Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2022-0224

Vol. **36**, No. 1, pp. 160-199
July 2024

Development of a Less Dissipative Interface Variable Reconstruction to Solve the Euler Equations by Q Learning Method

Shing-Ian Huang, Sheng Chang Wu, Tsung-Yu Yang, Chi-Heng Ting, Yi-Jhen Wu and Yang-Yao Niu[*]

Department of Aerospace Engineering, Tamkang University, New Taipei City, ROC.

Communicated by Kun Xu

Received 4 September 2022; Accepted (in revised version) 13 May 2023

Abstract. In this study, we propose a blend of the average of THINC-EM and MUSCL (ATM) methods based on the AUSMD scheme for solving detonation wave problems. It is well known that the simulation of the detonation problems can produce incorrect shock information or strong spurious due to the stiff source term. Accurate simulation of detonation problems plays a crucial role in the design of detonation engines. The proposed ATM method combines the MUSCL and THINC-EM methods with different weighting functions, the optimized parameters of which are determined by the Q-learning method in order to accurately capture detonation waves, shock waves, and expansion fans. To validate the proposed numerical method, one and two-dimensional shock tube and the detonation tube and nozzles are chosen as benchmark test cases. Our numerical results show that the proposed the ATM type AUSMD scheme has great potential for handling more complex detonation problems and pulse detonation engine flow problems.

AMS subject classifications: 65Y04

Key words: High resolution scheme, Euler equation, ATM method, discontinuity and detonation, Q-learning.

Nomenclature

U:	Conservation term
E:	Convection term of x axis
F:	Convection term of y axis
ψ:	Source term

[*]Corresponding author. *Email address:* `yyniu@mail.tku.edu.tw` (Y.-Y. Niu)

u:	Velocity of x axis
v:	Velocity of y axis
e_t:	Energy
q_0:	Chemical heat release
ρ:	Density
p:	Pressure
R:	Gas constant
ε:	Reaction time
r:	Specific heat ratio
z:	Fraction of unreacted fluid
T:	Temperature
T_0:	Ignition temperature
$\dot{m}$:	Mass flow
h:	Enthalpy per unit mass
Q:	Q value
s_τ:	Current state
a_τ:	Current action
$s_{\tau+1}$:	New state
$a_{\tau+1}$:	All possible actions at that new stat
η:	Learning rate
γ:	Discount rate
r:	Reward for taking that action at that state
τ:	Time period

1 Introduction

As demonstrated by numerous studies, the thermodynamic cycle of the engine driven by a detonation wave is highly efficient [1–3]. During operation of detonation engines, the detonation wave, which propagates at supersonic speed, typically transforms reactants into products and releases energy, resulting in sudden and sharp jumps of the thermodynamic states. It is noted that under certain conditions, the detonation wave propagates at or near the Chapman-Jouguet (CJ) velocity, which is induced by the reactions associated with shock waves and expansion fans within ducts or nozzles. There is a wide spectrum of length scales and times in the collision of transverse waves caused by high pressure regions and intensive reactions. However, it is not easy to observe the process of detonation and understand the relevant scientific theories inside the internal reactive flows. Numerical simulation is often the most economical and effective way to study internal detonation flows. The hyperbolic conservation laws are generally used in conjunction with the stiff source terms to model the discontinuity fronts of the chemical reactive flows. However, traditional schemes may produce spurious or incorrect wave information due to inhomogeneous source terms, making it difficult to accurately capture discontinuities when detonation waves occur (as first reported in [4]). Therefore, a precise numerical dissipation in the selected scheme is the necessary remedy for the spatial or time resolution of discontinuities in the detonation problem calculations.

During the past decades, significant efforts have been put in to eliminate the spurious numerical solutions of the chemical reactive flow problems. These studies have resulted in the development of modern schemes such as volume of fraction schemes [5–8], level set schemes [9, 10], and front tracking methods [11], which can track detonation fronts or interfaces successfully. However, these methods can be challenging to combine current shock-capturing schemes without losing conservation properties. In addition, Chorin [12] proposed an analytical solution for discontinuities, and it can determine the locations of reactive waves without requiring extra numerical viscosity. This approach has been successful in resolving detonation problems. However, it is not easy to extend to simulation of the complex flow problems. Yee et al. [13] discussed a simple scalar equation with the source term for understanding the behavior of numerical methods in reacting flow problems. MacCormack-type predictor-corrector methods with flux limiters and splitting are proposed for the fluid dynamics and chemistry by solving the homogeneous conservation law as the convection step and the reaction step separately. Hezel et al. [14] presented a modified fractional step method for detonation waves, and it uses the exact Riemann solution to determine where burning should occur. Bao and Jin [15] proposed a random projection method based on the fractional step method, which uses a standard shock-capturing scheme in the convection step and performs a projection in the reaction step to make the ignition temperature random. This method has been applied successfully to various problems in one- and two- dimensions, but it assumes an a priori stiff source. In 2005, Xiao, Honma, and Kono [16] presented a practical and straightforward numerical method for capturing the free boundary, and it was devised using the hyperbolic tangent function. This method, also known as THINC (tangent of hyperbola for interface capturing) [17], uses the hyperbolic tangent function to compute the numerical flux for the fluid fraction function and gives a conservative, oscillation-free, and smearing-free solution to the fluid fraction function even for very distorted interfaces. In 2018, Deng et al. [18] proposed a new shock-capturing scheme to solve stiff detonation wave problems. This method follows the sharp property of THINC to reduce the dissipative effect of the MUSCL method [19] based on the idea of boundary variation diminishing (BVD). It can significantly reduce the numerical dissipation errors, thereby preventing the false calculation of the discontinuities or spurious waves propagation speeds. By choosing the appropriate reconstruction polynomials from a given set of polynomials, the BVD approach can reduce numerical dissipation by minimizing jumps at the cell interfaces. As shown in [17], the degree of the discrepancy in the discontinuities propagation speed is highly dependent on how shock-capturing schemes deal with the discontinuities. The fundamental reason behind this spurious phenomenon is that shock-capturing schemes introduce excessive numerical dissipation errors around discontinuities. During the reconstruction process, a steplike tangent of hyperbola for interface capturing (THINC) function has been applied in many numerical approaches and models as evident in [20, 21]. The final reconstruction function effectively removes the numerical dissipations. The resulting scheme is referred to as MUSCL-THINC-BVD.

Reinforcement learning is a computational approach to understanding and automating goal-directed learning and decision-making. It differs from other computational approaches as it emphasizes an individual's learning from direct interaction with his/her environment without relying on exemplary supervision or a complete model of the environment. Reinforcement learning is the first to seriously address the computational problems that arise when learning from interactions with the environment to achieve long-term goals. In the recent years, it has gained significant attention in the machine learning (ML) and artificial intelligence communities. In a 2020 paper, Frank et al. [22] examined how ML can be used to accelerate or enhance the accuracy of simulation techniques such as computational fluid dynamics, molecular dynamics, and structural analysis. They have also investigated the potential of ML in generating computationally efficient surrogate models of physical systems, which could potentially eliminate the need for more costly simulation techniques.

Several advances have been observed for the WENO scheme in the recent years through the use of ML techniques. In 2019, Wang et al. [23] were the first to view numerical PDE solvers as a Markov Decision Process (MDP) and use reinforcement learning (RL) to learn new solvers, with a focus on solving 1D Scalar Conservation Laws. In 2021, Kossaczká et al. [24] improved the accuracy of the well-known fifth-order WENO shock-capturing scheme using deep learning techniques. Recently, in 2022, Way et al. [25] introduced Backpropagation Through Time and Space (BPTTS), a method that trains a recurrent spatiotemporal neural network, and is used in a homogeneous multi-agent reinforcement learning (MARL) setting to learn numerical methods for hyperbolic conservation laws.

There are also several developments for the MUSCL scheme in recent years through the use of ML techniques. In 2017, Singh et al. [26] proposed a modeling paradigm that aimed to improve the accuracy of predictive models for turbulence by effectively utilizing limited data obtained from physical experiments. In 2020, Grimberg et al. [27] applied ML to reduce the complexity of CFD models for turbulent flows. In 2021, Meuris et al. [28] proposed a method that combined deep neural networks and spectral methods to solve partial differential equations (PDEs). They used Deep Operator Network (DeepONet), a deep learning approach, to identify suitable functions to expand the solution of PDEs. Most recently, in 2022, Schwarz et al. [29] developed a new slope limiter based on deep learning and reinforcement learning techniques. The limiter satisfies several admissibility constraints to ensure the accuracy of the solution, including positivity and adherence to a relaxed discrete maximum principle.

The class of methods for solving optimal control problems by solving this equation is known as dynamic programming [30]. In 1957, Bellman [31] also introduced the discrete stochastic version of the optimal control problem known as Markovian decision processes (MDPs). In 1960, Ron Howard [32] devised the policy iteration method for

MDPs. These are all essential elements underlying the theory and algorithms of modern reinforcement learning.

Dynamic programming is widely considered to be the only feasible method for solving general stochastic optimal control problems. It suffers from what Bellman called the ″curse of dimensionality″, where the computational requirements grow exponentially with the number of state variables. Despite this, it is still more efficient and widely applicable than any other general method. Since the late 1950s, dynamic programming has been extensively developed, including extensions to partially observable MDPs [33], many applications (including those surveyed by White [34–36] in 1985, 1988, and 1993), approximation methods (including those surveyed by Rust [37] in 1996), and asynchronous methods (including those surveyed by Bertsekas [38,39] in 1982 and 1983).

Another thread that led to the modern field of reinforcement learning is centered on the idea of trial-and-error learning. This thread began in psychology, where ″reinforcement″ learning theories are common. The first person to succinctly express the essence of trial-and-error learning was Edward Thorndike. He proposed the idea that good or bad outcomes following an action tend to be reselected and changed accordingly in a process he referred to as the ″Law of Effect.″ This law describes the impact of reinforcing events on the tendency to select actions. Although sometimes controversial [40–42], the Law of Effect is widely regarded as a basic principle underlying such behavior [43–46].

The Law of Effect includes the two most important aspects of trial-and-error learning. Firstly, it is selective, which means that it involves trying alternatives and choosing by comparing their consequences. Secondly, it is associative, meaning that the alternatives found by selection are relevant to a particular situation. Natural selection in evolution is a classic example of a selection process, but it is not associative. Supervised learning is associative, but not selective. The combination of these two is critical to the law of effects and trial-and-error learning. Another way of saying this is that the Law of Effect is the fundamental way of combining search and memory: search in the form of trying and selecting among many actions in each situation, and memory in the form of remembering which actions worked best, associating them with the situations in which they were the best. Combining search and memory in this way is essential for reinforcement learning.

In the early days of artificial intelligence (AI), before it was distinguished from other branches of engineering, some researchers explored trial-and-error learning as an engineering principle. The earliest computational investigations of trial-and-error learning were perhaps by Minsky [47] as well as Farley and Clark [48], both in 1954. Minsky discussed the computational models of reinforcement learning and described his construction of an analog machine composed of components which he called SNARCs (Stochastic Neural-Analog Reinforcement Calculators). Farley and Clark described another neural-network learning machine designed to learn by trial and error. In the 1960s, the terms ″reinforcement″ and ″reinforcement learning″ were used in the engineering literature for the first time in [48–50]. Minsky's paper ″Steps Toward Artificial Intelligence″ [51] was particularly influential.

The third thread to the history of reinforcement learning is related to temporal-difference learning. Temporal-difference learning methods are distinctive as they are driven by the difference between temporally successive estimates of the same quantity. This thread is smaller and less distinct than the other two, but it has played a particularly important role in the field. It is partially because temporal-difference methods seem to be new and unique to reinforcement learning.

Finally, in 1989, Chris Watkins's [52] development of Q-learning fully brought together the temporal-difference and optimal control threads. This work extended and integrated all three main lines of prior efforts in reinforcement learning research. In 1987, Paul Werbos [53] also contributed to this integration by arguing for the convergence of trial-and-error learning and dynamic programming. By the time Watkins was working, reinforcement learning research had evolved significantly, primarily within the ML subfield of AI, but also more broadly within neural networks and AI. Over the past few years, the Q-learning Q-table learning algorithm has become increasingly popular due to its success in addressing challenging sequential decision-making problems. Previous Q-learning Q-table learning algorithm approaches faced a difficult design issue in terms of feature choice. However, the Q-learning Q-table learning algorithm has been successful in complicated tasks with lower prior knowledge due to its ability to learn different levels of abstractions from data. This opens up the possibility to mimic some human problem-solving capabilities, even in high-dimensional space which, only a few years ago, was difficult to conceive.

Reinforcement learning (RL) has gained attention due to its low computation cost and excellent energy-saving performance. However, the potential of RL has not yet been fully realized in computational fluid dynamics (CFD) applications because it can be difficult to design targets. This paper presents a study on the use of less dissipative interface variable reconstruction to solve the Euler equations via the Q-learning method with optimized parameters. The aim is to accurately capture discontinuous planes. In the recent years, Q-learning has been used to solve various problems, including the simultaneous online optimization of energy efficiency and battery life in battery/ultracapacitor electric vehicles (Xu et al. [58]), prediction of wind speed (Kosana et al. [59]), and permutation flow shop scheduling problems (He et al. [61]). Suratkar et al. [62] also presented a honeypot system that conducts a severity analysis of adversaries who attack it. Q-learning show its effectiveness in solving these problems, so it is a promising method to consider for use in CFD applications. In the following section, the paper would compare the accuracy of MUSCL, THINC, and the proposed three versions of ATM (Average of Thinc and MUSCL) methods. In the numerical verification, the Q-learning of the enforce learning was used to decide the weighing parameter in the proposed ATM method on the resolution of one-dimensional blast wave problem as compared to the reliable results. Then, the optimized weighting factor in the proposed ATM methods will extend to compute the benchmark test case of a two-dimensional detonation tube. Finally, the single-tube and multi-tube pulse detonation problems would be performed for complicated simulation of complex shock waves, slip lines and detonative waves in the unsteady state.

2 Governing equations

2.1 Numerical models

The problem of the detonation tube in the current study can be described by an ideal gas, unsteady, compressible fluids. The 2D stiff Euler equation with source term as in [1,2] can be written as:

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=\frac{1}{\varepsilon}\psi(U), \tag{2.1}$$

where

$$\begin{aligned} &U=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho e_t\\ \rho Z\end{pmatrix}, \quad E(U)=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (\rho e_t+p)u\\ \rho uZ\end{pmatrix},\\ &F(U)=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (\rho e_t+p)v\\ \rho vZ\end{pmatrix}, \quad \psi(U)=\begin{pmatrix}0\\ 0\\ 0\\ 0\\ -\frac{1}{\varepsilon}\rho zH(T-T_0)\end{pmatrix}, \end{aligned} \tag{2.2}$$

where U, E, F and ψ denote the column vectors of the conserved variables, convective flux vectors E and F, and source term, respectively.

The pressure p can be obtained through equations of state:

$$p=(\gamma-1)\left(\rho e_t-\frac{1}{2}\rho u^2-\frac{1}{2}\rho v^2-q_0\rho Z\right), \tag{2.3}$$

where ρ is the fluid density, u is fluid velocity, e_t is the energy, Z is the fraction of unreacted fluid, q_0 is the chemical heat release, ε is the reaction time, r is the specific heat ratio, T_0 is ignition temperature, p is the pressure, T is the temperature. The ρ fluid density is chosen as the main parameter for computing the reward in connection with the Q-learning Q-table algorithm. The method of setting the reward score will vary depending on the specific case and goal, and will be explained in more detail in the following chapters.

Where $H(x)=1$ for $x>$, and $H(x)=0$ for $x<0$. More detail can be found in [15].

2.2 Numerical method

For the sake of simplicity, the numerical procedure is presented in one dimension. However, this method can be easily extended to multi-dimensions on structured grids through a dimension-wise reconstruction process. It is well-known that the numerical simulation of the considered flows not only requires the avoidance of spurious numerical oscillations, but also accurate resolution of stationary contact discontinuities. In this study, we compare three different sharp interface algorithms: MUSCL, THINC-EM, and the proposed method, ATM. The proposed ATM method aims to reconstruct the solution functions so that the jumps at cell boundaries are minimized, which effectively reduces the numerical dissipation in the resulting schemes. Before the spatial discretization, the time evolution is replied on the Strang splitting suggested in [13]. In the current numerical simulation inside a PDE's single and multiple nozzle tubes, the primary challenge in the simulation of the detonation waves inside a nozzle tube is due to the chemical time scale problem and the detonation velocity initial condition. The chemical time scale may be orders of magnitude faster than the fluid dynamical time scale, making the problem numerically stiff. To solve it, we apply the Strang Splitting Method which is by Yee [13] to overcome the time scale problem in the detonation wave problem. The Strang splitting takes the form as

$$U^{n+1}=S_\psi(k)S_f(k)S_\psi(k/2)U^n. \tag{2.4}$$

To maintain second-order accuracy, the Strang splitting can be used, in which the U^{n+1} is computed from U^n by

$$U^{n+1}=S_\psi(k/2)S_f(k)S_\psi(k/2)U^n. \tag{2.5}$$

A split version of the method studied might take the form:

$$S_\varphi(k/2):\ \left[I-\frac{1}{4}\,k\psi'\left(U_j^n\right)\right]\Delta U^*=\frac{1}{2}k\psi\ (U_j^n), \tag{2.6}$$

$$U_j^*=U_j^n+\Delta U_j^*, \tag{2.7}$$

$$S_f(k):\ \Delta U_j^{(1)}=-\frac{k}{h}\left(f(U_j^*)-f(U_{j-1}^*)\right), \tag{2.8}$$

$$U_j^{(1)}=U_j^*+\Delta U_j^{(1)}, \tag{2.9}$$

$$\Delta U_j^{(2)}=-\frac{k}{h}\left(f(U_{j+1}^{(1)})-f(U_j^{(1)})\right), \tag{2.10}$$

$$U_j^{(2)}=U_j^*+\frac{1}{2}(\Delta U_j^{(1)}+\Delta U_j^{(2)}), \tag{2.11}$$

$$S_\psi(k/2):\ \left[I-\frac{1}{4}k\psi'\left(U_j^{(2)}\right)\right]\Delta U^{(2)}=\frac{1}{2}k\psi(U_j^2), \tag{2.12}$$

$$U_j^{n+1}=U_j^{(2)}+\Delta U_j^{(2)}, \tag{2.13}$$

where U is the conservation value, and the $S_\varphi(k/2)$ is the operator means to calculate the source term in a half of time step, and $S_f(k)$ is to calculate the flux term. The main ideal to calculate the flux by $S_f(k)$ is using the MacCormack's method. The method is easily modified to include source terms in an explicit manner and maintain second-order accuracy.

First, the extended Euler equations in conservation form and source term can be solved by the explicit conservative scheme as

$$U_i^{n+1}=U_i^n-\frac{\Delta t}{\Delta x}\left(F_{i+\frac{1}{2}}^{n+\frac{1}{2}}-F_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right)+\Delta t H_i^n. \tag{2.14}$$

Then, we can perform the third-order accurate Runge-Kutta (RK) methods with a constant time step. It can be

$$U^1=U^n-cL(U^n), \tag{2.15}$$

$$U^2=\frac{3}{4}U^{n}+\frac{1}{4}U^{1}-\frac{1}{4}cL\left(U^1\right), \tag{2.16}$$

$$U^{n+1}=\frac{1}{3}U^{n}+\frac{2}{3}U^{2}-\frac{2}{3}cL\left(U^2\right), \tag{2.17}$$

where L is the calculation of the flux and source term defined by

$$L(F,H)=\left(F_{i+\frac{1}{2}}^{n+\frac{1}{2}}-F_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right)-\Delta x H_i^n. \tag{2.18}$$

And,

$$c=\frac{\Delta t}{\Delta x}=\frac{CFL}{\max|u+a|}, \tag{2.19}$$

where CFL is the Courant Friedrichs Lewy number and a is the speed of sound. In addition, the CFL number must be set less than 0.1.

For the spatial discretization of $F_{i+\frac{1}{2}}$, the primitive variables on the cell interface are determined through the second-order or third-order spatial accurate type MUSCL, THINC-EM, and ATM method. In this study, it is found that there is no noticeable difference between solutions obtained from second- or third-order schemes. Only results obtained via the interpolation with the third-order spatial accuracy are presented. The three-order accurate MUSCL, THINC-EM and ATM methods to reconstruct the conservation term and resolve discontinuous solutions with more numerical dissipation effects. Once the primitive variables on the left side and the right side of the cell interface are determined. Interpolation methods are used to determine the values of left and right side at interface $U_{i+\frac{1}{2}}^L$ and $U_{i+\frac{1}{2}}^R$ from primitive variable for reconstruction by U. Then it uses those values to calculate the flux. The values of left and right side at interface $U_{i+\frac{1}{2}}^L$ and $U_{i-\frac{1}{2}}^R$ at cell interfaces are computed from the reconstruction functions in cell C_i. The detail formula of each schemes will be presented in the following sections.

In the first part, we introduce the MUSCL scheme which the simplest possible reconstruction of the data is piecewise constant. The interfacial values at meshes by MUSCL function can be appeared as

$$U^{L,MUSCL}_{i+\frac{1}{2}}=U_i+\frac{1}{4}\left[(1-\eta)\sigma\left(\Delta_{i+\frac{1}{2}},\Delta_{i-\frac{1}{2}}\right)+(1+\eta)\sigma\left(\Delta_{i+\frac{1}{2}},\Delta_{i-\frac{1}{2}}\right)\right], \tag{2.20}$$

$$U^{R,MUSCL}_{i-\frac{1}{2}}=U_i-\frac{1}{4}\left[(1-\eta)\sigma\left(\Delta_{i+\frac{1}{2}},\Delta_{i-\frac{1}{2}}\right)+(1+\eta)\sigma\left(\Delta_{i+\frac{1}{2}},\Delta_{i-\frac{1}{2}}\right)\right], \tag{2.21}$$

where η is the value defined the spatial order of accuracy, σ is the function of a slope limiter defined at

$$\sigma\left(\Delta_{i+\frac{1}{2}},\Delta_{i-\frac{1}{2}}\right)=\frac{\Delta_{i-\frac{1}{2}}\left(\Delta^2_{i+\frac{1}{2}}+\omega\right)+\Delta_{i+\frac{1}{2}}(\Delta^2_{i-\frac{1}{2}}+\omega)}{[\Delta^2_{i+\frac{1}{2}}+\Delta^2_{i-\frac{1}{2}}+2\omega]}, \tag{2.22}$$

with

$$\Delta_{i+\frac{1}{2}}=U_{i+1}-U_i, \quad \Delta_{i-\frac{1}{2}}=U_i-U_{i-1}, \tag{2.23}$$

and $\omega=10^{-6}$. We use a constant value of three-order upwind-biased scheme $\eta=\frac{1}{3}$ to simulate the example and calculate the function. In addition, when $\eta=-1$ the second-order upwind bias scheme can be obtained. We denote the reconstructed value at cell interfaces from MUSCL reconstruction as $U^{L,\ MUSCL}_{i+\frac{1}{2}}$ and $U^{R,\ MUSCL}_{i-\frac{1}{2}}$. The MUSCL scheme has more numerical dissipation and is partial to smear out flow structures despite use widely in various numerical models. It is maybe a fatal fault in solving interfacial multi-component flows.

In the THINC-EM method as in [20], the tangent hyperbola function is used as a model function for a discontinuous volume fraction within a mesh cell. In principle, however, any model function $U(x)$ that can connect two states U_{i-1} and U_{i+1} in a monotone, compact fashion, such that

$$U_i=\frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(x)dx, \tag{2.24}$$

can be used. The THINC-EM model proposed is considered, the function of the volume fraction is represented as:

$$U_i(x)=\frac{U_{\max}}{2}\left[1+\gamma\tanh\left(\beta\Big(\frac{x-x_{i-1/2}}{x_{i+1/2}-x_{i-1/2}}-\widetilde{x}_i\Big)\right)\right], \tag{2.25}$$

where β is chosen by Q-learning in order to localize the discontinuity to be approximately in one cell. One can determine the left / right state interpolations by considering the integrated average value of the flux of material crossing a cell boundary, rather than just the end-states. This leads to another representation for left and right states, defined as follows using THINC-EM:

$$U_{L,i+\frac{1}{2}}=-\frac{1}{u_{i+\frac{1}{2}}\Delta t}\int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}-u_{i+\frac{1}{2}}\Delta t}U_i(x)dx,\quad u_{i+\frac{1}{2}}\geq 0, \tag{2.26}$$

$$U_{R,i+\frac{1}{2}}=-\frac{1}{u_{i+\frac{1}{2}}\Delta t}\int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2}}-u_{i+\frac{1}{2}}\Delta t}U_{i+1}(x)dx,\quad u_{i+\frac{1}{2}}<0. \tag{2.27}$$

Generalizing to an arbitrary cell interface, we have:

$$U_{L,i+\frac{1}{2}}=U_{\min}+\frac{1}{2}U_{\max}\left(1-\gamma D^+\right), \tag{2.28}$$

$$U_{R,i-\frac{1}{2}}=U_{\min}+\frac{1}{2}U_{\max}\left(1+\gamma D^-\right), \tag{2.29}$$

with

$$D^+=\frac{1}{\beta(V_{i+\frac{1}{2}}+\varepsilon)}\ln\left(\cosh\left(\beta V_{i+\frac{1}{2}}\right)-\frac{\tanh(\beta)+C}{1+C\tanh(\beta)}\sinh\left(\beta V_{i+\frac{1}{2}}\right)\right), \tag{2.30}$$

$$D^-=\frac{1}{\beta(V_{i-\frac{1}{2}}+\varepsilon)}\ln\left(\cosh\left(\beta V_{i-\frac{1}{2}}\right)+C\sinh\left(\beta V_{i-\frac{1}{2}}\right)\right), \tag{2.31}$$

$$B=\exp\left[\gamma\beta\left(2\frac{(U_i-U_{\min}+\epsilon)}{(U_{\max}+\epsilon)}-1\right)\right],\quad \epsilon=1\times 10^{12}, \tag{2.32}$$

$$C=\frac{(\frac{B}{\cosh(\beta)}-1)}{\tanh(\beta)}, \tag{2.33}$$

$$V_{i+1/2}=\frac{\left|\overrightarrow{u}_{i+1/2}\bullet\overrightarrow{n}_{i+1/2}\right|}{\left|\Delta\overrightarrow{x}_{i+1/2}\bullet\overrightarrow{n}_{i+1/2}\right|}\Delta t,\quad \Delta\overrightarrow{x}_{i+1/2}=\overrightarrow{x}_{c,i+1}-\overrightarrow{x}_{c,i}, \tag{2.34}$$

and

$$U_{\min}=\min(U_{i+1},U_{i-1}), \tag{2.35}$$

$$U_{\max}=\max(U_{i+1},U_{i-1})-U_{\min}, \tag{2.36}$$

$$\gamma=sgn(U_{i+1}-U_{i-1}). \tag{2.37}$$

The pure THINC method may cause spurious pressure oscillations. In order to avoid unnecessary numerical oscillations for detonation problems, it is essential to reconstruct the solution functions in such a way that the jumps at cell boundaries are minimized. This effectively reduces the numerical dissipation in the resulting schemes.

Here, we proposed a blend of an average of THINC-EM and MUSCL (ATM) difference to determine the right and left interfacial data, based on the weight functions, to achieve reconstructed interfacial variables at cell interface. The weight functions of the ATM method can be expressed in many ways as followings:

ATM1:

$$U_{i+\frac{1}{2},j} = \omega U^{MUSCL}_{i+\frac{1}{2},j} + (1-\omega) U^{THINC-EM}_{i+\frac{1}{2},j}, \tag{2.38}$$

$$\omega = MAX\left(1-\alpha\left|\frac{P_{i-1,j}-2P_{i,j}+P_{i+1,j}}{P_{i-1,j}+2P_{i,j}+P_{i+1,j}}\right|,0\right); \tag{2.39}$$

ATM2:

$$V = [\rho,u,v,p], \tag{2.40}$$

$$U_{i+\frac{1}{2},j} = \omega U^{MUSCL}_{i+\frac{1}{2},j} + (1-\omega) U^{THINC-EM}_{i+\frac{1}{2},j}, \tag{2.41}$$

$$\omega = MAX\left(1-\alpha\left|\frac{V_{i-1,j}-2V_{i,j}+V_{i+1,j}+\epsilon}{V_{i-1,j}+2V_{i,j}+V_{i+1,j}+\epsilon}\right|,0\right), \tag{2.42}$$

where ω is a weighting factor of primitive variable gradients, when it is close to zero it means the system is more like THINC-EM. Conversely when it is close to 1, it means the system is more like MUSCL.

In addition, the ATM 3 suggested by Chiu et al. in [57] is evaluated. The ATM3 is employed only when the following condition is satisfied:

ATM 3:

$$(U_{i+1}-U_i)(U_i-U_{i-1}) > 0. \tag{2.43}$$

The face values evaluated by this scheme are modified as

$$U^L_{i+\frac{1}{2}} = \omega U^{L,MUSCL}_{i+\frac{1}{2}} + (1-\omega) U^{L,THINC}_{i+\frac{1}{2}}, \tag{2.44}$$

$$U^R_{i-\frac{1}{2}} = \omega U^{R,MUSCL}_{i-\frac{1}{2}} + (1-\omega) U^{R,THINC}_{i-\frac{1}{2}}, \tag{2.45}$$

where $U^R_{i-\frac{1}{2}}$ and $U^R_{i+\frac{1}{2}}$ are the face values reconstructed using MUSCL and THINC, respectively, and the weighted function ω is defined with a near zero parameter ε as

$$\omega = 1-\min\left(\frac{U^{L,MUSCL}_{i+\frac{1}{2}} - U^{R,MUSCL}_{i-\frac{1}{2}} + \varepsilon}{U_{i+1}-U_i+\varepsilon}, \frac{U^{L,MUSCL}_{i+\frac{1}{2}} - U^{R,MUSCL}_{i-\frac{1}{2}} + \epsilon}{U_i - U_{i-1}+\epsilon}\right). \tag{2.46}$$

After deciding the interpolation method, the reconstructed values at cell boundaries are used to evaluate the AUSMD Riemann solver [54–56]. The flux extrapolation of $F^n_{i+\frac{1}{2}}$, based on AUSMD, is utilized to discretize the hyperbolic fluxes of mass, momentum, energy, and volume fraction equations. In this method, a primitive variable Riemann solver in the AUSMD form is used to calculate the numerical fluxes across the interfaces of discontinuous flow structures. It is well-known that the basic idea behind the AUSM type scheme is to split the numerical flux into a convection flux and a pressure flux. he proposed flux difference method, used to evaluate the mass flux of the convection flux, can eliminate this surplus numerical dissipation at contact discontinuities. We applied the Advection Upwind Splitting Method (AUSMD) without any additional dissipative terms to simulate the numerical flux around the interfaces. The "D" denotes a flux-Difference-splitting-biased scheme, which is used to calculate the correct flow velocity in each time step.

It is possible to capture discontinuities in the form of weak solutions for hyperbolic equations using high-resolution algorithms for conservation. However, it is important to maintain high accuracy in spatial differences in these algorithms, as second-order approximations often result in spurious oscillatory solutions. This is especially relevant in the simulation of fluid dynamics problems, where it is necessary to avoid entropy decrease in expansion waves due to the second law of thermodynamics. Linear numerical dissipation terms with adjustable parameters have historically been used to address this issue, but they are not effective when discontinuities are present, as they can cause numerical instability and oscillations.

The original numerical flux of AUSMD is defined as

$$E_{1/2}=\frac{1}{2}\left[(\rho u)_{1/2}(\psi_L+\psi_R)-|(\rho u)_{1/2}|(\psi_R-\psi_L)\right]+P_{1/2}, \tag{2.47}$$

where

$$\psi_L=\begin{pmatrix}1\\u_L\\H_L\\Y_L\end{pmatrix},\quad \psi_R=\begin{pmatrix}1\\u_R\\H_R\\Y_R\end{pmatrix},\quad P_{1/2}=\begin{pmatrix}0\\p_{1/2}\\0\\0\end{pmatrix},$$

and

$$(\rho u)_{1/2}=m_{1/2}=a_{1/2}M_{1/2}\rho_{L/R}. \tag{2.48}$$

The subscript "L/R" means the variables are chosen in an upwind manner. For example, if $M_{\frac{1}{2}}\geq 0$, then $\rho_{L/R}=\rho_L$; otherwise $\rho_{L/R}=\rho_R$. The $a_{1/2}$ mean the speed of sound. Here we simply use simple average as

$$a_{1/2}=\frac{1}{2}(a_R-a_L). \tag{2.49}$$

The Advection Upwind Splitting Method (AUSMD) has been demonstrated to possess favorable properties such as high resolution for contact discontinuities and shock waves, and conservation of enthalpy for single-phase flows in recent years. Based on this idea, we propose that an analytic solution of a one-dimensional Riemann problem can be applied to the mass flux in AUSMD without the addition of any extra dissipative terms to simulate the numerical flux

$$\begin{aligned}
&\overline{p}=\frac{1}{C_L+C_R}\left[C_R p_L+C_L p_R+C_L C_R(u_L-u_R)\right],\\
&\overline{u}=\frac{1}{C_L+C_R}\left[C_L u_L+C_R u_R+(p_L-p_R)\right],\\
&\overline{\rho}_L=\rho_L+(\overline{p}-\rho_L)/a_L^2,\\
&\overline{\rho}_R=\rho_R+(\overline{p}-\rho_R)/a_R^2.
\end{aligned}\tag{2.50}$$

Notice that in this linearized solution we only need to specify constant values for $\overline{\rho}$ and $\overline{a}$. Selecting some average of the data values ρ_L,ρ_R,a_L,a_R appears sensible

$$\overline{\rho}=\frac{1}{2}(\overline{\rho}_L-\overline{\rho}_R),\quad \overline{a}=\frac{1}{2}(a_L-a_R).\tag{2.51}$$

In this approximation, we do not need to make a choice for the averages $\overline{\rho}$ and $\overline{a}$; their values are replaced by data values at the foot of the corresponding characteristics. If $C_L=C_R=\overline{\rho a}$, the approximations for the primitive variables on the both sides of the material interface becomes

$$\begin{aligned}
&p^*=\frac{1}{2}[p_L+p_R+(u_L-u_R)/\overline{\rho a}],\\
&u^*=\frac{1}{2}[u_L+u_R+(p_L-p_R)/\overline{\rho a}],\\
&\rho_L^*=\rho_L+(p^*-p_L)/a_L^*,\\
&\rho_R^*=\rho_R+(p^*-p_R)/a_R^*.
\end{aligned}\tag{2.52}$$

Based on the original AUSMD and the approximated Riemann solution of star region, a hybrid flux splitting of the mass flux can be proposed to modify for the computation of flows around the wave interface as

$$(\rho u)_{\frac{1}{2}}=\begin{cases}
m_{1/2}^+, & \text{if } 0\le S_L,\\
\rho_L^* u^*, & \text{if } S_L\le 0\le S_*,\\
\rho_R^* u^*, & \text{if } S_*\le 0\le S_R,\\
m_{1/2}^-, & \text{if } 0\ge S_R,
\end{cases}\tag{2.53}$$

where

$$m_{1/2}^{\pm}=\frac{1}{2}(m_{1/2}\pm|m_{1/2}|).\tag{2.54}$$

And the pressure flux in hybrid flux splitting can be proposed by

$$(p_{\frac{1}{2}}=\begin{cases}
p_L, & \text{if } 0\le S_L,\\
p^*, & \text{if } S_L\le 0\le S_*,\\
p^*, & \text{if } S_*\le 0\le S_R,\\
p_R, & \text{if } 0\ge S_R,
\end{cases}\tag{2.55}$$

where the pressure at the star region can be represented by

$$p^*=\frac{1}{2}[P_L+P_R+\rho_L(S_L-u_L)(S_*-u_L)+\rho_R(S_R-u_R)(S_*-u_R)], \tag{2.56}$$

and

$$S^*=\left[\frac{P_L+P_R+\rho_L u_L(S_L-u_L)+\rho_R u_R(S_R-u_R)}{\rho_L(S_L-u_L)+\rho_R(S_R-u_R)}\right]. \tag{2.57}$$

The lower and upper bounds of the fastest signal velocities are calculated by

$$S_L=\min\{u_L-a_L,\widetilde{u}-\widetilde{a}\}, \tag{2.58}$$

and

$$S_R=\min\{u_R+a_R,\widetilde{u}+\widetilde{v}\}, \tag{2.59}$$

where a_L and a_R are the mixture sound speeds at the cell face

$$\widetilde{u}=\frac{\sqrt{\rho_L}u_L+\sqrt{\rho_R}u_R}{\sqrt{\rho_L}+\sqrt{\rho_R}}, \tag{2.60}$$

$$\widetilde{a}=\left[(\gamma-1)\left(\widetilde{H}-\frac{1}{2}\widetilde{u}^2\right)\right]^{1/2}, \tag{2.61}$$

$$\widetilde{H}=\frac{\sqrt{\rho_L}H_L+\sqrt{\rho_R}H_R}{\sqrt{\rho_L}+\sqrt{\rho_R}}. \tag{2.62}$$

2.3 Reinforcement learning

For optimization, the Q-learning Q-table learning method in reinforcement learning was used to optimize the parameters α and β of the ATM1 to ATM3 in order to improve the numerical accuracy of the ATM methods. The whole procedure is described in Fig. 1.

The concept of Q-learning Q-table was first proposed by Watkins [52] as a model-free reinforcement learning method. The model can only wait for feedback from the real-world step-by-step and then take the next action based on the feedback. Q-learning Q-table uses behavioral values as the basis for outputting the next decision It is a single-step update learning method, allowing for the update of behavioral criteria at each step of the process rather than waiting until the end.

The Q-table is a data structure that is used to store the values of the Q function of every possible state and action in the environment. The Q function represents the expected return (or reward) that an agent will receive for taking a given action in a given state. The Q-table is updated as the agent takes measures in the environment and receives rewards. The Bellman equation is used to update the values in the table. The Q-learning algorithm updates the Q-table by considering the expected rewards for taking different

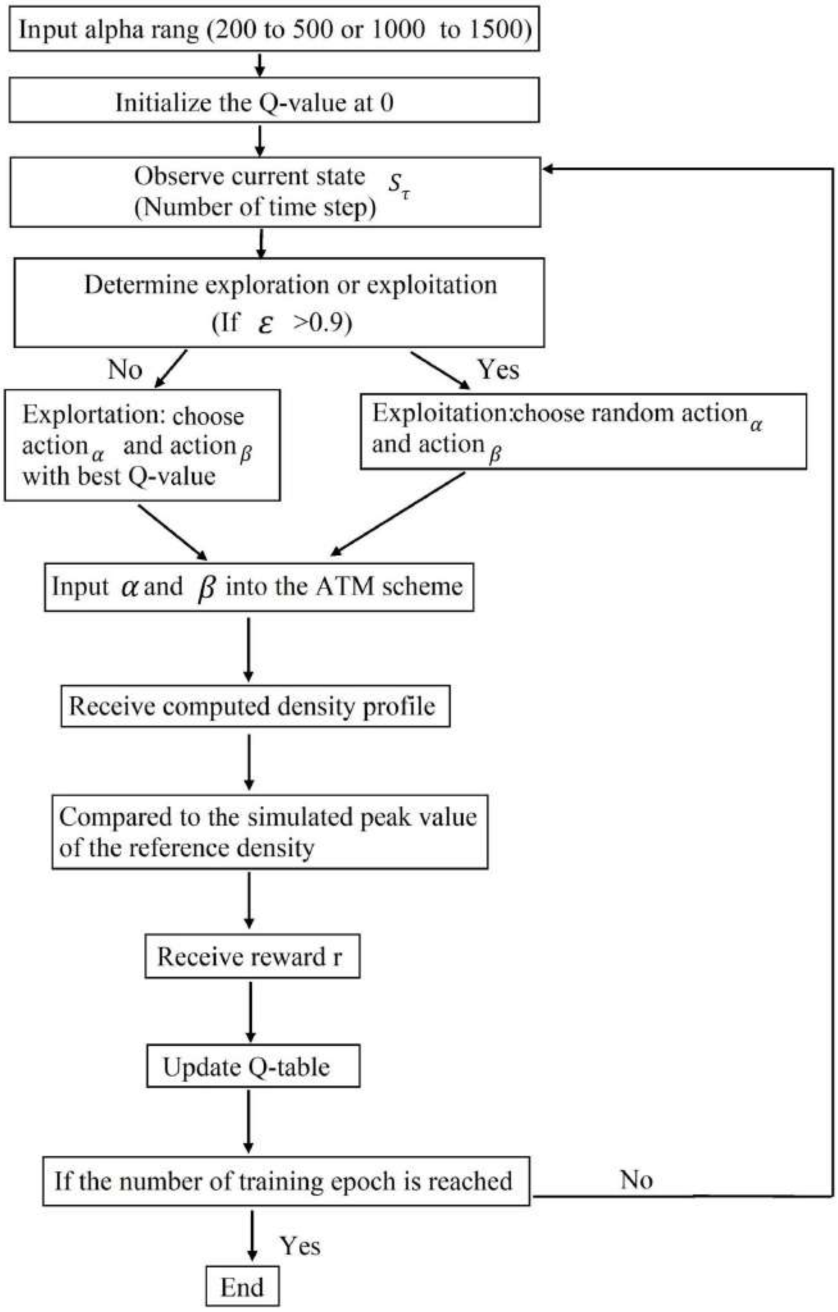

Input alpha rang (200 to 500 or 1000 to 1500)
Initialize the Q-value at 0
Observe current state s_τ
(Number of time step)
Determine exploration or exploitation
(If ε >0.9)
No
Yes
Explortation: choose action $_\alpha$ and action $_\beta$ with best Q-value
Exploitation:choose random action $_\alpha$ and action $_\beta$
Input α and β into the ATM scheme
Receive computed density profile
Compared to the simulated peak value of the reference density
Receive reward r
Update Q-table
If the number of training epoch is reached
No
Yes
End

Figure 1: The procedures of optimization flow chart.

actions in the current state, as well as the expected rewards for taking different actions in future states. As the agent takes actions and receives rewards, the values in the Q-table are updated, allowing the agent to learn which actions tend to lead to higher rewards in different states.

The necessary steps to converge the Q-value eventually to an optimal Q-value, Q, are as followed. For the given state-action pair, the Q-value should be as close to the right-hand side of the Bellman Optimality Equation as possible. For this, the loss between the Q-value and the optimal Q-value for the given state-action pair was compared iteratively. Each time the same state-action pair was encountered, the Q-value would be updated again to reduce the loss. The loss can be given as $Q(s_{\tau+1},a_{\tau+1})-Q(s_\tau,a_\tau)$

$$NewQ(s_\tau,a_\tau)\leftarrow Q(s_\tau,a_\tau)+\eta[r+\gamma\max Q(s_{\tau+1},a_{\tau+1})-Q(s_\tau,a_\tau)], \tag{2.63}$$

where $NewQ(s_\tau,a_\tau)$ is the new Q value for that state and the action, $\max Q(s_{\tau+1},a_{\tau+1})$ is the maximum expected future reward given the new state ($s_{\tau+1}$) and all possible actions ($a_{\tau+1}$) at that new state, $Q(s_\tau,a_\tau)$ is the current Q value, s_τm is the current state, $s_{\tau+1}$ is the new state, a_τ is the current action, $a_{\tau+1}$ is the new action, η is the learning rate, γ is the discount rate and r is the reward for taking that action at that state.

Eq. (2.63) shows that the Q-Learning Q-Table would receive a reward or penalty for every action taken in state s_τ. In each iteration, the Q-Learning Q-Table would select an action with the maximum Q-value at state, s_τ(exploitation), and then evaluate reward function when moving to the next state. In order to prevent the algorithm from being trapped inside a certain region, a greedy probability,ε was introduced. The greedy probability provides a probability to randomly choose an action from the action space which does not have the highest Q-value. Therefore, the agent may have a chance to explore a new environment (exploration). The newly computed Q-value could not be overwritten with an older value. Instead, a parameter called learning rate, as denoted by η, was used to determine the amount information from the previously computed Q-value for the given state-action pair had been retained over the newly computed Q-value calculated for the same state-action pair at a later time step. The range of the learning rate and discounting factor was between 0 and 1. The higher the learning rate, the more quickly the model adopted the newly computed Q-value. Therefore, the trade-off between new and old Q-value should be taken care of using the appropriate learning rate.

It is common to set the learning rate to a value between 0.01 and 0.1, the discount rate to a value between 0.8 and 0.95, and the greedy probability to a value between 0.1 and 1.0. The specific values to be chosen will depend on the particular problem and the desired behavior of the agent. In general, it is a good idea to try out different hyperparameter values to see how they affect the results. I. This study the learning rate, discount rate, and greedy probability, were set as 0.1, 0.9, and 0.9, respectively, and the number of time steps were declared as the states of the Q-learning algorithm. The actions in Q-learning algorithms were ones that enable the algorithm to do exploration within the states of Q-learning. Each action would lead the algorithm from the current state to the other states. The actions available for the developed Q-learning algorithm in this study were the $action_\alpha$ and the $action_\beta$. In the α range of 200-500, $action_\alpha$ was chosen an integer between 0 and 30, and then converted to the α parameter in Eq. (2.64). In addition, in the α range of 1000-1500, $action_\alpha$ is chosen as an integer between 0 and 50, and then converted to the α parameter in Eq. (2.65). At the same time, $action_\beta$ is selected as an integer between 0 and 100, and then converted to the β parameter in Eq. (2.66) as followed

$$\alpha = 200 + action_\alpha \times 10, \tag{2.64}$$

$$\alpha = 1000 + action_\alpha \times 10, \tag{2.65}$$

$$\beta = 1.2 + action_\beta \times 0.01. \tag{2.66}$$

Rewards were given to the actions when there is a small error in the simulated peak value of the density calculated by ATM as compared to the simulated peak value of the reference density. The optimization procedures can be seen clearly in the flow chart as shown in Fig. 1.

In this study, the error between the wave peak of the simulation results and the wave peak of the exact solutions was calculated. The value was used as the basis to judge the simulation results accuracy of each ATM method.

Initially, we attempted to guess the two factors using a range of values, which proved to be helpful for the Q-learning test. To avoid the local extremum problem, we implemented a condition to check for this case and handle it appropriately, such as by throwing an error to alert the user of an invalid input. It is important to anticipate and address these edge cases in order to ensure that the program functions correctly.

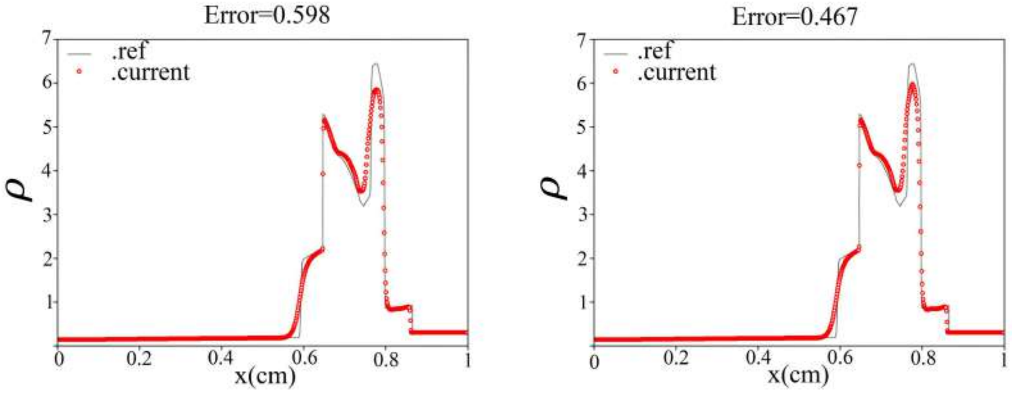
We employed an epsilon-greedy strategy, which is a simple method used in reinforcement learning to balance exploration and exploitation. This approach allows the agent interacting with the environment to occasionally choose a random action with a probability, known as epsilon, rather than the action that is believed to maximize the reward. The epsilon-greedy strategy is often used when the agent must learn about its environment and determine the optimal actions to maximize its reward. By introducing a small probability of random exploration, the agent can try out new actions and gather more information about the environment, potentially improving its performance over time. An error is a measure of the degree of inaccuracy on a measurement. The error in this essay is defined as the average difference between the optimal value and the reference value.

3 Result and discussions

3.1 Two interacting blast waves problem

This is a 1D benchmark test including more complicated interactions and discontinuous structures. Initially we set the following condition for state variables,

$$(\rho,u,p) = \begin{cases} (1,0,1000), & 0 < x \le 0.1, \\ (1,0,0.01), & 0.1 < x \le 0.9, \\ (1,0,100), & 0.9 < x \le 1.0. \end{cases}$$



(a)

(b)

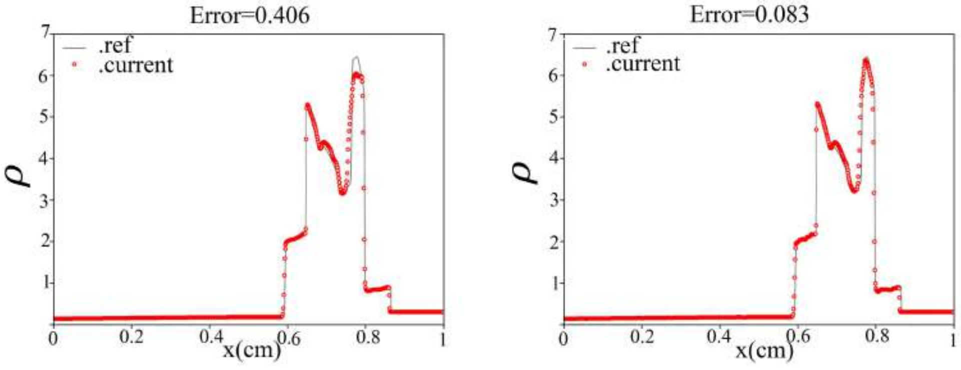
Figure 2: Numerical results by the MUSCL on 800 grid points. (a) Primitive variables; (b) Conservative variables.

The reflection boundary is set to the left and right ends of computational domain. This test involves the multiple interaction of strong shock waves, density discontinuities and rarefaction waves. A shock refraction problem describes a gas flow with three constant initial conditions. It is known that two strong blast waves developed and collided, and then produced complex interactions of the shock wave, rarefaction, and contact discontinuity. Therefore, the blast wave collision problem is often selected to verify the resolution of the peak of density distribution profile.

In this case, an investigation is conducted to assess the accuracy of three ATM schemes (ATM1, ATM2, and ATM3) in the computation of the flow behaviors near wave interfaces. The results of density containing exact solutions are depicted in Fig. 2. The solid line represents the exact solutions obtained through the MUSCL method, computed on 12800 meshes. The computed solutions are performed on uniformly grids with 800 cells. The circles represent the ATM solutions. A time step of $\Delta t/\Delta x = 0.005$ is chosen to ensure numerical stability. We first show the results of the MUSCL values calculated from the primitive variables and conservative variables respectively.

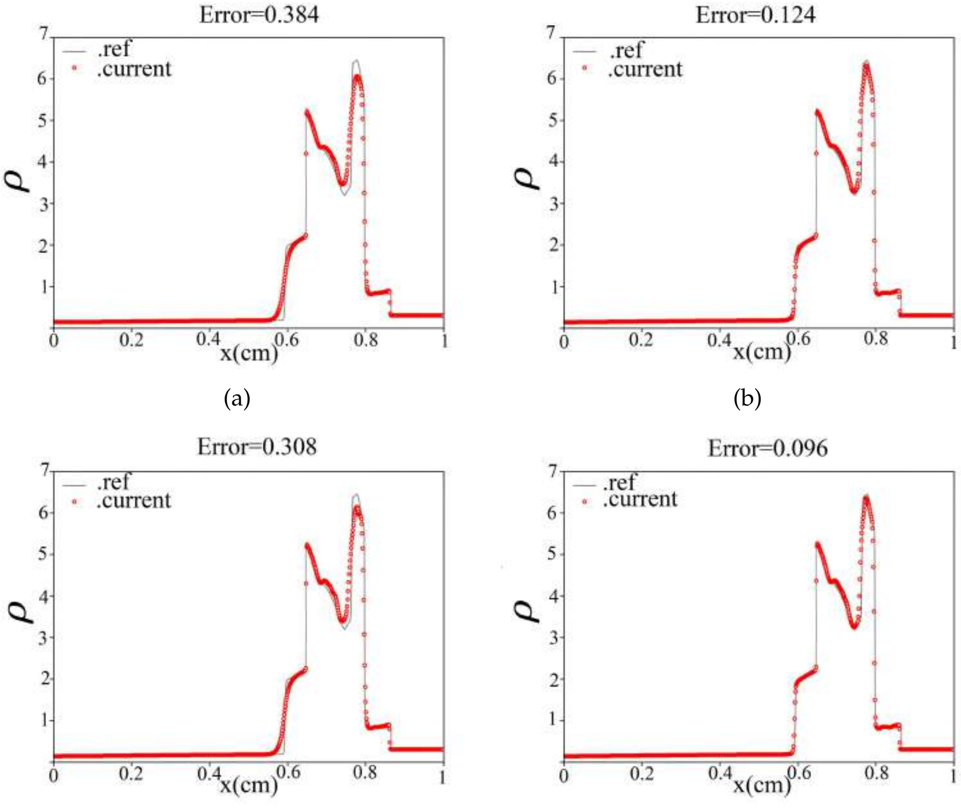
Next, we present the results of the THINC-EM method using a value of $\beta = 1.25$ in Fig. 3. We have improved upon the method by combining it with a flux splitting technique and the MUSCL + THINC-EM cell interface variable reconstruction method. This improved method, referred to as the ATM method, has been demonstrated to accurately resolve discontinuous solutions such as interfaces and rarefactions, as well as shock waves, with significantly less numerical dissipation. The results of three different ATM schemes are shown in Figs. 4, 5, and 6. First, we fix the $\alpha = 500$ and 1500, then fix the $\beta = 1.25$ for all ATM method in sequence as follows.

We present the numerical results of ATM1 after optimization in Fig. 4, which are compared with the exact solutions. The results agree with the exact solutions with an error of 0.096 in conservative variables when $\alpha = 1500$, $\beta = 1.25$ (as shown in Fig. (d)). The numer-



(a) (b)

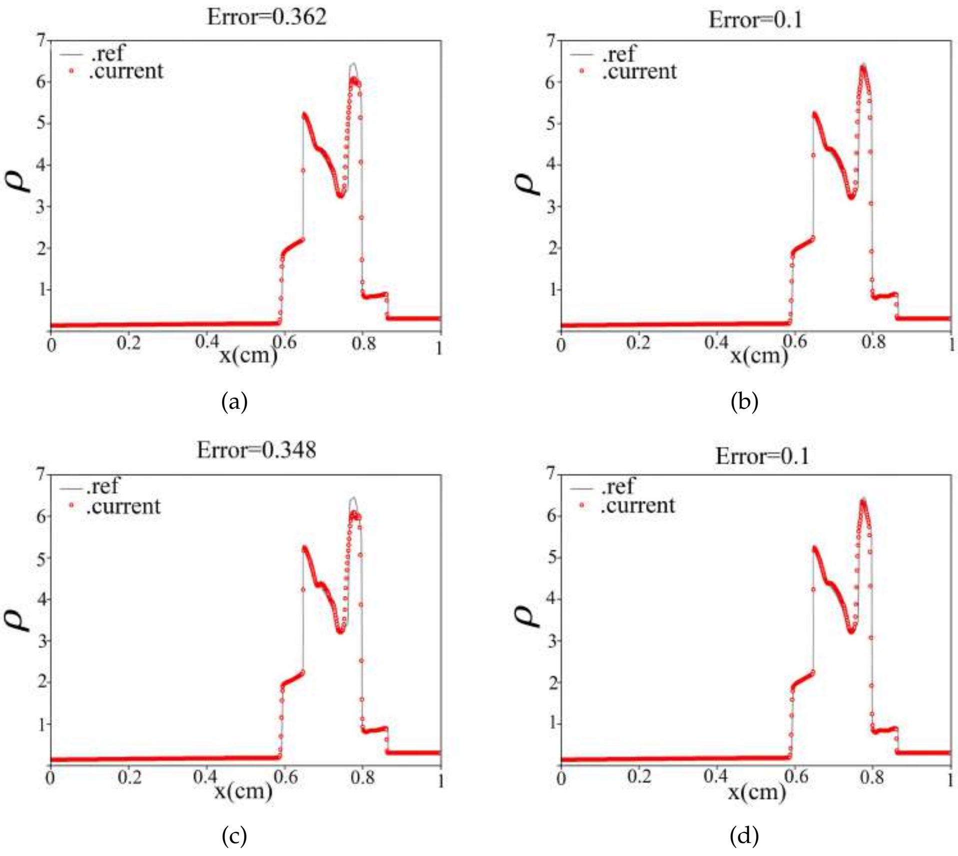
Figure 3: Numerical results by the THINC-EM use the $\beta=1.25$ on 800 grid points. (a) Primitive variables; (b) Conservative variables.



(a) (b)

(c) (d)

Figure 4: Numerical results by the ATM1 on 800 grid points. (a) Primitive variables, $\alpha=500$, $\beta=1.25$; (b) Conservative variables, $\alpha=500$, $\beta=1.25$; (c) Primitive variables, $\alpha=1500$, $\beta=1.25$; (d) Conservative variables, $\alpha=1500$, $\beta=1.25$.



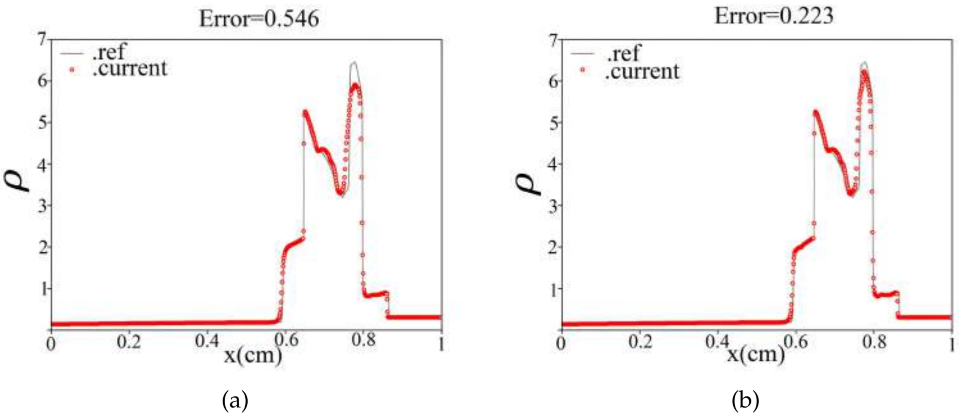
(a)

(b)

(c)

(d)

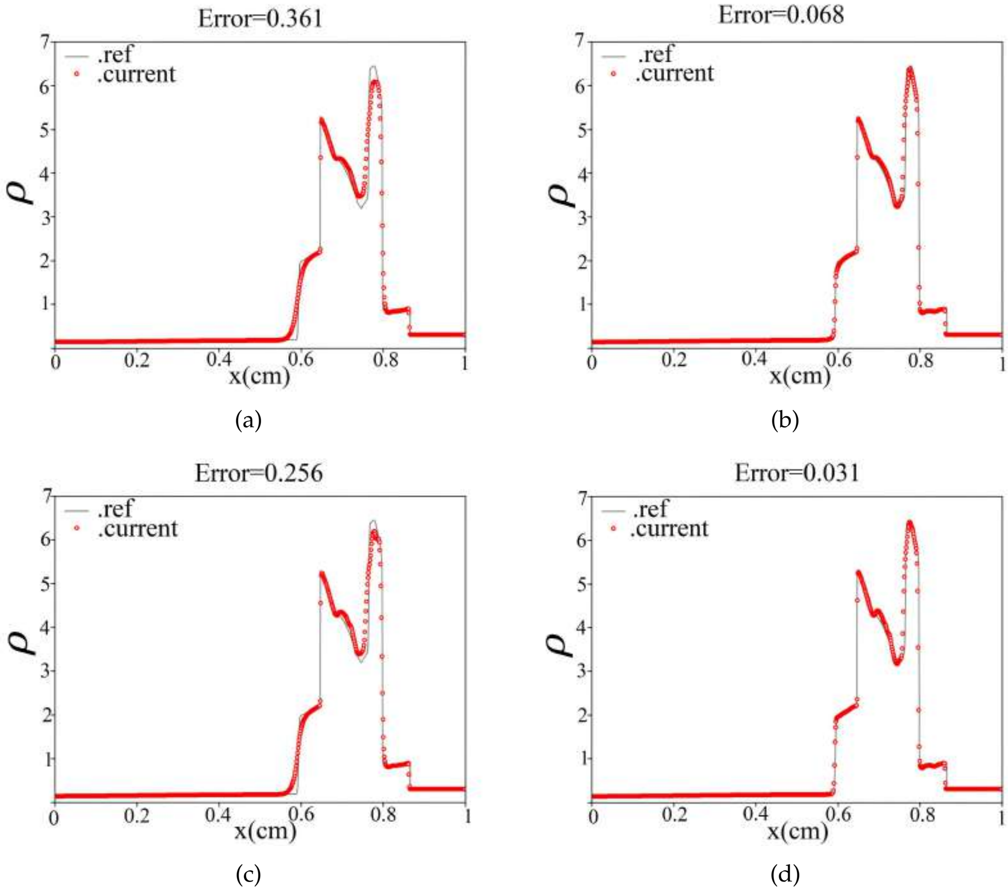
Figure 5: Numerical results by the ATM2 on 800 grid points. (a) Primitive variables, $\alpha=500$, $\beta=1.25$; (b) Conservative variables, $\alpha=500$, $\beta=1.25$; (c) Primitive variables, $\alpha=1500$, $\beta=1.25$; (d) Conservative variables, $\alpha=1500$, $\beta=1.25$.



(a)

(b)

Figure 6: Numerical results by the ATM3 use the $\beta=1.25$ on 800 grid points. (a) Primitive variables; (b) Conservative variables.



(a)

(b)

(c)

(d)

Figure 7: Numerical results of ATM1 after optimization by Q-Learning Q-table algorithm on 800 grid points. (a) Primitive variables, $\alpha=460$, $\beta=1.3$; (b) Conservative variables, $\alpha=500$, $\beta=1.3$; (c) Primitive variables, $\alpha=1160$, $\beta=1.3$; (d) Conservative variables, $\alpha=1360$, $\beta=1.3$.

ical results of ATM2 after optimization are also presented in Fig. 5, which are compared with the results of the exact solutions. The results agree with the exact solutions with an error of 0.1 in conservative variables when $\alpha=1500$ and $\beta=1.25$ (as shown in Fig. 5(d)). Finally, we show the comparison of the numerical results of the ATM3 optimization with the results of the exact solution in Fig. 3.

As can be seen in Figs. 7-10, ATM1 and ATM2 show the great results. In addition, ATM2 exhibits the most precise capture of the peak of the density profile and the lowest error factor in both primitive and conservative variables among all of the selected ATM methods. Furthermore, the density distribution along the rarefaction is well resolved using approximately four grid points in all versions of the ATM method. Overall, the resolutions of ATM1 and ATM2 are nearly identical when $\alpha=500$ and $\beta==1.25$. ATM2 exhibits particularly sharp capturing of the peak of the density profile and achieves a

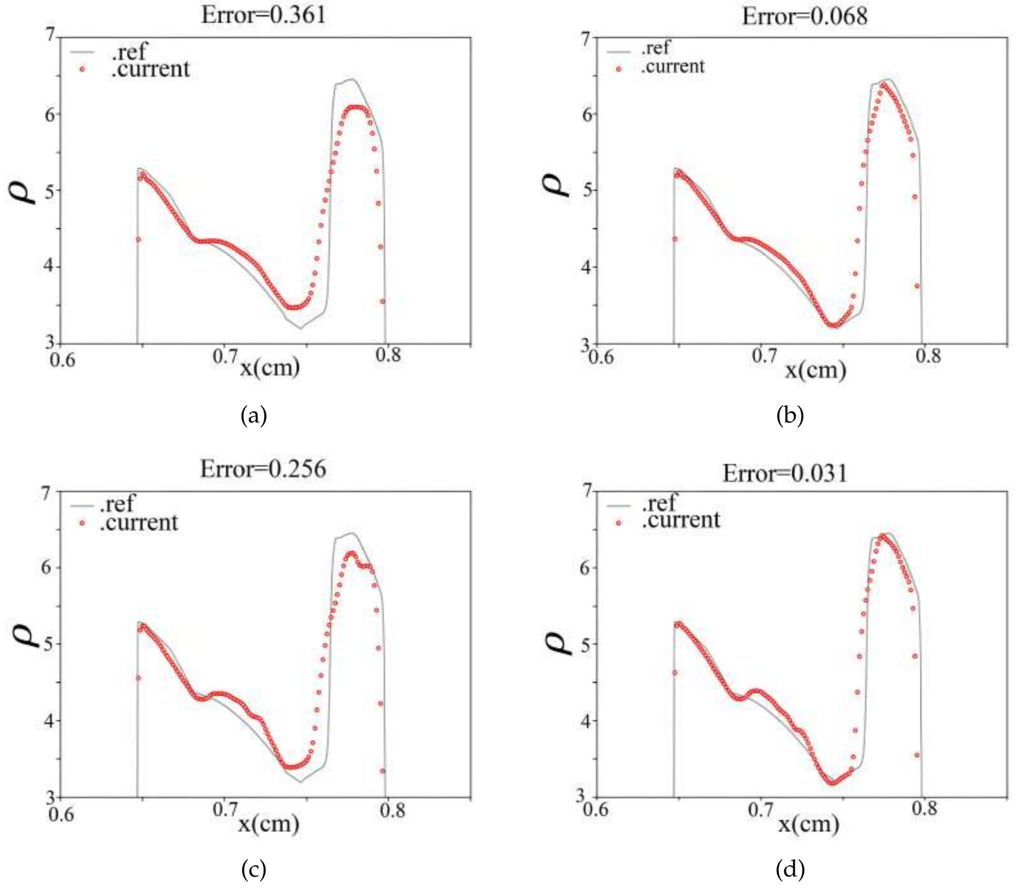
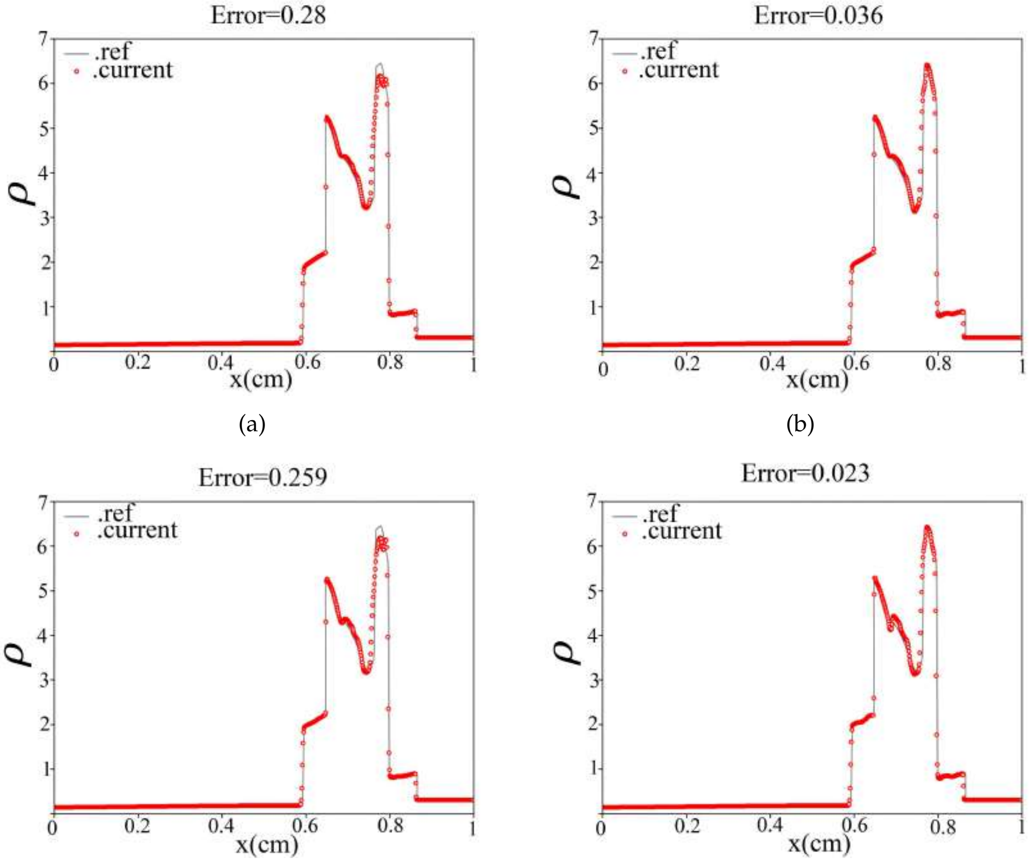


Figure 8: The enlarged views of ATM1 after optimization by Q-Learning Q-table algorithm on 800 grid points. (a) Primitive variables, $\alpha=460$, $\beta=1.3$; (b) Conservative variables, $\alpha=500$, $\beta=1.3$; (c) Primitive variables, $\alpha=1160$, $\beta=1.3$; (d) Conservative variables, $\alpha=1360$, $\beta=1.3$.

lower error than the other versions of the ATM. All three ATM methods demonstrate that the density distribution along the rarefaction can be accurately resolved using fewer grid points compared to the reference scheme using MUSCL.

Subsequently, the Q-learning of the enforcing learning is performed to get the optimized combination of the α and β from the $\alpha=460$ to 500 or $\alpha=1010$ to 1500 and $\beta=1.2$ to 1.3 for the ATM methods. In Figs. 7-8, it is shown that the ATM1 method with $\alpha=1360$ and $\beta=1.3$ demonstrates the most accurate in the case of calculating one-dimensional blast wave. On the other hand, the ATM2 method with $\alpha=1430$ and $\beta=1.3$ demonstrates the most accurate in the case of calculating one-dimensional blast wave case between the in Figs. 9-10. With regard to ATM3 in Figs. 11-12, the error factor of primitive variables conservative variables is both higher than ATM1 and ATM2. Overall, choosing $\alpha=1430$ and $\beta=1.299$ could be the optimized for the ATM2 method to calculate one-dimensional



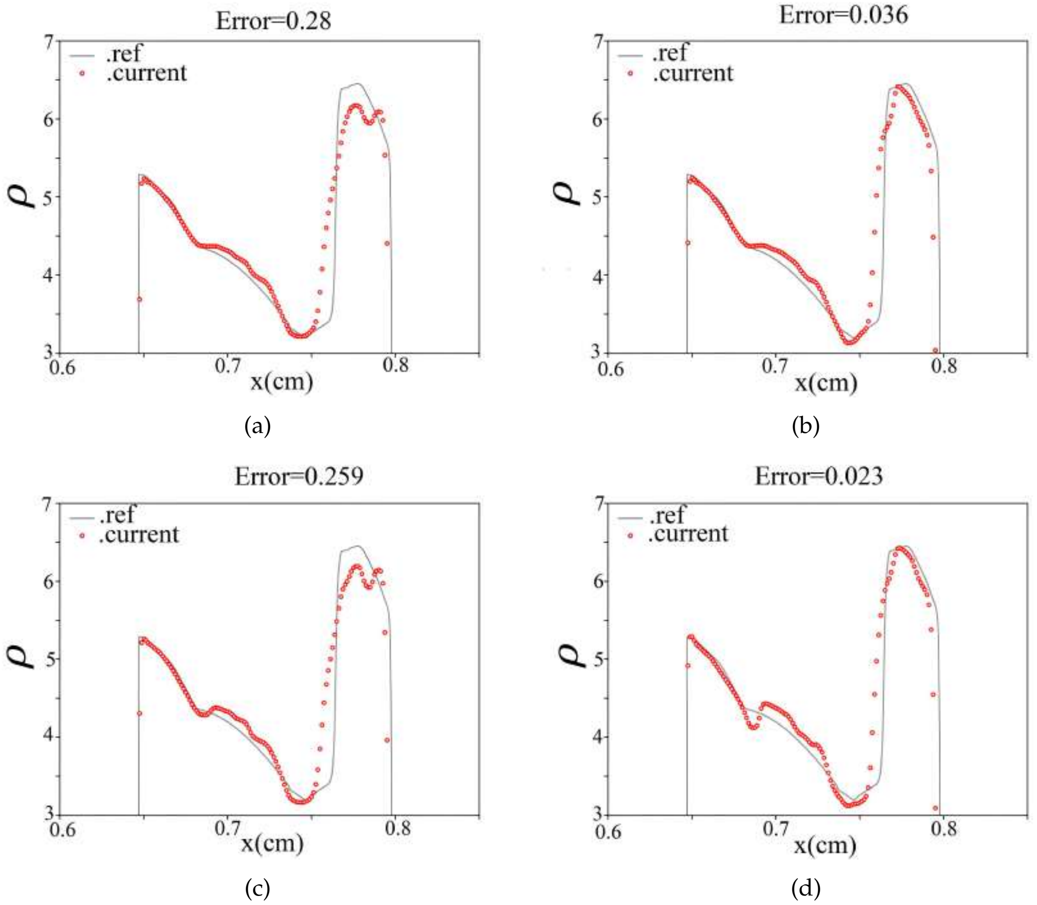
(a) (b)

(c) (d)

Figure 9: Numerical results of ATM2 after optimization by Q-Learning Q-table algorithm on 800 grid points. (a) Primitive variables, $\alpha=480$, $\beta=1.299$; (b) Conservative variables, $\alpha=480$, $\beta=1.298$; (c) Primitive variables, $\alpha=1010$, $\beta=1.299$; (d) Conservative variables, $\alpha=1430$, $\beta=1.299$.

blast wave case. To perform further verification of the ATM2 with $\alpha=1430$ and $\beta=1.3$, a grid independence study is using 400, 800, 1600, and 3200 grid points. In Fig. 13, it is seen that the results on 1600 grids is identical to the results of the 3200 grids, so we can understand that the required grid size about this model is 0.0.00625cm.

Upon comparing the error factors of the three ATM schemes (ATM1, ATM2, and ATM3), we found that ATM3 struggled to accurately capture discontinuous planes. In contrast, both ATM1 and ATM2 demonstrated better results with minimal error compared to ATM3. Moreover, since ATM2 with $\alpha=1430$, $\beta=1.299$ shows the best ability to capture discontinuities in the form of weak solutions for hyperbolic equations using high-resolution algorithms for conservation, we determined that ATM2 had the lowest error rate. As a result, we used ATM2 for case2 and case3.



(a)

(b)

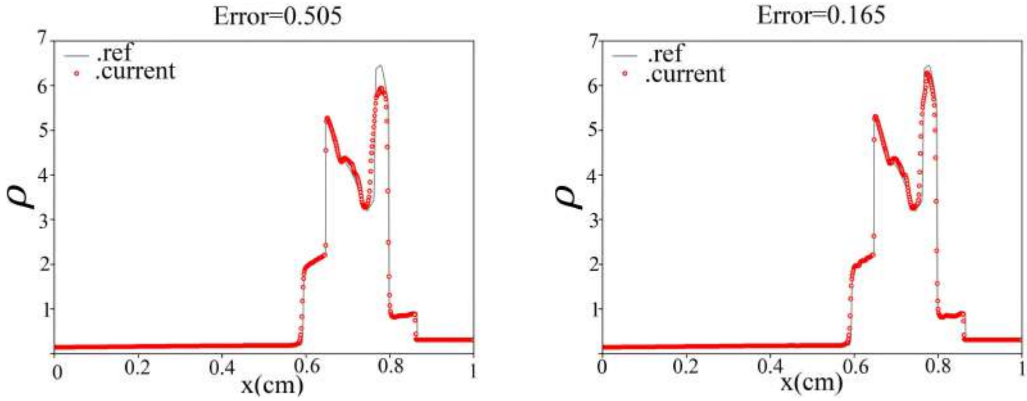
(c)

(d)

Figure 10: The enlarged views of ATM2 after optimization by Q-Learning Q-table algorithm on 800 grid points. (a) Primitive variables, $\alpha=480$, $\beta=1.299$; (b) Conservative variables, $\alpha=480$, $\beta=1.298$; (c) Primitive variables, $\alpha=1010$, $\beta=1.299$; (d) Conservative variables, $\alpha=1430$, $\beta=1.299$.

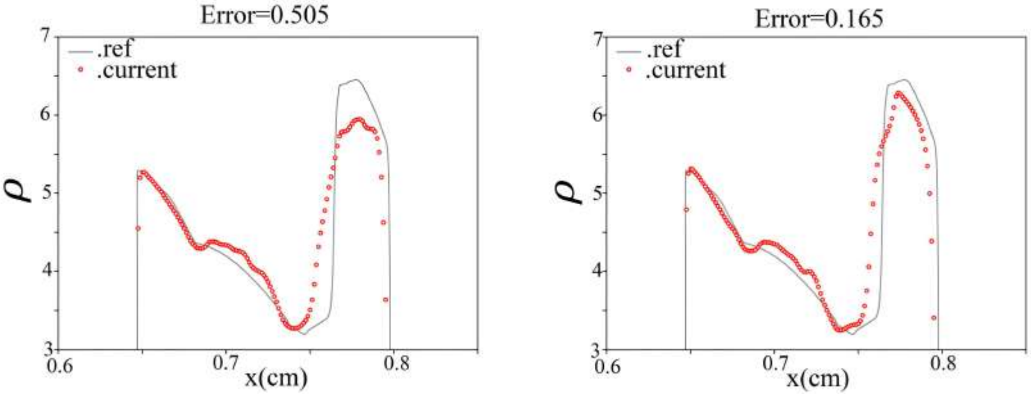
3.2 Two-dimensional detonation waves

Next, we consider the two-dimensional detonation wave problem, which has also been investigated by [17]. In this experiment case [15], a triangular detonation wave is set up within a computational domain of $[0,0.025]\times[0,0.005]$. The left side of the detonation wave is completely burned, while the right side is completely unburned. The upper and lower boundaries are assigned reflective boundary conditions, while the left and right boundaries are assigned zero-gradient boundary conditions. The main feature of this problem is that, due to its triangular shape, the central detonation wave will continue to move forward and the detonation wave of the upper half will hit the upper wall and bounce back. Similarly, the lower half of the detonation wave will hit the lower wall and bounce back, leading to the formation of two symmetrical vortices on the upper



(a) (b)

Figure 11: Numerical results of ATM3 after optimization by Q-Learning Q-table algorithm on 800 grid points. (a) Primitive variables, $\beta=1.278$; (b) Conservative variables, $\beta=1.291$.



(a) (b)

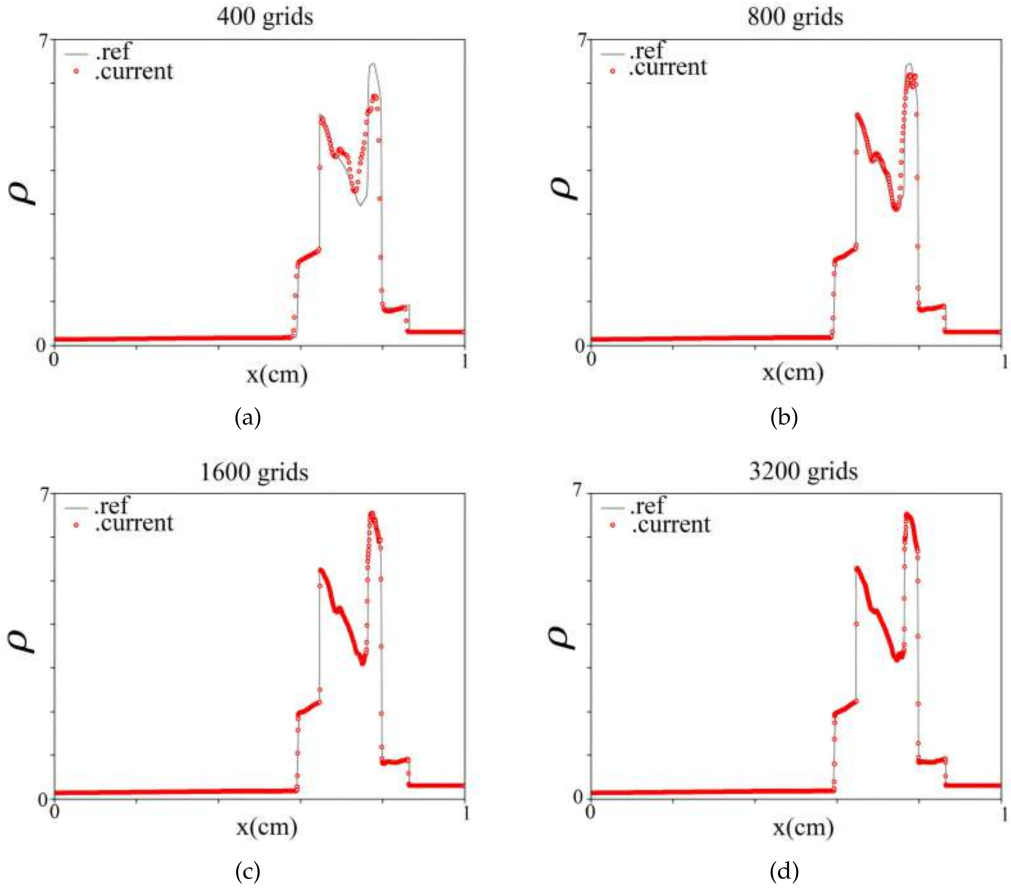
Figure 12: The enlarged views of ATM3 after optimization by Q-Learning Q-table algorithm on 800 grid points. (a) Primitive variables, $\beta=1.278$; (b) Conservative variables, $\beta=1.291$.

and lower sides. This makes the overall flow phenomenon very complicated. The state variables are shown in Fig. 14 as follows:

$$g(y)=\begin{cases} 0.004, & \text{if } |y-0.0025|\geq 0.001, \\ 0.005-|y-0.0025|, & \text{if } |y-0.0025|<0.001, \end{cases}$$

where

$$u_l=8.162\times 10^4>u_{CJ}, \quad \rho_l=\rho_{CJ}, \quad p_l=p_{CJ},$$



(a) (b) (c) (d)

Figure 13: Computed density profile of the two interacting blast waves problem by the grid independence study. (a) 400 grid points; (b) 800 grid points; (c) 1600 grid points; (d) 3200 grid point.

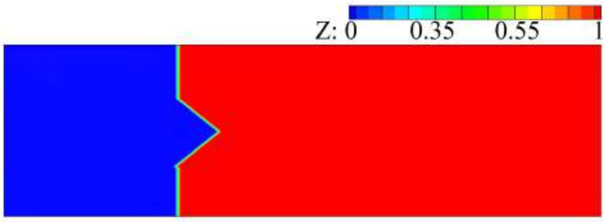


Figure 14: The initial state of 2D detonation problem.

and

$$(\rho,u,v,p,z)=\begin{cases} \left(1.945\times 10^{-3},\, 8.162\times 10^{4},\, 0,\, 6.27\times 10^{6},\, 0\right), & \text{if } x\le g(y), \\ \left(1.201\times 10^{-3},\, 0,\, 0,\, 8.321\times 10^{5},\, 1\right), & \text{if } x>g(y). \end{cases}$$

The parameters γ, K, q_0, T_0 in the Heaviside chemical reaction model can be found in [15]. We use the CGS units as [15]. The CFL number is set to 0.1. To provide a better comparison, the reference solution is calculated using the chosen ATM2 type AUSMD with $\alpha = 1430$, $\beta = 1.3$ scheme with 1600×320 grids. The density profiles at different times are presented in Fig. 15, with the reference solutions shown at the top. The solutions obtained using the MUSCL method, the ATM2 method, and the THINC method with 800×160 grid cells are shown below, respectively. The ATM2 method with $\alpha = 1430$ and $\beta = 1.3$ is able to produce a sharp detonation front and reflected waves similar to the reference solution on a coarse mesh, as shown in Fig. 15. However, the MUSCL method and THINC schemes produce spurious waves in front of the shock, and the THINC methods produce strong oscillations. The numerical results demonstrate that the proposed Q-learning approach can help the ATM2 scheme with suitable parameters to achieve sharp detonation waves with less numerical dissipation on coarse grids.

Due to the inherent blurriness of the internal reflected wave, it is challenging to accurately determine its degree of distortion. As such, we compare the density differences between the simulated values and the exact values at the centerline in Fig. 16 in order to evaluate the performance of difference methods.

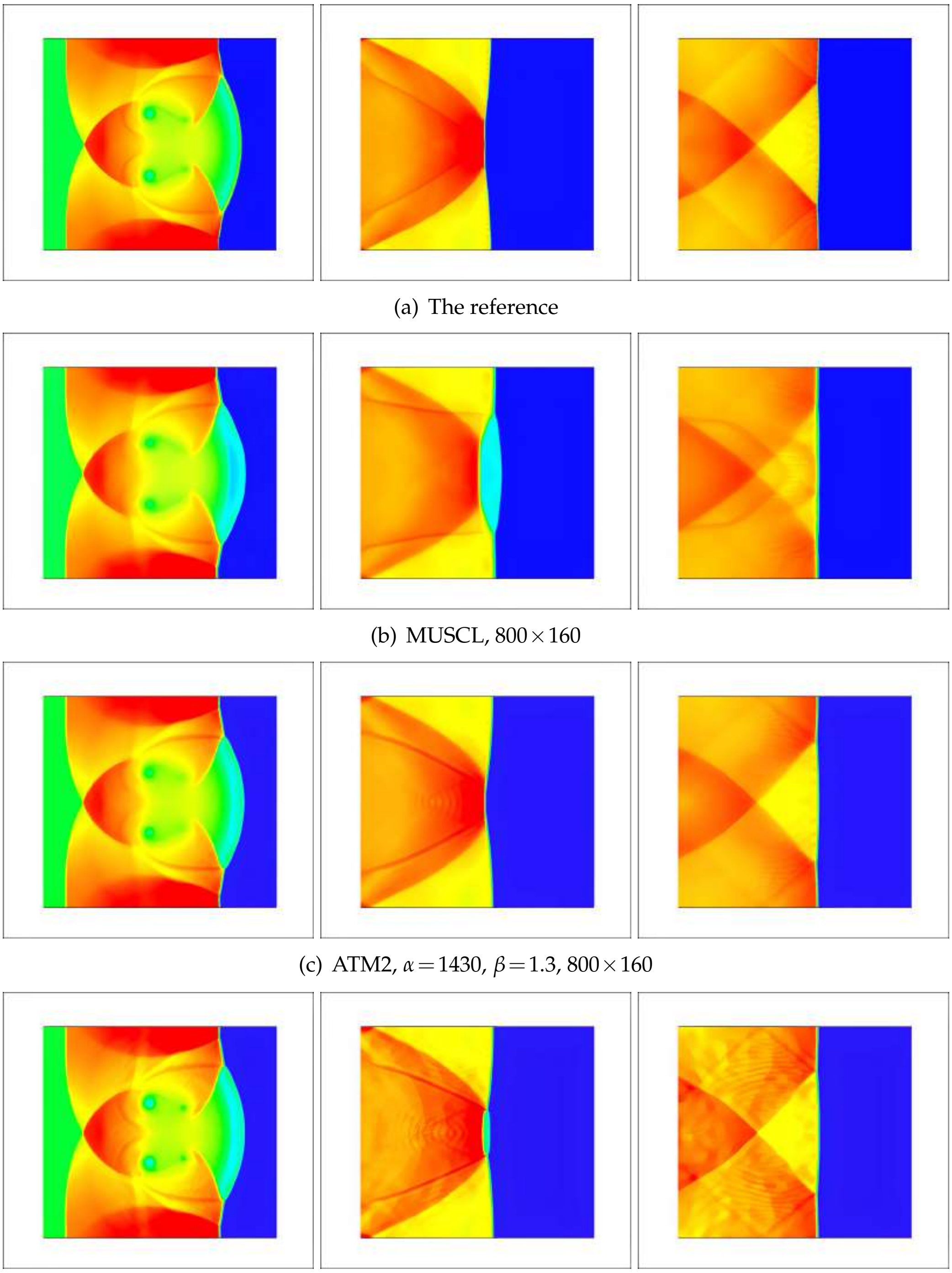
Upon examining Fig. 16, we can observe that the width of the virtual wave, which is generated by the MUSCL method at the wavefront, is quite thick. In contrast, the THINC method creates less virtual error at the wavefront, but it does produce numerical oscillations. Based on these results, we carefully select the appropriate parameter combination to use in the next case simulation.

3.3 Single-nozzle tube case

In the third case, we present the numerical simulations of single-tube PDE with a CD Nozzle, which has been invested by Fuhua Ma et al. in [1]. The configuration of the detonation tube is presented in Fig. 17. The computational domain is discretized into 170,000 grids. Initially, the detonation tube is filled with a stoichiometric hydrogen/air mixture at the ambient pressure (0.29 atm) and temperature (228 K). The boundary condition at the head end of the detonation tube are specified according to the engine operation. During the purging stage, the total temperature and total pressure are specified as 428 K and 2.12 atm, respectively. The axial velocity is extrapolated from the interior points, and the reactant mass fraction is set to zero. During the refilling, the parameters are the same, except that the reactant mass fraction is set to unity. A non-reflecting boundary condition is implemented along the open boundary of the external region. Detonation is initiated by a small amount of driver gas at 2000 K and 30 atm near the head end. This region spans a small length of 0.02 cm in order to minimize its effect on the assessment of the engine propulsive performance. For more details on the operation setting, see [1].

The parameters γ, K, q_0, T_0, in the Heaviside chemical reaction model are the same as mentioned in previous section, and the atmosphere data are as follow:

$$\rho_{atm} = 1.29 \times 10^{-3} (g/cm^3), \quad p_{atm} = 1.01325 \times 10^6 \ (bar).$$



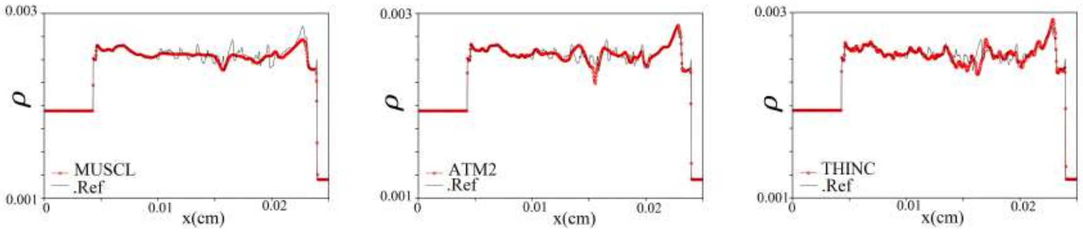
(a) The reference

(b) MUSCL, 800×160

(c) ATM2, $\alpha=1430$, $\beta=1.3$, 800×160

(d) THINC, $\beta=1.3$, 800×160

Figure 15: Density of 2D detonation problem. (Left column) time$=3\times 10^{-7}$ s; (middle column) time$=0.92\times 10^{-7}$ s; (right column) time$=1.7\times 10^{-7}$ s.



(a) MUSCL (b) ATM2, $\alpha=1430$, $\beta=1.3$ (c) THINC, $\beta=1.3$

Figure 16: Density field along the central line of x-direction at time$=1.7\times10^{-7}$ s.

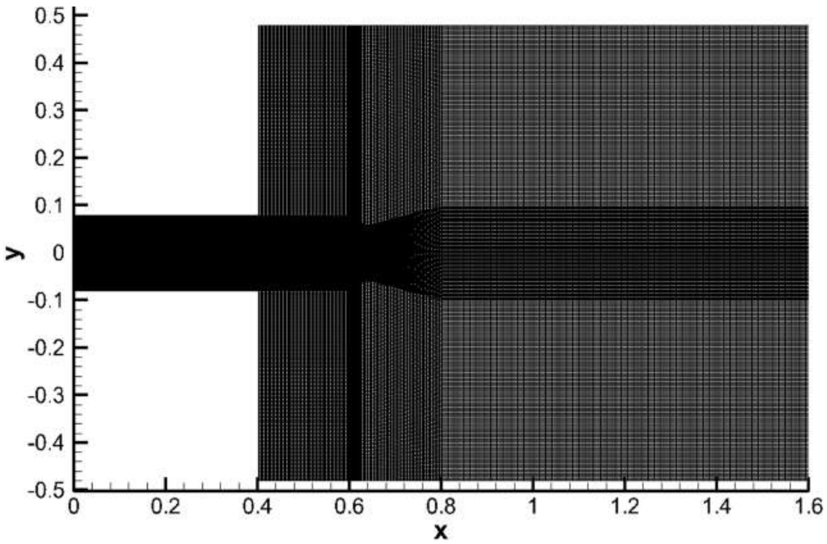
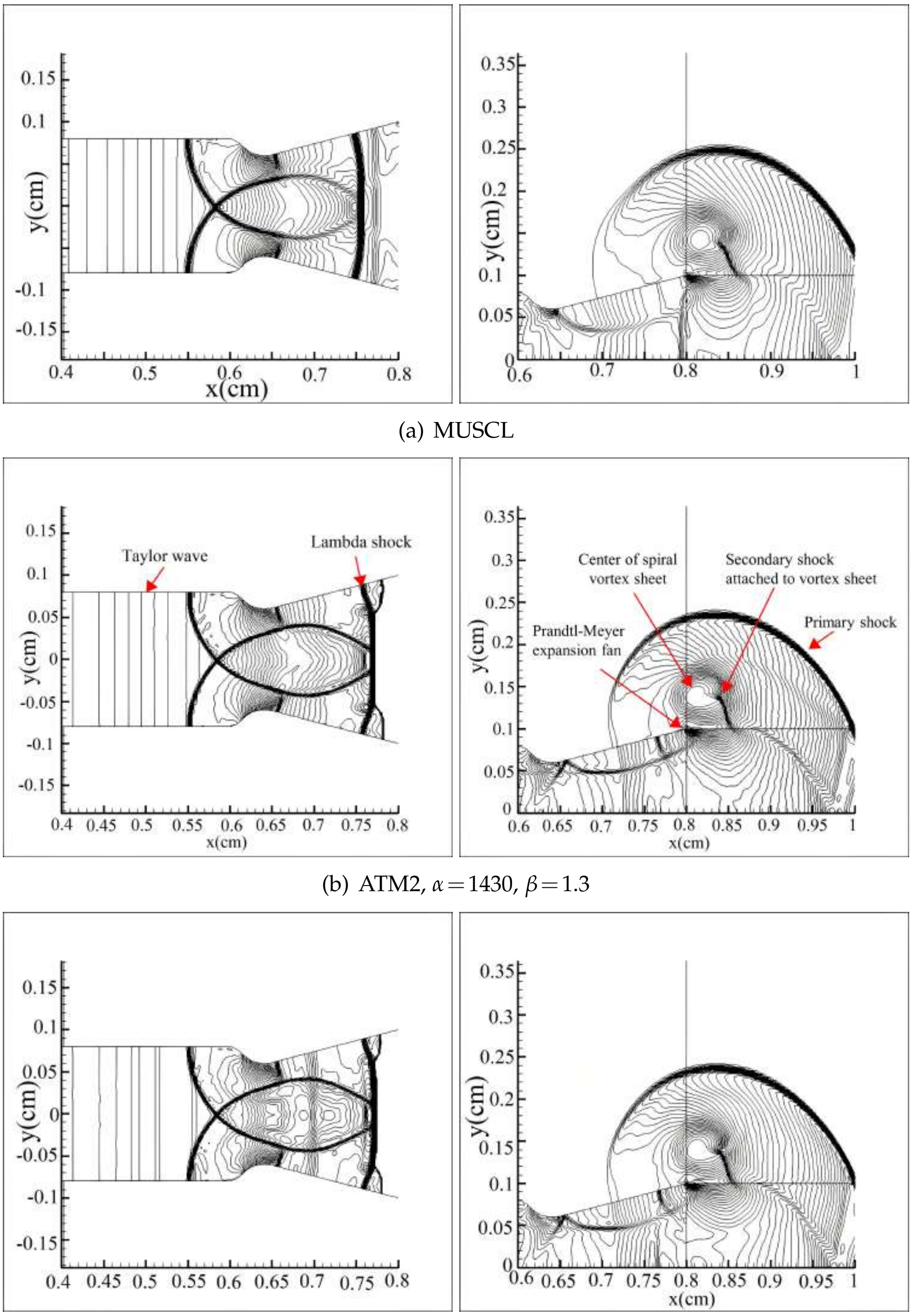


Figure 17: The configuration of the detonation tube.

The time evolution of the density-gradient field during one cycle of operation of the single-tube PDE is depicted in Figs. 18 and 19. Two time points, $6.5\mu s$ and $10.4\mu s$, have been selected to present enlarged pressure maps generated by the MUSCL and ATM2 methods. The primary shockwave, Taylor wave, and secondary shockwave emanating from the wall are shown in Fig. 16, along with the Prandtl-Meyer expansion fan and the center of the spiral vortex sheet.

In Fig. 18(a), numerical results computed by the MUSCL scheme with 170,000 grids, but physical phenomenon are not clear as Fig. 18(b), which computed by the ATM scheme ($\alpha=1430$, $\beta=1.3$) with 85,000 grids.

The primary shockwave, after being generated, travels downstream and reflects off the nozzle walls, creating a complex flow structure. Common features of this flow include the expanding primary shock, shear layers, a Prandtl–Meyer expansion fan that



(a) MUSCL

(b) ATM2, $\alpha=1430$, $\beta=1.3$

(c) THINC, $\beta=1.3$

Figure 18: The enlarged views of pressure contours at time$=6.5\mu$s (left column) and at time$=10.4\mu$s (right column).

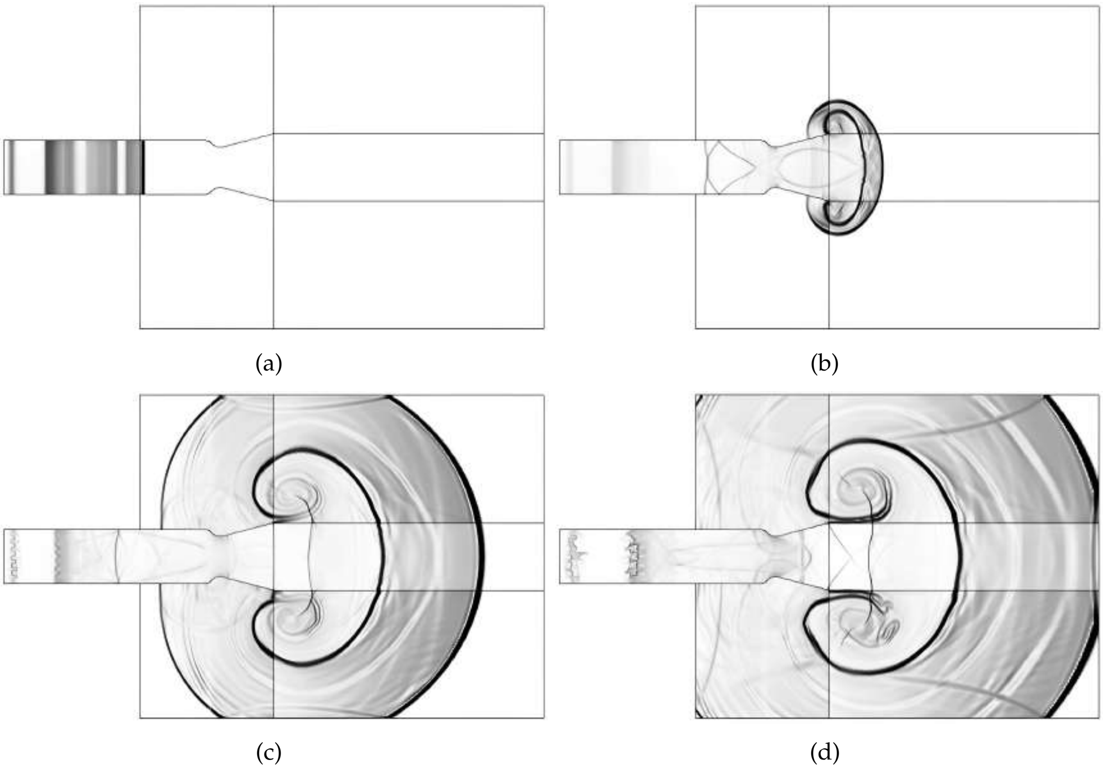


Figure 19: Time evolution of density-gradient field during one cycle of operation of single-tube PDE of ATM2, $\alpha = 1430$, $\beta == 1.3$.

originates from the edge of the nozzle exit at the beginning of the blowdown phase and oblique or normal shock waves in the later stages of the blowdown process, and numerous reflected shock waves. As seen in Figs. 18 and 19, the ATM method with the two parameters calculated through the Q-Learning method is able to simulate the results while incorporating the advantages of both the MUSCL method and the THINC method. This suggests that the combination of the ATM method and the Q-Learning method can be highly effective under the conditions of a coarse grid for various topics in the future. The selection of appropriate parameters can be adjusted for different problems and simulations to achieve accurate results.

3.4 Three-tubes PDE with CD nozzle

In this case, the three-tubes PDE system is configured as shown in Fig. 20. The computational domain is discretized into 170,000 grid cells. Initially, the bottom tube is partially filled with a quiescent stoichiometric hydrogen/air mixture at ambient pressure (0.29 atm) and temperature (228 K), while the rest of the region is filled with ambient air. The

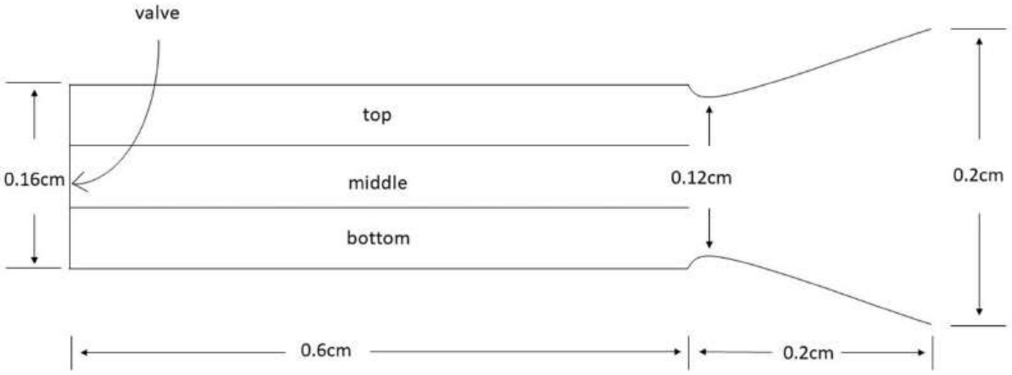
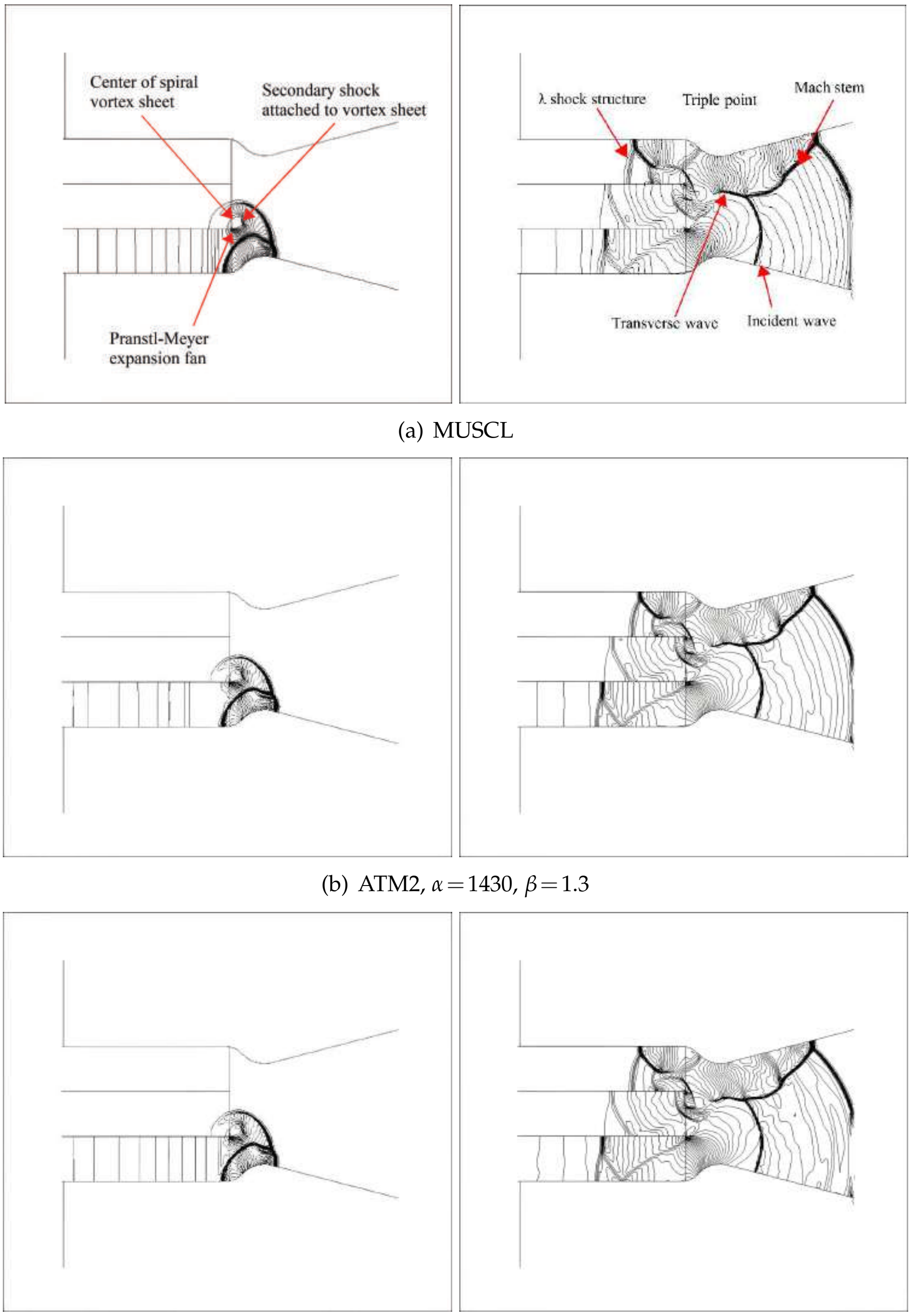


Figure 20: Configuration of three-tubes PDE.

operation cycle is the same as in the two-tubes PDE, and the weighting parameters used are $\alpha = 1430$ and $\beta = 1.3$.

At $t = 0.225$ ms, the detonation wave in the bottom tube reaches the reactant/air interface located 38 cm from the head end and degenerates into a non-reacting shock wave, known as the primary shock wave. At the same time, a series of expansion waves are generated at the interface and propagate downstream along with the Taylor wave to the tube exit and upstream to the head end. These expansion waves interact with and pass through the Taylor wave, reducing the length of the uniform region. As the expansion waves reflect off the head end, another series of expansion wave form and propagate downstream toward the tube exit, further reducing the pressure in the bottom tube. The primary shock wave reaches the bottom tube and diffracts at the tube exit, reflecting off the nozzle walls and causing complex waves to propagate upstream into all three tubes and downstream into the nozzle, as shown in the snapshot of t=0.75 ms in Fig. 21(b). The middle tube begins the ignition process at this time. At $t = 0.975$ ms, the primary shock wave has emerged from the nozzle into the external region, creating vortices near the edges of the nozzle exit. The external flow field is similar to that of the single-tube case, except for an asymmetrical pattern. At $t = 1.5$ ms, the top tube begins the ignition process, while the middle and bottom tubes are in the blowdown process. At t=1.725 ms, the detonation wave in the top tube interacts with the detonation waves in the bottom tube, making the flow field even more complex. At $t = 2.25$ ms, the shock wave has moved out of the nozzle to interact with the local flow field in the external region. We can also observe upstream-traveling reflected shock waves in all three tubes. At $t = 3.0$ ms, the end of the first cycle, the bottom tube is in the ignition process and the top and middle tubes are in the blowdown process.

To demonstrate the capability of the proposed method, a close-up view of the flow developments and the flow interactions among the tubes and nozzle are presented in Figs. 21 and 22. It is clear that the proposed methods can capture the flow structures in the detonation tube. In Fig. 21, the diffracted and reflected shock waves have propagated

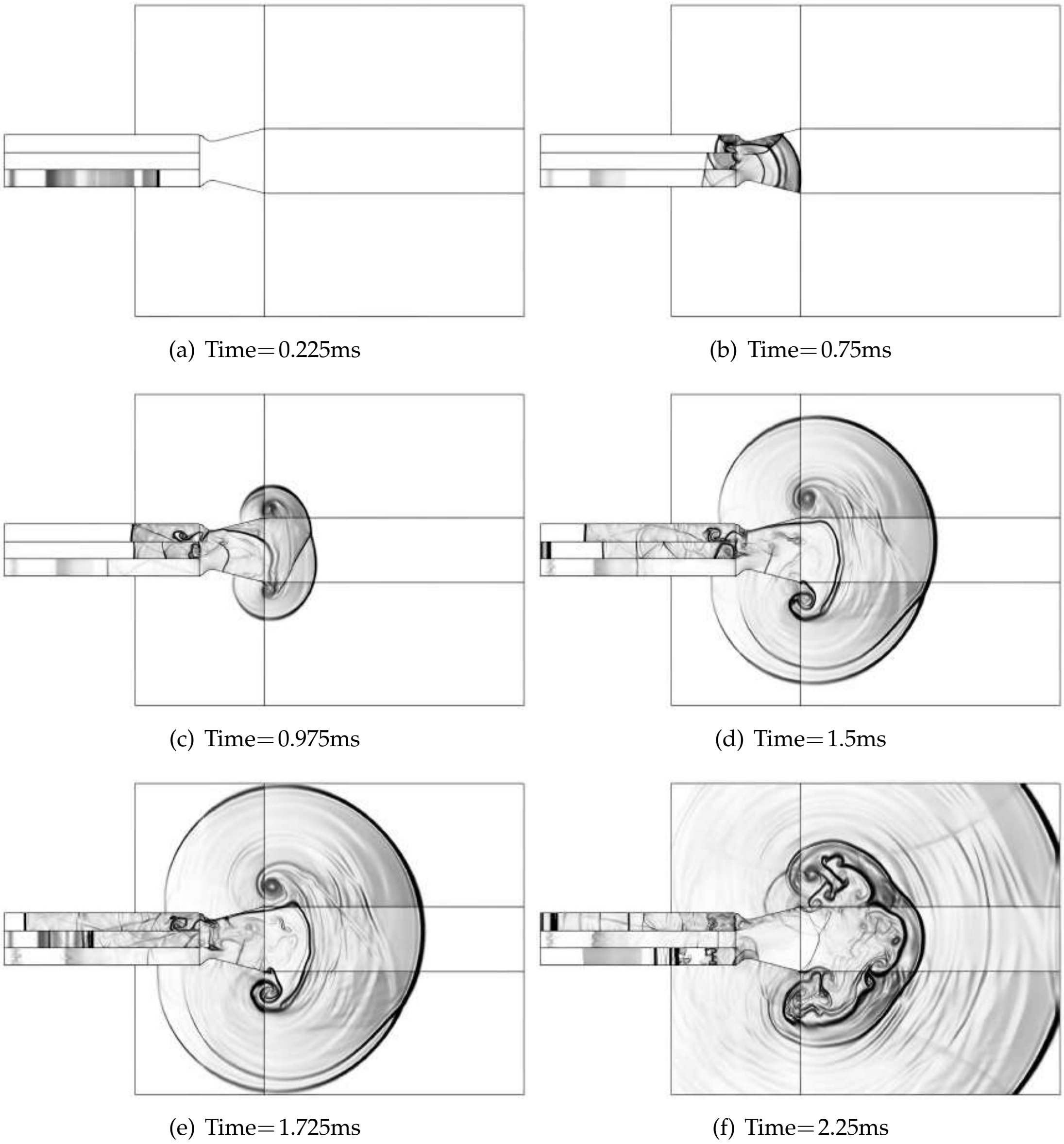


(a) MUSCL

(b) ATM2, $\alpha = 1430$, $\beta = 1.3$

(c) THINC, $\beta = 1.3$

Figure 21: Snapshots of pressure field showing flow interactions among tubes and nozzle during the first cycle at time$=6.5\mu$s (left column) and at time$=10.4\mu$s (right column).


(a) Time = 0.225ms

(b) Time = 0.75ms

(c) Time = 0.975ms

(d) Time = 1.5ms

(e) Time = 1.725ms

(f) Time = 2.25ms

Figure 22: Time evolution of density-gradient field during the first cycle of operation of three-tubes PDE of ATM2 of $\alpha = 1430$, $\beta == 1.3$.

into the middle and bottom tubes, respectively. On the other hand, the upper part of the leading shock hits the edge connecting the middle and top tubes, while the right part propagates in the divergent section of the nozzle. Along the curved wall, the flow behind the leading shock is locally expanded to supersonic, leading to the formation of a shock wave stemming out from the wall, as evidenced in Fig. 20. The upper part of the leading shock then hits and reflects off the upper wall of the nozzle. In the top tube, the first shock relates to the diffracted part and the second one to the reflected shock from the lower wall of the nozzle. The shock waves established in all three tubes propagate upstream and elevate the pressure therein. From Fig. 21, it can be seen that the vortex sheet, shock wave, and expansion fan are clearly captured by the current ATM2.

Fig. 21(c) can clearly show that starting with THINC, $\beta=1.3$, numerical oscillations will begin to occur. By comparing ATM2 with $\alpha=1430$ and $\beta=1.3$ in Fig. 21(b), it can be seen that the ATM method can maintain high accuracy without causing numerical oscillations. Finally, it is believed that the single-nozzle tube and three-tubes simulations demonstrate that the combination of the ATM method and the Q-Learning method is helpful in improving the accuracy of the numerical model for reconstructed boundary points.

Compressible flows, a common occurrence in various fields, typically comprise two distinct components: complex, multi-scale structures and discontinuities. These elements require the utilization of a highly precise numerical technique, particularly in smooth regions, to accurately depict the multi-scale structures. Furthermore, the method must be able to accurately capture discontinuities such as shock waves without any oscillations. This requirement has led to the development of high-resolution schemes. High-resolution schemes for the simulation of compressible flows must possess the following key characteristics in order to be deemed effective: (1) the capability to capture discontinuities without the presence of oscillations, (2) the maintenance of sharp resolution for various types of discontinuities, particularly the contact discontinuity, even over prolonged simulations, and (3) an acceptable level of accuracy, as determined by the convergence rate, in smooth regions. As seen in [63], our ATM2 method meets these goals.

4 Conclusion

In this study, we propose a hybrid AUSMD flux splitting combined with the average of the MUSCL and THINC cell interface variable reconstruction (ATM type AUSMD) method to solve the hyperbolic conservation laws with stiff source terms arising from chemically reactive flows. This scheme consists of two steps: solving the homogeneous hyperbolic conservation law by the ATM type AUSMD method and using the Strang splitting method for the stiff reaction term. The ATM method minimizes the boundary variations of the characteristic variables and produces a less dissipative resolution. Extensive numerical examples for one- and two- dimensional detonation waves problems have been tested, demonstrating the effectiveness of this novel method. We then further applied this method to simulate the two-dimensional single-, multi-tube- PDE with CD nozzles. The results show that the proposed method can clearly capture the flow interactions.

In summary, ATM2 type AUSMD with $\alpha=1430$, $\beta=1.3$ scheme is an effective and simple method with great practical significance for simulating stiff detonation problems.

The accuracy of the simulation has been improved after optimizing the ATM methods. The optimal combination of the two parameters (α and β) can be found simultaneously and automatically using the Q-learning Q-table. Therefore, it is suggested that the ATM2 method with $\alpha = 1430$ and $\beta = 1.3$ be used as the main method for simulating stiff, complicated reactive flow problems.

Acknowledgments

The author wishes to acknowledge the financial support sponsored by the National Science Council of Taiwan, ROC under Contract MOST103-2221-E-032-024-MY3 and 10-2221-E-032 -020 -MY3.

References

[1] F. Ma, J.-Y. Choi, V. Yang, Thrust chamber dynamics and propulsive performance of single-tube pulse detonation engines, Journal of Propulsion and Power, 21 (2005), 512-526.

[2] F. Ma, J.-Y. Choi, V. Yang, Propulsive performance of airbreathing pulse detonation engines, Journal of Propulsion and Power, 22 (2006), 1188–1203.

[3] E. Wintenberger, J. Shepherd, Model for the performance of airbreathing pulse-detonation engines, Journal of Propulsion and Power, 22 (2006), 593–603.

[4] A.J. Chorin, Random choice solution of hyperbolic systems, Journal of Computational Physics, 22 (1976), 517–533.

[5] H. B. Stewart, B. Wendroff, Two-phase flow: Models and methods, Journal of Computational Physics, 56 (1984), 363–409.

[6] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) methods for the dynamics of free boundaries, Journal of Computational Physics, 39 (1981), 201–225.

[7] Shyue, K. M, An efficient shock-capturing algorithm for compressible multi-component problems, Journal of Computational Physics, 142(1998), 208–242.

[8] O. Ubbink, R.I. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, Journal of Computational Physics, 153 (1999), 26–50.

[9] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proceedings of the National Academy of Sciences, 93 (1996), 1591–1595.

[10] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech., 35 (2003), 341–372.

[11] E. Puckett, A. Almgren, J. Bell, D. Marcus, W. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, Journal of Computational Physics, 130 (1997), 269–282.

[12] A.J. Chorin, Random choice methods with applications to reacting gas flow, Journal of Computational Physics, 25 (1977), 253–272.

[13] H.C. Yee, D.V. Kotov, W. Wang, C.-W. Shu, Spurious behavior of shock-capturing methods by the fractional step approach: Problems containing stiff source terms and discontinuities, Journal of Computational Physics, 241 (2013), 266–291.

[14] C. Helzel, R. LeVeque, and G. Warneke. A modified fractional step method for the accurate approximation of detonation waves. SIAM J. Sci. Stat. Comput., 22 (2000), 1489–1510.

[15] W. Bao, S. Jin, The random projection method for hyperbolic conservation laws with stiff reaction terms, Journal of Computational Physics, 163 (2000), 216–248.

[16] F. Xiao, Y. Honma, T. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, International Journal of Numerical Methods in Fluids, 48 (2005), 1023–1040.

[17] Z. Sun, S. Inaba, F. Xiao, Boundary variation diminishing (BVD) reconstruction: A new approach to improve Godunov schemes, Journal of Computational Physics, 322 (2016) 309–325.

[18] X. Deng, B. Xie, F. Xiao, H. Teng, New accurate and efficient method for stiff detonation capturing, AIAA Journal, 56 (2018), 4024–4038.

[19] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, Journal of Computational Physics, 32 (1979), 101–136.

[20] D.A. Cassidy, J.R. Edwards, M. Tian, An investigation of interface-sharpening schemes for multi-phase mixture flows, Journal of Computational Physics, 228 (2009), 5628–5649.

[21] K.M.Shyue, F. Xiao, An Eulerian interface sharpening algorithm for compressible two-phase flow: The algebraic THINC approach, Journal of Computational Physics, 268 (2014), 326–354.

[22] M. Frank, D. Drikakis, V. Charissis, Machine-learning methods for computational science and engineering, Computation, 8 (2020) 15.

[23] Y. Wang, Z. Shen, Z. Long, B. Dong, Learning to discretize: Solving 1D scalar conservation laws via deep reinforcement learning, arXiv preprint arXiv:1905.11079, (2019).

[24] T. Kossaczká, M. Ehrhardt, M. Günther, Enhanced fifth order WENO shock-capturing schemes with deep learning, Results in Applied Mathematics, 12 (2021) 100201.

[25] E. Way, D.S. Kapilivai, Y. Fu, L. Yu, Backpropagation through time and space: Learning numerical methods with multi-agent reinforcement learning, arXiv preprint arXiv:2203.08937, (2022).

[26] A.P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA Journal, 55 (2017) 2215–2227.

[27] S.J. Grimberg, C. Farhat, Hyperreduction of CFD models of turbulent flows using a machine learning approach, AIAA Scitech 2020 Forum, (2020), 0363.

[28] B. Meuris, S. Qadeer, P. Stinis, Machine-learning custom-made basis functions for partial differential equations, arXiv preprint arXiv:2111.05307, (2021).

[29] A. Schwarz, J. Keim, S. Chiocchetti, A. Beck, A reinforcement learning based slope limiter for second-order finite-volume schemes, Proceedings in Applied Mathematics and Mechanics, 7 (2022) 10.

[30] R. Bellman, Dynamic Programming, Princeton University Press, Princeton, New Jersey Google Scholar, 1957.

[31] R. Bellman, A Markovian decision process, Journal of Mathematics and Mechanics, (1957) 679–684.

[32] R.A. Howard, Dynamic programming and Markov processes, 1960.

[33] W.S. Lovejoy, A survey of algorithmic methods for partially observed Markov decision processes, Annals of Operations Research, 28 (1991) 47–65.

[34] D.J. White, Real applications of Markov decision processes, Interfaces, 15 (1985) 73–83.

[35] D.J. White, Further real applications of Markov decision processes, Interfaces, 18 (1988) 55–61.

[36] D.J. White, A survey of applications of Markov decision processes, Journal of the Operational Research Society, 44 (1993) 1073–1096.

[37] J. Rust, Numerical dynamic programming in economics, Handbook of Computational Economics, 1 (1996) 619–729.
[38] D. Bertsekas, Distributed dynamic programming, IEEE Transactions on Automatic Control, 27 (1982) 610–616.
[39] D.P. Bertsekas, Distributed asynchronous computation of fixed points, Mathematical Programming, 27 (1983) 107–120.
[40] G.A. Kimble, Hilgard and Marquis, Conditioning and Learning, 1961.
[41] G. A. Kimble, Foundations of Conditioning and Learning, Appleton-Century-Crofts, Inc., New York., 1967.
[42] J. E. Mazur, Learning and Behavior, Third Edition, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[43] E. R. Hilgard, G. H. Bower, Theories of Learning, Prentice-Hall, Englewood Cliffs, NJ, 1975.
[44] D. C. Dennett, Brainstorms, Chapter 5: Why the Law-of-Effect Will Not Go Away, pages 71–89. Bradford/MIT Press, Cambridge, MA, 1978.
[45] D.T. Campbell, Blind variation and selective survival as a general strategy in knowledge-processes, Self-Organizing Systems, (1960) 205–231.
[46] G. Cziko, Without Miracles: Universal Selection Theory and the Second Darwinian Revolution, MIT Press, 1997.
[47] M.L. Minsky, Theory of Neural-Analog Reinforcement Systems and its Application to the Brain-Model Problem, Princeton University, 1954.
[48] B. Farley, W.D. Clark, Simulation of self-organizing systems by digital computer, Transactions of the IRE Professional Group on Information Theory, 4 (1954) 76–84.
[49] M. Waltz, K. Fu, A heuristic approach to reinforcement learning control systems, IEEE Transactions on Automatic Control, 10 (1965) 390–398.
[50] J. Mendel, R. McLaren, 8 Reinforcement-Learning Control and Pattern Recognition Systems, Mathematics in Science and Engineering, Elsevier, 1970, pp. 287–318.
[51] M. Minsky, Steps toward artificial intelligence, Proceedings of the IRE, 49 (1961) 8–30.
[52] C.J. Watkins, P. Dayan, Q-learning, Machine Learning, 8 (1992) 279–292.
[53] P.J. Werbos, Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research, IEEE Transactions on Systems, Man, and Cybernetics, 17 (1987) 7–20.
[54] Y. Wada, M.-S. Liou, A flux splitting scheme with high-resolution and robustness for discontinuities, 32nd Aerospace Sciences Meeting and Exhibit, (1994), 83.
[55] Y.-Y. Niu, M.-S. Liou, Numerical simulation of dynamic stall using an improved advection upwind splitting method, AIAA Journal, 37 (1999) 1386–1392.
[56] Y.-Y. Niu, Computations of two-fluid models based on a simple and robust hybrid primitive variable Riemann solver with AUSMD, Journal of Computational Physics, 308 (2016) 389–410.
[57] T.-Y. Chiu, Y.-Y. Niu, Y.-J. Chou, Accurate hybrid AUSMD type flux algorithm with generalized discontinuity sharpening reconstruction for two-fluid modeling, Journal of Computational Physics, 443 (2021) 110540.
[58] B. Xu, Q. Zhou, J. Shi, S. Li, Hierarchical Q-learning network for online simultaneous optimization of energy efficiency and battery life of the battery/ultracapacitor electric vehicle, Journal of Energy Storage, 46 (2022) 103925.
[59] V. Kosana, K. Teeparthi, S. Madasthu, S. Kumar, A novel reinforced online model selection using Q-learning technique for wind speed prediction, Sustainable Energy Technologies and Assessments, 49 (2022) 101780.
[60] Y. Li, H. Wang, N. Wang, T. Zhang, Optimal scheduling in cloud healthcare system using

Q-learning algorithm, Complex & Intelligent Systems, 8 (2022) 4603–4618.

[61] Z. He, K. Wang, H. Li, H. Song, Z. Lin, K. Gao, A. Sadollah, Improved Q-learning algorithm for solving permutation flow shop scheduling problems, IET Collaborative Intelligent Manufacturing, 4 (2022) 35–44.

[62] S. Suratkar, K. Shah, A. Sood, A. Loya, D. Bisure, U. Patil, F. Kazi, An adaptive honeypot using Q-learning with severity analyzer, Journal of Ambient Intelligence and Humanized Computing, 13 (2022) 4865–4876.

[63] Z. He, Y. Ruan, Y. Yu, B. Tian, F. Xiao, Self-adjusting steepness-based schemes that preserve discontinuous structures in compressible flows, Journal of Computational Physics, 463 (2022) 111268.