CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.SO-2021-0011

Vol. **2**, No. 4, pp. 748-775
December 2021

Enforcing Exact Boundary and Initial Conditions in the Deep Mixed Residual Method†

Liyao Lyu[1,2], Keke Wu[1], Rui Du[1,3,*] and Jingrun Chen[1,3,*]

[1] *School of Mathematical Sciences, Soochow University, Suzhou, 215006, China.*
[2] *Department of Computational Mathematics, Science, and Engineering, Michigan State University, East Lansing, MI, 48824, USA.*
[3] *Mathematical Center for Interdisciplinary Research, Soochow University, Suzhou, 215006, China.*

Received 2 February 2021; Accepted 9 June 2021

Abstract. Boundary and initial conditions are important for the wellposedness of partial differential equations (PDEs). Numerically, these conditions can be enforced exactly in classical numerical methods, such as finite difference method and finite element method. Recent years, we have witnessed growing interests in solving PDEs by deep neural networks (DNNs), especially in the high-dimensional case. However, in the generic situation, a careful literature review shows that boundary conditions cannot be enforced exactly for DNNs, which inevitably leads to a modeling error. In this work, based on the recently developed deep mixed residual method (MIM), we demonstrate how to make DNNs satisfy boundary and initial conditions automatically by using distance functions and explicit constructions. As a consequence, the loss function in MIM is free of the penalty term and does not have any modeling error. Using numerous examples, including Dirichlet, Neumann, mixed, Robin, and periodic boundary conditions for elliptic equations, and initial conditions for parabolic and hyperbolic equations, we show that enforcing exact boundary and initial conditions not only provides a better approximate solution but also facilitates the training process.

AMS subject classifications: 65M99, 65N99

Key words: Machine learning, deep neural networks, enforcement of boundary/initial conditions, penalty.

1 Introduction

Partial differential equation (PDE) is one of the most important tools to model various phenomena in science, engineering, and finance. It has been a long history of developing

†The first two authors contributed equally to the work.

*Corresponding author. *Email addresses:* lyuliyao@msu.edu (L. Lyu), wukekever@gmail.com (K. Wu), durui@suda.edu.cn (R. Du), jingrunchen@suda.edu.cn (J. Chen)

reliable and efficient numerical methods for solving PDEs. Notable examples include finite difference method [19], finite element method [28], and discontinuous Galerkin method [8]. For low-dimensional PDEs, these methods are proved to be accurate and demonstrated to be efficient. However, they run into the curse of dimensionality for high-dimensional PDEs, such as Schrödinger equation in the quantum many-body problem [9], Hamilton-Jacobi-Bellman equation in the stochastic optimal control [1], and nonlinear Black-Scholes equation for pricing financial derivatives [16].

In the last decade, significant advancements in deep learning have driven the development of solving PDEs in the framework of deep learning, especially in the high-dimensional case where deep neural networks overcome the curse of dimensionality by construction; see [2, 3, 6, 10–13, 17, 21, 22, 24, 26, 27] for examples and references therein. Among these, deep Ritz method uses the variational form (if exists) of the corresponding PDE as the loss function [11] and deep Galerkin method (DGM) uses the PDE residual in the least-squares senses as the loss function [26]. In [24], physics-informed neural networks are proposed to combine observed data with PDE models. The mixed residual method (MIM) first rewrites a PDE into a first-order system and then uses the system residual in the least-squares sense as the loss function [22]. These progresses demonstrate the strong representability of deep neural networks (DNNs) for solving PDEs.

In classical numerical methods, basis functions or discretization stencils have compact supports or sparse structures. Machine learning methods, instead, employ DNNs as trial functions, which are globally defined. This stark difference makes DNNs overcome the curse of dimensionality while classical numerical methods cannot. However, there are still unclear issues for DNNs, such as the dependence of approximation accuracy on the solution regularity and the enforcement of exact boundary conditions. It is straightforward to enforce exact boundary conditions in classical numerical methods while it is highly nontrivial for DNNs due to their global structures. A general strategy is to add a penalty term in the loss function which penalizes the discrepancy between a DNN evaluated on the boundary and the exact boundary condition. Such a strategy inevitably introduces a modeling error which leads to a degradation of the approximation accuracy and typically makes the training process more difficult [7]. Therefore, it is always desirable to construct DNNs which automatically satisfy boundary conditions and there are several efforts towards this objective [4, 23, 25]. It is shown that Dirichlet boundary condition can be enforced exactly over a complex domain in [4]. This idea cannot be applied for Neumann boundary condition since the solution value on the boundary is not available. This issue is solved by constructing the trial DNN in a different way [23]. However, for mixed boundary condition, this construction has a serious issue at the intersection of Dirichlet and Neumann boundary conditions and an approximation has to be applied. Therefore, it is so far that an exact enforcement of mixed boundary condition for DNNs has still been lacking.

In this work, in the framework of MIM, we demonstrate how to make DNNs satisfy boundary and initial conditions automatically using distance functions and explicit constructions. As a consequence, the loss function in MIM is free of penalty term and