An improved finite difference method enhanced by deep neural networks

Wei Suo[1,2,3], **Weiwei Zhang**[1,2,3]
[1] School of Aeronautic, Northwestern Polytechnical University, Xi'an, 710072, China
[2] International Joint Institute of Artificial Intelligence on Fluid Mechanics, Northwestern Polytechnical University, Xi'an, 710072, China
[3] National Key Laboratory of Aircraft Configuration Design, Xi'an, 710072, China

Keywords: Neural networks, Numerical methods, Hybrid strategy, Partial differential equations

Abstract. Finite difference method (FDM) is a classic numerical method for solving partial differential equations (PDEs), while trade-offs between computational costs and accuracy are challenging. In recent years, neural networks (NNs) have attracted increasing interest in solving PDEs due to impressive nonlinear approximation capabilities. In this work, we propose an improved finite difference method enhanced by deep neural networks, called multi-scale neural computing (MSNC). The multi-scale concept in the MSNC refers to decomposing the solution of PDEs into two parts, i.e. the global scale solution and the local scale solution. The global scale solution is assumed to describe the general trends of the PDEs solution, which be can easily captured by a NN due to spectral bias. While the local scale solution is assumed to describe the local fluctuation details of the PDEs solution and solved by the FDM. Several tests are implemented to compare the MSNC with the standard FDM. Demonstrated advantages include higher accuracy and lower costs than the standard FDM. The MSNC also exhibits stable convergence, showcasing the potential for hybrid of NN and numerical method.

1. Introduction

The behavior of complex physical systems is described by partial differential equations (PDEs). Finite difference method (FDM) is a classic numerical method for PDEs, with applications ranging from aerodynamics, weather, astrophysics, and so on. However, high-fidelity simulations depend on fine-grained spatial and temporal discretization to resolve all phenomena of interest. It leads to inevitable trade-offs between computational costs and accuracy.

In recent years, neural networks (NNs) has attracted increasing interest in solving PDEs, leading to some popular works like Physics-informed neural networks (PINNs)[1], Fourier Neural Operator (FNO)[2] and Deep Operator Network (DeepONet)[3]. Despite the advantages in inverse problems, the current NNs-based PDEs solvers struggle to match the efficiency and accuracy of numerical methods in solving forward problems, as discussed in some literature[4]. Another approach is to hybrid numerical methods with neural networks, by utilizing the computational efficiency in numerical methods and nonlinear approximation capabilities of NNs[5].

In this work, we propose an improved finite difference method enhanced by deep neural networks (DNNs), called multi-scale neural computing (MSNC). The method is inspired by the spectral bias of NNs, which means a DNN in training first quickly captures the dominant low-frequency components and then relatively slowly captures the high-frequency ones[6]. The multi-scale concept in the MSNC refers to decomposing the solution of PDEs into two parts, i.e. the global scale solution and the local scale solution. The global scale solution is assumed to describe the general trends of the PDEs solution and be smooth, which be can easily captured by a NN due to spectral bias. While the local scale solution is assumed to describe the local fluctuation details of the PDEs solution and solved by the FDM, which can be more simply resolved than the whole solution. The motivation of the MSNC is to calculate the solutions of PDEs using an order-of-magnitude coarser grid than the standard FDM required for the same accuracy. Therefore, the MSNC is able to balance the computational costs and accuracy, making the solution of PDEs more efficient.

2. Methods

Consider a a general form of PDEs

$$L(u) = f$$
$$B(u) = g$$

where L denotes the differential operator, and B denotes the boundary conditions.

The overall procedure of the MSNC is illustrated in Figure 1. The solution process can be divided into two stages: offline stage and online stage. In the offline stage, the NN is trained based on historical data, which is assumed to be accumulated in other solved state. Leveraging the similarity among data, the obtained NNs can better approximate the solutions compared to random generation. In the online stage, the NN and FDM are combined to solve the PDE, with decomposing the unknown variable u into two parts, $u = \bar{u} + \hat{u}$, where denotes the solution at the global scale, predicted directly by the NN, and $\hat{u}$ denotes the solution at the local scale, solved by the FDM. According to the decomposition of u , the original PDE can be rewritten as $L(\bar{u}) + L(\hat{u}) = f$. For the calculation of derivatives at the global scale, ie. $L(\bar{u})$, automatic difference (AD) of the NN is used. Due to the NN is trained, the solution and derivatives related to $\bar{u}$ are all known and can be regraded as source term of the original PDE. Then the original PDE is transformed into a new PDE, ie. $L(\hat{u}) = f - L(\bar{u})$, with boundary conditions also changing into $B(\hat{u}) = g - B(\bar{u})$. Then the FDM is applied to solve the new PDE for the solution at the local scale . After discretization and solving the algebraic system of equations, solutions of global scale and local scale are composed to be the solution of the original PDE.

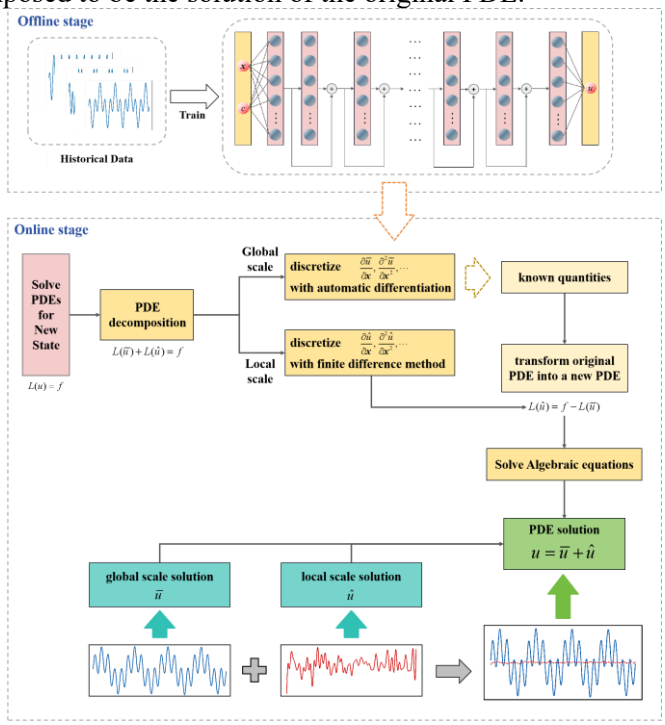

Figure 1. The overall procedure of the MSNC.

The NN in this paper is trained based on historical data to establish the mapping between the spatial coordinates x of the solution domain and the PDEs solution u. Simultaneously, to distinguish historical data for different state parameters and achieve generalization of state parameters, these state parameters c need to be included as inputs to the NN. In summary, the NN takes spatial coordinates x (the number of spatial coordinates depends on the dimensionality of the PDEs) and state parameters $\boldsymbol{c}$ as inputs, and outputs the solution u of the PDEs. Using the residual neural network as the basic model architecture, it improves upon the framework of fully connected neural networks. The basic building block is changed from a hidden layer to a residual block, as shown in Figure 2(a). The output of a residual block is the linear sum of the input and output of a hidden layer. Residual neural networks exhibit better training convergence efficiency compared to fully connected neural networks. Based on the residual neural

network, the NN model established in this paper is illustrated in Figure 2(b). The model takes the value of spatial coordinates and state parameters at a single point as inputs, and outputs the corresponding solution of the PDEs at that single point.

The loss function is set as the mean squared error (MSE) between the predicted values and the label values. In the actual training process of the NN model, data is read in batches, so the loss function can be expressed as

$$LOSS_{MSE}(u) = \frac{1}{N}\sum_{i=1}^{N}(u_{pred} - u_{ref})^2$$

where N denotes the size of a batch, u_{pred} denotes the the predicted values, and u_{ref} denotes the label values.

(a)

(b)

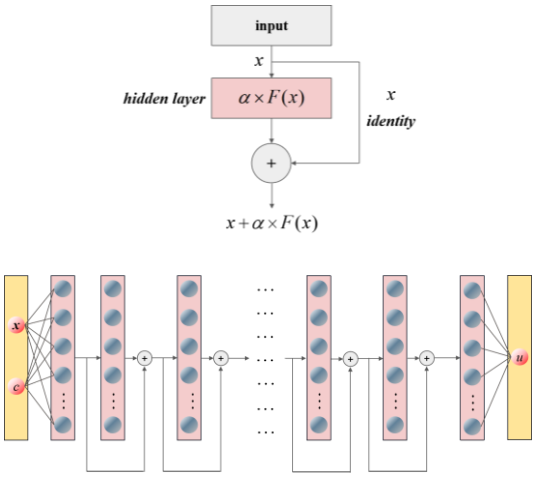

Figure 2. (a) Schematic diagram of a residual block. The output of a residual block is the linear sum of the input and output of a hidden layer. (b) Schematic diagram of the NN model established in this paper. The model takes the value of spatial coordinates x and state parameters $\boldsymbol{c}$ at a single point as inputs, and outputs the corresponding solution u of the PDEs at that single point.

3. Results and discussion

In this section, we demonstrate the capability of the MSNC to solve different PDEs. Various numerical examples are presented to showcase the performance advantages of the proposed MSNC compared to the standard FDM.

3.1. One-dimensional linear Helmholtz equation

Consider the one-dimensional linear Helmholtz equation of the form

$$\frac{\partial^2 u}{\partial x^2} - \lambda u = f(x), \quad x \in [a,b]$$
$$u(a) = g_1, \quad u(b) = g_2$$

The constant and the domain specification are $a = -4, \quad b = 4, \quad \lambda = 10$.

We choose the source term $f(x)$ such that the above equation has the following solution

$$u(x) = \sin(\alpha_1 \pi x)\cos(\alpha_2 \pi x)$$

where α_1 and α_2 are two state parameters used for sampling. We choose $\alpha_1 = 3$ and $\alpha_2 = 2$ as the test state for this example.

We introduce a uniform grid to discretize the 1D spatial domain $[a,b]$. The grid numbers is denoted by *gridNum*. In addition, the standard second-order central FDM is employed as a comparison to the MSNC. Figure 3 show the solving results, including comparison of solutions, point-wise error, and decomposition of the MSNC, of the one-dimensional linear Helmholtz equation with *gridNum* of 320.

We can observe that, the MSNC achieves higher accuracy compared to the standard FDM, with the magnitude of the point-wise error mainly between $O(10^{-4})$ and $O(10^{-2})$ in the standard FDM ,while mainly between $O(10^{-5})$ and $O(10^{-3})$ in the MSNC. Additionally, in the decomposition of the MSNC, we can see that the NN solution provides a global profile similar to the entire solution, while the FDM solution offers local refinement to capture more local details. This demonstrates the feasibility of the MSNC's starting point, where NNs are employed for efficient capture of global scale, while the FDM are utilized for detailed description of local scale.

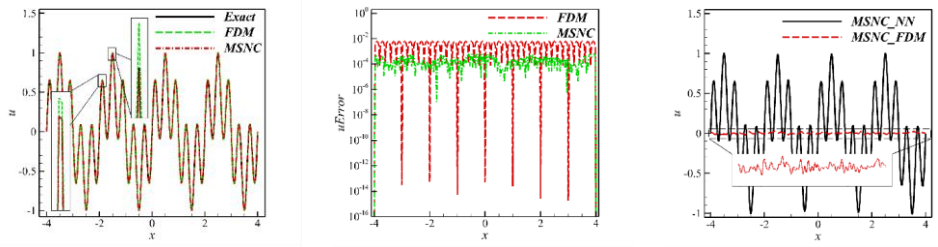

Figure 3. One-dimensional linear Helmholtz equation: comparison of solutions, point-wise error, and decomposition of the MSNC (*gridNum* = 320).

Table 1 shows the RMSE with different *gridNum* and convergence order. We can observe that, with an equivalent grid number, the MSNC consistently achieves higher solution accuracy than the standard FDM. We compute the ratio, defined as the RMSE of the standard FDM divided by the RMSE of the MSNC at same *gridNum*. We can see that, except for the minimum value of *gridNum*, the ratio is generally more than 10, demonstrating the significant improvement in accuracy achieved by the MSNC compared to the standard FDM. From another perspective, at the same level of accuracy, the MSNC requires approximately 4 times fewer grid numbers compared to the standard FDM, effectively reducing computational costs. Moreover, we can observe that, after *gridNum* exceeding a certain point, the MSNC exhibits an approximate 2-order convergence similar to the standard FDM.

Table 1 One-dimensional linear Helmholtz equation: RMSE with different gridNum and convergence order.

gridNum	RMSE FDM	Order	RMSE MSNC	Order	Ratio
80	7.821701e-02		8.889618e-03		8.80
160	1.792897e-02	2.18	1.188738e-03	3.74	15.08
320	4.392288e-03	2.04	2.717606e-04	3.74	16.16
640	1.092990e-03	2.01	6.652829e-05	3.74	16.43
1280	2.729848e-04	2.00	1.665229e-05	2.00	16.39
2560	6.823646e-05	2.00	4.432109e-06	1.88	15.40

3.2. Two-dimensional Poisson equation

Consider the two-dimensional Poisson equation of the form

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=f(x,y),\quad x,y\in[a_1,b_1]\times[a_2,b_2]$$

$$u(a_1,y)=g_1,\quad u(b_1,y)=g_2,\quad u(x,a_2)=g_3,\quad u(x,b_2)=g_4$$

The constant and the domain specification are $a_1=-1,\quad b_1=1,\quad a_2=-1,\quad b_2=1$.

We choose the source term $f(x)$ such that the above equation has the following solution

$$u(x,y) = (0.1\sin(\alpha_1 \pi x) + \tanh(10x)) \times \sin(\alpha_2 \pi y)$$

where α_1 and α_2 are two state parameters used for sampling. We choose $\alpha_1 = 2$ and $\alpha_2 = 2$ as the test state for this example.

We introduce a uniform grid to discretize the 2D spatial domain $[a_1, b_1] \times [a_2, b_2]$. The grid numbers of single dimension is denoted by *gridNum1D*, with the corresponding grid numbers of two-dimensional domain denoted by *gridNum2D*. In addition, the standard second-order central FDM is employed as a comparison to the MSNC.

Table 2 shows the RMSE with different *gridNum1D* and convergence order. We can see that the conclusions for one-dimensional linear Helmholtz equation can be extended to two-dimensional Poisson equation here. The MSNC achieves higher solution accuracy than the standard FDM. We compute the ratio, which is generally more than 50, with the maximum value being 71.09, demonstrating the significant improvement in accuracy achieved by the MSNC. From another perspective, to achieve same level of accuracy, the MSNC requires approximately 8 times fewer grid numbers in single dimension, meaning 64 times fewer grid numbers in two-dimensional spatial domain. Moreover, we can observe that, The MSNC exhibits an approximate 2-order convergence similar to the standard FDM after *gridNum1D* exceeding a certain point.

Table 2 Two-dimensional Poisson equation: RMSE with different gridNum1D and convergence order.

gridNum1D	RMSE FDM	Order	RMSE MSNC	Order	Ratio
40	5.138312e-03		7.242727e-05		70.94
80	1.286862e-03	2.00	1.810788e-05	2.00	71.07
160	3.228900e-04	1.99	4.541786e-06	1.99	71.09
320	8.092320e-05	2.00	1.169165e-06	1.94	69.21
640	2.025920e-05	2.00	3.641297e-07	1.61	55.64

3.3. Burgers' equation

Consider the Burgers' equation of the form

$$\frac{\partial u}{\partial t} + (bu - c)\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in [a,b]$$

$$u(a) = g_1, \quad u(b) = g_2$$

The constant and the domain specification are $a = 0, \quad b = 1, \quad c = 0.5, \quad \nu = 0.01$.

The above equation has the following solution

$$u(x) = \frac{c}{b}[1 - \tanh(\frac{c(x - x_0)}{2\nu})]$$

where we choose $x_0 = 0.5$ for this example. ν is the state parameter used for sampling.

We introduce a uniform grid to discretize the 1D spatial domain $[a,b]$. The grid numbers is denoted by gridNum. In addition, the standard second-order central FDM is employed as a comparison to the MSNC.

Table 3 shows the RMSE with different *gridNum* and convergence order. We can observe that, with an equivalent grid number, the MSNC consistently achieves higher solution accuracy than the standard FDM. We compute the ratio and can see that, except for the minimum value of *gridNum*, the ratio is generally more than 19, demonstrating the significant improvement in accuracy achieved by the MSNC compared to the standard FDM. From another perspective, at the same level of accuracy, the MSNC requires approximately 4 times fewer grid numbers compared to the standard FDM, effectively reducing

computational costs. Moreover, we can observe that, after *gridNum* exceeding a certain point, the MSNC exhibits an approximate 2-order convergence similar to the standard FDM.

Table 3 Burgers' equation: RMSE with different gridNum and convergence order.

gridNum	RMSE FDM	Order	RMSE MSNC	Order	Ratio
40	9.899922e-03		1.702047e-03		5.82
80	2.195426e-03	2.25	5.930346e-05	14.35	37.02
160	5.346864e-04	2.05	1.494199e-05	1.98	35.78
320	1.317274e-04	2.03	3.580371e-06	2.09	36.79
640	3.161834e-05	2.08	1.590218e-06	1.13	19.88

4. Conclusion

In this work, we propose a novel paradigm called MSNC for solving PDEs, with the consideration of the spectral bias of NNs and local approximation properties of FDM. Based on the idea of scale decomposition, NNs are employed for efficient capture of global scale, while numerical methods are utilized for detailed description of local scale. In addition, historical data is employed to train the NNs, and output layer weights of the trained NNs are released as free parameters in the new state of the PDEs. The advantages of the MSNC have been validated in various numerical examples. One is higher computational accuracy. Under an equivalent grid number, the MSNC exhibits higher solutions accuracy compared to the standard FDM. The other is lower computational cost. With equivalent accuracy, the MSNC significantly reduces the required grid number compared to the standard FDM.

5. References

1 Raissi, M., Perdikaris, P. & Karniadakis, G. E. 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).

2 Li, Z. *et al.* Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv preprint arXiv:2010.08895* (2020).

3 Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. Learning Nonlinear Operators Via Deeponet Based on the Universal Approximation Theorem of Operators. *Nature machine intelligence* **3**, 218-229 (2021).

4 Cai, S., Mao, Z., Wang, Z., Yin, M. & Karniadakis, G. E. Physics-Informed Neural Networks (Pinns) for Fluid Mechanics: A Review. *Acta Mechanica Sinica* **37**, 1727-1738 (2021).

5 Vinuesa, R. & Brunton, S. L. Enhancing Computational Fluid Dynamics with Machine Learning. *Nature Computational Science* **2**, 358-366 (2022).

6 Xu, Z.-Q. J., Zhang, Y. & Luo, T. Overview Frequency Principle/Spectral Bias in Deep Learning. *Communications on Applied Mathematics and Computation*, 1-38 (2024).