

Article

# A Simple and Robust Shock-Capturing Approach for Discontinuous Galerkin Discretizations

Jae Hwan Choi <sup>1,\*</sup>, Juan J. Alonso <sup>1</sup> and Edwin van der Weide <sup>2</sup>

<sup>1</sup> Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA

<sup>2</sup> Department of Mechanical Engineering, University of Twente, 7522 NB Enschede, The Netherlands

\* Correspondence: jhchoi89@stanford.edu

Received: 1 June 2019; Accepted: 3 July 2019; Published: 10 July 2019



**Abstract:** The discontinuous Galerkin (DG) method has become popular in Computational Fluid Dynamics mainly due to its ability to achieve high-order solution accuracy on arbitrary grids, its high arithmetic intensity (measured as the ratio of the number of floating point operations to memory references), and the use of a local stencil that makes scalable parallel solutions possible. Despite its advantages, several difficulties hinder widespread use of the DG method, especially in industrial applications. One of the major challenges remaining is the capturing of discontinuities in a robust and accurate way. In our previous work, we have proposed a simple shock detector to identify discontinuities within a flow solution. The detector only utilizes local information to sense a shock/discontinuity ensuring that one of the key advantages of DG methods, their data locality, is not lost in transonic and supersonic flows. In this work, we reexamine the shock detector capabilities to distinguish between smooth and discontinuous solutions. Furthermore, we optimize the functional relationships between the shock detector and the filter strength, and present it in detail for others to use. By utilizing the shock detector and the corresponding filtering-strength relationships, one can robustly and accurately capture discontinuities ranging from very weak to strong shocks. Our method is demonstrated in a number of two-dimensional canonical examples.

**Keywords:** discontinuous Galerkin method; shock-capturing; discontinuities; high-order methods; CFD

## 1. Introduction

Ever since Computational Fluid Dynamics (CFD) began to play an important role in analyzing fluid motion and designing industrial products, engineers have sought techniques which can increase the accuracy of a flow simulation without increasing its associated computational cost. Many CFD researchers have developed a growing list of numerical methods to meet these demands. In particular, CFD developments over the past half century have focused on finite volume, finite difference, and spectral methods [1–6]. However, conventional CFD methods have faced difficulties in achieving higher-order (order of solution convergence larger than 2) simulations from both the implementation and high-performance computing perspectives. For example because of their low arithmetic intensity and large computational stencils, higher-order Weighted Essentially Non-Oscillatory (WENO) methods [7] on unstructured meshes and higher-order finite difference methods [8] can lose efficiency when running large-scale simulations.

To overcome these shortcomings in parallel performance and scalability, during the past two decades researchers have focused on finite-element methods such as the discontinuous Galerkin (DG) method. Reed and Hill first introduced the DG method in their seminal paper [9]. Later on, a solid theoretical foundation for the DG method was established by Cockburn and Shu in a series of papers [10–14]. For an overview of the DG method, the reader is referred to [15], which provides many more details than is practical to discuss in this paper. Because of its mathematical formulation,

the DG method has high arithmetic intensity compared to the finite volume and the finite difference methods. Furthermore, higher-order accuracy in space can easily be obtained by adding more degrees of freedom in combination with higher-order polynomial bases within a given element. For these reasons, the DG method can obtain higher-order spatial accuracy relatively easily even on complex geometries and when compared to both finite volume and finite difference methods. Additionally, in the DG method only the nearest face-neighbors are needed, while many operations for a given element do not require information from neighboring elements at all, which is an attractive feature for modern computer hardware. Due to these advantages, the DG method has become popular in many fields.

Despite these advantages, the DG method has not yet been fully adopted in industry mainly because of some remaining challenges for practical use. One of these difficulties is the robust and accurate capturing of discontinuities within a flow solution, which is the main objective of the present work. Specifically, and like most other CFD methods, the polynomial nature of the basis functions used by the DG method lead to spurious oscillations in flow simulations if discontinuities emerge during a computation. Because of this fundamental issue, engineers and scientists have pursued methods that can help attain the two fundamental objectives for shock-capturing in the DG framework: the balancing of robustness and accuracy. In order to satisfy these goals, various approaches have been proposed in the literature [16–26]. Researchers have devised multiple methods to sense a discontinuity with information from a local stencil of surrounding neighbors. After the discontinuity is identified, a flow solution can be stabilized with any one several resolving methods including artificial viscosity, limiting, and filtering. A brief comparison of these approaches to shock detection and resolution can be found in [27].

In our previous work, we proposed an approach for discontinuity detection for the DG method and demonstrated its shock-capturing capabilities [27]. Our shock detector robustly distinguishes between smooth solutions and discontinuities even for polynomial orders as low as two and three, which are relatively low orders for such shock-capturing schemes in the DG literature. Furthermore, the shock detector parameter and the necessary filtering strengths were correlated through numerical experiments, enabling the user to avoid tedious parameter tuning processes. These correlations provide robust shock-capturing capabilities regardless of the strengths of shocks present in a flow simulation. However, in that early work, we did not provide a quantitative analysis of the mathematical relationships between the detector and the filtering strengths. At the time, it was not clear whether the preliminary relationships used provided excessive filtering which, despite its solution smoothness advantages, might lead to a severe loss in accuracy. For this reason, in this paper the relationships between the shock detector and the filtering strengths are optimized to obtain maximum accuracy while maintaining robustness of the shock-capturing capability. These relationships are presented for future use.

The remaining sections are organized as follows. The governing equations and the associated numerical discretization approach are introduced in Section 2. In Section 3, the shock detector proposed in [27] and a filtering method are briefly described. Additionally, functional relationships between the shock detector and the filtering method are optimized to obtain accurate solutions while maintaining robustness. Numerical simulation results are shown in Section 4. Finally, we close the paper with concluding remarks and ongoing work in Section 5.

## 2. Governing Equations and Numerical Discretization

This section briefly describes the governing equations and numerical discretization used in this paper. The majority of this section is restated, with some modifications to enhance clarity, from [27].

### 2.1. Governing Equations

In this work, we consider the Euler equations of gas dynamics in a differential conservative form

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot \vec{F} = 0, \quad (1)$$

where  $\vec{u} = \vec{u}(\vec{x}, t)$  is the vector of conserved variables and  $\vec{F} = \vec{F}(\vec{u})$  is the vector of convective fluxes.  $\vec{x}$  is the position vector in a physical domain  $\Omega$ , and  $\partial t$  represents a partial derivative in the temporal dimension. Standard inviscid flux definitions apply.

## 2.2. Discontinuous Galerkin Discretization

The physical domain  $\Omega$  and its boundary  $\partial\Omega$  are discretized with  $K$  non-overlapping elements

$$\Omega = \bigcup_{i=1}^K D_i, \quad (2)$$

where  $D_i$  is a discrete element in the domain. The index  $i$  ranges from 1 to the number of elements,  $K$ , in the domain. In two dimensions, the domain is divided into triangular or quadrilateral elements. For each discrete element, a numerical solution  $\vec{u}^h$  of (1) is constructed as

$$\vec{u} \approx \vec{u}^h(\vec{x}, t) = \sum_{i=1}^{N_p} \hat{u}_i(t) \psi_i(\vec{x}) = \sum_{j=1}^{N_p} \vec{u}_j(\vec{x}_j, t) l_j(\vec{x}). \quad (3)$$

$\vec{u}^h$  can be described either in a modal form with modal basis functions,  $\psi_i$ , or in a nodal form with Lagrange polynomials,  $l_j$ .  $\hat{u}_i(t)$  is the vector of modal coefficients for the corresponding  $\psi_i(\vec{x})$  and  $\vec{u}_j(\vec{x}_j, t)$  is the vector of nodal coefficients for the corresponding  $l_j(\vec{x})$ , where  $\vec{x}_j$  indicates the spatial location of  $\vec{u}_j$ . More details about the transformation between the two forms and the selection of the modal basis functions can be found in [15]. In this work, we use Legendre polynomials for  $\psi_i$  to construct the numerical solution for the modal form.  $N_p$  is the number of degrees of freedom in each discrete element. Similarly to the modal basis functions, various options exist to distribute the nodal points within an element. For simplicity, equidistant points are used to represent the degrees of freedom. Although distributing nodal points uniformly may lead to ill-conditioned interpolations (Runge's phenomenon), equidistant nodal points up to polynomial order 4, the upper limit considered in this work, perform robustly.

With this discontinuous Galerkin discretization, the governing Equation (1) can be converted into the following weak form at time  $t$  (integration only takes place over the spatial element,  $D_i$ ):

$$\int_{D_i} \frac{\partial \vec{u}}{\partial t} l_j dV - \int_{D_i} \vec{F}_k \frac{\partial l_j}{\partial \vec{x}_k} dV + \oint_{\partial D_i} \vec{n} \cdot \vec{F}^* l_j dA = 0, \quad j = 1, \dots, N_p, \quad (4)$$

where  $\vec{n}$  is the unit outward normal vector at the boundary of  $D_i$  and  $\vec{F}^*$  is a numerical flux at this boundary. In order to stabilize the numerical discretization, the Roe flux [28] is used for the surface integral calculation. The Lax–Friedrichs flux [29] is only used in Section 4.1. For more information about the Roe and the Lax–Friedrichs flux formulations, the reader is referred to [30].

## 2.3. Space-Time Discretization: ADER-DG

The DG discretization enables simple p-refinement, which is one of its major advantages compared to different higher-order methods. Higher accuracy in space can be achieved by adding more degrees of freedom within an element while the communication remains limited to its face neighbors.

For the integration in time, it is possible to use standard time integration schemes for Equation (4). As the mass matrix for a DG scheme is local for an element, explicit schemes of the Runge–Kutta type can also be used. However, for practical applications, the time step for explicit time integration schemes is very much limited by the CFL condition and only when combined with time-accurate local time-stepping can it be an alternative to implicit methods. Unfortunately, standard time integration schemes do not allow for an order of accuracy beyond first order when applied in combination with time-accurate local time-stepping, which is clearly too low for practical applications.

In order to achieve high-order accuracy in both space and time while using time-accurate local time-stepping, Toro et al. suggested the Arbitrary high-order DERivatives (ADER) method in [31]. In that work, the authors obtain high-order accuracy in the spatial and temporal dimensions by using the Cauchy–Kovalevskaya procedure. Dumbser implemented the ADER method in a finite volume solver in [32]. Similarly, in order to apply the ADER method in combination with DG, we need to introduce a space-time discretization. Here, we summarize the major steps for the DG space-time discretization. Further details can be found in [33,34].

### 2.3.1. The Local Space-Time DG Predictor

In the predictor step, we look for a solution vector  $\vec{q}(\vec{x}, t)$  which takes the classical DG discretization at time step  $n$  as an input and returns its evolution in time up to time step  $n + 1$ , the difference between these time levels being the time step  $\Delta t$ :

$$\vec{u}(\vec{x}, t^n) \rightarrow \vec{q}(\vec{x}, t) \quad t^n \leq t \leq t^{n+1} = t^n + \Delta t. \quad (5)$$

$\vec{q}(\vec{x}, t)$  can be expanded in space and time as follows:

$$\vec{q}(\vec{x}, t) = \vec{q}(\vec{\zeta}, \tau) = \vec{q}_i \theta_i(\vec{\zeta}, \tau), \quad (6)$$

where  $\theta_i$  is a polynomial basis function in space-time,  $\vec{q}_i$  is a vector of space-time degrees of freedom, and summation over repeated indices is implied.  $\vec{\zeta}$  and  $\tau$  are parametric coordinates in space and time, respectively. For simplicity, let us restrict ourselves to the one-dimensional case and use parametric coordinates ( $\tau$  in time and  $\zeta$  in space). After multiplication by a space-time test function and integrating over a space-time control volume, Equation (1) becomes

$$\int_{-1}^1 \int_{-1}^1 \theta_j \left( \frac{\partial \vec{q}}{\partial \tau} + \frac{\partial \vec{f}}{\partial \zeta} \right) d\zeta d\tau = 0, \quad (7)$$

where  $\vec{f} = \frac{\Delta t}{2} (\partial \zeta / \partial x) \cdot \vec{F}$ . Additionally, we assume that the convective fluxes can also be expanded over the space-time basis functions

$$\vec{f} = \vec{f}_i \theta_i. \quad (8)$$

Since we are using the nodal representation of a discrete solution,  $\vec{f}_i$  is just the evaluation of the convective fluxes at the space-time degrees of freedom. Therefore,  $\vec{f}_i = \vec{f}(\vec{q}_i)$ . Note that the assumption (8) is in principle not correct for a nonlinear system of equations and aliasing errors may be introduced. However, this assumption is only used in the predictor step of the algorithm and, as such, the errors do not amplify. With all these equations, we can obtain a final system of algebraic equations for

$$K_{ij}^1 \vec{q}_j + K_{ij}^\zeta \cdot \vec{f}_j = F_{ik}^{-1} \vec{u}_k^n, \quad (9)$$

where

$$\begin{aligned} K_{ij}^1 &= \int_{-1}^1 \theta_i(\zeta, 1) \theta_j(\zeta, 1) d\zeta - \int_{-1}^1 \int_{-1}^1 \frac{\partial \theta_i}{\partial \tau} \theta_j d\zeta d\tau, \\ K_{ij}^\zeta &= \int_{-1}^1 \int_{-1}^1 \theta_i \frac{\partial \theta_j}{\partial \zeta} d\zeta d\tau, \\ F_{ik}^{-1} &= \int_{-1}^1 \theta_i(\zeta, -1) \psi_k(\zeta) d\zeta. \end{aligned} \quad (10)$$

The time integration integrals in (10) are approximated by using a quadrature rule with Gauss–Legendre points. After evaluating (10), the unknown  $\vec{q}_j$  can be solved with an iterative procedure up to a desired tolerance. Further details can be found in [33,34]. Note that integration by

parts is not used in space to obtain (9). Also note that the space-time discretization described for  $\vec{q}$  in the predictor step is completely local to an element and, therefore, a CFL limit is required to ensure that the procedure is stable.

### 2.3.2. The Corrector Step

After the space-time predictor solution  $\vec{q}(\vec{x}, t)$  is obtained, we can use it to calculate the solution at the next time level,  $n + 1$ . By integrating the weak form (4) in time from  $t^n$  to  $t^{n+1}$ , the equation becomes

$$\left( \int_{D_i} l_j l_k dV \right) (\bar{u}_k^{n+1} - \bar{u}_k^n) - \int_{t^n}^{t^{n+1}} \int_{D_i} \vec{F}_k(\bar{u}) \frac{\partial l_j}{\partial \bar{x}_k} dV dt + \int_{t^n}^{t^{n+1}} \oint_{\partial D_i} \vec{n} \cdot \vec{F}^*(\bar{u}) l_j dAdt = 0. \quad (11)$$

Since we obtained a time evolution of  $\bar{u}$  until  $t_{n+1}$  from the predictor step, the integrals over the time-volume and time-surface can be calculated with the predictor solution  $\vec{q}$

$$\left( \int_{D_i} l_j l_k dV \right) (\bar{u}_k^{n+1} - \bar{u}_k^n) - \int_{t^n}^{t^{n+1}} \int_{D_i} \vec{F}_k(\vec{q}) \frac{\partial l_j}{\partial \bar{x}_k} dV dt + \int_{t^n}^{t^{n+1}} \oint_{\partial D_i} \vec{n} \cdot \vec{F}^*(\vec{q}) l_j dAdt = 0, \quad (12)$$

to obtain the desired solution at the end of the time step.

## 3. Shock-Capturing in the Discontinuous Galerkin Method

Shock-capturing in the DG method comprises two fundamental steps: (a) detecting elements which may contain a flow discontinuity or may exhibit spurious oscillations, and (b) stabilizing the numerical solution by applying resolving methods in those elements. In [27], we proposed a simple shock detector for the DG method and introduced the functional relationships between the shock detector and filtering strengths. In this section, we revisit the methodology and refine the correlations between the sensor and the filtering method for use by others.

### 3.1. Shock Detector

Multiple researchers have proposed strategies for detecting discontinuities in the DG method [19,21,22,24]. All shock detectors in the literature can be categorized into three groups: detectors that only use local information, detectors that utilize local information and direct neighbors' information, and detectors that exploit local information and the Voronoi neighbors' information. To preserve one of the advantages of the DG method concerning high-performance computing, it is highly desirable to develop shock detectors that only use information from the local element.

Persson and Peraire proposed a novel shock detector that fulfills these demands in [24]. In this shock detector, a spectral decay rate is measured by computing the ratio between the highest modal coefficient and all modal coefficients. By using this ratio, the detector distinguishes between smooth regions and a shock/discontinuity, and it does this particularly well for elements with high polynomial order. However, the detector requires parameter tuning procedures. Inspired by Persson and Peraire's work, we proposed a shock detector in [27] to circumvent calibrating steps. The basic implementation of the detector is revisited here.

Let  $\phi^n$  be the ratio between the norms of the modal coefficients of the highest and the lowest mode in the description of a flow quantity  $u$  at time level  $n$ :

$$\phi^n = \frac{|\hat{u}_N^n|_2}{|\hat{u}_0^n|_2}, \quad (13)$$

where  $\hat{u}_N^n$  is the modal coefficient of the  $N^{\text{th}}$  degree polynomial at  $t^n$ . For one-dimensional problems,  $\hat{u}_N^n$  is just one modal coefficient. In two-dimensional problems,  $\hat{u}_N^n$  are the modal coefficients for which the degree of the corresponding polynomial is greater than or equal to  $N$ . In addition,  $\hat{u}_0^n$  is the modal

coefficient of the lowest polynomial at time level  $n$ : it is the mean value of  $u$  within the element. In this work, we use the density in an element to calculate  $\phi^n$ .

The value of  $\phi^n$  is usually small unless an element has oscillatory variations in its solution. Indeed, even for an oscillatory solution,  $\phi^n$  is not large if the domain is sufficiently refined with discrete elements. However, if a discontinuity is convected from an adjacent element or if it emerges within an element, then  $\phi^n$  becomes larger. Therefore, by comparing the values of  $\phi$  at two different time levels, one can narrow down the candidates to flag as *troubled* elements: the elements that may contain a discontinuity that needs to be treated. Now let  $\beta$  be the ratio between  $\phi^{n+1}$  and  $\phi^n$ :

$$\beta = \frac{\phi^{n+1}}{\phi^n}. \quad (14)$$

We can examine how the strength of an oscillation within an element behaves between two time levels by measuring the  $\beta$  value. With these definitions of  $\phi$  and  $\beta$ , the shock detecting procedure can be stated as follows:

1. Calculate  $\phi^n$  from  $u(\vec{x}, t^n)$ ,
2. March solution forward one time-step and calculate a candidate solution  $u(\vec{x}, t^{n+1})$ ,
3. Calculate  $\phi^{n+1}$  from  $u(\vec{x}, t^{n+1})$ ,
4. If  $\phi^{n+1} \leq \phi_0$ , an element is not considered to be a troubled element. Otherwise, the current element might be a troubled element. Here,  $\phi_0$  is a threshold value that depends on the polynomial order of the element.
5. For elements with  $\phi^{n+1} \geq \phi_0$ , calculate  $\beta$ . If  $\beta \geq \beta_0$ , then the current element is a troubled element. Here,  $\beta_0$  is an empirically determined value.

We set  $\beta_0$  as 1.0 in order to capture a stationary shock in numerical simulations. In addition, the value of  $\phi_0$  is predetermined based on the type and polynomial order of an element. Details of determining appropriate  $\phi_0$  values are described in Section 3.3. Therefore, using the proposed sensor does not require tuning parameters from one problem to the next.

### 3.2. Filtering

After an element is flagged as a troubled element, additional treatment is required to stabilize or resolve a high-gradient solution near a shock or discontinuity. In this work, we use the filtering method. The filtering method is effectively equivalent to using artificial viscosity. However, it does not restrict the time step compared to directly adding artificial viscosity in the governing equations. In addition, the filtering method amounts to a simple matrix-vector multiplication that can be easily implemented. Therefore, it is not necessary to calculate a viscous flux to resolve a discontinuity, which is necessary when using artificial viscosity. Details of the theoretical background and implementation of the method can be found in [15]. The following second-order exponential filter is used in this work:

$$\sigma(\eta) = \exp(-\alpha\eta^2) \quad \text{where} \quad \eta = \min(n/N, 1.0). \quad (15)$$

Here,  $N$  is the polynomial order of a basis function and  $\alpha$  is a parameter to change the strength of the filter. The  $\sigma$  values are determined based on the polynomial order and they are multiplicative factors for the modal coefficients,  $\hat{u}$ , of the element to obtain filtered modal coefficients,  $\tilde{u}$

$$\tilde{u} = \sigma\hat{u}. \quad (16)$$

For example, let us assume that a filter with  $\alpha = 1.0$  is applied to a one-dimensional  $n = 2$  element. This element has three modal coefficients,  $\hat{u}_1$ ,  $\hat{u}_2$ ,  $\hat{u}_3$  and their polynomial orders are 0, 1, and 2, respectively. Then,  $\sigma(0/2) = \exp(-1.0 \times 0^2) = 1.0$  is the multiplicative factor for  $\hat{u}_1$ ,  $\sigma(1/2) = \exp(-1.0 \times 0.5^2) \approx 0.7788$  is the multiplicative factor for  $\hat{u}_2$ , and  $\sigma(2/2) = \exp(-1.0 \times 1^2) \approx 0.3679$  is the multiplicative factor for  $\hat{u}_3$ . Therefore,  $\tilde{u}_1 = 1.0 \hat{u}_1$ ,  $\tilde{u}_2 = 0.7788 \hat{u}_2$ , and  $\tilde{u}_3 = 0.3679 \hat{u}_3$ . Note that,

for quadrilateral elements in two dimensions, some basis functions have a larger polynomial order value than  $N$ . Therefore,  $\eta$  is the minimum value of  $n/N$  and 1.0:

### 3.3. Correlation between $\alpha$ and $\phi^{n+1}$

Researchers seek two objectives in capturing a discontinuity: stability and accuracy. Of these two goals, stability has higher priority because measuring accuracy becomes relevant only if the solution is stable. Therefore, it is necessary to obtain a relationship between  $\alpha$  and  $\phi^{n+1}$  that provides at least a stable solution when a shock exists within a flow, while adding the least amount of filtering to improve the accuracy of the computed solution.

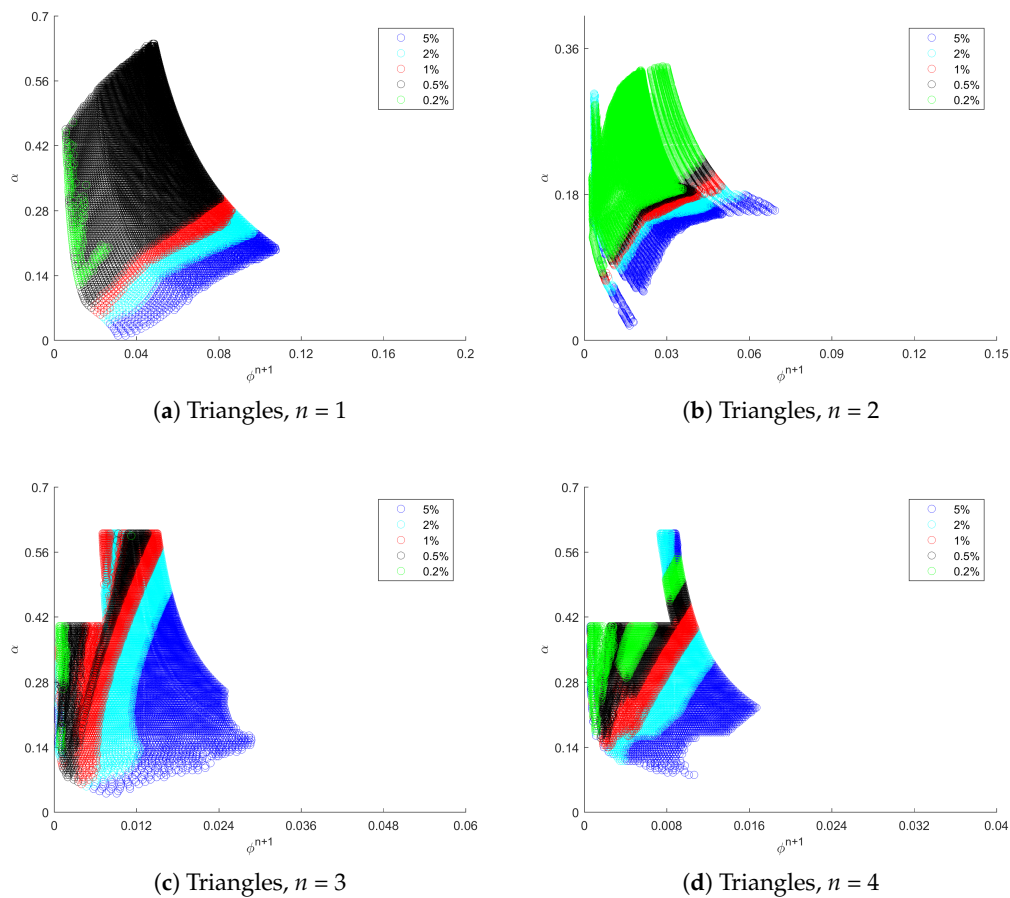
The shock tube problem introduced in [35] is selected as our benchmark problem to correlate  $\alpha$  and  $\phi^{n+1}$ . In the shock tube problem, one can change the strength of the right-running shock wave by varying the initial pressure ratio across the diaphragm. We fix the initial conditions just as was done in [35] except for the pressure on the left side of the diaphragm. Starting from an initial pressure ratio of 2, we gradually increased the pressure ratio up to a value of 15 in order to generate a series of shock waves with different strengths. For each initial pressure ratio, we performed numerical experiments with a set of  $\alpha$  values, where  $\alpha$  is constant in each experiment. During each simulation, we track the maximum  $\phi^{n+1}$  values as well as the undershoots that occur near a shock. The  $L_\infty$  error in density is then calculated with respect to the initial density value on the right side of the diaphragm. Near a discontinuity, it is critical to minimize the  $L_\infty$  error among other error measurements. This is because the  $L_\infty$  is directly related to undershoots near the right-running shock in this problem, and these undershoots must be suppressed to guarantee a stable solution throughout the simulations. By performing these numerical experiments, we can obtain a relationship between  $\alpha$  and  $\phi^{n+1}$  to suppress undershoots as a function of the  $L_\infty$  percent error.

In Figure 1, the  $L_\infty$  percent errors in density are depicted as a function of  $\alpha$  and  $\phi^{n+1}$  for each polynomial order of a triangular element. For a specific  $\phi^{n+1}$  value, the minimum  $\alpha$  value to ensure that the  $L_\infty$  error does not exceed a certain value can be found. For example, for  $n = 1$  triangles and  $\phi^{n+1} = 0.06$ , at least  $\alpha \approx 0.15$  is required to guarantee that undershoots do not surpass 2% error. If the  $L_\infty$  error from the undershoots is required to be lower than 0.5% for the same polynomial order and  $\phi^{n+1}$  value, the filtering strength must be larger than 0.25. To generate simple relationships between  $\alpha$  and  $\phi^{n+1}$ , we constructed a piecewise linear function for each element and polynomial order, provided in [27].

The functional relationships given in [27] ensure that a solution will not diverge due to the presence of a shock. However, if a solution is filtered out excessively, its accuracy will be degraded, thereby losing the main advantage of higher-order methods. Therefore, the functional relationships between  $\alpha$  and  $\phi^{n+1}$  need to be tailored to simulate a flow as accurately as possible while maintaining stability.

In designing the relationships between  $\alpha$  and  $\phi^{n+1}$ , we must determine two unknown factors. The first is whether such relationships return excessive filtering strengths. The filtering strengths must be strong enough to stabilize a solution. However, we should not excessively filter a solution to maintain accuracy. The other one is the threshold value selected,  $\phi_0$ . Theoretically, one can lower the  $\phi_0$  value all the way to zero, which effectively makes the solver apply filtering everywhere in the domain. On the other hand, if one selects too large a value for  $\phi_0$ , the solver cannot robustly distinguish the presence of a shock/discontinuity. Therefore, the  $\phi_0$  value should be sufficiently large up to a point where it does not harm the robustness of the solver.

To eliminate such ambiguities, we perform other numerical experiments with the shock-entropy interaction problem proposed by Shu and Osher in [36]. The objective of these numerical experiments is to minimize the  $L_2$  error in density with respect to a reference solution. The reference solution is obtained by a one-dimensional simulation with 3,200 equidistant elements of polynomial order two.



**Figure 1.**  $L_\infty$  percent errors in density from a series of shock tube problems with various initial pressure ratios. The errors are plotted as a function of  $\alpha$  and  $\phi^{n+1}$  for each polynomial order of triangular elements.

In these experiments, we allow two parameters to change:  $\alpha_0$ , an offset value to shift piecewise linear functions, and  $\phi_0$ , a threshold value for the sensor described in Section 3.1. We optimize the  $\alpha(\phi^{n+1})$  relationships by shifting the functions along the  $\alpha$  axis as well as by changing their  $\phi_0$  value.  $\alpha_0$  values range from  $-0.3$  to  $0.3$ , and a typical range for  $\phi_0$  depends on the element type and polynomial order to find a local minimum of the  $L_2$  error.

Figure 2 shows the normalized  $L_2$  error in density by solving the Shu–Osher problem with triangular elements and different polynomial orders. For a fixed  $\phi_0$  value, the  $L_2$  error increases if we increase the filtering strength. This is because high-gradient information is lost by excessive filtering. On the other hand, if the filtering strength decreases too much, the error also increases, and eventually the simulation diverges. For a constant  $\alpha_0$  value, the  $L_2$  error increases if we set a value of  $\phi_0$  that is too large. This is because spurious oscillations are introduced due to a failure of the shock detector. On the contrary, the  $L_2$  error also increases if  $\phi_0$  becomes too small. In this case, the shock detector determines too many elements as troubled elements and filtering is applied to elements that might not require such resolving processes. In Figure 2, red dots are selected as design points for each polynomial order of a triangular element. After repeating the same procedures for quadrilateral elements, we propose the following relationships between  $\alpha$  and  $\phi^{n+1}$ , which we use in all other test cases presented in this paper:

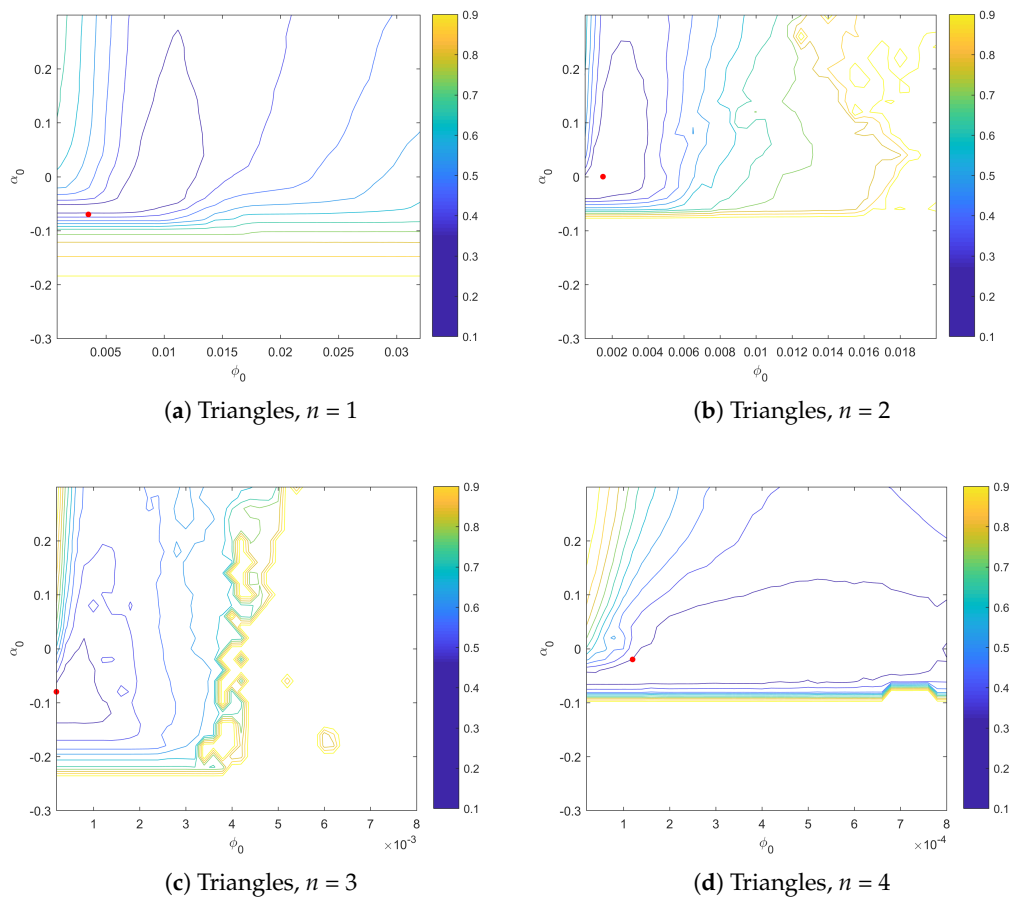


(1) Two dimensions, triangular elements:

$$\begin{aligned}
 \alpha_{tri,1}(\phi^{n+1}) &= (3.21 \times \phi^{n+1} - 0.01994) & 0.00350 \leq \phi^{n+1}, \\
 \alpha_{tri,2}(\phi^{n+1}) &= (6.18 \times \phi^{n+1} + 0.04383) & 0.00150 \leq \phi^{n+1} \leq 0.01980, \\
 &= (2.93 \times \phi^{n+1} + 0.10818) & 0.01980 \leq \phi^{n+1}, \\
 \alpha_{tri,3}(\phi^{n+1}) &= (35.7 \times \phi^{n+1} + 0.03362) & 0.00020 \leq \phi^{n+1}, \\
 \alpha_{tri,4}(\phi^{n+1}) &= (38.0 \times \phi^{n+1} + 0.05520) & 0.00012 \leq \phi^{n+1}.
 \end{aligned}
 \tag{17}$$

(2) Two dimensions, quadrilateral elements:

$$\begin{aligned}
 \alpha_{quad,1}(\phi^{n+1}) &= (4.00 \times \phi^{n+1} - 0.19400) & 0.06000 \leq \phi^{n+1}, \\
 \alpha_{quad,2}(\phi^{n+1}) &= (3.67 \times \phi^{n+1} + 0.01004) & 0.00300 \leq \phi^{n+1} \leq 0.019160, \\
 &= (2.46 \times \phi^{n+1} + 0.03323) & 0.019160 \leq \phi^{n+1}, \\
 \alpha_{quad,3}(\phi^{n+1}) &= (20.54 \times \phi^{n+1} - 0.014566) & 0.00200 \leq \phi^{n+1}, \\
 \alpha_{quad,4}(\phi^{n+1}) &= (30.15 \times \phi^{n+1} + 0.013116) & 0.00090 \leq \phi^{n+1}.
 \end{aligned}
 \tag{18}$$



**Figure 2.** Normalized  $L_2$  errors in density for the Shu–Osher problem for each polynomial order of triangular elements. Red dots are selected design points for  $\alpha_0$  and  $\phi_0$ . For diverged simulation cases, the normalized  $L_2$  error values are replaced with the maximum  $L_2$  error value among stable solutions.

## 4. Numerical Simulation Results

### 4.1. Isentropic Vortex Problem

We first solve the isentropic vortex problem introduced in [7] for measuring the performance of our shock detector in the absence of discontinuities. This problem is well suited for evaluating whether a numerical scheme can preserve the initial shape of a vortex as well any initial disturbances in flow quantities. To obtain the optimum accuracy from the DG discretization, the shock detector should not flag any element in the domain as a troubled element, as the solution to this problem is smooth. Otherwise, the accuracy of a simulation will degrade due to the misuse of the filtering process. The initial non-dimensional conditions for this problem are as follows:

$$(\rho, u, v, p) = \left( \left[ 1 - \frac{(\gamma - 1)\epsilon^2}{8\gamma\pi^2} e^{(1-r^2)} \right]^{\frac{1}{\gamma-1}}, 1 - \frac{\epsilon}{2\pi} e^{0.5(1-r^2)} \bar{y}, 1 + \frac{\epsilon}{2\pi} e^{0.5(1-r^2)} \bar{x}, \rho^\gamma \right), \quad (19)$$

where  $(\bar{x}, \bar{y}) = (x - 5, y - 5)$ ,  $r^2 = \bar{x}^2 + \bar{y}^2$ , and  $\epsilon = 5.0$ . The physical domain becomes  $\Omega = [0, 10] \times [0, 10]$  and it is discretized with two different types of meshes: a fully unstructured mesh with triangular elements and a structured mesh with quadrilateral elements. The CFL value is 0.5 for calculating a global time step. As the time step for stability does not vary much over the domain, it is not necessary to use the time-accurate local time-stepping capabilities of ADER-DG.

The  $L_2$  errors in density are listed in Tables 1 and 2. Except for grid levels 1 to 3 in the  $n = 2$  quadrilateral element grid, the  $L_2$  errors are identical regardless of the filtering. Even for the quadrilateral elements with polynomial order two, the optimum accuracy is recovered as grids are refined. From this numerical test, the sensor and the filtering method proposed in Section 3 can identify a smooth solution in a flow and does not degrade the accuracy of the solution for both triangular and quadrilateral elements.

**Table 1.**  $L_2$  error of density on unstructured grids with triangular elements.

Grid Level	Element Length	$n = 2$		$n = 3$		$n = 4$	
		Filtering	W/O Filtering	Filtering	W/O Filtering	Filtering	W/O Filtering
1	10/4	2.112E-02	2.112E-02	2.708E-02	2.708E-02	1.332E-02	1.332E-02
2	10/6	1.598E-02	1.598E-02	1.422E-02	1.422E-02	4.114E-03	4.114E-03
3	10/8	9.339E-03	9.339E-03	6.474E-03	6.474E-03	1.135E-03	1.135E-03
4	10/12	3.376E-03	3.376E-03	1.179E-03	1.179E-03	1.434E-04	1.434E-04
5	10/16	1.843E-03	1.843E-03	3.415E-04	3.415E-04	3.417E-05	3.417E-05

**Table 2.**  $L_2$  error of density on structured grids with quadrilateral elements.

Grid Level	Element Length	$n = 2$		$n = 3$		$n = 4$	
		Filtering	W/O Filtering	Filtering	W/O Filtering	Filtering	W/O Filtering
1	10/4	3.535E-02	3.931E-02	2.580E-02	2.580E-02	1.815E-02	1.815E-02
2	10/6	1.643E-02	1.648E-02	1.130E-02	1.130E-02	6.296E-03	6.296E-03
3	10/8	1.300E-02	1.251E-02	6.668E-03	6.668E-03	1.033E-03	1.033E-03
4	10/12	5.167E-03	5.167E-03	2.611E-03	2.611E-03	1.440E-04	1.440E-04
5	10/16	2.099E-03	2.099E-03	8.461E-04	8.461E-04	3.923E-05	3.923E-05

### 4.2. Shock Tube Problem

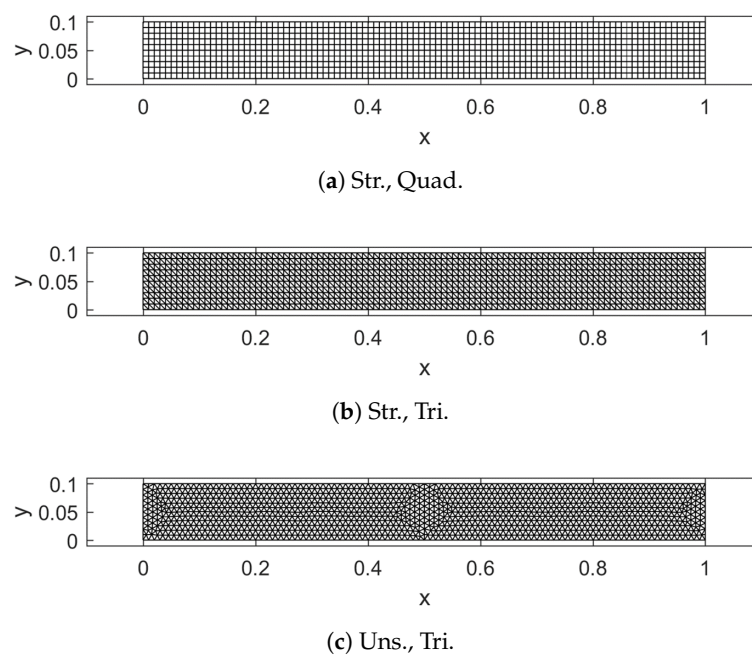
This numerical test case is a fundamental problem to prove whether a numerical method can capture discontinuities [35]. After the diaphragm is removed at  $t = 0$ , a left-running rarefaction wave, a right-running contact discontinuity, and a right-running shock wave are created. Since  $C^0$ -continuous

and discontinuous physical waves both exist in the problem, this test case is well suited to examine the performance of our shock-capturing capability. The initial conditions at  $t = 0$  are given by

$$\begin{aligned} (\rho, u, v, p) &= (1.0, 0.0, 0.0, p_L), & \text{if } x \leq 0.5, \\ &= (0.125, 0.0, 0.0, p_R), & \text{otherwise,} \end{aligned} \quad (20)$$

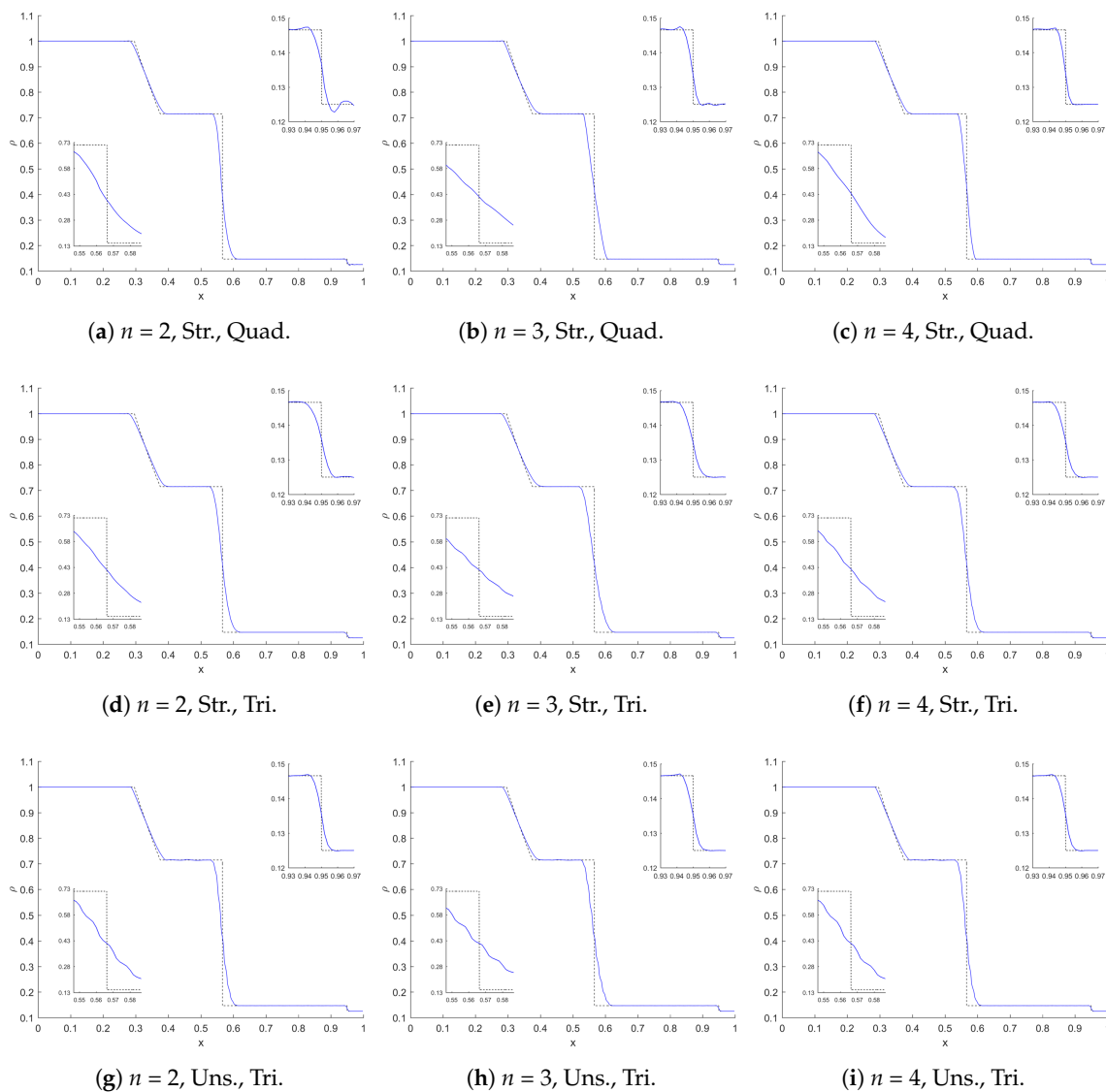
where  $p_L$  and  $p_R$  are pressure values on the left and right sides of the diaphragm, respectively. The CFL value used for calculating the explicit time step is 0.45 and each simulation runs until the right-running shock wave arrives at  $x = 0.95$ . For comparison, an exact solution is calculated using the method in [30] based on the selected initial pressure ratio at  $t = 0$ . In Sod's original work in [35],  $p_L = 1.0$  and  $p_R = 0.1$ . However, in our numerical experiments, we only fix the value of  $p_R$  to 0.1 and set  $p_L$  values to 0.2, 0.55, and 1.0. By selecting these  $p_L$  values, we can generate normal shock waves with normal Mach numbers of approximately 1.1, 1.38, and 1.66, respectively.

We choose the numerical domain as  $\Omega = [0, 1] \times [0, 0.1]$ . The domain is discretized with triangles or quadrilaterals of characteristic length 0.01. Figure 3 shows the three different meshes used in the simulations: a structured mesh with quadrilateral elements, a structured mesh decomposed into triangular elements, and an unstructured mesh with triangular elements. Periodic boundary conditions are imposed on the surface elements located at  $y = 0.0$  and  $y = 0.1$ .

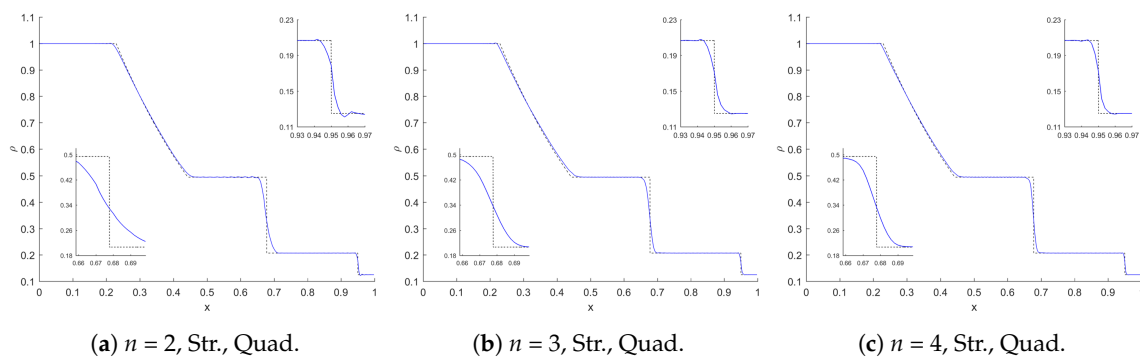


**Figure 3.** Three different meshes for shock tube problem: (a) structured mesh with quadrilateral elements; (b) structured mesh with triangular elements; (c) unstructured mesh with triangular elements.

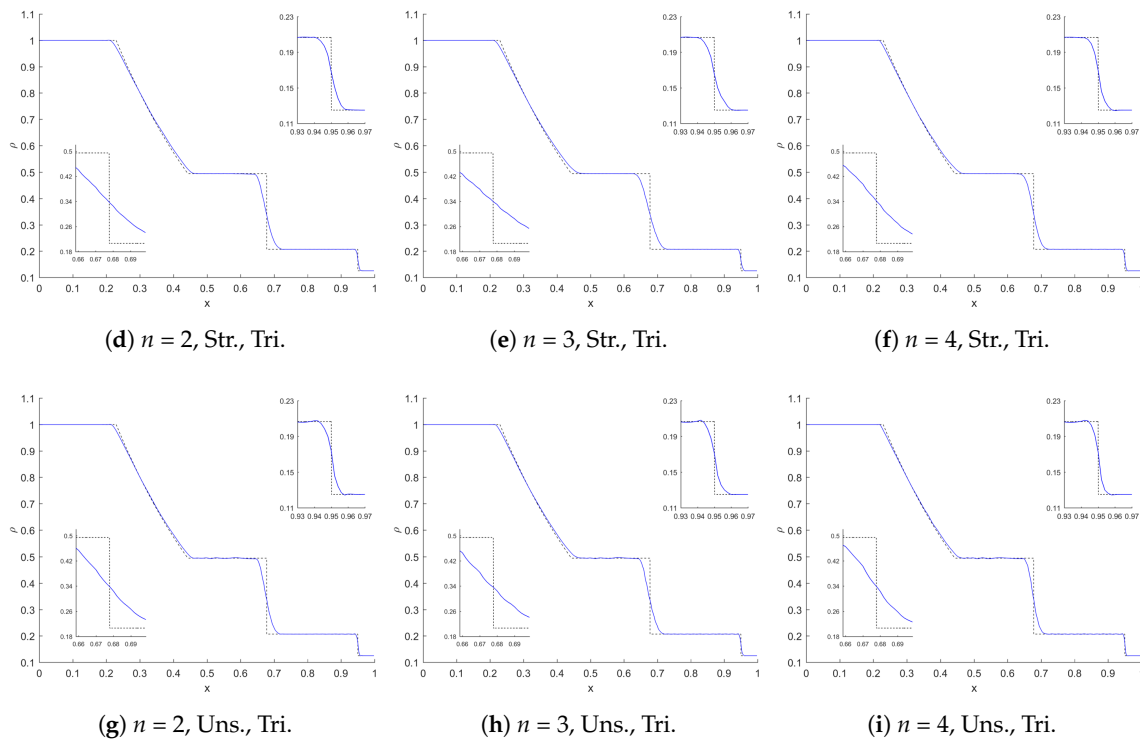
Numerical simulation results are shown in Figures 4–6. In each figure, the dashed black line represents the exact solution and the solid blue line is the numerical simulation result with the proposed sensor and filtering approach. Numerical solutions are interpolated along the line at  $y = 0.054$  with 501 equidistant sampling points. The shock detector identifies the shock and the contact discontinuity both in the two-dimensional structured and unstructured meshes. Furthermore, the filtering is appropriately applied to elements flagged as troubled elements to resolve both weak and strong discontinuities.



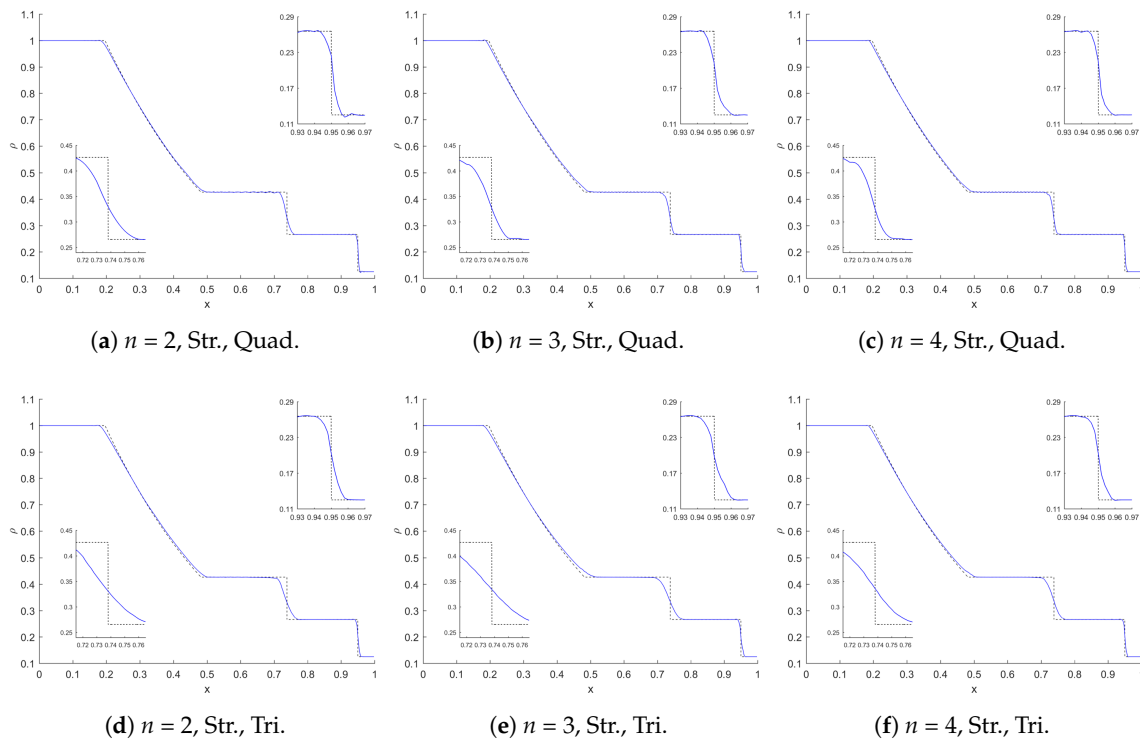
**Figure 4.** Two-dimensional shock tube problem with initial pressure ratio 2, various polynomial orders, and different types of grids. In each plot, the top-right inset figure is the result near the shock wave and the bottom-left inset figure is the result near the contact discontinuity.



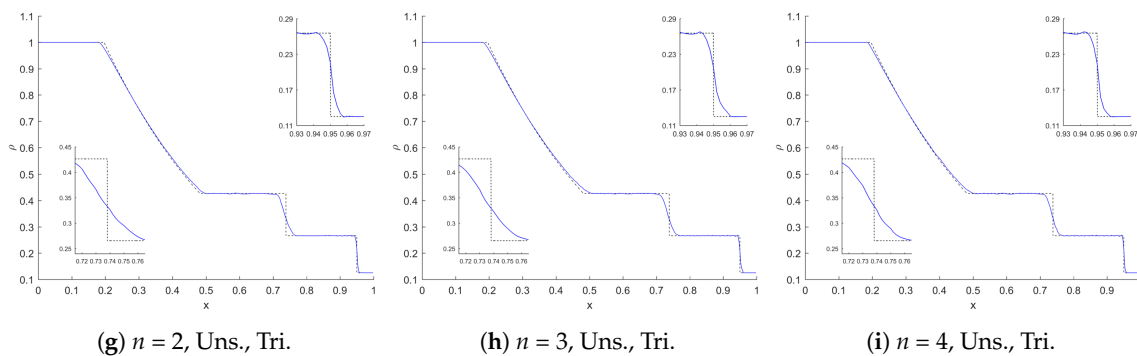
**Figure 5.** Cont.



**Figure 5.** Two-dimensional shock tube problem with initial pressure ratio 5.5, various polynomial orders, and different types of grids. In each plot, the top-right inset figure is the result near the shock wave and the bottom-left inset figure is the result near the contact discontinuity.



**Figure 6.** Cont.



**Figure 6.** Two-dimensional shock tube problem with initial pressure ratio 10, various polynomial orders, and different types of grids. In each plot, the top-right inset figure is the result near the shock wave and the bottom-left inset figure is the result near the contact discontinuity.

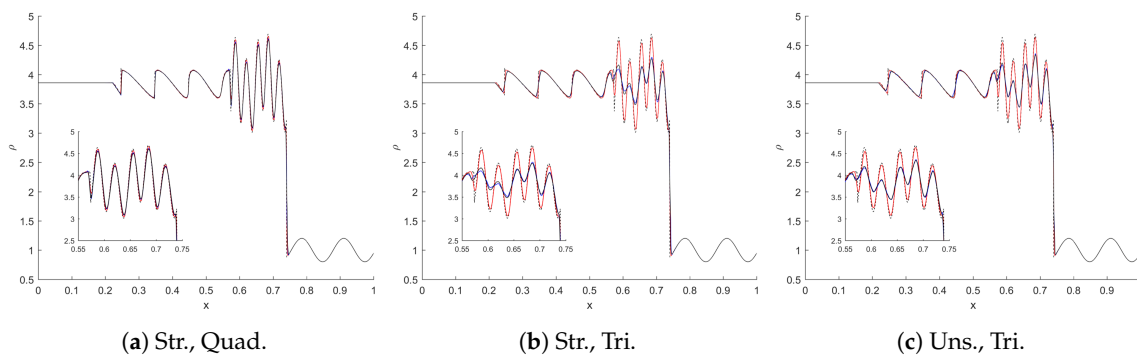
#### 4.3. Shu–Osher Problem

This problem was suggested by Shu and Osher in [36]. The physical domain is divided into a high-pressure part on the left side and a sinusoidal density wave on the right side of a diaphragm. Then, a Mach 3.0 shock wave propagates to the right and interacts with the sinusoidal wave after the diaphragm is removed. Robust and accurate shock-capturing is necessary to preserve the initial high-frequency behavior as well as local extrema after the shock is propagated. The initial non-dimensional conditions at  $t = 0$  are:

$$\begin{aligned}
 (\rho, u, v, p) &= \left( \frac{27}{7}, 2.629369, 0.0, \frac{31}{3} \right) & \text{if } x \leq 0.1, \\
 &= (1 + 0.2\sin(50x), 0.0, 0.0, 1.0) & \text{otherwise.}
 \end{aligned}
 \tag{21}$$

The numerical domain for this problem is  $\Omega = [0, 1] \times [0, 0.1]$  and the domain is discretized similarly as in Figure 3: a structured mesh with quadrilateral elements, a structured mesh decomposed into triangular elements, and an unstructured mesh with triangular elements. However, and unlike in the shock tube cases, we intentionally fix the total number of degrees of freedom along the  $x$ -direction to 1200. Consequently, elements with  $n = 2, 3$ , and  $4$  have characteristic lengths of 0.0025, 0.0033, and 0.004, respectively, regardless of the element type. By restricting the total number of degrees of freedom, we can compare the efficiency of the shock-capturing qualitatively as a function of the polynomial order of the discretized elements. Each simulation runs until  $t = 0.18$  and the CFL value is 0.3. A reference solution is obtained with 3,200 second-order elements in one dimension with the proposed sensor and filtering method in [27]. Periodic boundary conditions are imposed on the surface elements located at  $y = 0.0$  and  $y = 0.1$ .

Simulation results with different element types and polynomial orders are shown in Figure 7. The results are qualitatively the same for quadrilaterals regardless of the polynomial order. For triangular grids, polynomial order two elements preserve local extrema and high-frequency behavior almost the same as quadrilaterals, but the accuracy of the solutions for polynomial order three and four degrades significantly. However, in [27], we found that triangular elements with polynomial order three and four can also preserve accuracy after the shock if the characteristic length of the triangles is 0.0025. Therefore, we conclude that polynomial order two is cost effective in capturing shocks and pressure oscillations in these situations, compared to polynomial orders three and four for triangular elements with the sensor and the filtering methods.



**Figure 7.** Two-dimensional Shu–Osher problem with  $\beta_0 = 1.00$ . The dashed black line is the reference solution and the solid red, blue, black lines are interpolated solutions with 1,201 equidistant points for  $n = 2, 3$ , and 4 elements, respectively.

#### 4.4. Shock–Vortex Interaction

This problem was introduced by Jiang and Shu in [37]. A stationary shock resides within the domain and a right-running vortex interacts with a shock as it goes through it. To capture the highly-dynamic flow after the vortex traverses the shock, the detector and the filter must capture shocks robustly and precisely.

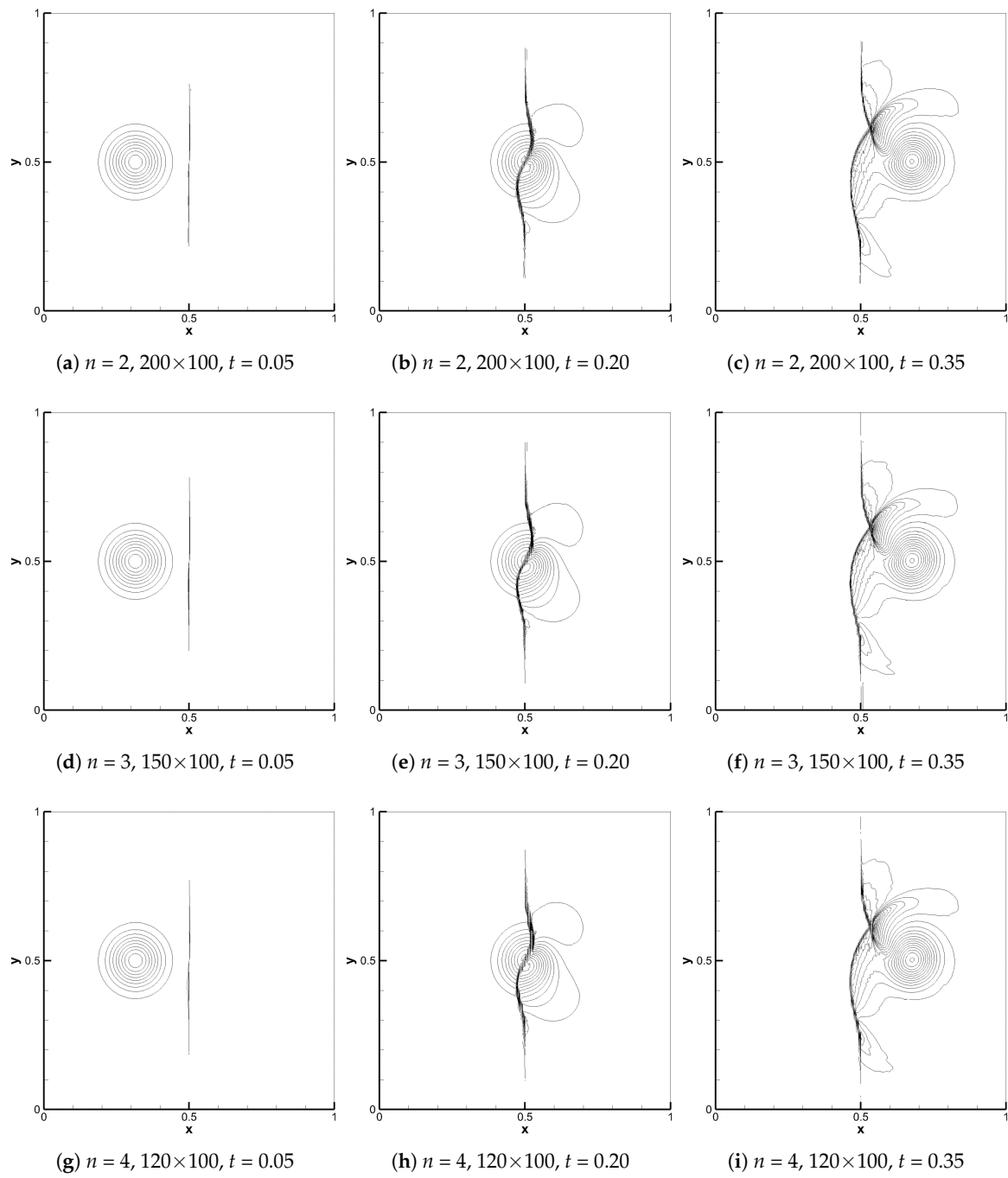
The spatial domain used is  $\Omega = [0, 2] \times [0, 1]$ , and it is discretized using a structured grid with quadrilateral elements. The grid spacing is uniform in the  $y$ -direction, but a finer spacing is used in the  $x$ -direction around the stationary shock location,  $x = 0.5$ . Similarly to the Shu–Osher problem, we fix the total number of degrees of freedom to 600 along the  $x$ -direction to evaluate the efficiency of capturing shocks as a function of the polynomial order of the quadrilateral elements. The non-dimensional initial conditions of the left side of the shock at  $t = 0$  are as follows:

$$(\rho, u, v, p) = \left( \left(1 - \frac{(\gamma - 1)\epsilon^2 e^{2\alpha(1-\tau^2)}}{4\alpha\gamma}\right)^{\frac{1}{\gamma-1}}, M\sqrt{\gamma} + \epsilon\tau e^{\alpha(1-\tau^2)} \sin\theta, -\epsilon\tau e^{\alpha(1-\tau^2)} \cos\theta, \rho^\gamma \right), \quad (22)$$

where  $M = 1.1$ ,  $\tan\theta = (y - y_c)/(x - x_c)$ ,  $\tau = r/r_c$ ,  $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ ,  $\epsilon = 0.3$ ,  $r_c = 0.05$ ,  $(x_c, y_c) = (0.25, 0.5)$ , and  $\alpha = 0.204$ . Initial conditions on the right side of the shock are calculated from the Rankine–Hugoniot conditions. Inviscid wall boundary conditions are imposed on the surface elements located at  $y = 0.0$  and  $y = 1.0$ . The CFL value used is 0.1 and each simulation runs until  $t = 0.8$ .

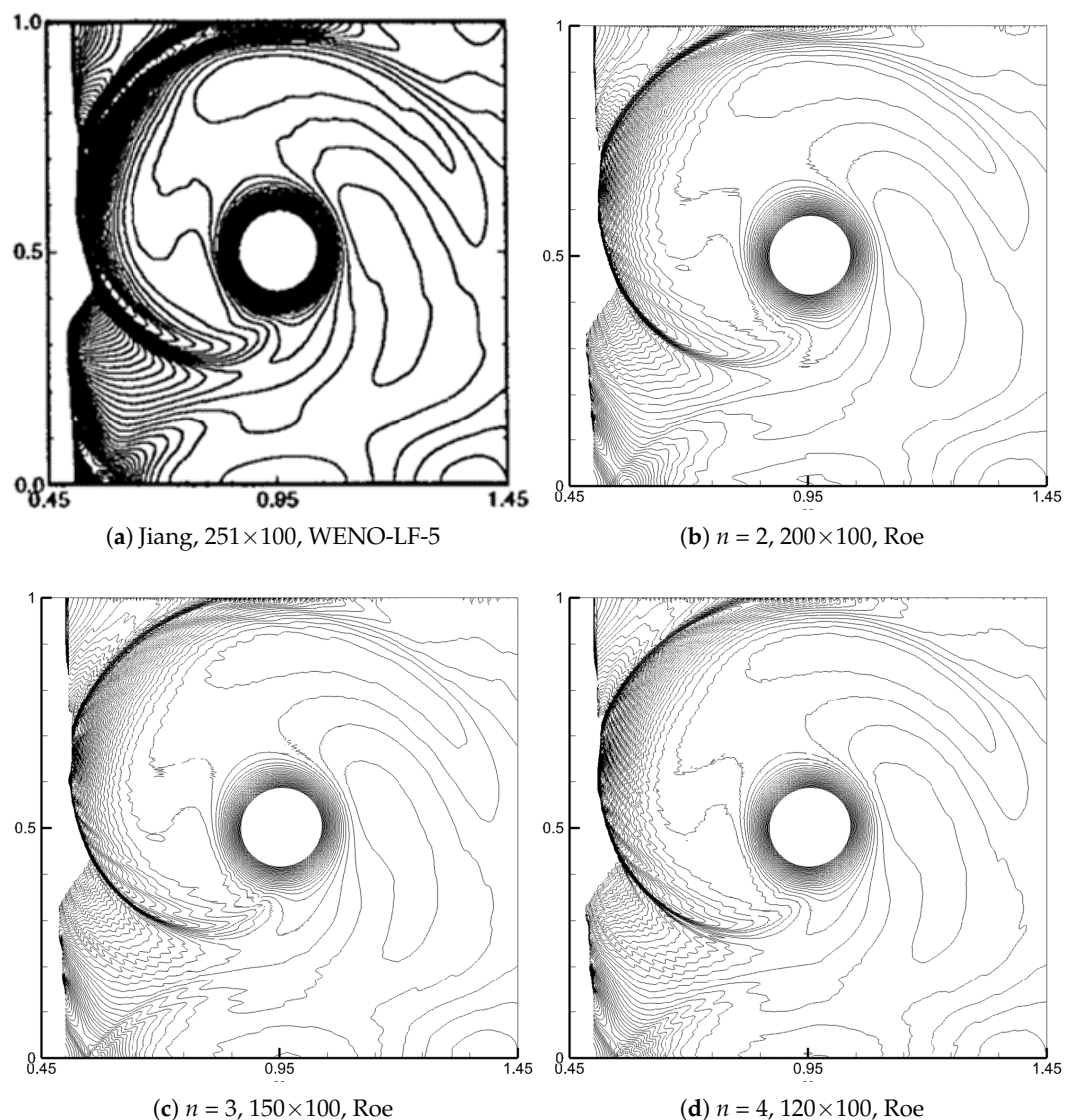
Figure 8 shows isolines of pressure at  $t = 0.05, 0.2$ , and  $0.35$ . The pressure contours are qualitatively identical for all polynomial orders at any time. Note that, in these figures, the isolines are not shown in some areas around the shock. This is especially the case for regions where the vortex has not disturbed the flow yet. As the value of the solution at the interfaces between elements is multiply defined in the DG discretization, i.e., allowing a jump in the solution, and the shock is located exactly at these interfaces, the postprocessing software does not show the isolines in these regions.

Pressure contours at  $t = 0.6$  are shown in Figure 9. For comparison purposes, we imported the result by [37] in Figure 9a. Pressure contours are qualitatively the same. Among three different polynomial orders, our  $n = 2$  solution exhibits less numerical noise compared to the  $n = 3$  and  $n = 4$  solutions.



**Figure 8.** Two-dimensional shock-vortex interaction problem at various times. Forty-one pressure contours are used from 0.84 to 1.50.





**Figure 9.** Two-dimensional shock-vortex interaction problem at  $t = 0.6$ . 90 pressure contours are used from 1.19 to 1.37. (a) WENO-LF-5 results by Jiang et al.

## 5. Conclusions

In this work, we further examined whether the shock detector in [27] can properly distinguish between shocks/discontinuities and regions of the flow with smooth solutions. In addition, we proposed problem-independent, optimized relationships between the shock detector and the filtering strength as a function of polynomial order and element type in two dimensions. By using the suggested relationships, weak and strong discontinuities can be captured robustly with no additional parameter tuning for two-dimensional elements with polynomial orders 2, 3, and 4. Moreover, the proposed method is based only on operations which utilize local information. The shock sensor and the filtering method have been implemented in the discontinuous Galerkin finite element method (DG-FEM) solver in the SU2 framework [38] for users to easily utilize and reproduce these results.

The whole method is extensible to three-dimensional elements as well as viscous problems in a straightforward way. In the future, we expect our implementation of the SU2 DG-FEM solver to be applied to problems where interactions between shock waves and turbulence are present. Additionally, we would like to exploit the proposed method in realistic industrial cases that require highly-scalable methods.

**Author Contributions:** Conceptualization, formal analysis, investigation, methodology, software, validation, visualization, and writing—original draft: J.H.C.; Supervision, project administration, writing—review and editing: J.J.A. and E.v.d.W.

**Funding:** This research was funded by the United States Department of Energy under the Predictive Science Academic Alliance Program 2 (PSAAP2) at Stanford University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, Q.; Ren, Y.X.; Li, W. Compact high order finite volume method on unstructured grids I: Basic formulations and one-dimensional schemes. *J. Comput. Phys.* **2016**, *314*, 863–882. [[CrossRef](#)]
2. Gao, X.; Owen, L.D.; Guzik, S.M. A high-order finite-volume method for combustion. In Proceedings of the AIAA Scitech 2016 Forum, San Diego, CA, USA, 4–8 January 2016; pp. 1–14.
3. Lele, S.K. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **1992**, *103*, 16–42. [[CrossRef](#)]
4. Deng, X.; Mao, M.; Tu, G.; Zhang, H.; Zhang, Y. High-order and high accurate CFD methods and their applications for complex grid problems. *Commun. Comput. Phys.* **2012**, *11*, 1081–1102. [[CrossRef](#)]
5. Allahyari, M.; Mohseni, K. Numerical simulation of flows with shocks and turbulence using observable methodology. In Proceedings of the AIAA Scitech 2018 Forum, Kissimmee, FL, USA, 8–12 January 2018; pp. 1–10.
6. Jause-Labert, C.; Godeferd, F.; Favier, B. Numerical validation of the volume penalization method in three-dimensional pseudo-spectral simulations. *Comput. Fluids* **2012**, *67*, 41–56. [[CrossRef](#)]
7. Shu, C.W. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 325–442.
8. van der Weide, E.; Giangaspero, G.; Svärd, M. Efficiency benchmarking of an energy stable high-order finite difference discretization. *AIAA J.* **2015**, *53*, 1845–1860. [[CrossRef](#)]
9. Reed, W.; Hill, T. *Triangular Mesh Methods for the Neutron Transport Equation*; Technical Report LA-UR-73-479; Los Alamos Scientific Laboratory: Los Alamos, NM, USA, 1973.
10. Cockburn, B.; Shu, C.W. The Runge–Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws. *Math. Model. Numer. Anal.* **1991**, *25*, 337–361. [[CrossRef](#)]
11. Cockburn, B.; Shu, C.W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comput.* **1989**, *52*, 411–435.
12. Cockburn, B.; Lin, S.Y.; Shu, C.W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comput. Phys.* **1989**, *84*, 90–113. [[CrossRef](#)]
13. Cockburn, B.; Hou, S.; Shu, C.W. The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case. *Math. Comput.* **1990**, *54*, 545–581.
14. Cockburn, B.; Shu, C.W. The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comput. Phys.* **1998**, *141*, 199–224. [[CrossRef](#)]
15. Hesthaven, J.S.; Warburton, T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, 1st ed.; Springer: New York, NY, USA, 2008.
16. Atkins, H.L.; Pampell, A. Robust and accurate shock capturing method for high-order discontinuous Galerkin methods. In Proceedings of the 20th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, USA, 27–30 June 2011; pp. 1–22.
17. Barter, G.; Darmofal, D. Shock capturing with higher-order, PDE-based artificial viscosity. In Proceedings of the 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, USA, 25–28 June 2007; pp. 1–14.
18. Burbeau, A.; Sagaut, P.; Bruneau, C.H. A problem-independent limiter for high-order Runge–Kutta discontinuous Galerkin methods. *J. Comput. Phys.* **2001**, *169*, 111–150. [[CrossRef](#)]
19. Clain, S.; Diot, S.; Loubère, R. A high-order finite volume method for systems of conservation laws—Multi-dimensional optimal order detection (MOOD). *J. Comput. Phys.* **2011**, *230*, 4028–4050. [[CrossRef](#)]
20. Dumbser, M.; Loubère, R. A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. *J. Comput. Phys.* **2016**, *319*, 163–199. [[CrossRef](#)]

21. Klöckner, A.; Warburton, T.; Hesthaven, J.S. Viscous shock capturing in a time-explicit discontinuous Galerkin method. *Math. Model. Nat. Phenom.* **2011**, *6*, 57–83. [[CrossRef](#)]
22. Lv, Y.; See, Y.C.; Ihme, M. An entropy-residual shock detector for solving conservation laws using high-order discontinuous Galerkin methods. *J. Comput. Phys.* **2016**, *322*, 448–472. [[CrossRef](#)]
23. Nguyen, C.; Peraire, J. An adaptive shock-capturing HDG method for compressible flows. In Proceedings of the 20th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, USA, 27–30 June 2011; pp. 1–20.
24. Persson, P.O.; Peraire, J. Sub-cell shock capturing for discontinuous Galerkin methods. In Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 9–12 January 2006; pp. 1–13.
25. Sheshadri, A. An analysis of stability of the flux reconstruction formulation with applications to shock capturing. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2016.
26. Vuik, M.J.; Ryan, J.K. Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes. *J. Comput. Phys.* **2014**, *270*, 138–160. [[CrossRef](#)]
27. Choi, J.H.; Alonso, J.J.; van der Weide, E. Simple shock detector for discontinuous Galerkin method. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019.
28. Roe, P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **1981**, *43*, 357–372. [[CrossRef](#)]
29. Lax, P.D. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Commun. Pure Appl. Math.* **1954**, *7*, 159–193. [[CrossRef](#)]
30. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*; Springer: Berlin/Heidelberg, Germany, 2009; Chapter 5,6,7,11.
31. Toro, E.F.; Millington, R.C.; Nejad, L. Towards very high order Godunov schemes. In *Godunov Methods: Theory and Applications*; Springer: Boston, MA, USA, 2001; pp. 907–940.
32. Dumbser, M.; Einaux, C.; Toro, E.F. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *J. Comput. Phys.* **2008**, *227*, 3971–4001. [[CrossRef](#)]
33. Rezzolla, L.; Zanotti, O. *Relativistic Hydrodynamics*; Oxford University Press: Oxford, UK, 2013.
34. Zanotti, O. *Discontinuous Galerkin Methods for Hyperbolic PDEs. Lecture 2*. In Proceedings of the Prospects in Theoretical Physics 2016, Princeton, NJ, USA, 18–29 July 2016; pp. 1–59. Available online: <https://static.ias.edu/pitp/2016/node/1026.html> (accessed on 10 July 2019).
35. Sod, G. Survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **1978**, *27*, 1–31. [[CrossRef](#)]
36. Shu, C.W.; Osher, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *J. Comput. Phys.* **1989**, *83*, 32–78. [[CrossRef](#)]
37. Jiang, G.S.; Shu, C.W. Efficient Implementation of Weighted ENO Schemes. *J. Comput. Phys.* **1996**, *126*, 202–228. [[CrossRef](#)]
38. SU2, The Open-Source CFD Code. Available online: <https://su2code.github.io> (accessed on 1 June 2019).

