Can Machine Learning Bridge the Gap Between Lagrangian Mesh-Free Methods and High-Order Interpolants?

Lucas Gerken Starepravo, Jack King,
Georgios Fourtakas, Ajay Harish
School of Engineering, University of Manchester, Manchester, UK
lucas.gerkenstarepravo@postgrad.manchester.ac.uk

Steven Lind
School of Engineering, Cardiff University, Cardiff, UK

I. Introduction

Lagrangian mesh-free numerical frameworks offer geometry flexibility, addressing limitations commonly encountered in traditional mesh-based methods. However, Lagrangian methods tend to operate with low-order convergence [1]. Although high-order approximations can be advantageous, yielding results at low resolution comparable to those of low-order approximations at much higher resolutions, the computational cost becomes prohibitive in Lagrangian frameworks due to the need for frequent matrix inversion to compute weights.

To bridge the gap between Lagrangian mesh-free simulations and Eulerian high-order mesh-free numerical methods, the authors proposed two approaches that leverage machine learning (ML) to surrogate specific segments of the latter. In principle, by achieving more accurate weight computations than current Lagrangian methods and lower computational cost compared to high-order mesh-free methods, these surrogates can be applied to Lagrangian frameworks, resulting in an ML-driven Lagrangian high-order mesh-free method.

The current research aims to report a limitation found in the application of multi-layer perceptrons (MLP) and physics-informed neural networks (PINNs) to surrogate segments of high-order mesh-free methods. In the first approach, neural networks were trained to surrogate the high-order kernel. In the second approach, neural networks were trained to solve the dense, ill-conditioned, dense, and low-rank linear system that arises in the high-order mesh-free methods. These approaches were tested with the Local Anisotropic Basis Function Method (LABFM) [2], a high-order mesh-free method.

II. Methodology

A. The Local Anisotropic Basis Function Method

LABFM is an Eulerian mesh-free approach that allows arbitrarily high-order convergence with disordered nodes. In this framework, a general discrete operator is defined to approximate differential operators

$$L_i^d(\cdot) = \sum_{j=1}^{\mathcal{N}} (\cdot)_{ji} w_{ji}^d \tag{1}$$

where $L_i^d(\cdot)$ is the differential operator; d identifies the differential operator being approximated; w_{ji}^d are the weighted ABFs for each support region; and the $\mathcal{N}$ is the number of nodes within a computational stencil. The weighted anisotropic basis functions (ABFs) are obtained by calculating the dot product between the weights, $\boldsymbol{\Psi_i^d}$ and the ABFs, $\boldsymbol{W_{ji}}$. The weights are obtained by solving the following linear system $\boldsymbol{M_i \Psi_i^d = C^d}$. The vector $\boldsymbol{C^d}$, known as the "LABFM moments", ensures that the weights $\boldsymbol{\Psi_i^d}$ correspond to the differential operator that is being approximated. The linear system $\boldsymbol{M_i}$ contains weighted distances of the nodes within the computational stencil, and it is constructed by taking the tensor product of the vector containing Taylor monomials, $\boldsymbol{X_{ji}}$ and the ABFs, and summing for all nodes

$$\boldsymbol{M_i} = \sum_{j=1}^{\mathcal{N}} \boldsymbol{X_{ji}} \otimes \boldsymbol{W_{ji}} \tag{2}$$

B. High-order Kernel Surrogate with Machine Learning

In this approach, the authors leveraged the universal approximation property of feed-forward neural networks to map the distance between support nodes and central node to the weighted ABFs. The input of the networks were the distances between the support nodes and the central node, $\boldsymbol{x}_{ij} \in \mathbb{R}^{\mathcal{N}}$ and $\boldsymbol{y}_{ij} \in \mathbb{R}^{\mathcal{N}}$ and the outputs were the weighted sum of the ABFs, $\boldsymbol{w}_{ji}^d = \boldsymbol{W}_{ji} \cdot \boldsymbol{\Psi_i}^d$.

Two types of networks were trained: fully connected feed-forward MLPs and feed-forward PINN, with LABFM moments included in the loss function. The training dataset consisted of approximately 100,000 points, where 80% where used for training and the remaining for validation.

The networks underwent offline training and their results were evaluated in three distinct phases. The first was a qualitative analysis of the predicted weighted ABFs. The second phase involved using the networks to compute the LABFM moments with the predicted weights. Lastly, a convergence study was conducted using the predicted weights with a test function.

C. Solving Linear Systems with Machine Learning

The second approach is a more direct way to accelerate LABFM with ML. We surrogate the solution of the dense, ill-conditioned, low-rank linear systems with a constant right-hand side vector. This approach contains fewer intermediate steps compared to the approach shown in Section II-B.

The models built in this approach had as input the flattened linear system $\boldsymbol{M_i}$, and as output the weights, $\boldsymbol{\Psi_i^d}$. Given the moments are constant for each differential operator being approximated, they can be omitted from the network.

The neural network architectures employed to solve the linear system included: MLPs, PINN with LABFM moments included in the loss function. The training dataset consisted of approximately 400,000 points, where 80% were used for training and the remaining for validation.

The networks underwent offline training, with accuracy assessed via LABFM moments of predicted weights, convergence analysis against LABFM with LU factorisation, and computational time comparison for solving the linear system.

III. Results

A. Analysis of Predicted Weighted Anisotropic Basis Functions

Figure 1 presents the weighted ABFs for a second-order Laplace operator approximation, while Fig. 2 shows the absolute error between predictions and ground truth. In the actual weight distribution, support nodes closer to the central node have larger weights, and the predicted weights successfully replicate this pattern. Errors were approximately two orders of magnitude smaller than the maximum weight, indicating that the neural networks accurately captured the overall weight distribution.

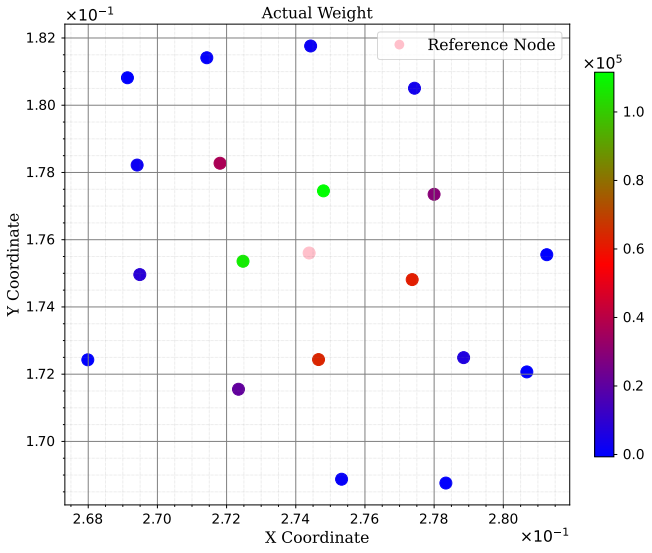

Fig. 1. LABFM computed weights of second-order approximation of Laplace operator. Reference node in pink.

To quantitatively evaluate the results, the LABFM moments of the predicted weighted ABFs were compared to the target moments corresponding to the second-order approximation of the Laplace operator. The LABFM moments were calculated, and the absolute errors between the predicted and target moments were computed for all points. These errors were then averaged to provide an overall measure of accuracy. Additionally, the standard deviation of the absolute errors was calculated to assess the variability in the predictions. The results are summarized in Table I.

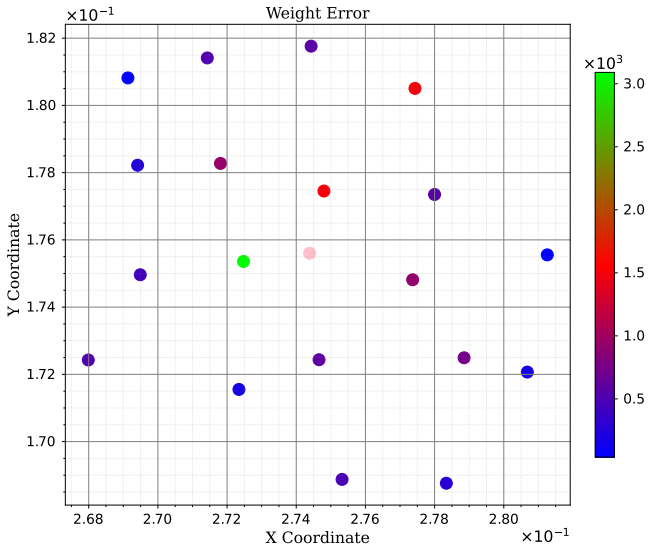

Fig. 2. Absolute error between LABFM computed and PINN predicted weights of second-order approximation of Laplace operator. Reference node in pink.

Table I shows that the first moments are predicted significantly worse for all models compared to the second-order moments. The PINN performed slightly better than the MLP, with results one order of magnitude more accurate. However, the high error in the first moments of both models indicates that second-order derivatives are being poorly approximated, since first-order derivatives are not eliminated by the predicted weighted ABFs. In practice, errors of $\mathcal{O}(s)$ are being generated for a second-order approximation, where s is the average spacing between nodes.

Lastly, a convergence analysis with both networks showed divergent behaviour. The divergence confirms that although the qualitative results showed the ANN learned the weight distribution as a function of the support nodes, the presence of low-order errors and contribution of first-order derivatives when calculating the Laplace operator makes the system divergent.

B. Analysis of Machine Learning-Based Linear System Solutions

Similarly to the approach described in Section III-A, but for the linear system surrogate, the LABFM moments were computed, and the absolute error, along with its standard deviation, was evaluated relative to the target moments. Table II summarizes the results for the predicted weights, $\boldsymbol{\Psi_i^d}$.

The first-order moments were all predicted with a mean error and standard deviation of $\mathcal{O}(10^{-4})$. The second-order moments

TABLE I
LABFM MOMENTS FOR THE SECOND-ORDER APPROXIMATION OF THE LAPLACE OPERATOR, COMPUTED USING WEIGHTED ABFS ($\boldsymbol{w_i^d}$) PREDICTED BY ANN AND PINN.

Model	**Corresponding Taylor Monomial**					
	Metric	x	y	$x^2/2!$	xy	$y^2/2!$
ANN	Mean Error	30.91	29.36	5.09×10^{-2}	8.98×10^{-2}	4.96×10^{-2}
	Error Std. Dev.	24.30	23.39	3.88×10^{-2}	6.73×10^{-2}	3.84×10^{-2}
PINN	Mean Error	2.26	2.33	5.58×10^{-3}	7.65×10^{-3}	5.30×10^{-3}
	Error Std. Dev.	2.06	2.22	4.62×10^{-3}	6.95×10^{-3}	4.48×10^{-3}

TABLE II
LABFM MOMENTS FOR THE SECOND-ORDER APPROXIMATION OF THE LAPLACE OPERATOR, COMPUTED USING WEIGHTS ($\boldsymbol{\Psi_i^d}$) PREDICTED BY ANN AND PINN.

Model	**Corresponding Taylor Monomial**					
	Metric	x	y	$x^2/2!$	xy	$y^2/2!$
ANN	Mean Error	3.00×10^{-4}	1.74×10^{-4}	1.45×10^{-4}	8.08×10^{-5}	1.14×10^{-4}
	Error Std. Dev.	2.41×10^{-4}	1.53×10^{-4}	7.48×10^{-5}	8.02×10^{-5}	8.96×10^{-5}
PINN	Mean Error	1.38×10^{-4}	1.99×10^{-4}	6.52×10^{-5}	4.11×10^{-5}	6.63×10^{-5}
	Error Std. Dev.	1.29×10^{-4}	1.62×10^{-4}	5.80×10^{-5}	5.73×10^{-5}	5.55×10^{-5}

achieved an accuracy of $\mathcal{O}(10^{-5})$ for the PINN and one of the second-order moments with the MLP, representing a significant improvement compared to the first-order moments predicted with the previous approach (shown in Table I).

Sequentially, the ANN and PINN were subjected to a convergence analysis, Fig. 3 shows the results obtained. However, this revealed that the level of accuracy obtained is insufficient, causing differential operator to diverge at a lower resolution and achieve a lower accuracy than LABFM theoretically allows.

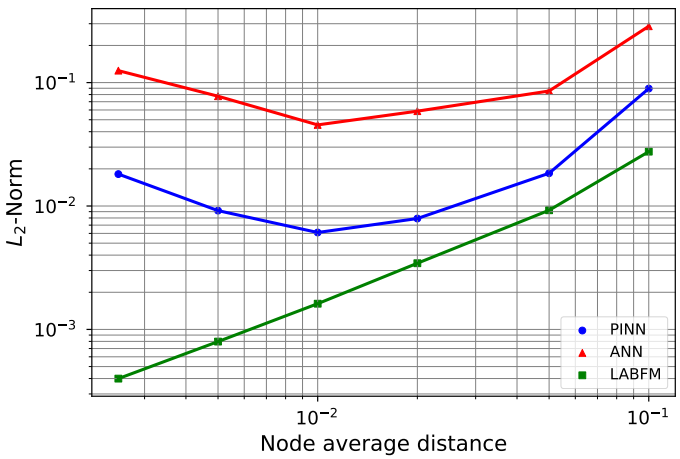

Fig. 3. Convergence plot of LABFM, ANN and PINN for a second-order approximation of the Laplace operator.

Furthermore, a computational cost analysis of ANNs and PINNs compared to LU factorisation was conducted. Both ML models took longer to perform predictions compared to LU factorisation to solve the linear system. These findings suggest that ANNs and PINNs to solve ill-conditioned dense linear systems are not advantageous due to the computational overhead and insufficient accuracy for the current state-of-the-art in mesh-free methods.

IV. CONCLUSION

This abstract presented two approaches to surrogate segments of high-order mesh-free methods. The aim was to compute high-order weights at a lower computational cost compared to current algorithms. Ultimately, these approaches seek to bridge the gap between Eulerian high-order mesh-free methods and Lagrangian mesh-free methods.

The results presented in this study reveal key limitations in the application of ML techniques, specifically MLPs and PINNs, to surrogate high-order mesh-free methods. The first approach generated qualitatively positive results, however a quantitative analysis indicated that the low-order LABFM moments caused the predicted weights to be divergent. In the second approach, the models demonstrated the ability to approximate weighted ABFs and solve dense, ill-conditioned, low-rank linear systems. However, their precision was insufficient to preserve the theoretical convergence properties of LABFM, and the computational cost of the ML-based methods exceeded that of LU factorisation.

These findings suggest that, despite the qualitative success in capturing weight distributions and solving linear systems, current ML approaches are not yet viable for replacing matrix inversion and the high-order kernel in high-order mesh-free methods.

REFERENCES

[1] S. J. Lind, B. D. Rogers, and P. K. Stansby, "Review of smoothed particle hydrodynamics:towards converged Lagrangian flow modelling," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, vol. 476, p. 2241, September 2020.

[2] J. King, S. Lind, and A. Nasar, "High order difference schemes using the Local Anisotropic Basis Function Method," *Journal of Computational Physics*, vol. 415, p. 109549, August 2020.