compared to empirical approaches, which are solely based on the statistical patterns of hydrological variables obtained from observed data. Despite the benefits of numerical flood modelling tools, significant computational costs of implementing such models at high spatio-temporal resolutions have limited their applications. Data-driven machine learning (ML) models, designed to learn the underlying governing equations of the numerical models provide a computationally robust and fast alternative to the existing flood prediction models. However, these ML-based models often struggle to generalise for 'small-data' regime tasks, which are common to simulation-based tasks in fluid dynamics. In this study, to overcome extrapolation and over-fitting challenges of data-driven surrogate models, a Physics-Informed Neural Network model is adopted for surrogate modelling of a hydrodynamic simulator developed to investigate wave characteristics and tidal dynamics in the English Channel. A novel approach to encoding the conservation of mass into a deep learning model is introduced by including additional terms in the optimisation criterion, acting to regularise the model, avoid over-fitting and produce more physically consistent predictions by the surrogate. The model outlined improved performance by 10-20% across a range of metrics compared to a data-driven alternative.*

Keywords: Physics-Informed Machine Learning, Surrogate Modelling, PINNs, Hydrodynamic Modelling, Coastal Resilience, Flood forecasting

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2023.
doi: 10.7712/120223.10379.19908

1 Introduction

Data-driven approaches to machine learning (ML) and deep learning (DL; [1, 2]) heavily rely on the available data to guide decisions regarding model selection and parameterization. In this paradigm, in order to select the best model structure, one needs to determine the number of hidden layers, neurons per layer and type of activation functions in a neural network (NN), and thus requires the state-of-art skills and considering all possible approaches, including trial-and-error, to reach the best model structure (e.g., neural network architecture or Bayesian network structure [3]). The selected model is mostly compatible to the data, and in some cases to the elicited expert opinions [4], in the sense that the best model structure minimises the goodness-of-fit criteria. After selecting the best model structure learned from the data, parameters will be estimated by minimising a loss function, possibly subject to some constraints on the parameters, which is often equivalent to minimising the data-fit' error on the training data. For instance, in supervised learning tasks, and in particular for regression problems, this usually involves selecting a set of parameters, $\boldsymbol{\theta}^*$, that minimise the mean of the squared error on the training data,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, L(\mathbf{y}, f(\mathbf{x};\boldsymbol{\theta})), \tag{1}$$

$$L(\mathbf{y}, f(\mathbf{x};\boldsymbol{\theta})) = \mathbb{E}[(\mathbf{y} - f(\mathbf{x};\boldsymbol{\theta}))^2], \tag{2}$$

where, $\mathbf{x}$ are the selected inputs, $\mathbf{y}$ are the corresponding observed/computed outputs, and $f(.;\boldsymbol{\theta})$ is the selected model using ML approaches, parameterised by $\boldsymbol{\theta}$. Using the loss function outlined in Eq. (2) for optimization is a data-driven approach that solely considers the disparity between the predicted and observed target values under a given parameterization, without incorporating external knowledge or constraints.

Supervised ML tasks can be reduced to attempting to learn some unknown distribution, $P(\mathbf{y}|\mathbf{x})$, allowing inferences to be made about some system or real-world problem. In the case of prediction, the task is to learn a predictive distribution, $P(\mathbf{y}^* \mid \mathbf{y}, \mathbf{x}, \mathbf{x}^*)$, given the training dataset $\mathcal{D} = (\mathbf{x}, \mathbf{y})$. The deep learning literature highlight that increasing the scale of models and data yields continuously better estimates $P(\mathbf{y}^* \mid \mathbf{y}, \mathbf{x}, \mathbf{x}^*)$, even in highly complex settings. However, when dealing with sparse data, an alternative method for improving the generalization performance of machine learning models is to incorporate prior knowledge about the outputs, denoted as $\pi(\mathbf{y})$. By incorporating prior knowledge into the modeling process, the resulting model can take into account additional information beyond the training data alone. This may allow the model to make more accurate predictions on new, unseen data, particularly when the available training data is limited.

When applying ML approaches to address problems in engineering and environmental science, the distribution being learned, $P(\mathbf{y}|\mathbf{x})$, is often a real physical system with known laws governing its behaviour. However, most applications of ML in these settings, for tasks such as inferring data-driven solutions to partial differential equations (PDEs) or applying inverse methods for PDE discovery [5, 6, 8, 7], overlook the abundance of prior knowledge about these systems and utilise purely data-driven approaches instead. Physics Informed Machine Learning ([9]) was developed to tackle the shortcomings of data-driven ML methods by incorporating information about the underlying physical laws governing a system in the development and training of the ML models. Through the incorporation of prior information extracted from the physical laws governing the system, the aim is to build more physically consistent models,

improve data efficiency, enhance generalisation, and promote transparency and interpretability [10].

This study extends the application of Physics-Informed Neural Networks (PINNs) to complex realistic problems in fluid dynamics by constructing a PINNs surrogate model [18, 11, 12, 13] for numerical hydrodynamic simulators. The model is developed for fast inference of latent solutions of fluid's free-surface elevations at high spatiotemporal resolutions, as generated by numerical simulators solving the two-dimensional Navier-Stokes equations. Further, rather than directly incorporating PDEs in the loss function, a novel approach to physics-informed regularisation is developed to account for the conservation of mass in predictions made by the surrogate model. The proposed approach is demonstrated by constructing surrogate PINNs models for a widely used hydrodynamic simulator, Delft3D [19, 20]. The performance of the proposed surrogate model is assessed against a data-driven approach.

1.1 PINNs surrogate model

Advanced numerical models (i.e. simulators) have rapidly become an indispensable tools in many scientific and engineering problems, such as fluid dynamics. Simulators can replicate intricate physical phenomena by effectively capturing the underlying physics of the problem. However, this typically requires a considerable amount of computational resources, in terms of CPU/GPU-hours. Flood modelling is an example of a domain where computational methods have gained rapid and widespread adoption, owing to the scarcity of empirical data in contrast to the substantial flexibility offered by computational methods. The hydrodynamic models adopted for flood modelling, such as Delft3D used here [19], implement a discrete version of the Navier Stokes equations to simulate the underlying fluid flow accurately. However, such models are often computationally very expensive, which highlights the necessity and significance of developing robust surrogate models as fast alternatives.

When utilising a hydrodynamic simulator, only the boundary conditions and outputs (e.g., fluid depths or velocity components) are known post-simulation, such that direct access to the derivatives is usually not possible. Previous PINNs implementations [22, 16, 6, 15, 21] overcome this problem by regressing outputs, $u(t, x)$, onto their spatiotemporal coordinates such that $\hat{u}(t, x) = f(t, x)$, where $f(\cdot, \cdot)$ is a neural network. Then by backpropagating back to the original inputs, it's possible to calculate $\frac{\partial \hat{u}}{\partial x}$ or $\frac{\partial \hat{u}}{\partial t}$ which can then be incorporated into the loss function via PDE residuals. For example, if inferring latent solutions to the heat equation in one-dimension, the governing PDE can be expressed as,

$$\frac{\partial u}{\partial t} - c\frac{\partial^2 u}{\partial^2 x} = 0. \tag{3}$$

By using a neural network to estimate $\hat{u}(t, x)$ and then backpropagating to obtain $\frac{\partial \hat{u}}{\partial t}$ and $\frac{\partial^2 \hat{u}}{\partial^2 x}$, repeating the process n times to obtain n^{th}-order derivatives, an additional term can be included into the loss function,

$$L(u, \hat{u}) = (u - \hat{u})^2 + \left| \frac{\partial \hat{u}}{\partial t} - c\frac{\partial^2 \hat{u}}{\partial^2 x} \right|^2, \tag{4}$$

where the first term is the data-fit error term. An illustration of this existing type of PINNs model and network architecture can be seen in Fig. (1).

The API (Application Programming Interface) of a model like this is limited in that during inference it can only robustly generalise to settings with similar boundary conditions to those

used during training. The standard PINNs model does not allow for flexibility in the parameterisation of boundary conditions during inference. [14] states that a PINNs model as outlined in their study is agnostic to boundary conditions. However, in general, for new boundary conditions, a new model or additional training data to update an existing model is likely required to obtain good generalisation. The model proposed in this study overcomes this limitation by explicitly parameterising the boundary conditions in the neural network's input, allowing straightforward inference of latent solutions under widely varying boundary conditions and without the backpropagation requirement to obtain derivatives.

In this study, a novel PINNs-based surrogate model of the following form is proposed:

$$\hat{Y}^t = f(\mathbf{s}^t, ..., \mathbf{s}^{t-N}), \tag{5}$$

where $\hat{Y}^t$ is a matrix representing predicted fluid depth in all cells (i.e., $n_x \times n_y$) in the domain, and $\mathbf{s}^t$ are the boundary conditions at time t, and f is a neural network. Based on the surrogate model presented in Eq. (5), the fluid depths at time t are regressed onto the boundary conditions at time t and the N previous time-steps. A flexible surrogate model of this kind has the capability to serve as a substitute for the original numerical model.

For hydrodynamic simulators the continuity equation can be written in the following form:

$$\Delta(\text{storage}) = \Delta(\text{sources} - \text{sinks}), \tag{6}$$

which forms the basis of the physics-informed loss function for the PINNs model proposed in this study. This condition can be re-expressed for fluid depths in cell-(i, j) predicted by the PINNs surrogate, $\hat{y}_{ij}^t$, and true fluid depths y_{ij}^t, as,

$$\Delta x \Delta y \sum_i \sum_j \hat{y}_{ij}^t - \Delta x \Delta y \sum_i \sum_j y_{ij}^{t-1} = \Delta t \sum_i \sum_j s_{ij}^t - \Delta t \sum_i \sum_j q_{ij}^t, \tag{7}$$

where q_{ij}^t denotes the inflow at time t, meaning that the aggregate volume of fluid in the domain predicted by the surrogate at time t minus the true volume at time $t-1$ should equal the aggregate inflow minus the aggregate outflow. Starting from these conditions, an appropriately scaled physics-informed loss function is derived,

$$L(Y^t, \hat{Y}^t) = \mathbb{E}[(Y^t - \hat{Y}^t)^2] + \left(\frac{\text{ReLu}(\hat{v}^t - v^{t-1} - \Delta t \sum_i \sum_j s_{ij}^t)}{\Delta x \Delta y \times n_x \times n_y} \right)^2 + \left(\frac{\text{ReLu}(v^{t+1} - \hat{v}^t - \Delta t \sum_i \sum_j s_{ij}^{t+1})}{\Delta x \Delta y \times n_x \times n_y} \right)^2, \tag{8}$$

where the denominator, $\Delta x \Delta y \times n_x \times n_y$, is the total area of modelling domain, and ReLU is the Rectified Linear Unit function, so that $\text{ReLU}(x) := \max(0, x)$. This total loss function consists of a data-fit error term, $\mathbb{E}[(Y^t - \hat{Y}^t)^2]$, and then adds additional terms to the loss function when the predictions made violate the conservation of mass constraints expressed through the second and third terms.

1.2 Neural network architecture

Previous PINNs works [5, 15, 16, 17] utilised a dense fully-connected MLP (multi-layer perceptron) architecture as the basis of the model (as seen in Fig. (1)). However, consideration for

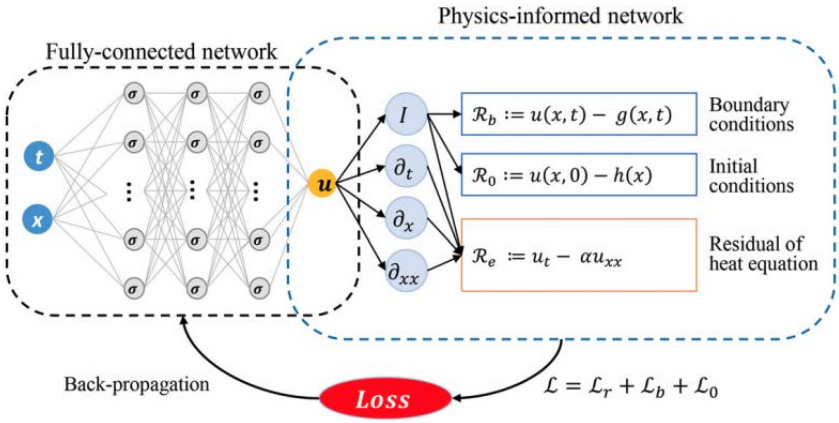

Figure 1: xample of an existing PINNs architecture, regressing latent solutions $\mathbf{u}$ onto spatiotemporal coordinates, $[x, t]$, as illustrated in [15].

alternative architectures was not made in the existing studies. In this study alternative architectures were explored and preliminary testing found that a convolutional neural network (CNN), with convolutions in 1D over vector inputs, was the best performing base architecture. Therefore this architecture was adopted for the surrogate models presented and also highlights that the physics-informed loss function outlined in this study does not depend on any one architecture and can be applied to any differentiable network. In order to benchmark the performance of the PINNs surrogate, the results will be compared against an ordinary data-driven CNN, i.e., the same network architecture but trained purely using the MSE loss function (herein referred to as 'CNN').

To ensure that any differences in performance between the PINN models and the CNN are solely attributed to the physics-informed loss function, we train both models multiple times using the same initial weight configuration. By repeatedly sampling random weight initialisations, $\boldsymbol{\theta}_i^0$, and then training the PINNs model and the CNN with their respective loss functions to estimate optimal parameters (as outlined in Eq. (1)), $\boldsymbol{\theta}_i^*$, the models' test performance is averaged over the subsequent networks to mitigate effects of favourable or unfavourable initial weights. The test performance of the models then becomes,

$$L(\mathbf{y}^*, f(\mathbf{x}^*)) = \mathbb{E}_i[L(\mathbf{y}^*, f(\mathbf{x}^*; \boldsymbol{\theta}_i^*)) \mid \boldsymbol{\theta}_i^0]. \tag{9}$$

where $(\mathbf{x}^*, \mathbf{y}^*)$ denote the test inputs and outputs. The physics-informed loss function is likely highly non-convex in the parameters and therefore predictive performance could be sensitive to the initialisation of the weights of the network. Taking expectations with respect to this variable allows for a high level of certainty that any deviations in performance against the CNN are entirely the result of modelling and not due to favourable/unfavourable initial conditions in the weight-space.

1.3 Performance metrics

In order to assess model predictive performance two metrics are utilised in this study. The first metric used is the Root Mean Squared Error (RMSE) of the aggregate water depth ($\mathbf{y}$) and predicted depths ($\hat{\mathbf{y}}$) resulted from the fitted PINNs or the CNN model:

$$\text{RMSE} = \sqrt{\mathbb{E}[(\mathbf{y} - \hat{\mathbf{y}})^2]}. \tag{10}$$

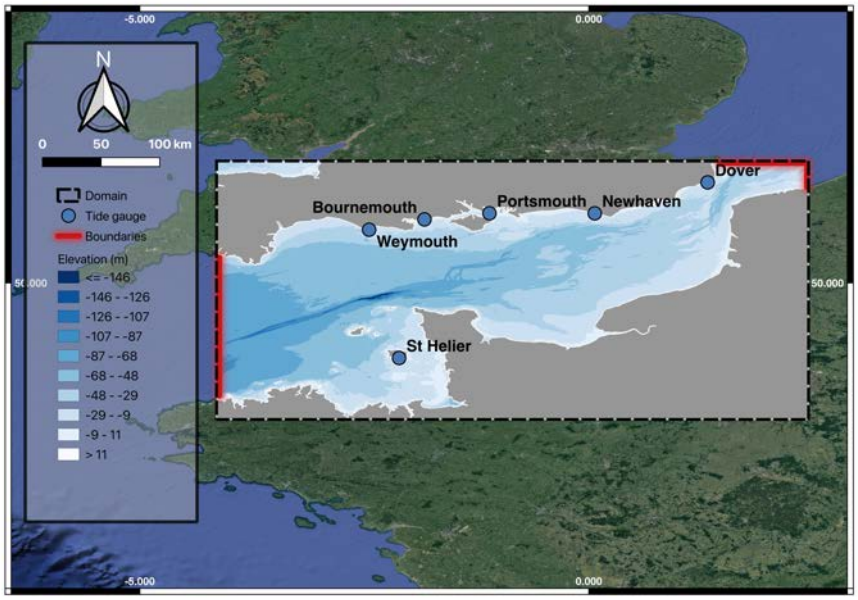

Figure 2: Delft3D modelling domain of the English Channel with bathymetry, boundaries and tide gauge validation locations illustrated.

RMSE is utilised because it scales the prediction errors back to the original unit of measurement, $[m]$, and so gives interpretable results. Second metrics used is a modified version of the RMSE described as:

$$\text{mRMSE} = \sqrt{\frac{\sum_{\Omega} (y_{\Omega_i} - \hat{y}_{\Omega_i})^2}{|\Omega|}}, \tag{11}$$

where $\Omega = \{(i,j) \mid y_{ij}^t \cup \hat{y}_{ij}^t > 0\}$. This metric gives a more realistic estimate of the true error by only considering cells where fluid is either predicted or observed at any time in the dataset, and so does not include cells that remain dry at all times, which can misleadingly reduce the mean error when included.

2 Results

2.1 Delft3D

The proposed model was used as a surrogate model for tidal simulations in the English Channel using Delft3D numerical model, with the simulation domains illustrated in Fig. (2). The simulator was used to generate accurate data on the spatiotemporal water level variations for five discrete 1-month periods, obtaining simulated data at hourly intervals (i.e. hourly time-step). The results obtained from the numerical simulator were validated using empirical tide gauge data collected at a number of validation locations to ensure consistency of the model outputs with the real measured data. By using the simulator's spatiotemporally-varying boundary conditions as input data and the hourly water level data for the domain as outputs, the PINNs and CNN models were trained to reproduce the mapping described by the original simulator. Cross-validation (CV) is performed to validate the surrogates by partitioning data according to the discrete simulations, which were used to train and test the model, resulting in 5-fold CV for model validation and assessment.

The computed performance metrics for the validated models are outlined in Table (1). Enhanced performance is observed in using the PINNs surrogate model compared to the CNN

Model	mRMSE	RMSE	Volume Discrepancy
CNN model (MSE)	0.154	0.150	2.26×10^6
PINNs	0.139	0.137	1.84×10^6

Table 1: Mean CV scores for the CNN and PINNs model on the Delft3D data.

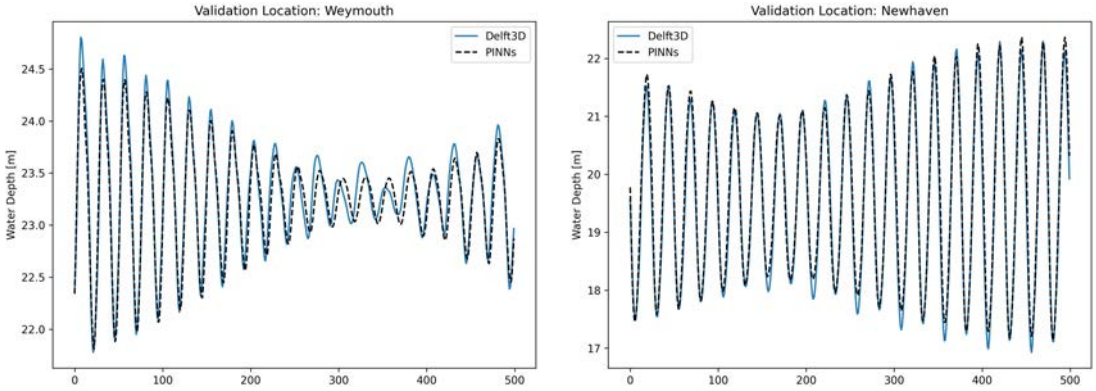

Figure 3: PINNs predictive performance over random 500-timestep periods at the Weymouth and Newhaven locations./

model. However, in this case, the average reduction in the error rate for the modified RMSE metric is approximately 11%. Hence, the PINNs still illustrate an improvement over the purely data-driven surrogate model (i.e. CNN). However, the expected reduction in error is not very significant. It can be interpreted that an average reduction in error of over 10% could not be due to the random noise, and the proposed PINNs-based approach is a more robust surrogate model in this sense. Additionally, the PINNs model produces a reduction in the volume discrepancy.

Fig. (3) shows the PINNs model's predictions at two of the validation tide gauge locations, showing a snapshot of the predicted values by highlighting a random period of 500 timesteps (approximately 10 days) for each of the locations. Fig. (3) highlights the robustness and appropriateness of the PINNs surrogate model, and how the model overcame a high degree of variability in the mean water depth in each cell, as shown by the widely varying average depth at each of the verification locations.

3 Conclusion

This study proposed a novel formulation of a PINNs model applied to the development of surrogate models for hydrodynamic simulators. The simulator and subsequent surrogate models were applied to tidal modelling at a high spatiotemporal scale. A CNN has been used as the underlying network architecture, highlighting that the PINNs approach can be straightforwardly applied to architectures beyond dense fully-connected networks. The obtained results showed that when testing the model on Delft3D data, the PINNs model outperformed the conventional CNN model in which the same network architecture was trained using an ordinary MSE loss function. Furthermore, the PINNs model was found to exhibit more stable optimisation behaviour, especially around the minima, although convergence took more weight updates. By conditioning for the effects of random weight initialisations in the networks, there is a high degree of certainty that any performance differences between the models arise purely from the optimisation criteria of the networks.

The primary cost associated with the PINNs model compared to the purely data-driven coun-

terpart has increased model development costs. The design of the custom loss function is more demanding than utilising the existing loss function provided by popular neural network frameworks. Furthermore, having recurrent connections to past and future target values required a modified training routine. However, by leveraging advanced autograd mechanics in frameworks such as PyTorch, fast, vectorised implementations of complex custom loss functions can easily be integrated into existing models without much additional computational overhead. This study extends existing PINNs approaches by not requiring compromises on the functional API of the network. Existing PINNs approaches tend to require spatiotemporal coordinates as input to the network, outputting a latent solution under the same boundary conditions as the training data. Here, the ability to flexibly parameterise boundary conditions in the network is maintained such that the surrogate model can be straightforwardly applied to entirely new boundary conditions.

REFERENCES

[1] K. Khosravi, F. Rezaie, J.R. Cooper, Z. Kalantari, S. Abolfathi, J. Hatamiafkoueieh, Soil water erosion susceptibility assessment using deep learning algorithms. *Journal of Hydrology*, 129229, 2023.

[2] B. Ghiasi, R. Noori, H. Sheikhian, A. Zeynolabedin, Y. Sun, C. Jun, S. Abolfathi, Uncertainty quantification of granular computing-neural network model for prediction of pollutant longitudinal dispersion coefficient in aquatic streams. *Scientific reports*, **12(1)**, 4610, 2022.

[3] J. Q. Smith, and A. Daneshkhah, On the robustness of Bayesian networks to learning from non-conjugate sampling. *International Journal of Approximate Reasoning*, **51(5)**, 558-572, 2010.

[4] A. Daneshkhah, and J. E. Oakley, Eliciting multivariate probability distributions. *Rethinking risk measurement and reporting*, **1**, 1-43, 2010.

[5] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561.*, 2019.

[6] 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. *Journal of Computational Physics*, **378**, 686–707, 2019.

[7] O. Chatrabgoun, M. Esmaeilbeigi, M. Cheraghi, A. Daneshkhah, Stable Likelihood Computation for Machine Learning of Linear Differential Operators with Gaussian Processes. *International Journal for Uncertainty Quantification*, **12(3)**, 2022.

[8] Y. Bar-Sinai, S. Hoyer, J. Hickey, M.P. Brenner, Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, **116**, 15344-15349, 2019.

[9] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning. *Nature Reviews Physics*, **3**, 422-440, 2021.

[10] K. Kashinath, M. Mustafa, A. Albert, J.L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, **379**, 20200093, 2021.

[11] A. O'Hagan, Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, **91**, 1290-1300, 2006.

[12] B. Sudret, S. Marelli, J. Wiart, Surrogate models for uncertainty quantification: An overview. *2017 11th European conference on antennas and propagation (IEEE-EUCAP)*, 793-797, 2017.

[13] S. Conti, J.P. Gosling, J.E. Oakley, A. O'Hagan, Gaussian process emulation of dynamic computer codes. *Biometrika*, **96**, 663-676, 2009.

[14] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, **367**, 1026-1030, 2020.

[15] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, **143**, 2021.

[16] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, **360**, 112789, 2020.

[17] D. Zhang, L. Lu, L. Guo, G.E. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, **397**, 108850, 2019.

[18] J. Donnelly, S. Abolfathi, J. Pearson, O. Chatrabgoun, A. Daneshkhah, Gaussian process emulation of spatio-temporal outputs of a 2D inland flood model. *Water Research*, **125**, 119100, 2022.

[19] D. F. D. Manual, Delft3D FLOW: Simulation of multi-dimensional hydrodynamic flows and transport, User Manual. Deltares System. **V4.05**, 2022.

[20] M. Fanous, J. M. Eden, R. Remesan, R., A. Daneshkhah, Challenges and Prospects of Climate Change Impact Assessment on Mangrove Environments Through Mathematical Models. *Environmental Modelling and Software,*, https://doi.org/10.1016/j.envsoft.2023.105658, 2023.

[21] F. Sahli Costabal, Y. Yang, P. Perdikaris, D.E. Hurtado, E. Kuhl. Physics-informed neural networks for cardiac activation mapping. *Fontiers in Physics*, **8**, 42, 2020.

[22] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 1-12, 2022.