

Riemann solver modeled by artificial neural networks for Euler equations

1. Introduction

The Riemann problem, first proposed by Bernhard Riemann in 1860 [?], is the simplest, non-trivial initial value problem for the hyperbolic system [?]. Although the Riemann problem is an old and classical topic, it become hot again in the domain of numerical computations, after Godunov developed a famous finite-difference scheme (Godunov method) in 1959 [?] and used the solution of Riemann problem as the key ingredient in the Godunov method. After more than two decades researching about the Riemann problem, nowadays, it is well recognized that the Riemann problem plays the role of "building blocks" for all fields of theory, numerics and applications [? ?].

For numerics, the original Godunov method employs an exact Riemann solver to define the numerical flux at each computational cell interface. However, as iterative procedures are needed in the construction of exact Riemann solvers, it always costly to use the exact solver in a practical computation, besides, for some complex equations systems, for example, magnetohydrodynamics (MHD)[?] and elastic-plastic flows [?], the exact solution is not that easy to construct. For these reasons, approximate, non-iterative solutions are preferred in current computational fluid dynamics (CFD).

There is a huge amount of approximate Riemann solvers are developed in recent two decades, including approximate state-based Riemann solvers, for example, double-shock Riemann solver, double-rarefaction Riemann solver

, double-shock Riemann solver [? ?], approximate fluxed-based Riemann solvers, such as the HLL-type solver [?] and Osher Riemann solver. But in this

paper, we try to propose a different idea of constructing approximate Riemann solvers with the powerful technique of artificial neural networks (ANN) and deep learning.

Deep learning, which uses multi-layer neural networks (deep neural networks) to model nonlinear functions, recently has revolutionized many fields such as image, text, and speech recognition. Many researchers also have extended the neural networks to solve the PDEs and ODEs. We think the deep learning is also very suitable to modelling Riemann solvers. There are some practical reasons.

Firstly, limited to the simplification, approximate Riemann solvers always loss important features of the Riemann problem, for example, most of the approximate Riemann solvers can not resolve the contact wave or multi-materials interface well. Secondly, too much numerical dissipation exist in most of the approximate Riemann solvers, especially simulating a weak wave, such as a rarefaction wave. However, if we turn into an exact Riemann solver, the complexity increases the computer's burden and is difficult to accelerate by a modern computing technique such as the GPU acceleration due to its algorithm structure. Different from the existing classical methods, machine learning is a data-driven approach. We do not need to simplify the real structures in the solution of the Riemann problem. With enough training of the networks, we can get an accurate enough model of the exact Riemann solver. The key of the modelling is how to get the data sets. Fortunately, as the Riemann problem is so simple that even for a complex equation system, it is easy to get data by a refined mesh numerically. For a simple equation system such as the Euler equations studied in this paper, an exact Riemann solver can be used to get the data set. Besides, coding the deep neural networks Riemann solver with GPU is naturely and directly. Up till now, as far as we know, there is no study about using deep learning to solve the Riemann problem.

This paper is organized as follows. In Section 2, the Riemann problem and exact Riemann solver are given for the Euler equation system. In Section 3, we will introduce the deep neural network architecture and training algorithms

we used. Other numerical methods used in the comparison examples is given in Section 4. Numerical examples and comparisons are taken in Section 5. In Section 6, some conclusions are listed.

2. The Riemann problem and the exact solver of Euler equations

2.1. The one-dimensional Euler equations [?]]

In this paper, we study the one-dimensional time-dependent Euler equations with an ideal Equation of State. The conservative formulation of the Euler equations, in differential form, is

$$\mathbf{U} + \mathbf{F}(\mathbf{U})_x = \mathbf{0}, \quad (1)$$

where \mathbf{U} and $\mathbf{F}(\mathbf{U})$ are the vectors of conserved variables and fluxes, given respectively by

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

where ρ is density, p is pressure, u is particle velocity and E is total energy per unit volume

$$E = \rho \left(e + \frac{1}{2} u^2 \right), \quad (3)$$

where e is the specific internal energy given by a caloric Equation of State (EOS)

$$e = e(p, \rho). \quad (4)$$

For ideal gases one has the simple expression as

$$e = \frac{p}{(\gamma - 1)\rho}, \quad (5)$$

where γ is the ratio of specific heats.

The conservation laws (1) - (2) may also be written in quasi-linear form as

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x = \mathbf{0}, \quad (6)$$

where the coefficient matrix $\mathbf{A}(\mathbf{U})$ is the Jacobian matrix

$$\mathbf{A}(\mathbf{U}) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma - 3)u^2 & (3 - \gamma)u & \gamma - 1 \\ \frac{1}{2}(\gamma - 1)u^3 - \frac{a^2 u}{\gamma - 1} & \frac{3 - 2\gamma}{2}u^2 + \frac{a^2}{\gamma - 1} & \gamma u \end{bmatrix}, \quad (7)$$

where a is the sound speed, for ideal gases, it can be solved as

$$a = \sqrt{\frac{\gamma p}{\rho}}. \quad (8)$$

The eigenvalues of the Jacobian matrix \mathbf{A} are

$$\lambda_1 = u - a, \quad \lambda_2 = u, \quad \lambda_3 = u + a. \quad (9)$$

and the corresponding right eigenvectors are

$$\mathbf{H}_1 = \begin{bmatrix} 1 \\ u - a \\ H - ua \end{bmatrix}, \mathbf{H}_2 = \begin{bmatrix} 1 \\ u \\ \frac{1}{2}u^2 \end{bmatrix}, \mathbf{H}_3 = \begin{bmatrix} 1 \\ u + a \\ H + ua \end{bmatrix}. \quad (10)$$

2.2. The Riemann problem and wave structures

The Riemann problem for the one-dimensional, time dependent Euler equations (1)-(2) is the initial value problem with data $(\mathbf{U}_L, \mathbf{U}_R)$

$$\begin{cases} \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0}, \\ \mathbf{U}(x, 0) = \begin{cases} \mathbf{U}_L & \text{if } x < 0, \\ \mathbf{U}_R & \text{if } x > 0. \end{cases} \end{cases} \quad (11)$$

The structure of the exact solution of the Riemann problem has three waves,

associated with the eigenvalues $\lambda_1 = u - a$, $\lambda_2 = u$ and $\lambda_3 = u + a$. The three waves separate four constant states, shown in Fig., from left to right the values on the states are: \mathbf{U}_L , \mathbf{U}^*L , \mathbf{U}_R^* and \mathbf{U}_R . Where \mathbf{U}_L^* and \mathbf{U}_R^* are unknown.

According to the analysis in Ref.[?]], the middle wave is always a contact wave while the left and right waves are either shock or rarefaction waves. Therefore there can be four possible wave patterns which are shown in Fig.

2.2.1. Equations for pressure and particle velocity

According to the relation of the contact wave, the pressure and velocity of \mathbf{U}_L^* and \mathbf{U}_R^* are equal. We can establish an equation for pressure p^* as

$$f(p, \mathbf{U}_L, \mathbf{U}_R) = f_L(p, \mathbf{U}_L) + f_R(p, \mathbf{U}_R) + \Delta u = 0, u = u_R - u_L, \quad (12)$$

where the functions f_L and f_R are given by

$$f_L(p, \mathbf{U}_L) = \begin{cases} (p - p_L) \left[\frac{A_L}{p + B_L} \right]^2 & \text{if } p > p_L (\text{shock}), \\ \frac{2a_L}{(\gamma - 1)} \left[\left(\frac{p}{p_L} \right)^{\frac{\gamma-1}{2\gamma}} - 1 \right] & \text{if } p \leq p_L \quad (\text{rarefaction}), \end{cases} \quad (13)$$

and

$$f_R(p, \mathbf{U}_R) = \begin{cases} (p - p_R) \left[\frac{A_R}{p + B_R} \right]^2 & \text{if } p > p_R (\text{shock}), \\ \frac{2a_R}{(\gamma - 1)} \left[\left(\frac{p}{p_R} \right)^{\frac{\gamma-1}{2\gamma}} - 1 \right] & \text{if } p \leq p_R \quad (\text{rarefaction}), \end{cases} \quad (14)$$