INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 19, Number 4, Pages 587–601

A MESH-LESS, RAY-BASED DEEP NEURAL NETWORK METHOD FOR THE HELMHOLTZ EQUATION WITH HIGH FREQUENCY

ANDY L YANG† AND FENG GU‡

Abstract. This article introduces a mesh-less, ray-based deep neural network method to solve the Helmholtz equation with high frequency. This method does not use an adaptive mesh refinement method, nor does it design a numerical scheme using some specially designed basis function to calculate the numerical solution, but it has the advantages of easy implementation and no mesh. We have carried out various numerical examples to prove the accuracy and efficiency of the proposed numerical method.

Key words. Deep learning, plane wave, deep Neural Network, loss, high frequency, Helmholtz equation.

1. Introduction

In mathematics, the eigenvalue problem of the Laplace operator is called the Helmholtz equation, which has many applications in physics, including the wave equation and diffusive equation. It also has applications in other scientific fields, including electromagnetic radiation [4], acoustics [2], and plasma [16], etc. When the Helmholtz equation is applied to a wave, the eigenvalue value is called the wavenumber. The most obvious feature of the Helmholtz equation is that it is not positive definite, which makes the solution of the equation have strong oscillations when the wavenumber is large. In numerical calculations, the high oscillatory property of the exact solution under high-frequency conditions will cause the approximate solution obtained by the numerical calculation to only have very low accuracy, which is called the "pollution effect", cf. [1]. Therefore, from the perspective of algorithm design, the highly oscillating nature of the solution makes it very challenging to obtain an effective numerical method for this equation, which is also the purpose of this article.

We recall that there exist many available numerical algorithms for the Helmholtz equation with various boundary conditions including, for instance, the finite element method (FEM), Discontinuous Galerkin method, Spectral method, hybridizable discontinuous Galerkin method, weak Galerkin method, etc., see [1, 7, 13, 24, 29, 31] and reference therein. Due to the high oscillating nature of the solution, some commonly-used numerical methods based on low-order polynomials cannot resolve the solution well. Instead, they will produce the so-called pollution effect, that is, for a fixed number of grid points for each wavelength, the numerical error increases with the increase of wavenumbers, see [1]. Therefore, while using the numerical method based on the low-order polynomials, unless a certain number of grid points are used for discretization for each wavelength, the calculation accuracy is relatively poor for high-frequency waves. Therefore, it is natural to use higher-order polynomials or oscillatory non-polynomial basis to replace the low-order polynomials. It has been shown that higher-order polynomials can effectively reduce the pollution effect, see

Received by the editors January 21, 2022 and, in revised form, April 23, 2022.
2000 *Mathematics Subject Classification.* 65L15, 34L16.

[24, 33], however, the computational cost is high due to the increase of degrees of freedom.

In this article, we try to break away from the traditional numerical methods based on variational framework using low/high-order polynomial basis, and use a novel mesh-less deep neural network (DNN) method to solve the high-frequency Helmholtz equation. We recall that the DNN method has attracted many attentions in recent years to many classic problems involved in scientific computing, especially the numerical solution of ordinary or partial differential equations, cf. [23, 5, 3, 8, 10, 12, 11, 18, 19, 21, 28, 30, 32] and references therein. Whether the algorithms of DNN can be applied to the field of scientific computing to obtain effective and accurate numerical algorithms has been confirmed by some recent research works. For example, in [11, 27], the authors give the quantitative relationship between neural network algorithms and low/high-order finite element methods; in [15], the authors discuss the approximate properties of the function classes given by the feedforward neural network using a single hidden layer; and in [17], the authors give the framework of deriving error estimates for a class of neural network algorithms according to the number of neurons. Based on these works, some novel methods on applying the DNN to solving ordinary/partial differential equations are developed, including the so-called PINN (the physics-informed neural network) method given in [21, 18], DGM (Deep-Galerkin method) give in [30] and DRM (Deep-Ritz method) given in [8]. Therefore, inspired by the DLSM (Deep-Least Squares Method) given in [5], in this article, we introduce a mesh-less, ray-based DNN method to solve the Helmholtz equation, and to investigate whether the method can be applied to high-frequency situations well. The mesh-free nature of this method allows us to easily get rid of designing adaptive grids or special spatial discretization methods, so it is very easy to implement. For the large wavenumber case, the obtained numerical results show that the designed DNN method can efficiently and accurately approximate the exact solution of the Helmholtz equations.

The rest of this paper is organized as follows. In Section 2, we review the basic idea of DNNs. In Section 3, the derivation of the methodology for the Helmholtz equation is developed. In Section 4, we present some numerical results to demonstrate the performance of our method. Some concluding remarks are given in Section 5.

2. DNN method

In this section, we briefly discuss the definition and approximation properties of the DNNs.

A DNN is a sequential alternative composition of linear functions and nonlinear activation functions. A n-layer neutral network $\mathcal{N}^n$ can be defined as

- Input layer: $\mathcal{N}^0 = \boldsymbol{x}$,
- Hidden layers: $\mathcal{N}^l = \sigma_l(\mathbf{W}^l\mathcal{N}^{l-1} + \mathbf{b}^l)$, $l = 1, 2, \cdots, n-1$,
- Output layer: $\mathcal{N}^n = \mathbf{W}^n\mathcal{N}^{n-1} + \mathbf{b}^n$,

where σ denotes the activation function, $\mathbf{W}^l$ denote the weights and $\mathbf{b}^l$ denote the biases. The most common used types of activation functions include the sigmoid function $\sigma(t) = (1 + e^{-t})^{-1}$ and the rectified linear unit (ReLU) $\sigma(t) = \max(0, t)$. For simplicity, we denote all the parameters in DNN by a parameter vector $\boldsymbol{\Theta}$, i.e.,

$$\boldsymbol{\Theta} = \{\mathbf{W^1}, \cdots, \mathbf{W^n}, \mathbf{b^1}, \mathbf{b^2}, \cdots, \mathbf{b^n}\}.$$

In Fig. 1, we sketch a simple fully connected DNN example with 3 hidden layers and 8 neurons in each hidden layer. The number m_l denotes the number of neurons in the l-th layer.