Discontinuity Computing with Physics-Informed Neural Network

Li Liu[1,*],Shengping Liu[1], Heng Yong[1], Fansheng Xiong[1],Tengchao Yu[1]

[1] *Institute of Applied Physics and Computational Mathematics, Beijing 100094, China*

Abstract

How to simulate shock waves and other discontinuities is a long history topic. As a new and booming method, the physics-informed neural network (PINN) is still weak in calculating shock waves than traditional shock-capturing methods. In this paper, we propose a 'retreat in order to advance' way to improve the shock capturing ability of PINN by using a weighted equations (WE) method with PINN. The primary strategy of the method is to weaken the expression of the network in high compressible regions by adding a local positive and compression-dependent weight into governing equations at each interior point. With this strategy, the network will focus on training smooth parts of the solutions. Then automatically affected by the compressible property near shock waves, a sharp discontinuity appears with the wrong inside-shock-points 'compressed' into well-trained smooth regions just like passive particles. In this paper, we study one-dimensional and two-dimensional Euler equations. And illustrated by the comparisons with high-order classical WENO-Z method in numerical examples, the proposed method can significantly improve the discontinuity computing ability.

Keywords: PINN, Shock Capturing, Compressible flow, Euler equations, Discontinuity calculating

1. Introduction

Calculating shock waves and other discontinuities sharply and without oscillations is a core topic in solving hyperbolic equations. The first work of

shock-capturing can be dated back to von Neumann and Richtmeyer [1], who introduced an artificial viscosity into a staggered Lagrangian scheme to solve compressible flows. Nowadays, there are variables advanced and high-order methods can simulate problems with shock waves, such as the essential non-oscillatory (ENO) methods [2], the weighted essentially non-oscillatory (WENO) methods [3], and the discontinuous Galerkin method [4], etc. We refer readers to the following literature [5, 6] for more information about the development of shock-capturing methods.

Correspondingly to the above classical methods, due to the rapid development of neural networks and other machine learning methods, an alternative way has emerged that using machine learning methods to solve partial differential equations (PDEs), such as [7, 8, 9, 10, 11]. Among those methods, the physics-informed neural network (PINN) method attracts much attention. PINN is a neural network (NN) that encodes PDEs or other model equations as a component of the neural network itself. Due to its generality, PINN has been used in solving different kinds of equations in many fileds [12]. Specifically, in the study of hyperbolic equations, Patel et al. [13] propose a PINN for discovering thermodynamically consistent equations that ensure hyperbolicity for inverse problems in shock hydrodynamics. Mao et al. [14] study 1D and 2D Euler equations with shock waves, and they use appropriate clustering of training data points around a high gradient area to improve solution accuracy in that area and reduce error propagation to the entire domain. Jagtap et al [15] propose cPINN that splits the computing domain into several small subdomains with different NNs to solve Burgers equation and Euler equations. Jagtap et al. [16] especially study inverse problems in supersonic flows. The above literature shows that PINN is really effective in handle inverse problems with prior information about the development of the flow structures, such as the density gradient. However, in the study of forward problems, the original PINN only performed well in simple problems such as tracking moving shock waves. Borrowing the strategy from traditional methods, Patel et al. [13] construct a mesh-based control volume PINN and introduce an entropy condition and TVD (total variation diminishing) condition

into the scheme. Papados [17] simulates the shock-tube problem with the computing domain extended and gains an outstanding result without introducing non-physics viscosity into equations.

This paper aims to enhance the shock-capturing ability of PINN, especially for those complex problems with shock generations. The idea of this work is based on the following two facts and a rough and qualitative analysis. **Fact 1**: the shock wave has no thickness in theory without considering viscosity. So it can not be governed by the differential equations for its infinite gradient but can be controlled by a physics compression process from the left and right statuses (Rankine–Hugoniot conditions). **Fact 2**: NNs are also bad at approximating first-order discontinuous functions(the universal approximation theorem is only effective with continuous functions). Then the training of NNs tends to reduce the gradient in a steep jumping region. So if points (collections) 'wrongly' fall inside a shock region, large equation losses exist in those collections for their huge gradients, in this paper we call them 'trouble points'. The NNs may focus on handling those trouble points as they always carry most of the losses. However, NNs can not increase the gradient to decrease the thickness of the shock because of the above two facts. While the physics process attends to compress the region with the time development. As a result, trouble points fall into a paradoxical status that increasing the gradient will seriously increase the total loss as they are not governed by the equations, however, decreasing the gradient will also increase the total loss as it conflicts with the physical compression process. More seriously, the total loss is the sum or average of all the collections' losses. So trouble points will attract the training and influence the convergence of other collections in smooth regions. Here we can compare the above process with a classical high-order method, such as the finite volume (FV) WENO scheme, to illustrate the difficulty of PINN. In a WENO scheme, when a cell is inside a discontinuous region, the order of the scheme will automatically reduce to no more than 2nd, thus a large dissipation is appended into the cell to get a non-oscillatory result. We can see the main difference with PINN is that the given cell with large dissipation and consequently large error will not

influence the accuracy order of other cells further from the discontinuity.

Then we introduce a 'retreat in order to advance' strategy into PINN. In order to break the paradoxical status of trouble points and get a sharp discontinuity, the main idea is that we let the NN in PINN abandon to train the shock waves and focus on training other smooth regions by weakening the expression of it in strong compression regions. Then automatically effected by the compressing property near shock waves, a sharp and exact shock appears. This strategy is done by multiplying a local positive and compression-related weight to the governing equations to adjust the expressions of NN in different collocations. We show the details of weighted-equations (WE) method in Section 3. Before that, in Section 2, we introduce the equations and notations studied in this paper. Then in Section 4, several 1D and 2D forward examples are considered to show the effect of the new method. At the end of the paper, conclusions and more discussions are given in Section 5.

2. Governing Equations

In this paper, we consider the 1D and 2D Euler equations in conservative forms

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{1}$$

where for 1D case

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}, \tag{2}$$

and for 2D cases $F = (F_1, F_2)$, where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \mathbf{F_1} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \mathbf{F_2} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}. \tag{3}$$

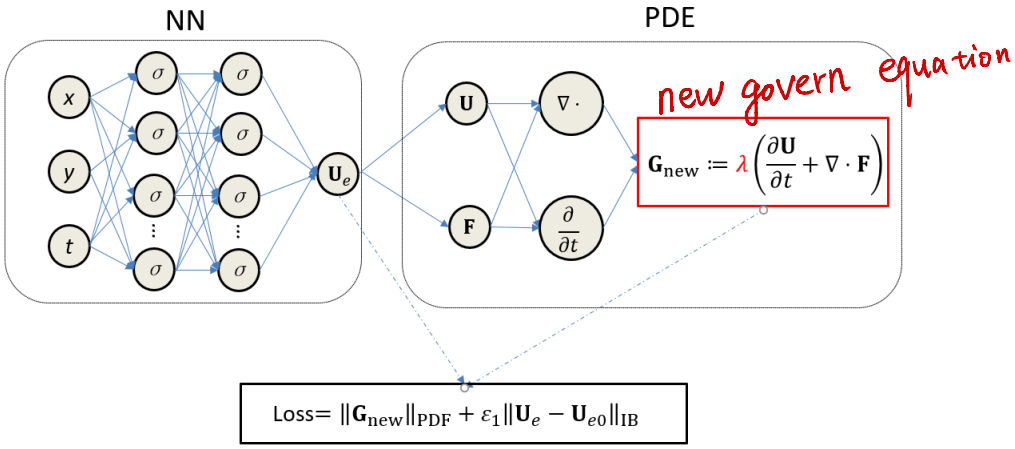


Figure 1: PINN-WE building blocks for conservative hyperbolic equations.

Here ρ is the density, u is the velocity and (u, v) is the velocity in 2D, p is the pressure, and E is the total energy. In order to close the equations, we use the equation of state (EOS) for ideal gas

$$E = \frac{1}{2}\rho u^2 + \frac{p}{\gamma - 1}, \tag{4}$$

where $\gamma = 1.4$ is radio of specific heat.

3. Weighted equations method for PINN

Here we first briefly introduce PINN for solving Euler equations. More details can ref to[14, 17]. The basic architecture of PINN mainly contains two parts, one is the uniformed NN associated with the variables $\mathbf{U}_e = (\rho, u, p)$ for 1D or $\mathbf{U}_e = (\rho, u, v, p)$ for 2D. While the other part is the informed one with the equations. The process of $\partial/\partial t$ and $\nabla\cdot$ is got by the automatic-derivative process.

The loss function used to train the model contains at least two parts to definite the problem. One is controlled by the equations and another is given by the initial and boundary conditions (also all can be seen as boundary conditions

if we do not distinguish temporal and spatial dimensions) of the problem:

$$\text{Loss} = ||\mathbf{G}||^2 + \varepsilon_1 ||\mathbf{U}_e - \mathbf{U}_{e0}||^2_{\text{IB}}, \tag{5}$$

where $\mathbf{G} := \partial_t \mathbf{U} + \nabla \cdot \mathbf{F}$ and $\mathbf{G} = 0$ is the governing equations. $\mathbf{U}_{e0}$ is the given initial and boundary conditions.

Many studies suggest a factor ε_1 to adjust the effects of different parts of losses based on the fact that points in the interior and on boundaries as not in the same quantities or significance. So a popular treatment is giving more weight to the points on boundaries.

However, inside each separate part of loss functions, studies always take an average of the effects of every point to the total loss. It may be suitable and without much significance to the problems with just smooth solutions. But when a discontinuity appears, as we discussed in the introduction, the gradient will become theoretically infinity and can not be described directly by the differential equations. So trouble points inside the discontinuity may introduce large errors and function losses.

By this basic analysis, we think taking an average of all the points' losses may not be a good idea in treating shock waves. Subsequently, we weaken the effect of points in high compressible regions by giving a local positive weight λ into the governing equations.

For a general conservative equation Eq.1

$$\mathbf{G}_{\text{new}} := \lambda(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}). \tag{6}$$

We name $\mathbf{G}_{\text{new}} = 0$ as **weighted equations (WE)**. Then obviously $\mathbf{G}_{\text{new}} = 0$ has the same solutions as $G = 0$ if λ is always positive. And we can adjust the NN's expression in different collocations by the design of a **gradient-dependent weight** λ. Correspondingly, the new loss is defined as

$$\text{Loss} = ||\mathbf{G}_{\text{new}}||^2 + \varepsilon_1 ||\mathbf{U}_e - \mathbf{U}_{e0}||^2_{\text{IB}}. \tag{7}$$

The structure of PINN-WE for conservative hyperbolic equations is shown in

Fig 1.

Specially, in this paper, we construct the gradient-dependent weight λ as

$$\lambda = \frac{1}{\varepsilon_2(|\nabla \cdot \vec{u}| - \nabla \cdot \vec{u}) + 1} \tag{8}$$

In work[1], an artificial viscosity is added to the function according to the magnitude of $\nabla \cdot u$ based on the fact that the flow field is compressed when $\nabla \cdot u < 0$. But here we only use it to detect shocks and active λ in compressing regions but do not add any dissipation into the equations.

We will show the effect in the following numerical examples.

4. Numerical examples

In this section, we use PINN with weighted Euler equations $\mathbf{G}_{\text{new}} = 0$. We study the classical 1D Sod problem and Lax problem. Then we consider two 2D problems with discontinuities. We compare all the results with a classical high-order finite differential scheme (5th-order WENO-Z method in spatial and 3rd-order Runge-Kutta method in temporal).

4.1. Sod problem

This problem has been extensively studied in the literature, and is one-dimensional Riemann problem with the initial constant states in a tube with unit length as

$$(\rho, u, p) = \begin{cases} (1, 0, 1), & if \quad 0 \le x \le 0.5, \\ (0.1, 0, 0.125), & if \quad 0.5 < x \le 1. \end{cases} \tag{9}$$

In this case, we use the NN with 7 hidden layers with 50 neurons in each layer. Random take 10000 collocations from a uniform mesh 100×200 in $X \times T$ space. And the number of boundary points is 1000. After the training. We choose a test set with 100 uniform points in $x \in [0, 1]$ at the end time $t = 0.2$.

We first compare the results with the classical high-order WENO-Z method with 100 cells in spatial. Showed in Fig.1, PINN with weighted equations (PINN-WE) can achieve similar results to the WENO-Z method, and are even better

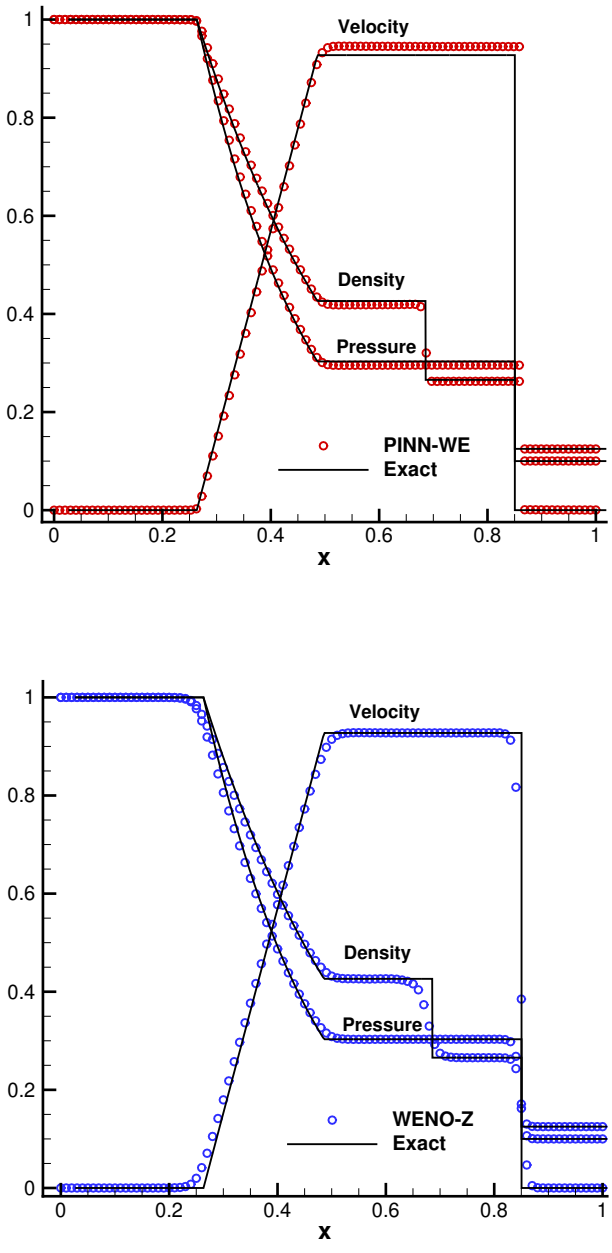


Figure 2: Results for Sod problem, comparing with the classical high-order WENO-Z method.

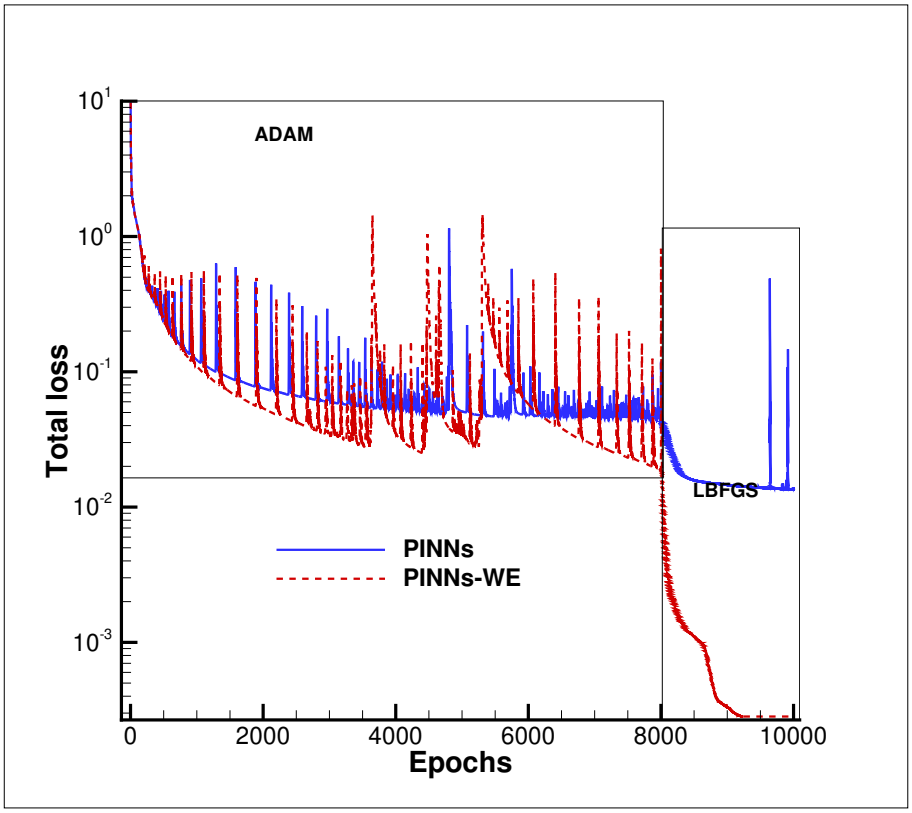


Figure 3: Loss histories for Sod problem

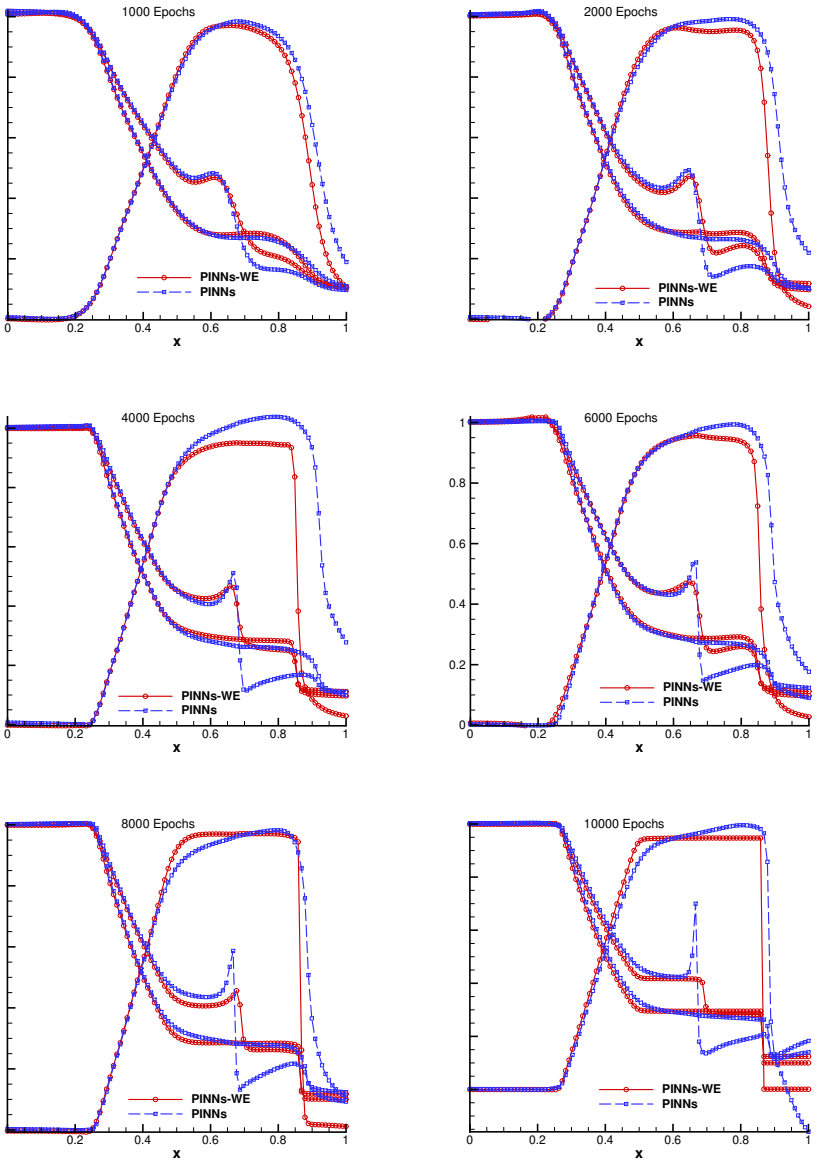


Figure 4: The results at different training epochs for Sod problem

in capturing shock waves without any transition points inside the shock. This is due to the feature that there has no dissipation introduced into the equations system.

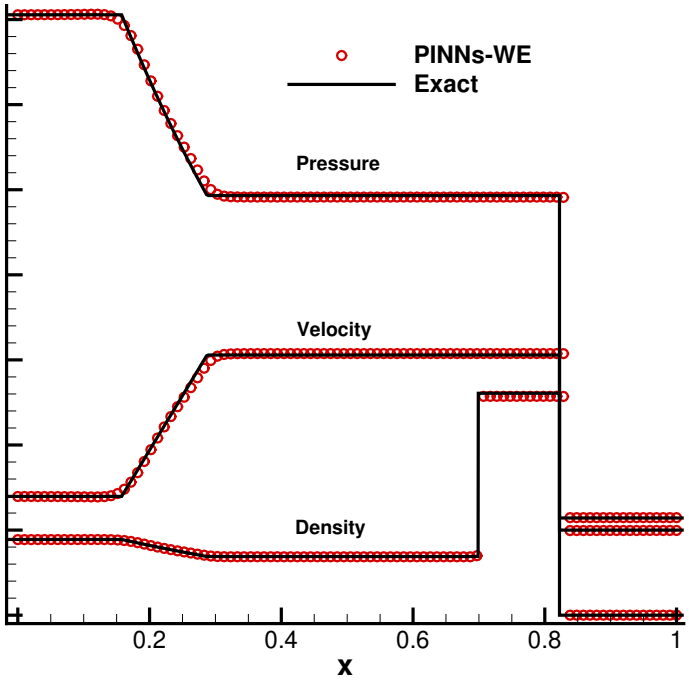
Then Fig.2 gives the loss histories of the training and compared the results with the original PINN. In Fig.3, we take some results at different epochs to show the development of training. The results illustrate that, at the first 1000 epochs, there is little difference between original PINN and PINN-WE as there still no strong compression region appears. Then original PINN fall into a straggle at training the shock. While PINN-WE can easily avoid this process. With the loss decreasing at smooth regions, a sharp shock appears with the trouble points compressed out by the the left and right smooth regions.

4.2. Lax problem

The second problem is a Lax problem, it is also a Riemann problem contains a strong shock and a strong contact, the initial condition is given as

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528), & if \quad 0 \leq x \leq 0.5 \\ (0.5, 0, 0.571), & if \quad 0.5 < x \leq 1 \end{cases} \tag{10}$$

We use a same network structure with the first example. The number of interior points is 50000 random taken from an uniformed 1000×5000 mesh in $X \times T$ space. And the initial points is 1000. This problem is more difficult to train than the Sod. The total loss is 0.01 at the end of the training with 20000 epochs. We also compare the result with the classical high-order WENO-Z scheme in Fig.4 We can get a similar conclusion in this example, the PINN-WE is effective in capturing shocks.



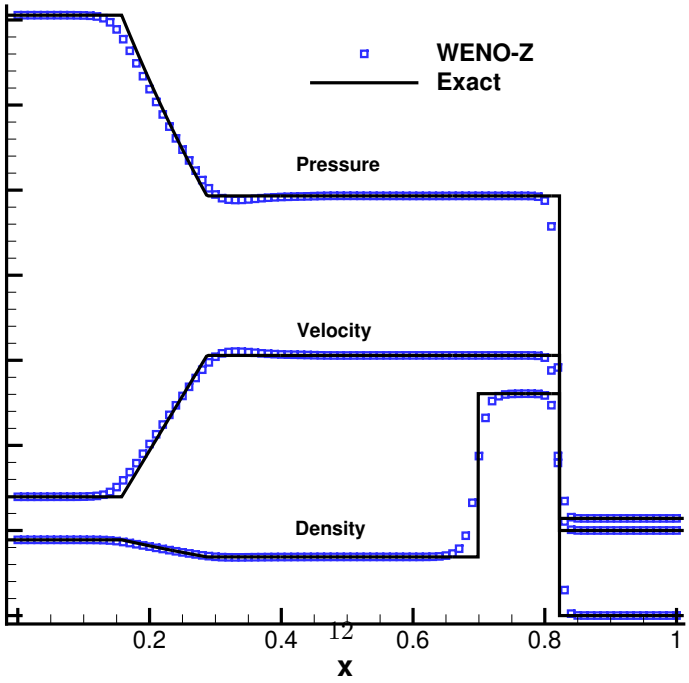


Figure 5: Results and loss history of the Lax problem

4.3. 2D Riemann problem

Then we consider a 2D problem. The basic setting can ref to case 8 in [18]. The initial condition is given as

$$(\rho,u,v,p)=\begin{cases}(1,-0.75,0.5,1) & \text{if} \quad 0\le x\le 0.5, 0\le y\le 0.5,\\ (2,0.75,0.5,1) & \text{if} \quad 0\le x\le 0.5, 0.5< y\le 1,\\ (3,-0.75,-0.5,1) & \text{if} \quad 0.5< x\le 1, 0\le y\le 0.5,\\ (1,0.75,-0.5,1) & \text{if} \quad 0.5\le x\le 0.5, 0.5< y\le 1.\end{cases} \tag{11}$$

We use the NN with 6 hidden layers and 60 neurons in each. The training points are sampled by LHS with 100,000 interior points in $T\times X\times Y$ and 10,000 initial points in $X\times Y$. The final loss is 0.08 when the training is ended. We test the model with meshed points 100×100 in $X\times Y$ at time 0.4. Then compare the results with the WENO-Z scheme in a 100×100 mesh. The result is shown in Fig.5-6.

Results illustrate that the new method can capture the discontinuities sharply and without oscillations that can not be totally avoided by a classical high-order method.

4.4. Shock diffraction over a cylinder

At last, we consider another 2D example with strong shock. In this example, a Mach $M_s=1.65$ planar shock interacts with a stationary circular cylinder. Similar problem and detailed analysis can ref to [19]. The post-shock Mach number is $M_2=0.728$, the front-shock state is $(\rho_1,u_1,v_1,p_1)=(1,0,0,1)$ and the post shock state is $(\rho_2,u_2,v_2,p_2)=(2,112,1.028,0,3.011)$. The basic initial condition is illustrated in Fig.7, the computing time is $t=0.4$. A NN with 7 hidden layers is used with 90 neurons in each layer. The collocations is sampled with LHS method in the 3D domain $T\times X\times Y$. And the number is $100,000$. The boundary points are random sampled on the cylinder and the number is $10,000$. The initial points are also sampled with LHS method and the number is $10,000$. The total loss is 0.048 after 20000 steps optimizing with ADAM and 2000 steps optimizing with LBFGS.

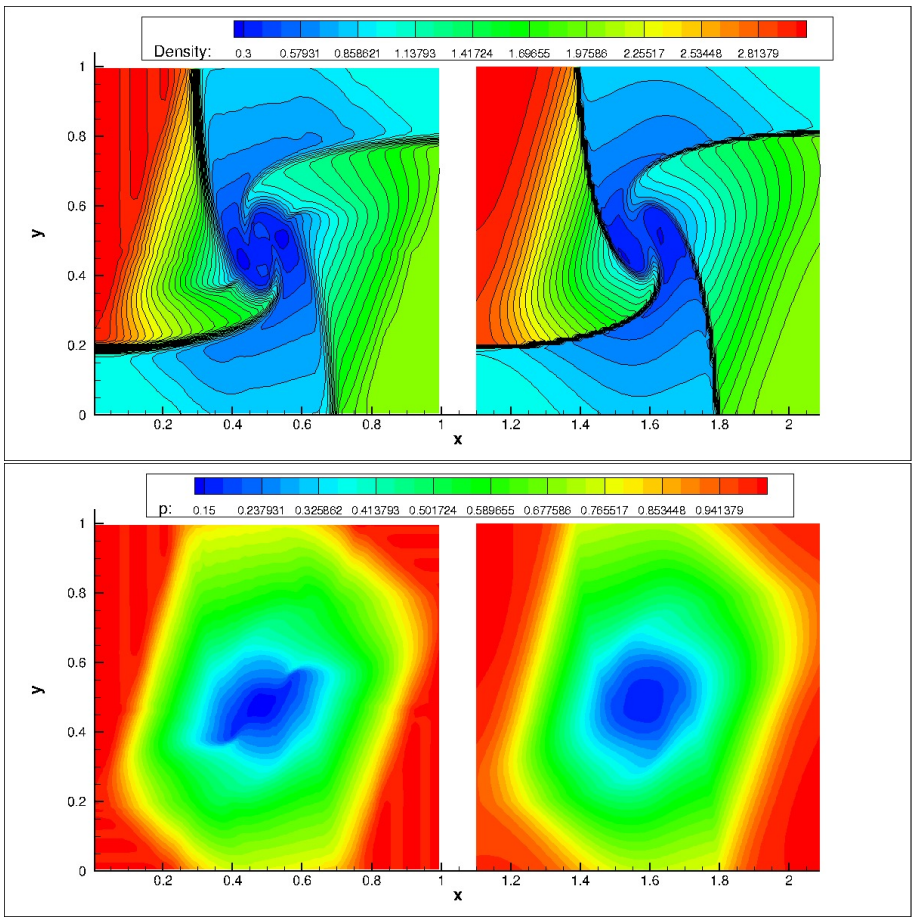


Figure 6: Results for 2D Riemann problem (part 1), the first is the density, second is the pressure,left is with the traditional method and right is with the new PINN-WE method.

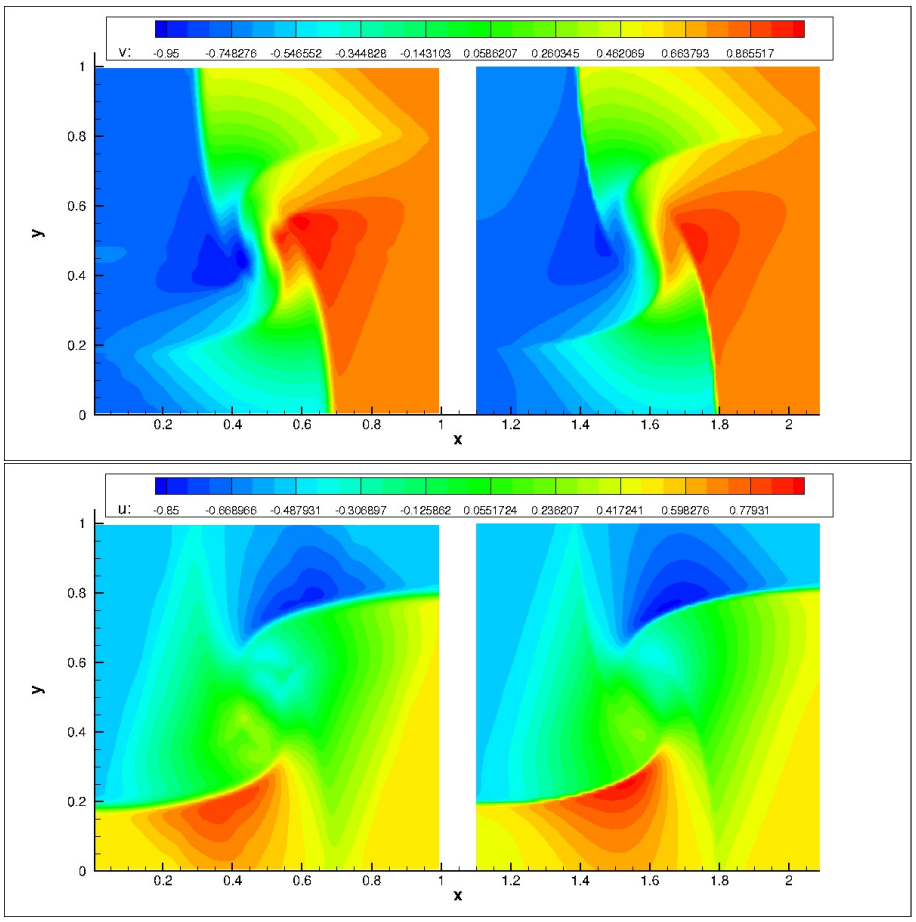


Figure 7: Results for 2D Riemann problem (part 2), the third and fourth are the velocities, left is with the traditional method and right is with the new PINN-WE method.

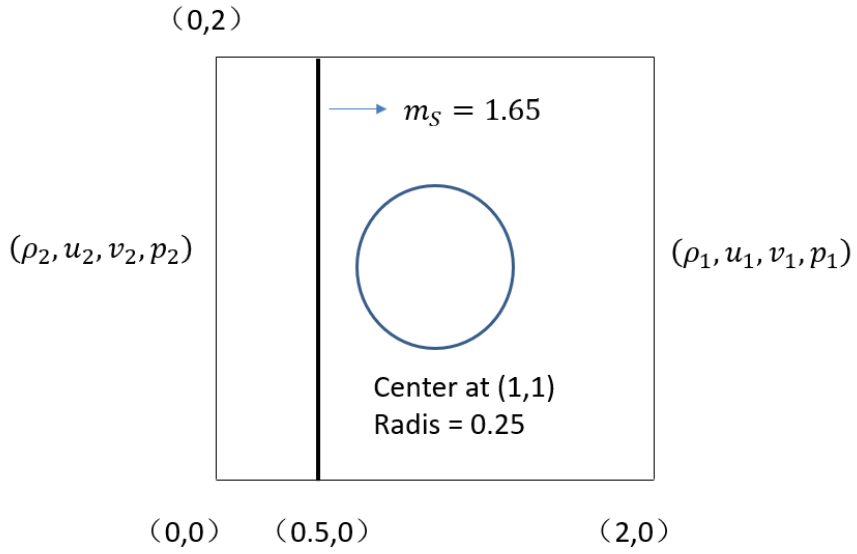


Figure 8: Results for shock diffraction over a cylinder, density and pressure, left is with the traditional method and right is with the new PINN-WE method.

Then we show the result at time 0.4 in Fig.6. We can get the result that the new PINN-WE method also effective in the relatively complex high-dimensional problems. Compared with the result by the classical high-order WENO-Z method with 100×100 grids, more sharp shocks can get by the new method.

5. Conclusions and discussions

In this paper, we propose a new idea to capture the discontinuities, especially shock waves using PINN in solving Euler equations. Different and may even be opposite from the idea that enhances the NNs' expression in the large gradient domain. We guess there is a paradoxical problem within the wrongly inside shock points (trouble points), that no matter increasing or decreasing gradients all may increase the total loss. So the NNs may let the training falls into a confrontation. In this paper, we introduce a positive gradient-dependent weight into the governing equations to adjust the expression of the network in different regions with different physical features. And then in solving Euler equations,

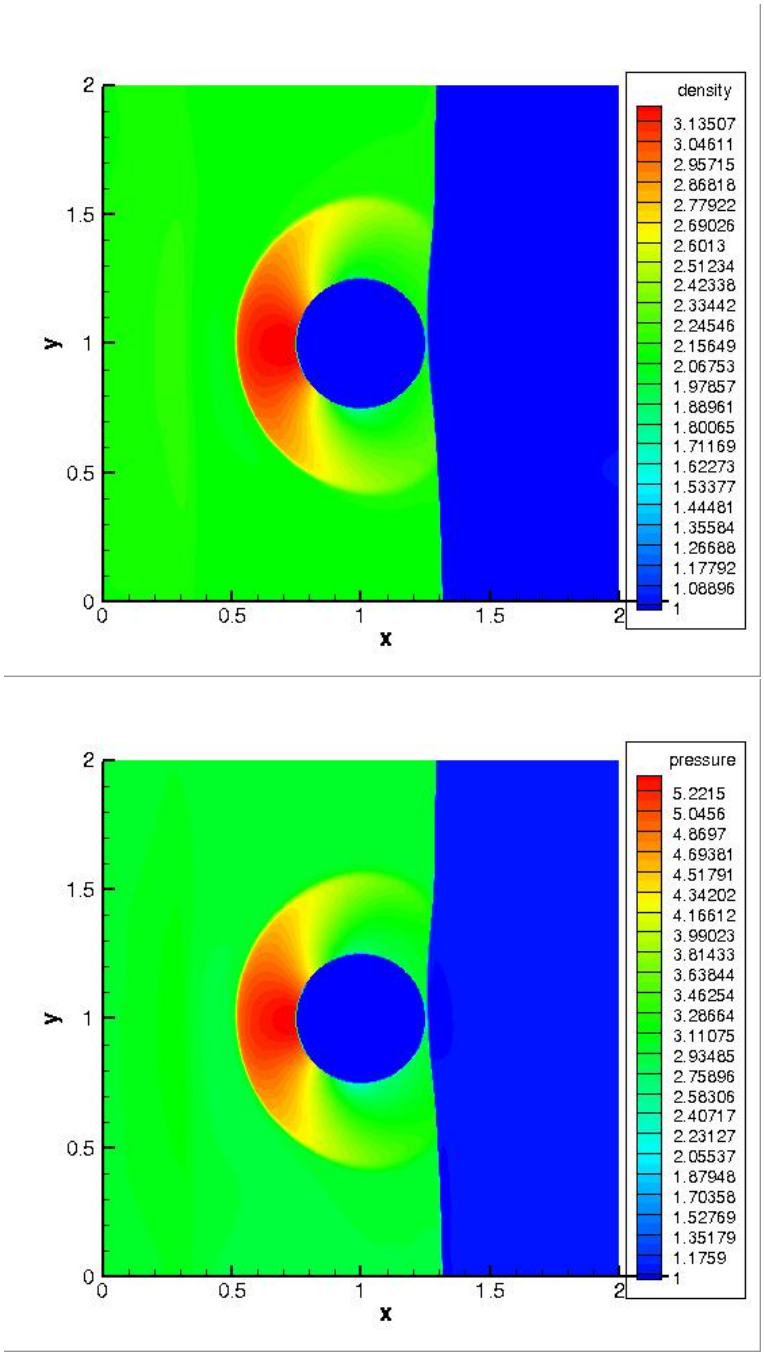


Figure 9: Results for shock diffraction over a cylinder, density and pressure, left is with the traditional method and right is with the new PINN-WE method.

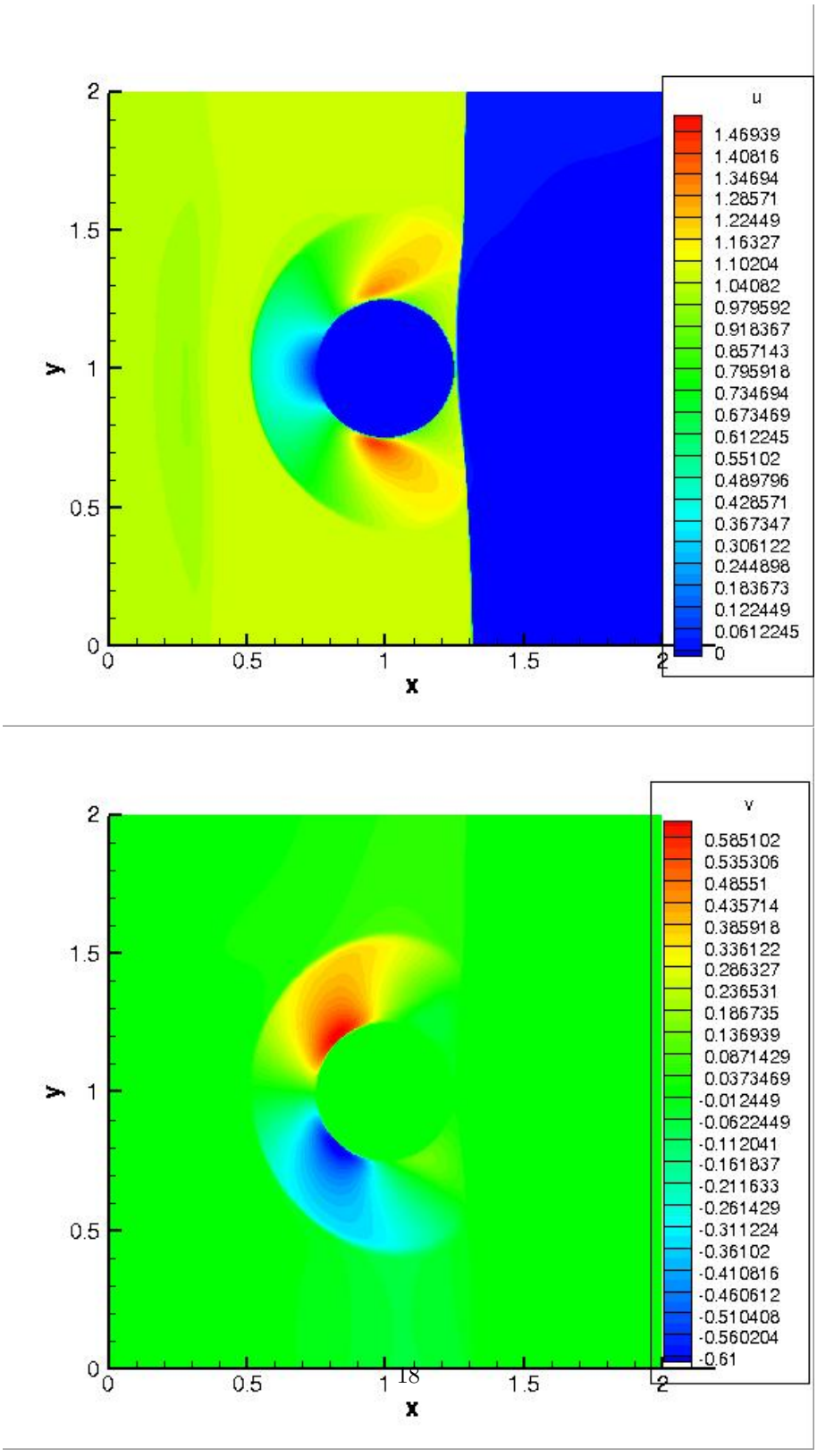


Figure 10: Results for shock diffraction over a cylinder, velocities u and v, left is with the traditional method and right is with the new PINN-WE method.

we construct a weight that is inverse to the local physics compression which can be measured by the divergence of the velocity.

By solving the weighted equations (WE) with PINN, the network will focus on training the smooth regions only as the shock regions has very small weights, then relay on the real physics compression from the trained smooth regions, the discontinuities automatically appear as the trouble points move out into smooth regions like passive particles.

This work only focuses on capturing discontinuities. However, there are still many open questions in solving Euler equations. Here we may give some discussions about other problems that we have faced but do not truly solve in simulating complex discontinuous problems.

Firstly, it is hard to converge to a real physical weak solution. Using WE, PINN can fast convergent to a discontinuous result but it has a chance to be a nonphysical weak solution of Euler equations. So how to introduce effective entropy conditions and other physical limitations into PINN still need more work. Secondly, the shock's position may not exactly be right without exactly conservation of mass, momentum and the total energy in PINN. Thirdly, as a global method, PINN could simulate big structures well but is still weak in capturing detailed structures as shown in Fig.5. So how to improve the resolutions of PINN is important to many problems and needs further study.

References

[1] J. VonNeumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, Journal of applied physics 21 (3) (1950) 232–237.

[2] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, in: Upwind and high-resolution schemes, Springer, 1987, pp. 218–290.

[3] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted eno schemes, Journal of computational physics 126 (1) (1996) 202–228.

[4] B. Cockburn, C.-W. Shu, The local discontinuous galerkin method for time-dependent convection-diffusion systems, SIAM Journal on Numerical Analysis 35 (6) (1998) 2440–2463.

[5] S. Pirozzoli, Numerical methods for high-speed flows, Annual review of fluid mechanics 43 (2011) 163–194.

[6] D. Zhang, C. Jiang, D. Liang, L. Cheng, A review on TVD schemes and a refined flux-limiter for steady-state calculations, Journal of Computational Physics 302 (2015) 114–154.

[7] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational physics 378 (2019) 686–707.

[8] G. Pang, L. Yang, G. E. Karniadakis, Neural-net-induced gaussian process regression for function approximation and PDE solution, Journal of Computational Physics 384 (2019) 270–288.

[9] K. O. Lye, S. Mishra, D. Ray, Deep learning observables in computational fluid dynamics, Journal of Computational Physics 410 (2020) 109339.

[10] J. Magiera, D. Ray, J. S. Hesthaven, C. Rohde, Constraint-aware neural networks for riemann problems, Journal of Computational Physics 409 (2020) 109345.

[11] H. Huang, Y. Liu, V. Yang, Neural networks with local converging inputs (NNLCI) for solving conservation laws, part II: 2D problems, arXiv preprint arXiv:2204.10424.

[12] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what's next, arXiv preprint arXiv:2201.05624.

[13] R. G. Patel, I. Manickam, N. A. Trask, M. A. Wood, M. Lee, I. Tomas, E. C. Cyr, Thermodynamically consistent physics-informed neural networks for hyperbolic systems, Journal of Computational Physics 449 (2022) 110754.

[14] Z. Mao, A. D. Jagtap, G. E. Karniadakis, Physics-informed neural networks for high-speed flows, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112789.

[15] A. D. Jagtap, E. Kharazmi, G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113028.

[16] A. D. Jagtap, Z. Mao, N. Adams, G. E. Karniadakis, Physics-informed neural networks for inverse problems in supersonic flows, arXiv preprint arXiv:2202.11821.

[17] A. Papados, Solving hydrodynamic shock-tube problems using weighted physics-informed neural networks with domain extension.

[18] A. Kurganov, E. Tadmor, Solution of two-dimensional riemann problems for gas dynamics without riemann problem solvers, Numerical Methods for Partial Differential Equations: An International Journal 18 (5) (2002) 584–608.

[19] H. Mo, F.-S. Lien, F. Zhang, D. S. Cronin, An immersed boundary method for solving compressible flow with arbitrarily irregular and moving geometry, International Journal for Numerical Methods in Fluids 88 (5) (2018) 239–263.