2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) | 978-1-7281-1179-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/EMBC46164.2021.9630259

A Combination of Deep Neural Networks and Physics to Solve the Inverse Problem of Burger's Equation

Shaikhah Alkhadhr[1] and Mohamed Almekkawy[1]

***Abstract*— One of the most basic nonlinear Partial Differential Equations (PDEs) to model the effects of propagation and diffusion is Burger's equation. This puts great emphasize on seeking efficient versatile methods for finding a solution to the forward and inverse problems of this equation. The focus of this paper is to introduce a method for solving the inverse problem of Burger's equation using neural networks. With recent advances in the area of deep learning, a Physics-Informed Neural Network (PINN) is a category of neural networks that proved efficient for handling PDEs. In our work, the 1D and 2D Burger's equations are simulated by applying a PINN to a set of domain points. The training process of PINNs is governed by the PDE formula, the initial conditions (ICs), the Boundary Conditions (BCs), and the loss minimization algorithm. After training the network to predict the coefficients of the nonlinear PDE, the inverse problem of the 1D and 2D Burger's equations are solved with an error as low as 0.047 and 0.2 for 1D and 2D case studies, respectively. The wave propagation model is accomplished with an approximate training loss value of $1 \times e^{-4}$. The utilization of PINNs for modeling Burger's equation is a mesh-free approach that competes with the commonly used numerical methods as it overcomes the curse of dimensionality. Training the PINN model to predict the propagation and diffusion effects can also be generalized to address further detailed applications of Burger's equation with complex domains. This contributes to clinical applications such as ultrasound therapeutics.**

***Index Terms*— Neural Networks, Burger's Equation, Inverse Problem, Mathematical Modelling.**

I. INTRODUCTION

Mathematical modelling is the designated method used to meet the demand in simulating clinical applications in hopes of achieving efficient and accurate conforming of the physical reality in applied procedures or inferred diagnostics [6], [9]. The real physics of the phenomenon cannot often tolerate the assumptions implied by common analytic procedures as linearization can occur or they can relatively undermine system nonlinearities. Traditional methods tend to transform the equation to a simpler version, provide the solution in the form of a converging series, or proceed in a converging iterative scheme to reach the solution [5]. This explains the increasing importance of seeking a solution for nonlinear Partial Differential Equations (PDEs) that should align correctly with the physical phenomenon [1], [2], [3], [4]. Consequently to this need in this field, numerical solutions of PDEs have been a research highlight for the last couple of decades. Conventionally, PDEs are solved via numerical methods like finite difference methods (FDM) or finite element methods [10], [11], [20]. These methods can suffer from the curse of dimensionality [7].

Burger's equation is a non-linear diffusive PDE seen very often in applied science fields. It is frequently used to describe a system of propagation and diffusion effects. The solution to the forward and inverse problems of the Burger's equation have been a significant contribution in research, and it remains in study due to the beneficial insights it presents in physical and clinical applications [16]. The inverse problem in precise is naturally ill-posed (small changes in data causes major changes in the solution). In ultrasound therapeutics like the noninvasive HIFU procedures [17], [18], [19], the propagation and diffusion of ultrasound waves travelling in different layers of the human body needs to be closely observed and studied in order to apply therapy or induce diagnostics.

Previous literature include the proposals of a numerical solution by Gamzaev et al. [12] which presents an implicit difference scheme algorithm that avoids iteration of the solution aiming to restore the boundary behaviour and the source term. There is also the approach proposed by Raissi et al. [8] which is a solution for the 1D inverse problem of Burger's equation using PINN that encourages the extension of this approach to the 2D case. Gosse et al. [13] proposed filtered gradient method that focuses on solving the inverse problem of the 1D Burger's equation with an iterative strategy. Most of these traditional methods handle a single dimensionality at a time. In addition to this restriction, their grid-based origin drives them to suffer from the curse of dimensionality.

Physics-Informed Neural Networks (PINNs) are presented as a novel contribution in that field. PINNs are designed for solving PDEs through sufficient training of the deep learning network. The concept of PINNs relies on building Feed-forward Neural Network (FNN) to take in spacial and temporal variables as an input, applying a chain of linear and nonlinear transformations according to the weights and biases, and then propagate the results through these layers via an activation function to produce a prediction in the output layer. The proposed method of solving the inverse problem of Burger's equation using PINN is mesh-free,therefore overcoming the weakness existing in the conventional grid-based methods. It also performs with rivaling accuracy due to training and prediction processes [7].

*This work was not supported by any organization

[1]Shaikhah Alkhadhr and Mohamed Almekkawy are with the School of Electrical Engineering and Computer Science, Pennsylvania State University, University Park, PA 16802, USA sua391@psu.edu

II. Background

A. Burger's Equation

Due to the major improvements in computational capability, there has been an increased interest in studying the physical and mathematical significance of Burger's equation. It is one of the most common hyperbolic problems and nonlinear PDEs to describe the propagation and diffusion effects of a wave in 1D or 2D in formulas shown in (1) and (2) respectively:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - (\nu)\frac{\partial^2 u}{\partial x^2} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + u(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}) - (\nu)(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = 0 \tag{2}$$

where velocity, space, time and kinematic viscosity are represented by u, x, t and ν, respectively. Remaining still is the challenges for the accuracy and convergence of various numerical solutions. The engineering difficulty of acquiring a solution comes from the nonlinear term $u\frac{\partial u}{\partial x}$ and from the diffusive term $(\nu)\frac{\partial^2 u}{\partial x^2}$ in the original equation (1). This is similar but with an expanded number of dimensions in (2). Burger's equation is famous for describing many physical phenomena like shock wave theory, longitudinal elastic waves in isotropic solids, and nonlinear wave propagation [5], [14], [15].

B. Physics-Informed Neural Networks (PINNs)

Solving this problem requires a brief overview of Deep Neural Networks (DNNs) and the method of Automatic Differentiation (AD). Out of the many different types of Neural Networks (NN) such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in our approach, we selected the Feed-forward Neural Networks (FNNs) as it is appropriate for handling the majority of PDE problems.

$$\mathcal{N}^L(x) : \mathcal{R}^{d_{input}} \rightarrow \mathcal{R}^{d_{output}} \tag{3}$$

The notation in (3) represents an NN composed of L-layers, $L-1$-hidden layers with $\mathcal{N}_l$, $\mathcal{W}_l$, and b_l that represent neurons, weights, and biases, respectively in every layer l. Fig. 1 shows a simplified diagram of a PINN. A series of nonlinear and linear transformations are applied to the input of the FNN to produce the output in the final layer. The training and testing losses are optimized through minimizing the error by incorporating known minimization methods like L-BFGS-B and Adam [7]. The DNN uses AD to construct the equation and gradually predict its coefficients. This is done by handling Burger's equation as a compositional function. The output error of PINN is maintained by formulating the PDE, the Initial conditions (ICs), and Boundary Conditions (BCs). The number of training points, the distribution of data samples, the number of training epochs, and the PINN architecture (layers and neurons) are highly dependent on the explored problem. The PINN output for the case of the Burger's equation inverse problem is the predicted coefficient values of the PDE.

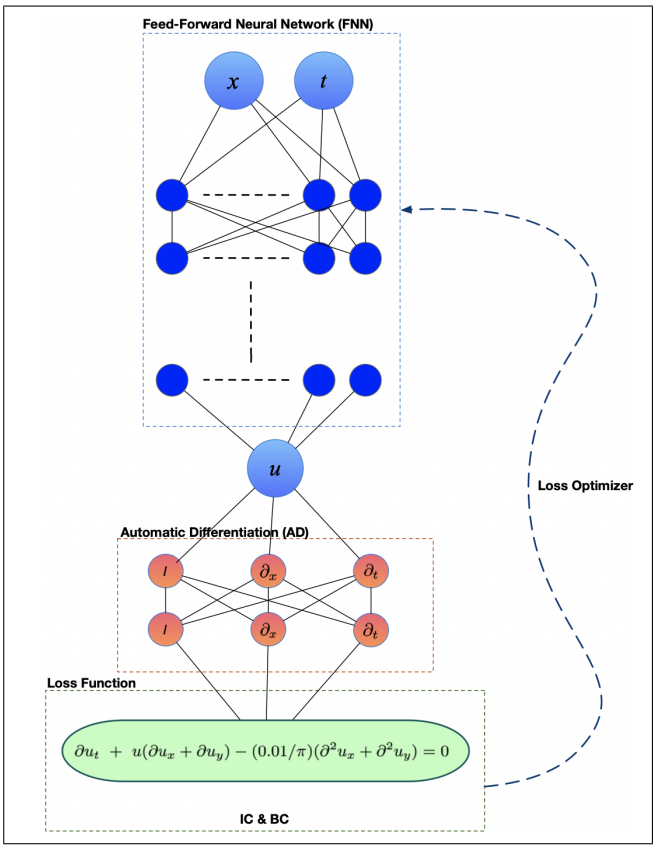

Fig. 1. A diagram of the PINN algorithm used for predicting the domain behaviour according to Burger's Equation. The upper neural network represents the solution $u(x,t)$ of the PDE.

III. Methodology

To model this problem, DeepXDE [7] is utilized. DeepXDE is a python library for implementing PINNs. Using this library, the forward problem can be implemented taking into consideration ICs, BCs, and the formula of the PDE. Fig. 2 and Fig. 3 show the construction of training data in the forward problem in both 1D and 2D along a timeline of multiple snapshots. In turn, the inverse problem is solved with minor changes and with the addition of the training data points induced from the forward problem.

Burger's forward problem is solved as an initial step to create the training data for the inverse problem. A random selection of the solution points from the forward solution is then exported as training data to plug into the inverse problem. Performing this procedure provides the advantage of defining the geometry of study, the number of training points, and other variables needed to prepare the training set for the inverse problem. To formulate the problem, we define the coefficients of the equation as $\lambda = [\lambda_1, \lambda_2]$ as shown in (4) and ((5)). Then, the PINN architecture is implemented in DeepXDE with the previously specified ICs and BCs. The PINN is then trained with the data acquired from the forward problem. For the exact solution coefficients, the values of λ_1 and λ_2 are set to 1 and $0.01/\pi$, respectively. These are also the same coefficient values used for the forward problem.

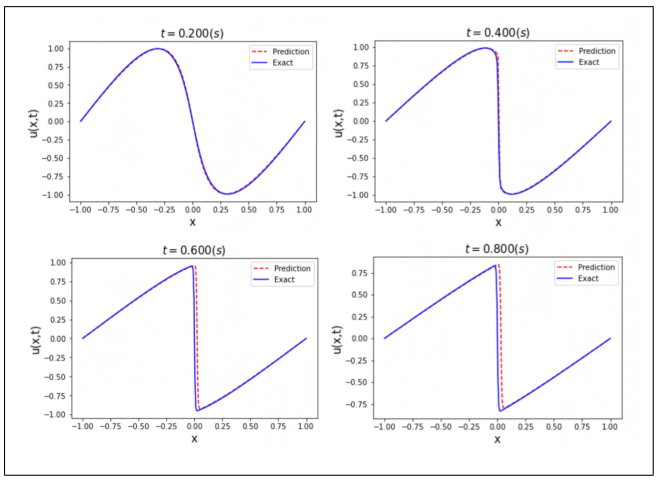

Fig. 2. Solution of the forward nonlinear wave propagation. The predicted solution of the 1D Burger's equation forward problem is used as a training data pool for the 1D inverse problem.

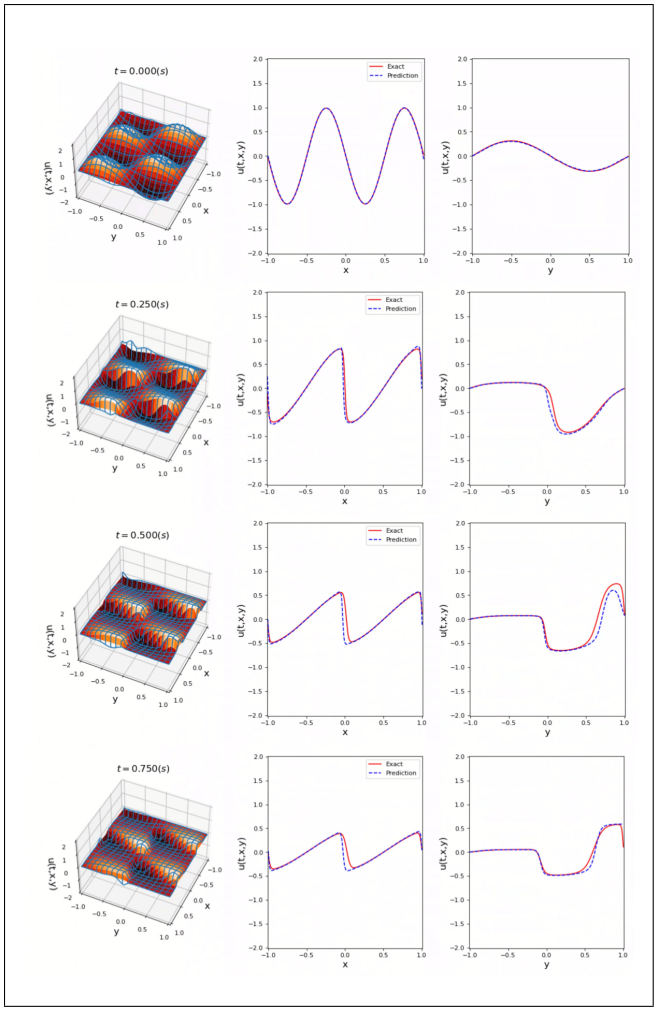

Fig. 3. Solution of the forward nonlinear wave propagation. The predicted solution is used as a training data pool for the 2D inverse problem.

$$1D: \quad \frac{\partial u}{\partial t} + (\lambda_1)\, u\frac{\partial u}{\partial x} - (\lambda_2)\frac{\partial^2 u}{\partial x^2} = 0 \tag{4}$$

$$2D: \quad \frac{\partial u}{\partial t} + (\lambda_1)\, u\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) - (\lambda_2)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{5}$$

IV. Results and Discussion

The results discussed here is the best of a series of performed PINN executions using different number of PINN architectures, loss weights, and training epochs. For both the 1D and 2D inverse problems setups, the PINN activation function is fixed to be a $tanh$ function.

A. 1D Burger's Equation

For the setup of the 1D Burger's equation model, the IC is as in (6) and we set Dirichlet BC as in (7) in a line geometry domain that spans the space [-1,1] and time [0,1].

$$f_{IC}(x,0) = -sin(\pi x) \tag{6}$$

$$f_{BC}(-1,t) = f_{BC}(1,t) = -sin(\pi x) \tag{7}$$

The PINN Architecture is built of 7 layers with 50 neurons in each layer, and it trains through 40,000 epochs. The learning rate is kept constant to $1e^{-3}$. The number of training points utilized from the forward problem is 10,000 randomly selected from the pool of the forward predicted solution. The training points are assigned a higher loss weight than the PDE, IC, and BC loss weights even though they all contribute to the training accuracy. This allows the DeepXDE framework to emphasize learning from the data points. The low training and testing loss values measuring at $8.3e^{-04}$ and $3.5e^{-04}$, respectively asserted the prediction of the λ's in the PDE. The predicted and the exact coefficients are presented in Table. I. The first row shows the exact 1D Burger's equation, and the second row shows the predicted coefficients. The last two rows are the computed error values for the first coefficient (λ_1) and the second coefficient (λ_2).

TABLE I
PREDICTED SOLUTION OF THE 1D BURGER'S INVERSE PROBLEM

Exact PDE	$u_t + 1.000uu_x + 0.003183u_{xx} = 0$
Predicted PDE	$u_t + 0.999uu_x + 0.003336u_{xx} = 0$
λ_1 error	0.001347
λ_2 error	0.047933

Assigning the appropriate weights for learning the solution points, the domain geometry, the initial state, and boundary of the defined geometry made the PINN architecture declared above capable of predicting an output λ_1 that is almost equal to the exact solution with an error in the order of $1e^{-3}$ and a λ_2 error as low as 0.047 which exceeds the accuracy of a previously recorded value in the literature [8].

B. 2D Burger's Equation

In the 2D Burger's equation, the IC and Dirichlet BC are set as in (8) and (9), respectively. The simulation is implemented to be in a rectangle geometry domain that spans the space x [-1,1] , y [-1,1] and time [0,1].

$$f_{IC}(x, y, 0) = sin(2\pi x)sin(\pi y) \tag{8}$$

$$\begin{aligned} f_{BC}(-1, y, t) &= f_{BC}(x, -1, t) \\ &= f_{BC}(x, 1, t) \\ &= f_{BC}(1, y, t) \\ &= sin(2\pi x)sin(\pi y) \end{aligned} \tag{9}$$

The PINN architecture metrics are slightly increased here to offer improved accuracy as the studied domain is extended from a line geometry to a rectangular domain. The PINN Architecture is composed of 7 layers with 55 neurons in each layer, and 17,000 training epochs. The learning rate for the 2D domain is fixed to $1e^{-3}$. The number of training points is 10,000 randomly sampled points created by the previously solving the forward problem. This PINN performed at a training loss of $7.5e^{-2}$ and a testing loss of $3.3e^{-2}$. The convergence rate for this problem is not as high as the 1D case. This implies the need for a larger PINN and greater number of epochs that was not performed due to limited available computational resources. Table. II shows the solution of the 2D Burger's inverse problem for setup values explained in this section. The first row shows the exact 2D Burger's equation, and the second row shows the predicted PDE coefficients. The prediction shows a λ_1 error of 0.08 with the given PINN architecture. The λ_2 error in 2D Burger's equation requires tuning the architecture and loss weights further to achieve an improved accuracy such as in the 1D case. However, for a modest setup close to the setup metrics of the 1D case, the resulting error by itself is motive for optimizing with deeper PINN architectures and a further tested weight loss distribution.

TABLE II

PREDICTED SOLUTION OF THE 2D BURGER'S INVERSE PROBLEM

Exact PDE	$u_t + 1.000u(u_x + u_y) + 0.003183(u_{xx} + u_{yy}) = 0$
Predicted PDE	$u_t + 0.919u(u_x + u_y) + 0.003867(u_{xx} + u_{yy}) = 0$
λ_1 error	0.081209
λ_2 error	0.214896

V. CONCLUSIONS

Burger's equation is the nonlinear wave propagation PDE model attempted in this paper. The reconstruction of the PDE was made possible from a provided set of data points along with specified BCs and ICs fed into a PINN algorithm to predict the closest values of coefficients to the ones in the exact PDE. This makes this approach open for continuous error optimization. Nevertheless, the proposed approach for solving the inverse problem of Burger's equation is flexibly applicable for more complex geometrical domains contributing significantly for clinical applications such as improved image reconstruction in ultrasound therapeutics. Further research to this proposed solution of the inverse problem can adopt a better inclusion scheme for the randomly sampled points to the training set of the PINN. This feature is referenced in [7] as Residual-based Adaptive Refinement (RAR). RAR implementation includes sampling points in training according to the mean residual error instead of the Randomness in point selection. This offers an improved technique in the DeepXDE framework.

REFERENCES

[1] C. Basdevant et al., "Spectral and finite difference solutions of the Burgers equation," Comput. Fluids, vol. 14, no. 1, pp. 23–41, 1986.

[2] J. H. He, "Application of homotopy perturbation method to nonlinear wave equations," Chaos, Solitons and Fractals, vol. 26, no. 3, pp. 695–700, 2005.

[3] M. Moghimi and F. S. A. Hejazi, "Variational iteration method for solving generalized Burger-Fisher and Burger equations," Chaos, Solitons and Fractals, vol. 33, no. 5, pp. 1756–1761, 2007.

[4] D. Qiu, Y. Zhang, and J. He, "The rogue wave solutions of a new (2+1)-dimensional equation," Commun. Nonlinear Sci. Numer. Simul., vol. 30, no. 1–3, pp. 307–315, 2016.

[5] M. P. Bonkile, A. Awasthi, C. Lakshmi, V. Mukundan, and V. S. Aswin, "A systematic literature review of Burgers' equation with recent advances," Pramana - J. Phys., vol. 90, no. 6, pp. 1–21, 2018.

[6] M. K. Almekkaway, I. A. Shehata, and E. S. Ebbini, "Anatomical-based model for simulation of HIFU-induced lesions in atherosclerotic plaques," Int. J. Hyperth., vol. 31, no. 4, pp. 433–442, 2015.

[7] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," SIAM Rev., vol. 63, no. 1, pp. 208–228, 2021.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys., vol. 378, pp. 686–707, 2019.

[9] M. K. I. Almekkawy, "Optimization of Focused Ultrasound and Image Based Modeling in Image Guided Interventions," 2014.

[10] I. A. Hassanien, A. A. Salama, and H. A. Hosham, "Fourth-order finite difference method for solving Burgers' equation," Appl. Math. Comput., vol. 170, no. 2, pp. 781–800, 2005.

[11] T. Öziş, E. N. Aksan, and A. Özdeş, "A finite element approach for solution of Burgers' equation," Appl. Math. Comput., vol. 139, no. 2–3, pp. 417–428, 2003.

[12] K. M. Gamzaev, "Numerical Solution of Combined Inverse Problem for Generalized Burgers Equation," J. Math. Sci. (United States), vol. 221, no. 6, pp. 833–839, 2017.

[13] L. Gosse and E. Zuazua, "Filtered gradient algorithms for inverse design problems of one-dimensional burgers equation," Springer INdAM Ser., vol. 16, pp. 197–227, 2017.

[14] J. I. Ramos, "Shock waves of viscoelastic Burgers equations," Int. J. Eng. Sci., vol. 149, p. 103226, 2020.

[15] E. R. Benton and G. W. Platzman, "A table of solutions of the one-dimensional Burgers equation," Q. Appl. Math., vol. 30, no. 2, pp. 195–212, 1972.

[16] M. Almekkawy et al., "Therapeutic Systems and Technologies: State-of-the-Art Applications, Opportunities, and Challenges," IEEE Rev. Biomed. Eng., vol. 13, pp. 325–339, 2020.

[17] X. Liu and M. Almekkawy, "An Optimized Control Approach for HIFU Tissue Ablation Using PDE Constrained Optimization Method," IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 68, no. 5, pp. 1555–1568, 2021.

[18] D. McMahon and M. Almekkawy, "Elements Selection for Transcostal HIFU Refocusing Method: Simulation Study," IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 67, no. 7, pp. 1366–1376, 2020.

[19] I. A. S. Elhelf, H. Albahar, U. Shah, A. Oto, E. Cressman, and M. Almekkawy, "High intensity focused ultrasound: The fundamentals, clinical applications and research trends," Diagn. Interv. Imaging, vol. 99, no. 6, pp. 349–359, 2018.

[20] M. Almekkawy, D. McMahon, H. Alqarni, and J. He, "Optimization of transcostal phased-Array refocusing using iterative sparse semidefinite relaxation method," 2017 IEEE Signal Process. Med. Biol. Symp. SPMB 2017 - Proc., vol. 2018-Janua, pp. 1–3, 2018.