INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 0000; **00**:1–44
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld

IMPICE Method for Compressible Flow Problems in Uintah.

L.T. Tran [1,2,*,†] and M. Berzins[1,2,‡]

[1]*School of Computing, University of Utah, Salt Lake City, UT 84112, U.S.A.*
[2]*The Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, U.S.A.*

SUMMARY

The Implicit Continuous-fluid Eulerian (ICE) method is a successful and widely used semi-implicit finite-volume method that applies to flows that range from supersonic to subsonic regimes. The classical ICE method has been expanded to problems in multiphase flow which span a wide area of science and engineering. The ICE method is utilized by the C-SAFE code Uintah written at the University of Utah to simulate explosions, fires and other fluid and fluid-structure interaction phenomena. The ICE method used in Uintah (referred to here as Production ICE) is described in many papers by Kashiwa at Los Alamos and Harman at Utah. However, Production ICE does not perform as well as many current methods for compressible flow problems governed by the Euler equations. We show, via examples, that changing the non-conservation form of the solver in Production ICE to a conservation form improves the numerical solutions. In addition, the use of slope limiters makes it possible to suppress the nonphysical oscillations generated by the ICE method in conservation form. This new form of ICE is referred to as IMPICE, the IMproved Production ICE method. The accuracy of IMPICE for one dimensional Euler equations is investigated by using a number of test cases. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Implicit Continuous-fluid Eulerian (ICE) method; Numerical method for Compressible Flow Problems; Improved ICE method using limiters; IMPICE Method for Compressible Flows; Improve Production ICE in Uintah; IMPICE spatial and temporal errors.

1. INTRODUCTION

The ICE method was developed by Harlow and Amsden in 1968 [11] with the aim of calculating the flows in all velocity ranges. The key idea for the method is to use a semi-implicit time discretization, in which the acoustic waves are treated implicitly while the advection terms are treated explicitly. As a result, the ICE method is able to remove the Courant stability limitation based on the speed of sound in the fluid. According to Harlow and Amsden [11], this is a numerically stable and efficient method for calculating transient, viscous fluid flows in several space dimensions. Harlow

†ltran@cs.utah.edu
‡mb@cs.utah.edu
*Correspondence to: L.T. Tran, School of Computing, University of Utah, 50 S. Central Campus Drive, Salt Lake City, UT 84112, U.S.A.

Copyright © 0000 John Wiley & Sons, Ltd.

and Amsden in 1971 [12] simplified the method and also greatly extended its scope of applicability. The ICE method by Harlow and Amsden was intended to simulate single-phase fluid dynamics problems. There are several improved versions of the ICE method using a pressure-correction solution procedure as seen in [4, 18, 19, 31, 37], and one typical pressure-correction method is referred as PISO (Pressure Implicit with Splitting of Operations). The ICE method was later extended by Harlow and Amsden in 1975 [13] and Kashiwa *et al.* in 1994 [21] to work with multiphase flow simulations. The ICE method by Harlow and Amsden used a staggered grid with normal velocity components at cell faces and all other variables at cell centers. As mentioned in Kashiwa and Lee [20], the main difficulties in the use of the staggered mesh include adding the artificial terms corresponding to a bulk viscosity to the equations in order to obtain reasonably smooth variation in density near shock waves and the development of spurious fluid as a result of a purely nonphysical circumstance. Kashiwa *et al.* [21] devised a new implementation of the ICE method by using a cell-centered scheme in which velocity is also located at the cell-center in an ongoing effort to deal with the difficulties in the classical approach. In the cell-centered approach, the velocity at cell faces is not computed directly, but is defined using the flow field or other dependent variables. Kashiwa and Lee [20] mention that definition of the face-centered velocity in the cell-centered scheme is a crucial matter for the robustness of the method. The fully cell-centered implementation of the ICE method in Kashiwa *et al.* [21] employs a conservative advection operator and a Lagrangian part which leaves a degree of freedom in the choice of conservation variables. The conservation laws used include at least those of mass, linear momentum, and internal energy (or alternatively the total energy). The Lagrangian part in most standard ICE implementations is fully conservative and it usually conserves the internal energy rather than the total energy.

Due to its general applicability, the ICE method has been used to study numerous complex flow problems, for example, fluidized dust beds, the flow of a liquid with entrained bubbles, atmospheric condensation with the fall of precipitation, the expansion and compression of a bubble formed by high-explosive gases under water and dynamics resulting from intense atmospheric explosions from the early time highly compressible flow [13, 21]. With its ability to handle complex flow problems, the ICE method for multiphase flows is also utilized by the C-SAFE code Uintah written at the University of Utah to simulate explosions, fires and other fluid and fluid-structure interaction phenomena [9]. The ICE code in the Uintah Framework will be referred to hereafter as Production ICE. The implementation of Production ICE is based on the cell-centered ICE method by Kashiwa *et al.* [21] with a few exceptions that will be discussed in detail in Section 3. The numerical scheme used in Production ICE [9, 10, 14, 27, 28] solves the conservation of mass, linear momentum and internal energy. However, the Lagrangian part in Production ICE is in non-conservative form which appears to be an exception to the standard ICE method. While this may not be a problem for some cases, it appears to be a problem when applying this Production ICE code to single-fluid cases that are governed by the Euler equations in which the obtained numerical solutions exhibit some discrepancies in the shock speeds and they additionally show unphysical oscillations. The

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Production ICE method for single-fluid cases solves the fluid flows that are governed by the one-dimensional, time-dependent Euler equations of gas dynamics given by

$$\begin{bmatrix} \rho \\ \rho u \\ \rho e \end{bmatrix}_t + \left(u \begin{bmatrix} \rho \\ \rho u \\ \rho e \end{bmatrix} \right)_x + \begin{bmatrix} 0 \\ p_x \\ p u_x \end{bmatrix} = 0, \tag{1}$$

where $\rho(x,t)$ is the density, $u(x,t)$ is the velocity, $p(x,t)$ is the pressure, and $e(x,t)$ is the internal energy per unit mass. The following equation of state is also employed:

$$p = (\gamma - 1)\rho e, \tag{2}$$

where γ is the specific heat ratio with the value of 1.4 for ideal gas.

It has been mentioned in [17, 26, 35] that nonconservative schemes approximating hyperbolic conservation laws do not converge to the correct solution in general. So the existence of discrepancies in the numerical solutions for non-linear hyperbolic systems using Production ICE is quite understandable. Therefore, in order to improve the Production ICE method, we first change the method to solve the system of one-dimensional Euler equations in conservation form where the total energy instead of the internal energy is conserved. The Improved Production ICE method, will be referred to hereafter as IMPICE, is a cell-centered ICE method which solves the system of one-dimensional Euler equations in conservation form that is given by

$$\begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix}_t + \left(u \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix} + \begin{bmatrix} 0 \\ p \\ p u \end{bmatrix} \right)_x = 0, \tag{3}$$

and the equation of state is given by

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

where $E(x,t)$ is the total energy per unit mass and $E = e + \frac{1}{2}u^2$. As a result of changing the Production ICE method to solve the system of Euler equations in conservation form, the computational results show the disappearance of the discrepancies in the obtained numerical solutions (refer to Appendix C). In Appendix C, the original method of Kashiwa *et al* [21] is referred to as Cell-centered ICE. However, Cell-centered ICE suffers from unphysical oscillations when there are moving contact discontinuities. Typically, methods in the literature use a variety of techniques such as constrained data reconstruction so as to avoid spurious oscillations; for example, [1, 2, 5, 39, 40, 41, 42]. To suppress oscillations, we will use a similar approach in which the data at the cell interface is the approximate Riemann solution to the local Riemann problem that is constructed by using slope-limited interpolation of the left and right cell-centered data. The approximate Riemann solver which was proposed by Harten *et al.* [15] and used by Davis [6] to satisfy consistency with the integral forms of the conservation law and entropy condition will be used to solve the local Riemann problem. The slope limiter used is selected from the extensive literature on the functions for slope limiters in the last few decades; see, for example,

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

[7, 16, 33, 36, 39, 40, 41, 43]. The IMPICE method is Cell-centered ICE with the oscillations being suppressed using the above-mentioned technique. In effect, although the original ICE method is a von-Neumann type method in which the fluxes are fully dependent on the time increment, we have now introduced a Riemann solver approach in which the fluxes depend directly on the approximate solution to the Riemann problem. The discussion of the methods in this paper will need the definition of the speed of sound. The speed of sound, $c(x,t)$, in an ideal gas is defined by

$$c = \sqrt{\frac{\gamma p}{\rho}}. \tag{5}$$

As for many numerical methods for the solution of PDEs, the IMPICE method approximates the Euler equations by a finite volume method using a spatial discretization of the problem and a time integration technique. Besides the importance of having a non-oscillating numerical solution, the numerical accuracy of the method in time and space is also important. The IMPICE method is first order in space and time. However, first-order methods are known to be not accurate enough to be used for large problems on relatively coarse grids. In order to increase the order of accuracy in time and space, we need to employ a nonlinear spatial discretization combined with a high order time discretization. Our goal is to obtain an IMPICE method with second-order accuracies in both time and space. In this paper, we will study numerical accuracy of the IMPICE method and discuss how to change the IMPICE method to achieve second-order accuracy in both space and time.

The content of this paper is organized as follows. In Section 2, we recap the cell-centered ICE method by Kashiwa *et al.* [21] which includes the spatial discretization and essential steps in the time integration. In Section 3, we present a detail implementation of the Production ICE method and describe differences between it and the cell-centered ICE method by Kashiwa *et al.* [21]. In Section 4, we discuss the proposed method by Kwatra *et al.* [23] that can be applied to calculate the time integration step of the semi-implicit ICE method. This method removes the restriction of sound speed in calculating the time step, but still maintains stability. In Section 5, we propose a modification to the Production ICE method to remove the unphysical oscillations in the numerical solutions. The numerical solutions of the IMPICE method are presented and compared to the numerical solutions of the Production ICE method in Section 6. The spatial and temporal accuracies of the IMPICE method are shown in Section 7. The issue of how to obtain second-order accuracy in time and space is discussed in Section 8 and Section 9. As it will be discussed in Section 2, there are many degrees of freedom in the implementation of the cell-centered ICE method by Kashiwa *et al* [21], so the purpose of this paper is not only to find an improved implementation of the Production ICE method but also to discuss how various choices in the implementation of the cell-centered ICE method affect the obtained numerical solutions.

2. THE CELL-CENTERED ICE METHOD BY KASHIWA *ET AL.* [21]

2.1. General description of the cell-centered ICE method

The cell-centered ICE method described in detail in [21] is a finite-volume solver in which space is discretized into N uniform cells of width $\Delta x = (b-a)/N$ where $[a,b]$ is the spatial domain. The

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

cells are centered around $x_j = a + (j - 1/2)\Delta x$ where $j = 1..N$. The boundaries of these cells are located at $x_{j+\frac{1}{2}} = a + j\Delta x$ where $j = 0..N$ and also called face-centers or cell interfaces. With this discretization, the domain boundaries are aligned with the first and last cell edges. A time integration method is used to estimate the averages of cell variables at some time $t = T_{end}$ from the averages of cell variables at $t = 0$. For each time integration step, assuming that the cell averages at time t_n are known, the goal is to compute the averages of cell variables at the next time step t_{n+1}.

The ICE method invokes operator splitting in which the solution consists of a Lagrangian phase and an Eulerian phase. The Lagrangian phase advances cell values without advection and maps new values to cell variables and the Eulerian phase advects the cell variables. The essential point that makes the ICE method an all-speed scheme is to use an implicit scheme for the Lagrangian phase and an explicit scheme for the Eulerian phase. The cell-centered ICE method comprises the following phases:

The Primary Phase:

With the spatial discretization as discussed above, the spatial derivatives in the governed equations are approximated using finite differences of quantities at face-centers. Since an implicit scheme is used for the Lagrangian phase, the variables involved in the Lagrangian phase are those evaluated at face-centers at $t_n + \frac{\Delta t}{2}$ and are determined in this phase. It is also necessary in this phase to estimate the fluxing velocity which is going to be used in the Eulerian phase. The fluxing velocity, $u^*_{j+\frac{1}{2}}$, is the flux of volume across the cell interface. In order to make clear which variables are defined at face center, the superscript * is used here for these variables as required.

The Lagrangian Phase:

Let V_j^n be the volume of cell j and U_j^n be the vector of averaging cell variables at t_n. In particular, V_j^n is equal to Δx for the above discretization. Assume that the cell volume is changed during the Lagrangian phase to V_j^L and $V_j^L = V_j^n + \Delta t(u^*_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}})$ where $u^*_{j-\frac{1}{2}}$ and $u^*_{j+\frac{1}{2}}$ are fluxing velocities at cell interfaces. There is also a change in the vector of averaging cell variables to U_j^L after the Lagrangian phase has been completed. A numerical scheme obtained from neglecting the convective effects is used to evaluate the change in the material state and in turn evaluate U_j^L.

The Eulerian Phase:

For this phase, we have to evaluate the change in the solution due to advection. Let V_j^{n+1} be the cell volume at t_{n+1} and assume that the mesh is stationary, then $V_j^{n+1} = \Delta x$. The change in the solution due to advection is as follows

$$V_j^{n+1}U_j^{n+1} = V_j^L U_j^L - \Delta t(u^*_{j+\frac{1}{2}}\langle U\rangle_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}}\langle U\rangle_{j-\frac{1}{2}}), \tag{6}$$

where $\langle U\rangle_{j+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} U(x_{j+\frac{1}{2}}, t)dt$ is the vector of advected quantities and is numerically determined. As suggested by [21], this numerical value may be determined using U_j^n or U_j^L; however, how to numerically determine this is not explicitly presented.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

State Variables Update Phase:

In this phase we update the primitive cell variables using U_j^{n+1} and the equation of state as these will be used for the following time integration step.

2.2. The cell-centered ICE implementation of Kashiwa et al. *[21]*

Kashiwa *et al.*'s improvements, [21], to ICE consist of changes to the above primary and Lagrangian phases. There are two important quantities that will be used in these phases, these are the face-centered velocity and pressure as denoted by $u^*_{j+\frac{1}{2}}$ and $p^*_{j+\frac{1}{2}}$. The face-centered velocity, $u^*_{j+\frac{1}{2}}$, is also the advected speed for the Eulerian phase, so [21] refers this as the fluxing velociy. The face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, is calculated based on the time-advanced equation for velocity. For the system of the Euler equations, the derived equation for velocity evolution is

$$u_t + uu_x = -\frac{p_x}{\rho}. \tag{7}$$

Equation (7) is written in Lagrangian form as

$$\frac{Du}{Dt} = -\frac{p_x}{\rho}. \tag{8}$$

The fluxing velocity in the cell-centered ICE method is obtained using the flow field and semi-implicit Euler scheme in the Lagrangian frame by

$$u^*_{j+\frac{1}{2}} = \langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho - \frac{\Delta t}{2\Delta x}\frac{p_{j+1}^{n+\frac{1}{2}} - p_j^{n+\frac{1}{2}}}{\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle}, \tag{9}$$

where $p_j^{n+\frac{1}{2}}$ is the cell-centered pressure at $t_n + \frac{\Delta t}{2}$, $\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho$ is the mass-weighted average face-centered velocity and $\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle$ is the average face-centered density at time t_n. The mass-weighted average velocity, $\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho$, of left and right states at face-center is given by

$$\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho = \frac{\rho^n_j u^n_j + \rho^n_{j+1}u^n_{j+1}}{\rho^n_j + \rho^n_{j+1}}, \tag{10}$$

and the average face-centered density, $\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle$, of left and right cell-centered densities is:

$$\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle = \frac{\rho^n_j + \rho^n_{j+1}}{2}. \tag{11}$$

As the face-centered pressure at $t_n + \frac{\Delta t}{2}$, $p_j^{n+\frac{1}{2}}$, is not readily available, it needs to be obtained from correcting the face-centered pressure at t_n, p^n_j. Let $\delta p^n_j = p_j^{n+\frac{1}{2}} - p^n_j$ be the difference between the cell-centered pressures at these two time levels and which is referred to in Kashiwa *et al.* [21] as "pressure corrector", equation (9) is then rewritten as

$$u^*_{j+\frac{1}{2}} = \tilde{u}^*_{j+\frac{1}{2}} - \frac{\Delta t}{2\Delta x}\frac{\delta p^n_{j+1} - \delta p^n_j}{\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle}, \tag{12}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

where

$$\tilde{u}^*_{j+\frac{1}{2}} = \langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho - \frac{\Delta t}{2\Delta x}\frac{p^n_{j+1} - p^n_j}{\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle}. \tag{13}$$

In order to determine the face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, the "pressure corrector" values, δp^n_{j+1} and δp^n_j, need to be determined using the equation of pressure evolution,

$$p_t + up_x = -c^2\rho u_x, \tag{14}$$

which is written in Lagrangian form as

$$\frac{Dp}{Dt} = -c^2\rho u_x. \tag{15}$$

In [21], two different ways to determine the "pressure corrector" values are discussed. The explicit "pressure corrector" uses the explicit discrete form and the implicit "pressure corrector" uses the semi-implicit discrete form of (15). The explicit form of "pressure corrector" is given by

$$\delta p^n_j = -\frac{\Delta t}{2}u^n_j\left(\frac{p^n_{j+1} - p^n_{j-1}}{2\Delta x}\right) - \frac{\Delta t}{2\Delta x}(c^2\rho)^n_j(\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho - \langle\langle u^n_{j-\frac{1}{2}}\rangle\rangle^\rho). \tag{16}$$

The implicit "pressure corrector" is obtained using

$$\delta p^n_j = -\frac{\Delta t}{2}u^n_j\left(\frac{p^n_{j+1} - p^n_{j-1}}{2\Delta x}\right) - \frac{\Delta t}{2\Delta x}(c^2\rho)^n_j(u^*_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}}). \tag{17}$$

The implicit "pressure corrector" is complicated since equations (12) and (17) show that the face-centered fluxing velocities and the "pressure corrector" values are interrelated, so there is a need to calculate the fluxing velocities using these two equations and this is not explicitly discussed in [21]. After having determined the "pressure corrector" values at cell centers, the face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, is determined using equation (12) where $\tilde{u}^*_{j+\frac{1}{2}}$ is defined in (13).

There are several suggested choices for calculating the face-centered pressure, $p^*_{j+\frac{1}{2}}$, in [21] and [22]. Two of these choices, which are derived from the pressure equation in Kashiwa *et al* in 1994 [21], will be discussed in Appendix D. In this paper, we employ the choice that is mentioned in Kashiwa in 2001 [22]. This choice aims to satisfy continuity of acceleration by equating acceleration increments for the left/right half spaces. The equation as specified in [22] is:

$$p^*_{j+\frac{1}{2}} = \left(\frac{\frac{1}{\rho^n_{j+1}}p^{n+\frac{1}{2}}_{j+1} + \frac{1}{\rho^n_j}p^{n+\frac{1}{2}}_j}{\frac{1}{\rho^n_j} + \frac{1}{\rho^n_{j+1}}}\right). \tag{18}$$

In the above equation, the face-centered pressure is thus calculated using specific volumes-weighted of the left and right time-advanced cell-centered pressures.

The Primary Phase is executed after the face-centered velocity and pressure, $u^*_{j+\frac{1}{2}}$ and $p^*_{j+\frac{1}{2}}$, are calculated. The choice of the vector of conserved variables, U, and the numerical procedure to determine the vector of the face-centered advected quantities, $\langle U\rangle_{j+\frac{1}{2}}$, are neccessary for the implementation of the Lagrangian phase and the Eulerian phase.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

3. PRODUCTION ICE IN UINTAH COMPUTATIONAL FRAMEWORK

The term Production ICE is used to denote the ICE method as implemented in Uintah Computational Framework by [9, 10, 14, 27, 28] to simulate fluid flows that are governed by the Euler and Navier Stokes equations. Production ICE solves the Euler system in non-conservation form (1) with the vector of variables $U = [\rho, \rho u, \rho e]^T$. The detail implementation of the phases in Production ICE follows the description given in Kashiwa *et al.* [21] with some exceptions that will be pointed out explicitly in the following discussion.

3.1. The Primary Phase

The first exception is that the face-centered quantities in Production ICE are not time-centered. The face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, and pressure, $p^*_{j+\frac{1}{2}}$, for the time step $[t_n, t_{n+1}]$ are approximated at the face-center at time t_{n+1}. For this reason, we use the notations $u^{**}_{j+\frac{1}{2}}$ and $p^{**}_{j+\frac{1}{2}}$ for the face-centered fluxing velocity and pressure in Production ICE. Using a different approach from equation (9), the face-centered fluxing velocity in Production ICE, $u^{**}_{j+\frac{1}{2}}$, is approximated as:

$$u^{**}_{j+\frac{1}{2}} = \langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho - \frac{\Delta t}{\Delta x}\frac{p^n_{j+1} - p^n_j}{\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle}, \tag{19}$$

where the mass-weighted average velocity and the average face-centered density, $\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho$ and $\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle$, are defined in (10) and (11). So another exception in calculating the fluxing velocity in Production ICE is that the scheme in (19) is not semi-implicit when the pressures used are defined at t_n.

The face-centered pressure, $p^{**}_{j+1/2}$, in Production ICE is calculated using the following equation

$$p^{**}_{j+\frac{1}{2}} = \left(\frac{\frac{1}{\rho^n_{j+1}}p^{n+1}_{j+1} + \frac{1}{\rho^n_j}p^{n+1}_j}{\frac{1}{\rho^n_j} + \frac{1}{\rho^n_{j+1}}}\right). \tag{20}$$

This is similar to equation (18), but with the cell-centered pressures at time t_{n+1}, p^{n+1}_j, where p^{n+1}_j is evaluated using an explicit scheme applied to the Lagrangian form of the equation of pressure evolution in (15), which is given by

$$p^{n+1}_j = p^n_j - \frac{\Delta t}{\Delta x}(c^2\rho)^n_j(\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho - \langle\langle u^n_{j-\frac{1}{2}}\rangle\rangle^\rho). \tag{21}$$

3.2. The Lagrangian Phase

Production ICE chooses the vector of conserved variables to include mass, linear momentum and internal energy. The use of the non-conservative form of the system of Euler equations in (1) means that the Lagrangian part in Production ICE is given by

$$V^L_j U^L_j = V^n_j U^n_j - \Delta t \begin{bmatrix} 0 \\ p^{**}_{j+1/2} - p^{**}_{j-1/2} \\ p^{n+1}_j(u^{**}_{j+1/2} - u^{**}_{j-1/2}) \end{bmatrix}. \tag{22}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

3.3. The Eulerian Phase

The change in solution values due to advection over the step $[t_n, t_{n+1}]$ is given by

$$V_j^{n+1}U_j^{n+1} = V_j^L U_j^L - \Delta t(u^{**}_{j+\frac{1}{2}}\langle U\rangle_{j+\frac{1}{2}} - u^{**}_{j-\frac{1}{2}}\langle U\rangle_{j-\frac{1}{2}}). \tag{23}$$

However, the numerical values of face-centered advected quantities in the following definition:

$$\langle U\rangle_{j+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} U(x_{j+\frac{1}{2}}, t)dt \tag{24}$$

has not been quantified so far in this paper and we will now show how to approximate it. Normally $U(x_{j+\frac{1}{2}}, t)$ is not constant for the step $[t_n, t_{n+1}]$, but a first-order accuracy is obtained by assuming that this is constant and is an upwinded cell-centered value. However, there are the cell-centered values at two different time levels that are available for the Eulerian phase. These are at t_n and the value after the Lagrangian step. By chosing the upwinded cell-centered values at time t_n for the face-centered advected quantities, we have

$$\langle U\rangle_{j+\frac{1}{2}} = \begin{cases} U^n_{j+1} & if \quad u^*_{j+\frac{1}{2}} < 0 \\ U^n_j & otherwise. \end{cases} \tag{25}$$

Alternatively, if the upwinded cell-centered values at Lagrangian time level are considered for face-centered advected quantities, we have

$$\langle U\rangle_{j+\frac{1}{2}} = \begin{cases} U^L_{j+1} & if \quad u^*_{j+\frac{1}{2}} < 0 \\ U^L_j & otherwise. \end{cases} \tag{26}$$

The Production ICE code uses (25) to define the face-centered advected quantities, [14, 27, 28].

3.4. State Variables Update Phase

The averages of cell primitive variables ρ,u,e, and p are then updated using the averages of cell variables ρ,ρu,ρe and the equation of state (2).

4. CFL CONDITION

The choice of the time step Δt in time integration affects the stability of the ICE method. As mentioned in [34], one requirement for the method to be stable is the fastest wave at a given time is allowed to travel, at most, one cell length Δx in the chosen time step Δt. For the system of Euler equations, the time step Δt is chosen to satisfy the condition:

$$\Delta t = \frac{C_{cfl}\Delta x}{S^n_{max}}, \tag{27}$$

where C_{cfl} is a Courant or CFL coefficient satisfying $0 < C_{cfl} < 1$ and S^n_{max} is the largest wave speed present through the domain at time t_n. A practical choice of S^n_{max} as mentioned in Toro [34]

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

is

$$S^n_{max} = \max_j (|u^n_j| + c^n_j). \tag{28}$$

However, Kwatra *et al.* [23] proposed a novel method for alleviating the stringent CFL condition imposed by the sound speed in simulating highly nonlinear compressible flow with shocks, contacts and rarefactions. It is mentioned in [23] that the maximum speed in equation (28) is too restrictive for flows where the sound speed, c, may be much larger than $|u|$, so the stringent CFL time step restriction imposed by the acoustic waves can be avoided and only the material velocity CFL restriction is used in calculating the maximum speed. The proposed method of [23] is well suited to the semi-implicit solver like the ICE method where only the advection step is the explicit part. The proposed maximum speed calculation in [23] is:

$$S^n_{max} = \max_j |u^n_j|. \tag{29}$$

The time step used in this paper for both Production ICE method and IMPICE method is determined using (27) where S^n_{max} is calculated using (29).

5. IMPICE METHOD

The IMPICE method improves the numerical solutions to the one-dimensional, time-dependent Euler equations of gas dynamics by solving the conservation form in (3). However, as will be validated by numerical experiments in Appendix C, the numerical solutions obtained using the cell-centered ICE method in conservation form have unphysical oscillations that need to be reduced or eliminated. The oscillations in the numerical solutions of the cell-centered ICE method cannot be diminished by decreasing the time step , so in this section, we will describe the algorithm used to suppress these oscillations numerically by using a simple approximate Riemann solver.

5.1. Numerical Discussion

To help explain the IMPICE method, we start with a discussion of schemes that approximate conservation laws as follows and consider a one-dimensional system in a conservation law form

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial F(U(x,t))}{\partial x} = 0, \quad x \in [a,b] \quad and \quad t \geq 0, \tag{30}$$

where $U(x,t)$ is the vector of conserved variables and $F(U(x,t))$ is the vector of fluxes. In order to approximate the solution of (30) with the initital condition

$$U(x,0) = U^0(x), \tag{31}$$

we discretize space into N uniform cells as in Section 2. The cell average of the cell $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ at time t_n is denoted by U^n_j, where

$$U^n_j = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(x,t_n)dx. \tag{32}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

A standard approach is used in integrating system (30) in space and time in the control volume $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [t_n, t_{n+1}]$ to give:

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} [U(x,t_{n+1}) - U(x,t_n)]\, dx = -\int_{t_n}^{t_{n+1}} \left[F(U(x_{j+\frac{1}{2}},t)) - F(U(x_{j-\frac{1}{2}},t))\right] dt.$$

This can then be written in the standard conservation form:

$$\Delta x U_j^{n+1} = \Delta x U_j^n - \Delta t \left(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\right) \tag{33}$$

where

$$F_{j+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} F(U(x_{j+\frac{1}{2}},t))dt. \tag{34}$$

Equation (33) is used by finite volume methods to solve the system (30) approximately. In order to use this relation, a spatial integration of the initital condition is required and the approximations of the fluxes at the cell interfaces are needed.

The numerical flux derivation follows the cell-centered ICE method of Kashiwa *et al.* [21] will be derived shortly. The system of Euler equations (3) of gas dynamics is written in the form (30) where $U = (\rho, \rho u, \rho E)^T$ and $F(U) = (\rho u, \rho u^2 + p, \rho u E + up)^T$. The face-centered flux (34) in this case is written as

$$F_{j+\frac{1}{2}} = \begin{bmatrix} \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u)(x_{j+\frac{1}{2}},t)dt \\ \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u^2 + p)(x_{j+\frac{1}{2}},t)dt \\ \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u E + pu)(x_{j+\frac{1}{2}},t)dt \end{bmatrix}. \tag{35}$$

A Taylor series approximation of $u(x_{j+\frac{1}{2}},t)$ is given by

$$u(x_{j+\frac{1}{2}},t) = u(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}}) + (t - t_{n+\frac{1}{2}})u_t(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}}) + O(\Delta t^2). \tag{36}$$

Using the notations $u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = u(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}})$ and $(u_t)_{j+\frac{1}{2}}^{n+\frac{1}{2}} = u_t(x_{j+\frac{1}{2}},t_{n+\frac{1}{2}})$, we have:

$$\begin{aligned} \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u)(x_{j+\frac{1}{2}},t)dt &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \rho(x_{j+\frac{1}{2}},t)\left[u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + (t - t_{n+\frac{1}{2}})(u_t)_{j+\frac{1}{2}}^{n+\frac{1}{2}} + O(\Delta t^2)\right] dt \\ &= u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \rho(x_{j+\frac{1}{2}},t)dt\right) + O(\Delta t^2). \end{aligned}$$

With a similar approach, we also have

$$\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u^2 + p)(x_{j+\frac{1}{2}},t)dt = u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u)(x_{j+\frac{1}{2}},t)dt\right) + p_{j+\frac{1}{2}}^{n+\frac{1}{2}} + O(\Delta t^2)$$

and

$$\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u E + up)(x_{j+\frac{1}{2}},t)dt = u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho E)(x_{j+\frac{1}{2}},t)dt\right) + (up)_{j+\frac{1}{2}}^{n+\frac{1}{2}} + O(\Delta t^2).$$

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

Then face-centered flux $F_{j+\frac{1}{2}}$ vector in (35) is rewritten as

$$F_{j+\frac{1}{2}} = u_{j+\frac{1}{2}}^{n+\frac{1}{2}} \begin{bmatrix} \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \rho(x_{j+\frac{1}{2}},t)dt \\ \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho u)(x_{j+\frac{1}{2}},t)dt \\ \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} (\rho E)(x_{j+\frac{1}{2}},t)dt \end{bmatrix} + \begin{bmatrix} 0 \\ p_{j+\frac{1}{2}}^{n+\frac{1}{2}} \\ (up)_{j+\frac{1}{2}}^{n+\frac{1}{2}} \end{bmatrix} + O(\Delta t^2). \tag{37}$$

The reader should note that equation (33) with the terms $F_{j-\frac{1}{2}}$ and $F_{j+\frac{1}{2}}$ are defined by equation (37) will be used in Section 7 to assess the numerical accuracy of the IMPICE method.

5.2. IMPICE Implementation

The scheme used for approximating the fluxing velocity, $u^*_{j+\frac{1}{2}}$, in the IMPICE method is similar to equation (9), that is:

$$u^*_{j+\frac{1}{2}} = u^n_{j+\frac{1}{2}} - \frac{\Delta t}{2\Delta x}\frac{\left(p_{j+1}^{n+\frac{1}{2}} - p_j^{n+\frac{1}{2}}\right)}{\rho^n_{j+\frac{1}{2}}}. \tag{38}$$

This equation was obtained by replacing $\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho$ in (9) with $u^n_{j+\frac{1}{2}}$, and $\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle$ with $\rho^n_{j+\frac{1}{2}}$. While these quantities denote the velocity and density at face-center at t_n, their numerical values are determined differently. While $\langle\langle u^n_{j+\frac{1}{2}}\rangle\rangle^\rho$ and $\langle\langle \rho^n_{j+\frac{1}{2}}\rangle\rangle$ are determined using the weighted averages in (10) and (11), the values $u^n_{j+\frac{1}{2}}$ and $\rho^n_{j+\frac{1}{2}}$ are determined based on the simple approximate Riemann solver that will be discussed in Section 5.3 below.

Since the pressures used in the (38) are time-advanced values, we need to perform a "pressure corrector" to obtain these. The explicit "pressure corrector" in (16) is the one used in our implementation of the IMPICE method. However, it is worth looking at the implicit "pressure corrector" and seeing the difference between the solutions of these two methods. By substituting (12) into (17), the equation for cell-centered "pressure corrector" now becomes

$$\begin{aligned} \delta p^n_j = &-\frac{\Delta t}{2}u^n_j\left(\frac{p^n_{j+1} - p^n_{j-1}}{2\Delta x}\right) - \frac{\Delta t}{2\Delta x}(c^2\rho)^n_j(\tilde{u}^*_{j+\frac{1}{2}} - \tilde{u}^*_{j-\frac{1}{2}}) \\ &+ \left[\frac{\Delta t}{2\Delta x}\right]^2 (c^2\rho)^n_j\left[\frac{\delta p^n_{j+1} - \delta p^n_j}{\rho^n_{j+\frac{1}{2}}} - \frac{\delta p^n_j - \delta p^n_{j-1}}{\rho^n_{j-\frac{1}{2}}}\right]. \end{aligned} \tag{39}$$

Let $\sigma = \frac{\Delta t}{2\Delta x}$ and rearrange the terms of above equation to get

$$\begin{aligned} &\left[1 + \sigma^2\frac{(c^2\rho)^n_j}{\rho^n_{j+\frac{1}{2}}} + \sigma^2\frac{(c^2\rho)^n_j}{\rho^n_{j-\frac{1}{2}}}\right]\delta p^n_j - \sigma^2\frac{(c^2\rho)^n_j}{\rho^n_{j+\frac{1}{2}}}\delta p^n_{j+1} - \sigma^2\frac{(c^2\rho)^n_j}{\rho^n_{j-\frac{1}{2}}}\delta p^n_{j-1} \\ &\qquad = -\sigma u^n_j\left(\frac{p^n_{j+1} - p^n_{j-1}}{2}\right) - \sigma(c^2\rho)^n_j(\tilde{u}^*_{j+\frac{1}{2}} - \tilde{u}^*_{j-\frac{1}{2}}). \end{aligned} \tag{40}$$

Therefore, the values δp^n_j are the solutions of the tri-diagonal linear system:

$$Ax = b \tag{41}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

where:

$$A = \begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & . & \\ & & . & . & c_{N-1} \\ 0 & & & a_N & b_N \end{bmatrix} \quad b = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ d_N \end{bmatrix} \quad x = \begin{bmatrix} \delta p_1 \\ \delta p_2 \\ \delta p_3 \\ . \\ \delta p_N \end{bmatrix} \tag{42}$$

and a_j, b_j, c_j, d_j are defined as follows:

$$a_j = \sigma^2 \frac{(c^2\rho)_j^n}{\rho_{j-\frac{1}{2}}^n} \quad for \quad j = 2..(N-1), \tag{43}$$

$$b_j = 1 + \sigma^2 \frac{(c^2\rho)_j^n}{\rho_{j+\frac{1}{2}}^n} + \sigma^2 \frac{(c^2\rho)_j^n}{\rho_{j-\frac{1}{2}}^n} \quad for \quad j = 2..(N-1), \tag{44}$$

$$c_j = \sigma^2 \frac{(c^2\rho)_j^n}{\rho_{j+\frac{1}{2}}^n} \quad for \quad j = 2..(N-1), \tag{45}$$

$$d_j = -\sigma u_j^n \left(\frac{p_{j+1}^n - p_{j-1}^n}{2} \right) - \sigma (c^2\rho)_j^n (\tilde{u}_{j+\frac{1}{2}}^* - \tilde{u}_{j-\frac{1}{2}}^*) \quad for \quad j = 2..(N-1), \tag{46}$$

and b_1, b_N, d_1 and d_N are obtained from the boundary condition. So in order to use the implicit "pressure corrector", we have to solve the tri-diagonal linear system in (41). While this obviously takes more time to calculate than the explicit "pressure corrector" in (16), the results computed using the two methods show that there is not much different between the numerical solutions obtained from using the implicit and explicit "pressure corrector" in the IMPICE method for the Euler equations examples used here.

The IMPICE method calculates the face-centered pressure, $p_{j+\frac{1}{2}}^*$, the same way as the cell-centered ICE method does in Section 2.2. It uses the calculation described in equation (18).

The IMPICE method chooses to conserve the total energy instead of internal energy, so the vector of conserved variables is $U = [\rho, \rho u, \rho E]^T$. The Lagrangian and Eulerian phases of the IMPICE method are then given by

$$V_j^L U_j^L = V_j^n U_j^n - \Delta t \begin{bmatrix} 0 \\ p_{j+1/2}^* - p_{j-1/2}^* \\ p_{j+1/2}^* u_{j+1/2}^* - p_{j-1/2}^* u_{j-1/2}^* \end{bmatrix}, \tag{47}$$

and

$$V_j^{n+1} U_j^{n+1} = V_j^L U_j^L - \Delta t (u_{j+\frac{1}{2}}^* \langle U \rangle_{j+\frac{1}{2}} - u_{j-\frac{1}{2}}^* \langle U \rangle_{j-\frac{1}{2}}), \tag{48}$$

in which the terms $\langle U \rangle_{j-\frac{1}{2}}$ and $\langle U \rangle_{j+\frac{1}{2}}$ are given by equation (26) and $V_j^L = V_j^n + \Delta t (u_{j+\frac{1}{2}}^* - u_{j-\frac{1}{2}}^*)$.

5.3. Application of Slope Limiters in the IMPICE Method

In common with many methods for conservative laws, slope limiters may be applied to the calculation of face-centered fluxing velocity, $u_{j+1/2}^*$. For the face-centered fluxing velocity, slope limiters are used in the estimation of face-centered quantities at t_n; in particular, they are used in

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

the calculation of $\rho^n_{j+\frac{1}{2}}$ and $u^n_{j+\frac{1}{2}}$. This approach originates from the idea of approximating the cell-centered state by the reconstructed states obtained from the left and right cell-averaged states of the previous time-step. The slope limited, reconstructed states are used as inputs to a Riemann solver to determine the state at the cell interface. This will be discussed in detail below.

While U is often used to denote the vector of conserved variables, the vector W, $W = (\rho, u, p)$, is often used to denote of the vector of primitive variables, see Laney [24] and Toro [34]. Let W^n_j be the vector of average cell-centered values of primitive variables of cell j at time t_n, then the value of W on the spatial domain at t_n is represented by the piecewise constant data $\left\{W^n_j\right\}$. The simplest and widely-used way to modify the piecewise constant data $\left\{W^n_j\right\}$ is to replace the constant state W^n_j by a piecewise linear functions $W^n_j(x)$. The construction of the piecewise linear functions can be found in many papers; the construction in Toro [34] will be used as described below.

As for the first-order Godunov method, one assumes that W^n_j represents an integral average in cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ as given by

$$W^n_j = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} W^n_j(x)dx. \tag{49}$$

A piecewise linear, local reconstruction of W^n_j is

$$W^n_j(x) = W^n_j + (x - x_j)\Delta W^n_j, \quad x \in I_j, \tag{50}$$

where ΔW^n_j is a suitably chosen slope of $W^n_j(x)$ in cell I_j. The integral of $W^n_j(x)$ in cell I_j is identical to that of W^n_j and thus the reconstruction process is conservative. The slope ΔW^n_j can be approximated by a simple finite difference formula given by

$$\Delta W^n_j = \frac{W^n_{j+1} - W^n_j}{\Delta x}. \tag{51}$$

However, to achieve a higher order scheme and to maintain bounded solutions, the slope at the current node is usually limited based on adjacent slopes. The obtained slope is a "limited slope" $\bar{\Delta} W^n_j$ which is used as :

$$W^n_j(x) = W^n_j + (x - x_j)\bar{\Delta} W^n_j, \quad x \in I_j, \tag{52}$$

to approximate W on I_j. The ratio r_j represents the ratio of successive gradients on the solution mesh at x_j,

$$r_j = \frac{W^n_j - W^n_{j-1}}{W^n_{j+1} - W^n_j}, \tag{53}$$

and the limited slope ΔW^n_j may be written in the form:

$$\bar{\Delta} W^n_j = \phi(r_j)\Delta W^n_j, \tag{54}$$

where $\phi(r_j)$ is some flux limiter function. For the results in this paper, we choose the Monotonized Central(MC) limiter function for calculating the limited slope in (54). The MC limiter function by

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Van Leer [40] is

$$\phi(r) = max[0, min(2r, 0.5 + 0.5r, 2)]. \tag{55}$$

At each interface $x_{j+\frac{1}{2}}$, we now may consider the so called Generalized Riemann Problem(GRP) as follows

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0, \tag{56}$$
$$W(x, t_n) = \begin{cases} W_j^n(x) & x < x_{j+\frac{1}{2}} \\ W_{j+1}^n(x) & x > x_{j+\frac{1}{2}}, \end{cases}$$

where $W_j^n(x)$ is the limited local reconstruction in (52). Naturally, for non-linear systems the exact solution of the GRP is exceedingly complicated, but for the purpose of evaluating face-centered states, an approximate solution may be suffice. In this approach, we are not trying to evaluate the solution of the GRP in (56) analytically but rely on the boundary extrapolated values at the interfaces. The values of $W_{j+\frac{1}{2}}^n$ at cell boundaries using local reconstructions $W_j^n(x)$ and $W_{j+1}^n(x)$ are denoted as $W_{j+\frac{1}{2}}^{n(L)}$ and $W_{j+\frac{1}{2}}^{n(R)}$ where

$$W_{j+\frac{1}{2}}^{n(L)} = W_j^n(x_{j+\frac{1}{2}}); \quad W_{j+\frac{1}{2}}^{n(R)} = W_{j+1}^n(x_{j+\frac{1}{2}}). \tag{57}$$

The values $W_{j+\frac{1}{2}}^{n(L)}$ and $W_{j+\frac{1}{2}}^{n(R)}$ are left and right extrapolated values at the boundary $x_{j+\frac{1}{2}}$ at time t_n. In this way, one may instead consider the conventional Riemann Problem with piecewise constant data in a new coordinate (ξ, τ) where $\xi = x - x_{j+\frac{1}{2}}$ and $\tau = t - t_n$ as

$$\frac{\partial U}{\partial \tau} + \frac{\partial F}{\partial \xi} = 0, \tag{58}$$
$$W(\xi, 0) = \begin{cases} W_{j+\frac{1}{2}}^{n(L)} & \xi < 0 \\ W_{j+\frac{1}{2}}^{n(R)} & \xi > 0. \end{cases}$$

The face-centered state at t_n, $W(0,0)$, is the value at the origin immediately after the interaction of the piecewise constant data $W_j^{n(L)}$ and $W_{j+1}^{n(R)}$ where

$$W(0,0) = \lim_{\tau \to 0^+} W(0, \tau). \tag{59}$$

By determining $W(0,0)$ in (ξ, τ) coordinate, we have the values of face-centered states given by $W_{j+\frac{1}{2}}^n = [\rho_{j+\frac{1}{2}}^n, u_{j+\frac{1}{2}}^n, p_{j+\frac{1}{2}}^n]^T$ at t_n. There are several ways to approximate the solution to the piecewise constant data Riemann problem (58) and therefore to approximate $W(0,0)$. In this paper, we use the simple approximate Riemann solver which was proposed by Harten *et al.* [15] and discussed in Davis [6] to approximate $W(0,0)$. In order to use the approximate Riemann solver

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

described in these papers, we rewrite equation (58) as

$$\frac{\partial U}{\partial \tau} + \frac{\partial F}{\partial \xi} = 0, \tag{60}$$
$$U(\xi, 0) = \begin{cases} U_L & \xi < 0 \\ U_R & \xi > 0, \end{cases}$$

where U_L and U_R are obtained from $W^{n(L)}_{j+\frac{1}{2}}$ and $W^{n(R)}_{j+\frac{1}{2}}$ respectively. The approximate Riemann solution of (60) is given by

$$U(x/t; U_L, U_R) = \begin{cases} U_L & for \quad x/t < a_L \\ U_{LR} & for \quad a_L < x/t < a_R \\ U_R & for \quad a_R < x/t \end{cases} \tag{61}$$

where a_L and a_R are lower and upper bounds, respectively, for the smallest and largest signal velocity and

$$U_{LR} = \frac{a_R U_R - a_L U_L}{a_R - a_L} - \frac{F(U_R) - F(U_L)}{a_R - a_L}. \tag{62}$$

The bounds a_R and a_L for the Euler equations are defined in Davis [6] as:

$$a_L = u_L - c_L, \quad a_R = u_R + c_R, \tag{63}$$

where u_L, c_L are the velocity and wave speed respectively obtained from U_L, and u_R, c_R are the velocity and wave speed obtained from U_R. The solution $W(0, 0)$ in (59) is derived from the approximate solution $U(0; U_L, U_R)$ in (61) which includes the approximations of $u^n_{j+\frac{1}{2}}$ and $\rho^n_{j+\frac{1}{2}}$; these in turn are used in equation (38) instead of using the mass-weighted quantities in equations (10) and (11).

In summary, the face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, is estimated via the following steps. First, using the local recontruction in (52), the left and right extrapolated values at this cell-center are obtained using (57). These extrapolated values then form the piecewise constant data to the Riemann problem (58). Second, this Riemann problem is solved approximately using the approach of Harten *et al.* [15] and Davis [6]. The approximate Riemann solution includes the approximate face-centered density, $\rho^n_{j+\frac{1}{2}}$, and face-centered velocity, $u^n_{j+\frac{1}{2}}$. Third, the "pressure correctors", δp^n_j, are calculated using equation (17). Finally, equations (12) and (13) are used to calculate $u^*_{j+\frac{1}{2}}$.

6. NUMERICAL RESULTS AND COMPARISONS

The following well-known test problems are often used to test the accuracy and robustness of many numerical methods in fluids. These tests for the one-dimensional, time-dependent Euler equations for ideal gases can be found in Toro [34] where they are used to access the performance of the numerical schemes being presented in the book and also employed here to illustrate the performance of the Production ICE method and the IMPICE method. In these chosen problems, two constant states, $W_L = [\rho_L, u_L, p_L]^T$ and $W_R = [\rho_R, u_R, p_R]^T$, are separated by a discontinuity

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Table I. Data for one-dimensional test problems with known exact solutions, for the time-dependent, one dimensional Euler equations

Problem	ρ_L	u_L	p_L	ρ_R	u_R	p_R	x_0	T_{end}
P1	1.0	0.0	1.0	0.125	0.0	0.1	0.3	0.2
P2	1.0	0.75	1.0	0.125	0.0	0.1	0.3	0.2
P3	1.0	0.0	1000	1.0	0.0	0.1	0.5	0.011
P4	5.99924	19.5975	460.894	5.99242	-6.19633	46.0950	0.4	0.034
P5	0.445	0.698	3.528	0.15	0.0	0.571	0.3	0.16

at a position $x = x_0$. The states W_L and W_R are given in Table I and the problem domain is $(x, t) \in [0, 1] \times (0, T_{end}]$.

P1 is the well known Sod's problem and **P2** is a modified version of **P1**. These tests are considered very mild, but as mentioned in Toro [34] they are useful for assessing the entropy satisfaction property of numerical methods . **P3** is considered a very hard problem for numerical methods. As mentioned in Toro [34], the solution to **P4** represents the collision of two strong shocks and consists of a left facing shock, a right travelling contact discontinuity and a right travelling shock wave. Problem **P5** is Lax's test problem [25].

Beside the problems with known exact solutions in Table I, we also include here the numerical solutions to the Shu and Osher [32] test problem. This test problem contains detailed features and structures and is considered by Greenough and Rider [8] to be a good one-dimensional surrogate for the interaction of a shock wave with a turbulent field. The initial condition at $t = 0$ of the problem is defined as

$$(\rho, u, p)(x, 0) = \begin{cases} (3.85714, 2.62936, 10.33333) & if \quad x < -4.0 \\ (1.0 + 0.2sin(5x), 0.0, 1.0) & otherwise, \end{cases} \tag{64}$$

on spatial domain $[-5.0, 5.0]$. The final time for this problem is $T_{end} = 1.8$. As the analytical solution of this test problem is not readily available, we use the "exact solution" in Martin *et al.* [29] to show how accurate of the numerical methods in this paper. The "exact solution" of Shu and Osher test problem in Martin *et al.* [29] is obtained with the unmodified WENO-JS scheme with $r = 3$ and $p = 2$ on 1600 grid points.

The numerical results for the above test problems of the IMPICE method are compared against those of the Production ICE method and shown in Figures 1–6. A significant improvement in numerical solutions of the IMPICE method is clearly shown through these figures. As shown in these figures, the profiles of the numerical solutions of Production ICE are not close to the exact solutions. This is due to the use of the non-conservative form in Production ICE. It can also be seen that, there are no existing oscillations at shock-front in the numerical solutions of the IMPICE method. The improvement in the numerical solutions of the IMPICE method compared to the Production ICE method comes from the use of a conservation form in the Lagrangian phase and the application of slope limiters in the data reconstruction of the Riemann problem. In order to see how the contribution of each source into the improvement of the IMPICE method, we include in Appendix C the comparision between the numerical results of the IMPICE method and the cell-centered ICE method. The cell-centered ICE method denotes the method that is implemented using the cell-centered ICE method of Kashiwa *et al.* [21] described in Section 2.2 which conserves mass,

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

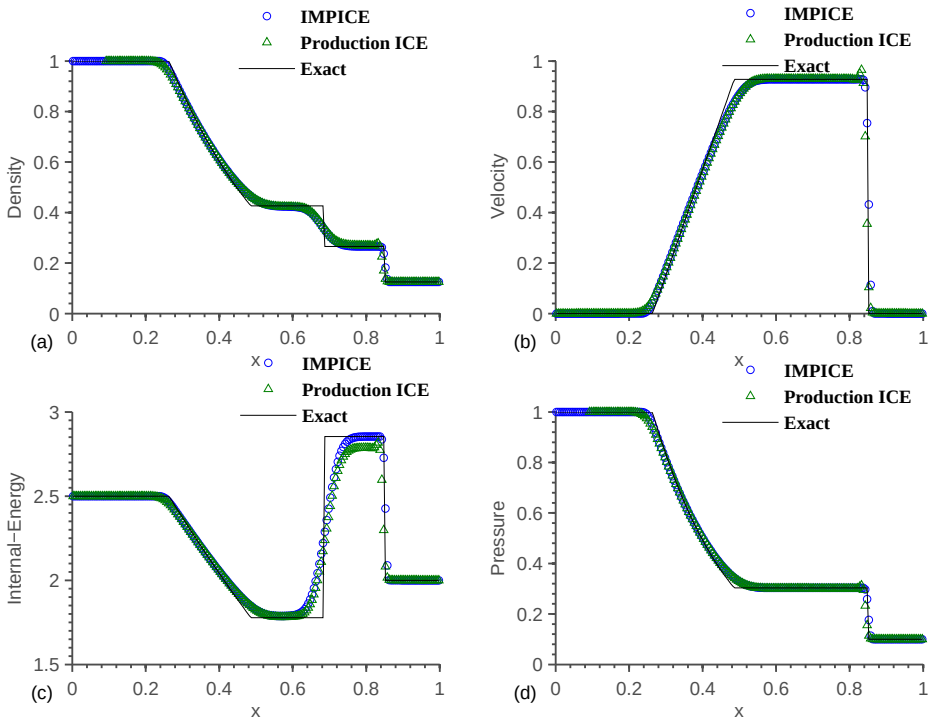

Figure 1. Production ICE and IMPICE numerical solutions for **P1** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

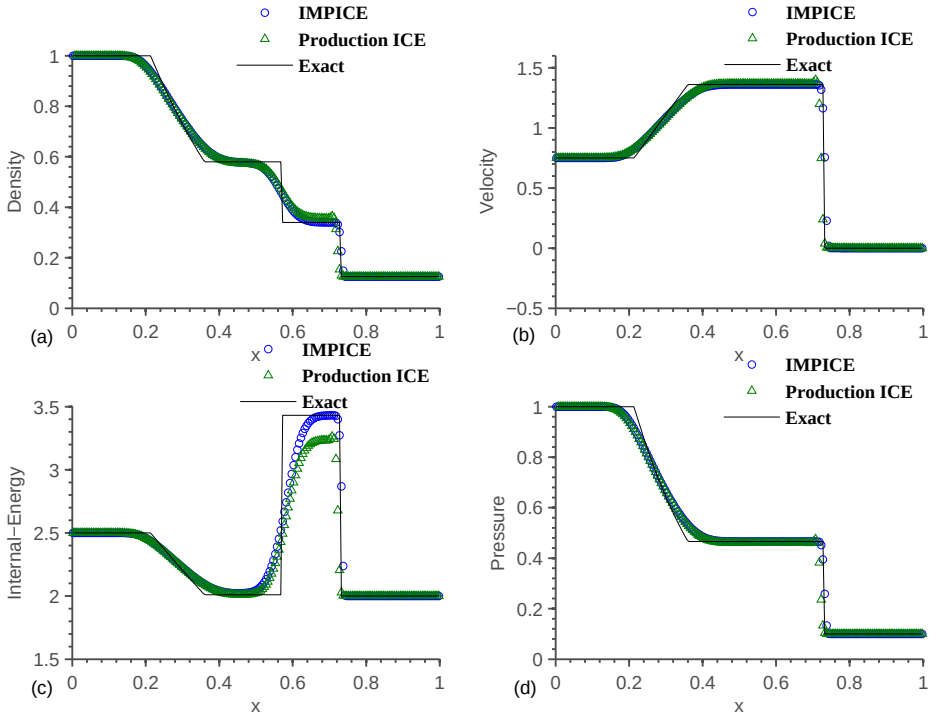

Figure 2. Production ICE and IMPICE numerical solutions for **P2** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

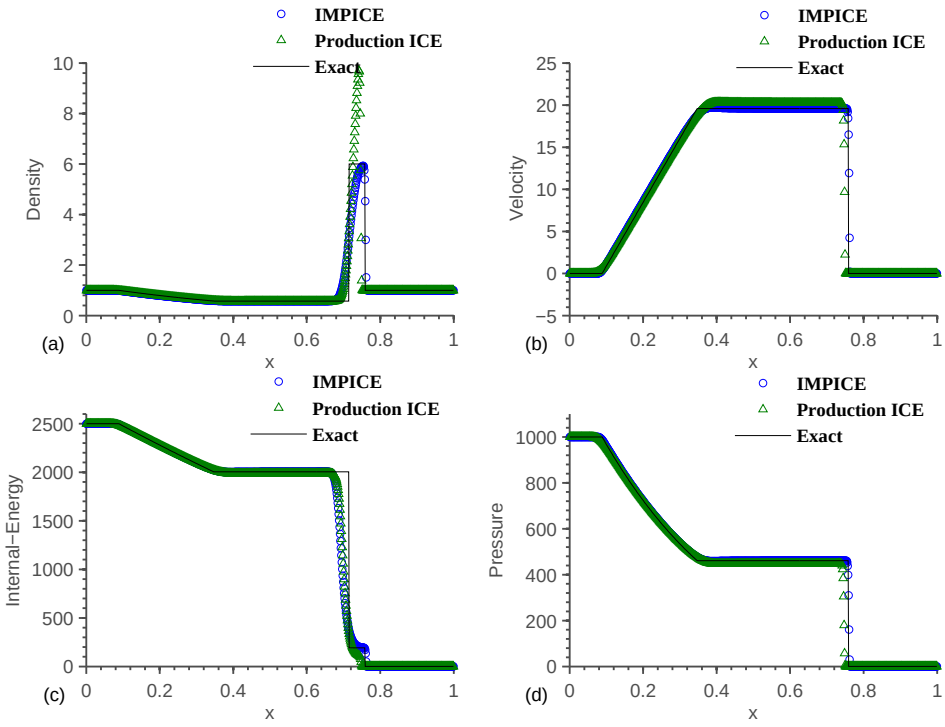

Figure 3. Production ICE and IMPICE numerical solutions for **P3** test problem with N=800(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

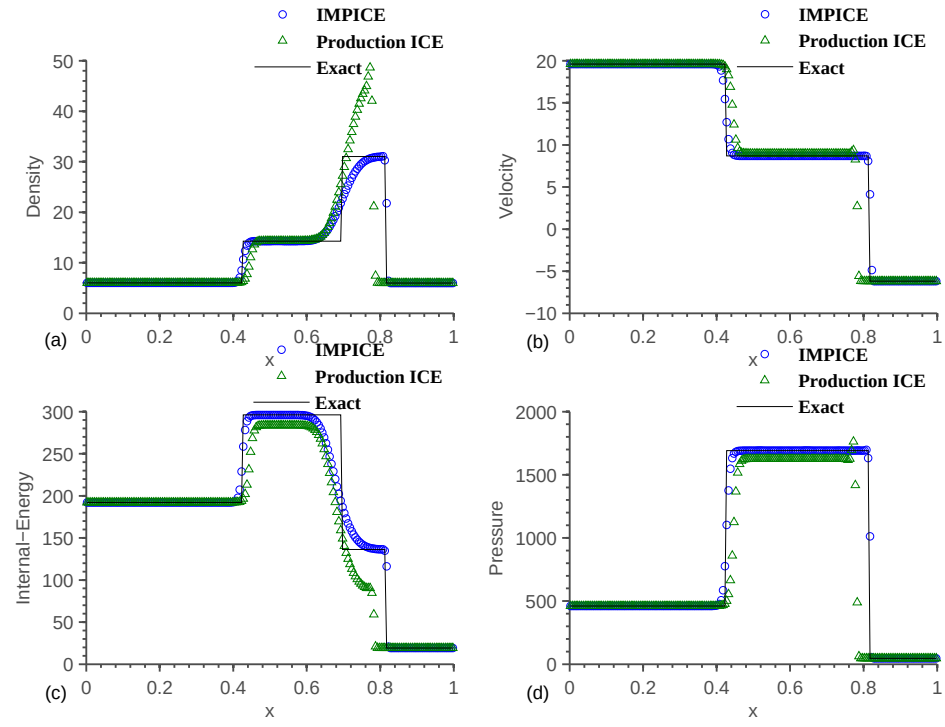

Figure 4. Production ICE and IMPICE numerical solutions for **P4** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

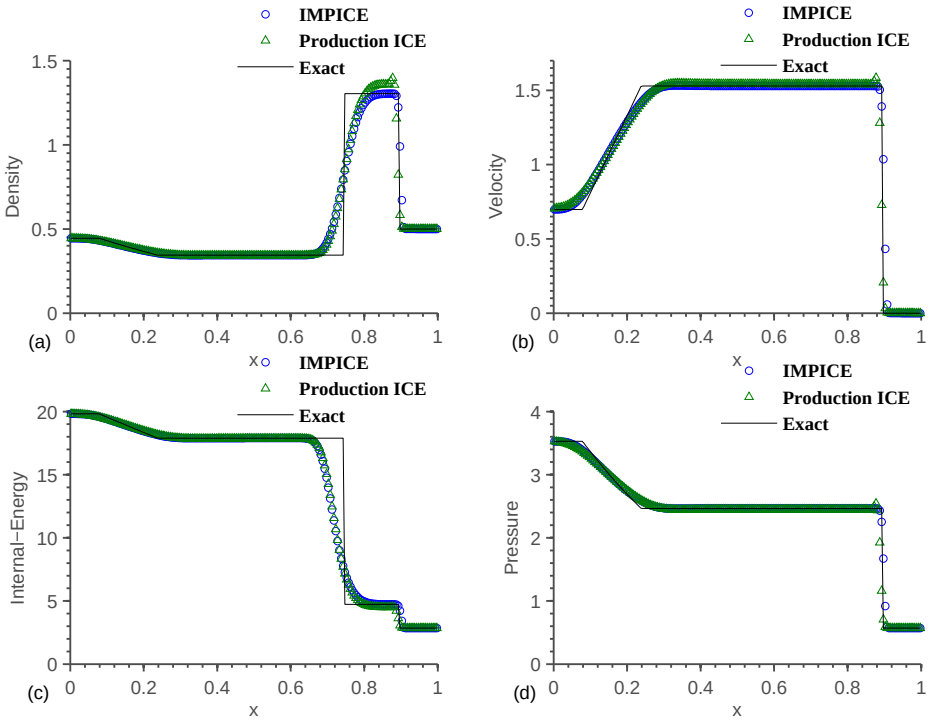

Figure 5. Production ICE and IMPICE numerical solutions for **P5** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

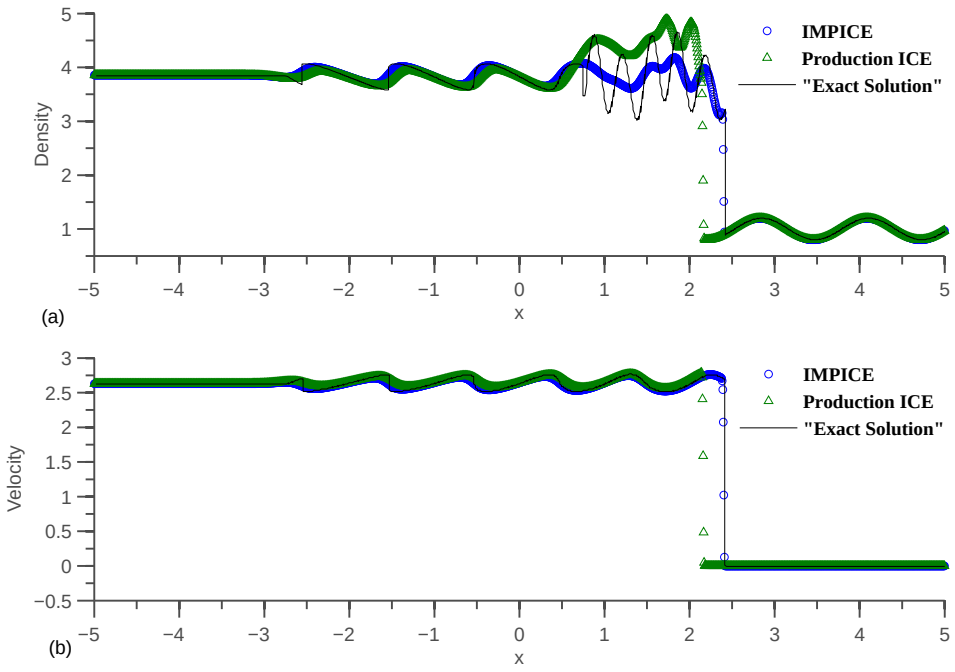

Figure 6. Production ICE and IMPICE numerical solutions for Shu and Osher test problem with N=1600(cells) and $C_{cfl} = 0.2$: (a) density and (b) velocity.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

linear momentum and total energy. From the results in Figures 1–6 and Figures 11–15, it may be seen that the use of conservation form improves the solution profiles and the reconstruction of the Riemann problem with the slope limiters helps to eliminate the non-physical oscillations.

7. ACCURACY IN SPACE AND TIME

There are two sources of errors in the numerical solutions of the IMPICE method: spatial and temporal errors. The spatial error comes from the spatial discretization of the PDEs and the temporal error comes from the time integration method. Let $e_j^U(t_n)$ be the total error in approximating U_j^n of cell j at t_n which is given by

$$e_j^U(t_n) = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(x,t_n)dx - U_j^n. \tag{65}$$

Let $\boldsymbol{U_j}\left[t_n;t_0,U_j^0\right]$ be the exact solution of discretized system with the initial condition $U_j(t_0) = U_j^0$. We then have

$$e_j^U(t_n) = es_j^U(t_n) + et_j^U(t_n), \tag{66}$$

where

$$es_j^U(t_n) = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(x,t_n)dx - \boldsymbol{U_j}\left[t_n;t_0,U_j^0\right] \tag{67}$$

and

$$et_j^U(t_n) = \boldsymbol{U_j}\left[t_n;t_0,U_j^0\right] - U_j^n \tag{68}$$

are the spatial error and the temporal error respectively.

7.1. Temporal Error

Let $le_j^U(t_n)$ be the time local error of the step $[t_{n-1},t_n]$ of the IMPICE method which is defined by

$$le_j^U(t_n) = \boldsymbol{U_j}\left[t_n;t_{n-1},U_j^{n-1}\right] - U_j^n, \tag{69}$$

and where from equation (33), the exact IMPICE local solution $\boldsymbol{U_j}\left[t_n;t_{n-1},U_j^{n-1}\right]$ is given by

$$\boldsymbol{U_j}\left[t_n;t_{n-1},U_j^{n-1}\right] = U_j^{n-1} - \frac{\Delta t}{\Delta x}\left(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\right), \tag{70}$$

where $F_{j+\frac{1}{2}}$ is defined in (37). On the other hand, the IMPICE solution at t_n is

$$U_j^n = U_j^{n-1} - \frac{\Delta t}{\Delta x}\left(F_{j+\frac{1}{2}}^{IMPICE} - F_{j-\frac{1}{2}}^{IMPICE}\right), \tag{71}$$

where

$$F_{j+\frac{1}{2}}^{IMPICE} = u_{j+\frac{1}{2}}^*\langle U\rangle_{j+\frac{1}{2}} + \begin{bmatrix} 0 \\ p_{j+\frac{1}{2}}^* \\ u_{j+\frac{1}{2}}^* p_{j+\frac{1}{2}}^* \end{bmatrix}. \tag{72}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Therefore

$$le_j^U(t_n) = \frac{\Delta t}{\Delta x}\left[\left(F_{j+\frac{1}{2}}^{IMPICE} - F_{j+\frac{1}{2}}\right) - \left(F_{j-\frac{1}{2}}^{IMPICE} - F_{j-\frac{1}{2}}\right)\right]. \tag{73}$$

From equations (37) and (72), we have:

$$\begin{aligned} F_{j+\frac{1}{2}} - F_{j+\frac{1}{2}}^{IMPICE} = \quad & u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\frac{1}{\Delta t}\int_{t_{n-1}}^{t_n} U(x_{j+\frac{1}{2}},t)dt - \langle U\rangle_{j+\frac{1}{2}}\right) + \left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} - u_{j+\frac{1}{2}}^{*}\right)\langle U\rangle_{j+\frac{1}{2}} \\ & + \begin{bmatrix} 0 \\ p_{j+\frac{1}{2}}^{n+\frac{1}{2}} - p_{j+\frac{1}{2}}^{*} \\ (up)_{j+\frac{1}{2}}^{n+\frac{1}{2}} - u_{j+\frac{1}{2}}^{*}p_{j+\frac{1}{2}}^{*} \end{bmatrix} + O(\Delta t^2). \end{aligned} \tag{74}$$

As $\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} - u_{j+\frac{1}{2}}^{*}\right) = O(\Delta t^2)$ and $\left(p_{j+\frac{1}{2}}^{n+\frac{1}{2}} - p_{j+\frac{1}{2}}^{*}\right) = O(\Delta t^2)$, we then have

$$F_{j+\frac{1}{2}} - F_{j+\frac{1}{2}}^{IMPICE} = u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\frac{1}{\Delta t}\int_{t_{n-1}}^{t_n} U(x_{j+\frac{1}{2}},t)dt - \langle U\rangle_{j+\frac{1}{2}}\right) + O(\Delta t^2). \tag{75}$$

By considering the expansion of $U(x_{j+\frac{1}{2}},t)$ about $\langle U\rangle_{j+\frac{1}{2}}$:

$$U(x_{j+\frac{1}{2}},t) = \langle U\rangle_{j+\frac{1}{2}} + (t-t_n)\left(\langle U\rangle_{j+\frac{1}{2}}\right)_t + \frac{(t-t_n)^2}{2}\left(\langle U\rangle_{j+\frac{1}{2}}\right)_{tt} + ..., \tag{76}$$

equation (75) now becomes

$$F_{j+\frac{1}{2}} - F_{j+\frac{1}{2}}^{IMPICE} = \frac{\Delta t}{2}u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\langle U\rangle_{j+\frac{1}{2}}\right)_t + O(\Delta t^2). \tag{77}$$

Therefore

$$\begin{aligned} \left(F_{j+\frac{1}{2}}^{IMPICE} - F_{j+\frac{1}{2}}\right) - \left(F_{j-\frac{1}{2}}^{IMPICE} - F_{j-\frac{1}{2}}\right) = \frac{\Delta t}{2}\left[u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\langle U\rangle_{j+\frac{1}{2}}\right)_t - u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\left(\langle U\rangle_{j-\frac{1}{2}}\right)_t\right] \\ + O(\Delta t^2). \end{aligned} \tag{78}$$

From equations (73) and (78), $le_j^U(t_n)$ is second-order in Δt for a fixed Δx. Therefore, $et_j^U(t_n)$ is first-order in Δt. As U is a vector of conserved variables, $et_j^U(t_n)$ is also a vector of temporal errors of these conserved variables where $et_j^U(t_n) = \left[et_j^{\rho}(t_n), et_j^{\rho u}(t_n), et_j^{\rho E}(t_n)\right]^T$. Define the approximate $L1$-error norm of the vector of temporal errors for variable q, where $q = \rho, \rho u, \rho E, u,$ or p, as follows

$$\|et^q(t_n)\|_{L_1} = \Delta x\sum_{j=1}^{N}|et_j^q(t_n)|, \tag{79}$$

where $et^q(t_n) = [et_1^q(t_n), et_2^q(t_n), et_3^q(t_n), ..., et_N^q(t_n)]^T$.

In order to calculate the temporal error in equation (68), we need to determine the IMPICE time-integrated exact solution $\boldsymbol{U_j}\left[T_{end}; t_0, U_j^0\right]$ in this equation. As we do not have the exact solution $\boldsymbol{U_j}\left[T_{end}; t_0, U_j^0\right]$, we assume that the calculated solution U_j^n converges to the time-integrated exact solution $\boldsymbol{U_j}\left[T_{end}; t_0, U_j^0\right]$ when reducing C_{cfl}. Therefore, we use a highly resolved solution as

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

Table II. Temporal Error: L_1-norms and the order of accuracy n of the conserved and primitive variables for the test cases in Table I using N=200(cells). The time-integrated exact solutions $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$ for the discretized problems of these test cases are obtained by using $C_{cfl}=0.0001$. The notation aE-b used here stands for $a\times10^{-b}$.

		$et^{\rho}(T_{end})$		$et^{\rho u}(T_{end})$		$et^{\rho E}(T_{end})$		$et^{u}(T_{end})$		$et^{p}(T_{end})$	
	C_{cfl}	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n
	0.4	6.92E-04	—	5.71E-04	—	1.16E-03	—	1.05E-03	—	4.42E-04	—
P1	0.2	3.37E-04	1.04	2.74E-04	1.06	5.54E-04	1.07	5.14E-04	1.03	2.11E-04	1.07
	0.1	1.67E-04	1.01	1.36E-04	1.01	2.76E-04	1.00	2.55E-04	1.00	1.05E-04	1.00
	0.05	8.36E-05	1.00	6.82E-05	1.00	1.39E-04	1.00	1.27E-04	1.00	5.29E-05	1.00
	0.4	1.53E-03	—	1.42E-03	—	3.15E-03	—	1.98E-03	—	1.11E-03	—
P2	0.2	7.55E-04	1.02	6.96E-04	1.03	1.56E-03	1.01	9.90E-04	1.00	5.55E-04	1.00
	0.1	3.74E-04	1.01	3.45E-04	1.01	7.78E-04	1.01	4.93E-04	1.00	2.76E-04	1.00
	0.05	1.86E-04	1.01	1.71E-04	1.01	3.88E-04	1.00	2.46E-04	1.00	1.38E-04	1.00
	0.4	1.08E-02	—	2.18E-01	—	3.44E-00	—	3.12E-02	—	6.61E-01	—
P3	0.2	5.20E-03	1.06	1.04E-01	1.06	1.63E-00	1.08	1.53E-02	1.03	3.11E-01	1.09
	0.1	2.57E-03	1.02	5.16E-02	1.02	8.06E-01	1.02	7.64E-03	1.00	1.53E-01	1.02
	0.05	1.27E-03	1.02	2.55E-02	1.01	4.00E-01	1.01	3.81E-03	1.01	7.59E-02	1.01
	0.4	3.94E-02	—	3.24E-01	—	5.26E-00	—	1.85E-02	—	2.05E-00	—
P4	0.2	1.94E-02	1.02	1.59E-01	1.03	2.59E-00	1.02	8.87E-03	1.06	9.76E-00	1.07
	0.1	9.45E-03	1.04	7.90E-02	1.01	1.26E-00	1.04	4.41E-03	1.01	4.85E-01	1.01
	0.05	5.01E-03	0.91	3.94E-02	1.00	6.10E-01	1.05	2.23E-03	0.98	2.47E-01	0.98
	0.4	2.11E-03	—	3.53E-03	—	6.32E-03	—	1.39E-03	—	1.64E-03	—
P5	0.2	1.04E-03	1.02	1.75E-03	1.02	3.21E-03	0.98	7.25E-04	0.94	8.55E-03	0.94
	0.1	5.18E-04	1.01	8.67E-04	1.01	1.61E-03	1.00	3.66E-04	0.98	4.30E-04	0.99
	0.05	2.58E-04	1.01	4.32E-04	1.01	8.03E-04	1.00	1.83E-04	1.00	2.15E-04	1.00

the time-integrated exact solution $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$ with $C_{cfl}=0.0001$. This solution meets the criterion mentioned in Greenough and Rider [8] that the grid converged solutions should be at least 8 times finer than the finest grid examined for error. The temporal error norms and their orders of accuracy of the conserved and primitive variables for the above test cases at T_{end} are shown in Table II. The results in Table II show that the orders of accuracy of the conserved variables for these test cases are very close to 1, this is consistent with the above analysis. The orders of accuracy of the primitive variables are also very close to 1.

7.2. Spatial Error

With the linear spatial discretization as discussed above, the spatial error of the vector of conserved variables U is first-order in Δx. In order to access the spatial errors of a test case, we need the exact solution $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$, see equation (67). The result in Section 7.1 gives the rate at which the computed solution approaches the true solution, so a more accurate approximation to the exact solution $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$ might be obtained by comparing the numerical solution to a finer-mesh numerical solution. Therefore, we estimate the exact solution $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$ in Table III using the computed solutions of the IMPICE method with $C_{cfl}=0.025$ and one with $C_{cfl}=0.0125$.

Define the $L1$-error norm of the vector of spatial errors for variable q, where $q=\rho,\rho u,\rho E,u,$ or p, as follows

$$\|es^q(t_n)\|_{L_1}=\Delta x\sum_{j=1}^{N}|es_j^q(t_n)| \tag{80}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Table III. Spatial Error: L_1-norms and the order of accuracy m of the conserved and primitive variables for the test cases in Table I. The exact solutions $\boldsymbol{U_j}\left[T_{end}; t_0, U_j^0\right]$ are the converged numerical solutions discussed in Section 7.2.

		$es^{\rho}(T_{end})$		$es^{\rho u}(T_{end})$		$es^{\rho E}(T_{end})$		$es^{u}(T_{end})$		$es^{p}(T_{end})$	
	N	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m
	100	1.38E-02	—	1.15E-02	—	2.68E-02	—	2.15E-02	—	1.06E-02	—
	200	9.16E-03	0.59	7.44E-03	0.63	1.56E-02	0.79	1.20E-02	0.84	6.10E-03	0.79
P1	400	5.83E-03	0.65	4.65E-03	0.68	9.10E-03	0.77	6.67E-03	0.85	3.48E-03	0.81
	800	3.68E-03	0.66	2.92E-03	0.67	5.29E-03	0.78	3.63E-03	0.88	1.95E-03	0.84
	1600	2.40E-03	0.62	1.89E-03	0.62	3.06E-03	0.79	1.95E-03	0.90	1.09E-03	0.84
	100	2.17E-02	—	1.89E-02	—	4.47E-02	—	2.83E-02	—	1.66E-02	—
	200	1.45E-02	0.58	1.26E-02	0.59	2.87E-02	0.64	1.63E-02	0.80	1.02E-02	0.71
P2	400	9.87E-03	0.55	8.93E-03	0.50	1.80E-02	0.67	9.25E-03	0.82	6.11E-03	0.74
	800	6.45E-03	0.61	5.98E-03	0.58	1.10E-02	0.71	5.15E-03	0.85	3.58E-03	0.77
	1600	4.24E-03	0.60	4.13E-03	0.53	7.04E-03	0.64	3.13E-03	0.72	2.17E-03	0.72
	100	1.80E-01	—	3.49E+00	—	8.06E+01	—	5.56E-01	—	1.16E+01	—
	200	1.47E-01	0.29	2.96E+00	0.24	4.84E+01	0.73	3.38E-01	0.72	6.93E+00	0.74
P3	400	1.09E-01	0.43	2.20E+00	0.43	3.48E+01	0.48	1.95E-01	0.79	4.10E+00	0.76
	800	7.60E-02	0.52	1.50E+00	0.55	2.26E+01	0.62	1.06E-01	0.87	2.33E+00	0.81
	1600	5.44E-02	0.48	1.07E+00	0.49	1.29E+01	0.81	5.63E-02	0.92	1.25E+00	0.90
	100	7.70E-01	—	6.80E+00	—	6.93E+01	—	2.11E-01	—	1.73E+01	—
	200	5.76E-01	0.42	5.21E+00	0.39	4.23E+01	0.71	9.28E-02	1.19	8.83E+00	0.97
P4	400	3.93E-01	0.55	3.46E+00	0.59	2.47E+01	0.77	5.32E-02	0.80	4.42E+00	1.00
	800	2.69E-01	0.55	2.39E+00	0.53	1.58E+01	0.65	2.47E-02	1.10	2.19E+00	1.01
	1600	1.91E-01	0.50	1.67E+00	0.52	9.99E+00	0.66	1.41E-02	0.81	1.29E+00	0.76
	100	3.90E-02	—	6.47E-02	—	1.53E-01	—	3.04E-02	—	3.35E-02	—
	200	3.00E-02	0.38	4.93E-02	0.39	8.37E-02	0.87	1.82E-02	0.74	2.10E-02	0.68
P5	400	2.03E-02	0.56	3.26E-02	0.60	5.19E-02	0.69	1.00E-02	0.86	1.16E-02	0.86
	800	1.41E-02	0.53	2.26E-02	0.53	3.52E-02	0.56	5.90E-03	0.77	6.95E-03	0.73
	1600	9.80E-03	0.53	1.54E-02	0.55	2.17E-02	0.70	3.25E-03	0.86	3.82E-03	0.86

where $es^q(t_n) = [es_1^q(t_n), es_2^q(t_n), es_3^q(t_n), ..., es_N^q(t_n)]^T$. The spatial error norms and their orders of accuracy for the above test cases at T_{end} are shown in Table III.

Theoretically the spatial error order of accuracy is first-order. However it is shown in Table III that the order of accuracy is mostly below 1 due to the discontinuities in the solutions of these test cases. Greenough and Rider [8] mention earlier work showing less than first-order accuracy for the first-order version of Godunov's method and suggest that this is due to the low resolution computed solutions being very different from the highly resolved solution. For reference purposes, we include in Appendices A and B of this paper the spatial errors and the orders of accuracy for the inviscid and viscous Burgers' problems. For the inviscid Burgers' problem, the order of accuracy is below 1. For the viscous Burgers' problem, the order is around 1.

As numerical solutions obtained with first-order methods are diffusive and not accurate enough to be used for some large problems on relatively coarse grids; for example, the numerical solution to Shu and Osher test problem shown in Figure 6, we improve the order of accuracy of the IMPICE method to 2nd-order in both space and time in the following sections.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Table IV. Temporal Error using the 2nd-order-in-time IMPICE: L_1-norms and the order of accuracy n of the conserved and primitive variables for the test cases in Table I using N=200(cells). The exact solutions $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$ for the discretized problems of these test cases are obtained by using $C_{cfl}=0.0001$.

		$E_T^{\rho}(T_{end})$		$E_T^{\rho u}(T_{end})$		$E_T^{\rho E}(T_{end})$		$E_T^{u}(T_{end})$		$E_T^{p}(T_{end})$	
	C_{cfl}	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n
P1	0.4	6.64E-005	—	5.20E-005	—	1.63E-004	—	1.19E-004	—	6.75E-005	—
	0.2	1.47E-005	2.17	1.16E-005	2.16	3.65E-005	2.15	2.71E-005	2.14	1.50E-005	2.17
	0.1	2.90E-006	2.35	2.39E-006	2.29	7.44E-006	2.30	5.71E-006	2.25	3.03E-006	2.31
	0.05	7.11E-007	2.03	5.98E-007	2.00	1.84E-006	2.01	1.46E-006	1.97	7.58E-007	2.00
P2	0.4	1.56E-005	—	2.33E-005	—	4.99E-005	—	4.10E-005	—	1.45E-005	—
	0.2	4.89E-006	1.67	6.24E-006	1.90	1.50E-005	1.74	1.20E-005	1.77	4.72E-006	1.62
	0.1	1.06E-006	2.21	1.49E-006	2.07	3.38E-006	2.15	2.90E-006	2.05	1.02E-006	2.21
	0.05	2.92E-007	1.86	3.84E-007	1.96	9.04E-007	1.90	7.31E-007	1.99	2.88E-007	1.83
P3	0.4	2.26E-004	—	4.91E-003	—	3.11E-001	—	4.96E-003	—	1.27E-001	—
	0.2	6.05E-005	1.90	1.27E-003	1.95	7.49E-002	2.05	1.18E-003	2.08	3.01E-002	2.07
	0.1	2.07E-005	1.55	3.82E-004	1.74	2.10E-002	1.84	3.39E-004	1.79	8.30E-003	1.86
	0.05	7.32E-006	1.50	1.19E-004	1.68	6.80E-003	1.63	1.10E-004	1.62	2.88E-003	1.53
P4	0.4	5.99E-003	—	3.38E-002	—	9.27E-001	—	1.88E-003	—	3.93E-001	—
	0.2	1.65E-003	1.86	1.08E-002	1.65	2.35E-001	1.98	6.42E-004	1.55	1.07E-001	1.88
	0.1	6.23E-004	1.41	4.25E-003	1.35	0.67E-001	1.81	2.56E-004	1.32	4.06E-002	1.40
	0.05	2.37E-004	1.40	1.55E-003	1.46	2.10E-002	1.67	0.93E-004	1.46	1.40E-002	1.54
P5	0.4	3.11E-005	—	5.99E-005	—	2.94E-004	—	8.45E-005	—	1.14E-004	—
	0.2	7.00E-006	2.15	1.44E-005	2.06	6.69E-005	2.13	2.02E-005	2.06	2.62E-005	2.13
	0.1	1.91E-006	1.87	3.72E-006	1.95	1.84E-005	1.86	5.44E-006	1.89	7.19E-006	1.87
	0.05	4.54E-007	2.08	8.77E-007	2.08	3.96E-006	2.22	1.11E-006	2.29	1.52E-006	2.25

8. HIGHER-ORDER ACCURACY IN TIME

In order to raise the order of accuracy globally in time, we use the method of extrapolation to raise the order of accuracy locally. By raising the local order of accuracy of the temporal error to 3rd-order, we raise the order of accuracy of the temporal error to 2nd-order globally. The 2nd-order-in-time IMPICE method is achieved using Richardson extrapolation. The steps in IMPICE method with 2nd-order temporal error to obtain the solution for next time step U_j^{n+1} from current time step solution U_j^n are:

- Perform one step of the 1st-order IMPICE method with stepsize Δt to obtain the solution $U1_j^{n+1}$ at t_{n+1}.
- Perform two consecutive steps of the 1st-order IMPICE method with stepsize $\frac{\Delta t}{2}$ to obtain the solution $U2_j^{n+1}$ at t_{n+1}.
- Set the solution at t_{n+1} of 2nd-order-in-time IMPICE method to $\left(2U2_j^{n+1}-U1_j^{n+1}\right)$.

The temporal error norms and the orders of accuracy of the conserved and primitive variables for the above test cases using the 2nd-order-in-time IMPICE method are shown in Table IV. We use a highly resolved solution as the exact solution $\boldsymbol{U_j}\left[T_{end};t_0,U_j^0\right]$ by setting $C_{cfl}=0.0001$ when calculating temporal errors. It is shown in Table IV that the time integration accuracy for both conserved and primitive variables is very close to second-order.

In doing so we note that this extrapolated method corresponds to the Runge-Kutta method whose positivity properties are described by Mehdizadeh Khalsaraei [30].

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

9. HIGHER-ORDER ADVECTION

The solutions with first-order accuracy of advection where advected quantities obtained from (26) are highly smeared at contact discontinuities. We have improved the spatial errror accuracy of the IMPICE method by using a higher-order advection method. A higher-order Van Leer advection method is discussed in VanderHeyden and Kashiwa [38], in which the compatible fluxes are also derived for this type of advection method. In this paper, the higher-order advection scheme is based on a higher-order approximation of the advected quantities in (24). This is done by assuming that $U(x_{j+\frac{1}{2}},t)$ in equation (24) is not a constant for the time step $[t_n,t_{n+1}]$. The advection equations of conserved variables in the Eulerian phase in Section 5.2 are given by

$$U_t + (uU)_x = 0. \tag{81}$$

In order to determine $U(x_{j+\frac{1}{2}},t)$, we will use equation (81) and the constructed values, $W_j^n(x,t)$, of primitive variables in the control volume $[x_{j-\frac{1}{2}},x_{j+\frac{1}{2}}]\times[t_n,t_{n+1}]$. Within this control volume, the constructed values, $W_j^n(x,t)$, are obtained by using Taylor series:

$$W_j^n(x,t) = W_j^n + (x-x_j)\left(\frac{\partial W}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial W}{\partial t}\right)_j^n + O(\Delta x^2,\Delta t^2). \tag{82}$$

The extrapolated values at cell boundaries obtained by using the constructed values, $W_j^n(x,t)$, are:

$$W_j^n(x_{j-\frac{1}{2}},t) = W_j^n - \frac{\Delta x}{2}\left(\frac{\partial W}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial W}{\partial t}\right)_j^n + O(\Delta x^2,\Delta t^2), \tag{83}$$

$$W_j^n(x_{j+\frac{1}{2}},t) = W_j^n + \frac{\Delta x}{2}\left(\frac{\partial W}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial W}{\partial t}\right)_j^n + O(\Delta x^2,\Delta t^2). \tag{84}$$

Therefore, there are two existing extrapolated values at the cell boundary at $x_{j+\frac{1}{2}}$ for the time interval $[t_n,t_{n+1}]$. These values are denoted as $W_j^n(x_{j+\frac{1}{2}},t)$ and $W_{j+1}^n(x_{j+\frac{1}{2}},t)$, and one may be chosen for the face-centered value based on the face-centered fluxing velocity at this cell boundary. The value of the vector of primitive variables at face-center is determined using

$$W(x_{j+\frac{1}{2}},t) = \begin{cases} W_{j+1}^n(x_{j+\frac{1}{2}},t) & if \quad u^*_{j+\frac{1}{2}} < 0 \\ W_j^n(x_{j+\frac{1}{2}},t) & otherwise. \end{cases} \tag{85}$$

Now, as the extrapolated primitive variables at the cell boundary at $x_{j+\frac{1}{2}}$ are readily available, we will show how to obtain the vector of advected quantities in (24). We derive the advected quantities for the case $u^*_{j+\frac{1}{2}} > 0$. The advected quantities for the case $u^*_{j+\frac{1}{2}} < 0$ are derived similarly. The vector of advected quantities $\langle U\rangle_{j+\frac{1}{2}}$ includes $\langle\rho\rangle_{j+\frac{1}{2}}$, $\langle\rho u\rangle_{j+\frac{1}{2}}$ and $\langle\rho E\rangle_{j+\frac{1}{2}}$. Equation (81) is

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

rewritten as follows

$$\frac{\partial \rho}{\partial t} = -u\frac{\partial \rho}{\partial x} - \rho\frac{\partial u}{\partial x}, \tag{86}$$

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}, \tag{87}$$

$$\frac{\partial E}{\partial t} = -u\frac{\partial E}{\partial x}. \tag{88}$$

Equations (84) and (86) are used to derive the mass advected quantity in equation (24); for the case of $u^*_{j+\frac{1}{2}} > 0$, we have

$$\begin{aligned}
\langle\rho\rangle_{j+\frac{1}{2}} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \rho(x_{j+\frac{1}{2}}, t)dt \\
&= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \left(\rho_j^n + \frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial \rho}{\partial t}\right)_j^n\right)dt + O(\Delta x^2, \Delta t^2) \\
&= \rho_j^n + \frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n + \frac{\Delta t}{2}\left(\frac{\partial \rho}{\partial t}\right)_j^n + O(\Delta x^2, \Delta t^2) \\
&= \rho_j^n + \frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n - \frac{\Delta t}{2}\left(u_j^n\left(\frac{\partial \rho}{\partial x}\right)_j^n + \rho_j^n\left(\frac{\partial u}{\partial x}\right)_j^n\right) + O(\Delta x^2, \Delta t^2).
\end{aligned}$$

Therefore

$$\langle\rho\rangle_{j+\frac{1}{2}} = \rho_j^n + \left(\frac{\Delta x}{2} - u_j^n\frac{\Delta t}{2}\right)\left(\frac{\partial \rho}{\partial x}\right)_j^n - \frac{\Delta t}{2}\rho_j^n\left(\frac{\partial u}{\partial x}\right)_j^n + O(\Delta x^2, \Delta t^2). \tag{89}$$

Equation (84) with $W = \rho$ and $W = u$ gives us

$$\begin{aligned}
\rho u(x_{j+\frac{1}{2}}, t) &= \left(\rho_j^n + \frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial \rho}{\partial t}\right)_j^n\right) \\
&\quad \times \left(u_j^n + \frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial u}{\partial t}\right)_j^n\right) + O(\Delta x^2, \Delta t^2) \\
&= \left(\rho_j^n + \frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial \rho}{\partial t}\right)_j^n\right)u_j^n \\
&\quad + \rho_j^n\left(\frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial u}{\partial t}\right)_j^n\right) \\
&\quad + \left(\frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial \rho}{\partial t}\right)_j^n\right)\left(\frac{\Delta x}{2}\left(\frac{\partial u}{\partial x}\right)_j^n + (t-t_n)\left(\frac{\partial u}{\partial t}\right)_j^n\right) \\
&\quad + O(\Delta x^2, \Delta t^2).
\end{aligned}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

The fluxed momentum at the face-center is then given by

$$\begin{aligned}\langle \rho u \rangle_{j+\frac{1}{2}} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \rho u(x_{j+\frac{1}{2}}, t)dt \\ &= \langle \rho \rangle_{j+\frac{1}{2}} u_j^n + \rho_j^n \left(\frac{\Delta x}{2} - u_j^n \frac{\Delta t}{2}\right)\left(\frac{\partial u}{\partial x}\right)_j^n + \frac{\Delta x}{2}\left(\frac{\partial \rho}{\partial x}\right)_j^n \left(\frac{\Delta x}{2} - u_j^n \frac{\Delta t}{2}\right)\left(\frac{\partial u}{\partial x}\right)_j^n \\ &\quad + \frac{\Delta x}{2}\frac{\Delta t}{2}\left(\frac{\partial u}{\partial x}\right)_j^n \left(\frac{\partial u}{\partial t}\right)_j^n + \frac{\Delta t^3}{3}\left(\frac{\partial u}{\partial t}\right)_j^n \left(\frac{\partial u}{\partial t}\right)_j^n + O(\Delta x^2, \Delta t^2).\end{aligned}$$

This gives us the approximation

$$\langle \rho u \rangle_{j+\frac{1}{2}} = \langle \rho \rangle_{j+\frac{1}{2}} u_j^n + \rho_{j+\frac{1}{2}}^n \left(\frac{\Delta x}{2} - u_j^n \frac{\Delta t}{2}\right)\left(\frac{\partial u}{\partial x}\right)_j^n + O(\Delta x \Delta t) + O(\Delta x^2, \Delta t^2). \quad (90)$$

With a similar derivation, we also have

$$\langle \rho E \rangle_{j+\frac{1}{2}} = \langle \rho \rangle_{j+\frac{1}{2}} E_j^n + \rho_{j+\frac{1}{2}}^n \left(\frac{\Delta x}{2} - u_j^n \frac{\Delta t}{2}\right)\left(\frac{\partial E}{\partial x}\right)_j^n + O(\Delta x \Delta t) + O(\Delta x^2, \Delta t^2). \quad (91)$$

We thus obtain second-order accuracy in space if C_{cfl} remains contant. Equations (89), (90), and (91) are used to calculate the face-centered fluxed quantities for the time step $[t_n, t_{n+1}]$ when the face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, is greater than 0. A set of similar equations can be easily derived for the case when the fluxing velocity is less than 0. However, when using these equations to estimate the face-centered advected quantities, we need to have numerical estimations for $\left(\frac{\partial \rho}{\partial x}\right)_j^n$, $\left(\frac{\partial u}{\partial x}\right)_j^n$, and $\left(\frac{\partial E}{\partial x}\right)_j^n$. These spatial numerical derivatives are limited to eliminate artificial extrema and preserve monotonicity [38]. In this paper, we choose one limiter from one-parameter family of minmod limiters [16, 41],

$$\left(\frac{\partial W}{\partial x}\right)_j^n = minmod(\theta \frac{W_j^n - W_{j-1}^n}{\Delta x}, \frac{W_{j+1}^n - W_{j-1}^n}{2\Delta x}, \theta \frac{W_{j+1}^n - W_j^n}{\Delta x}), \quad (92)$$

to estimate the spatial derivatives of primitive variables by setting $\theta = 1$. The multivariable minmod limiter in (92) is defined as

$$minmod(z_1, z_2, z_3, ...) = \begin{cases} min(z_1, z_2, z_3, ...) & if \quad z_j > 0 \quad \forall j \\ max(z_1, z_2, z_3, ...) & if \quad z_j < 0 \quad \forall j \\ 0 & otherwise. \end{cases} \quad (93)$$

The numerical solution of Shu and Osher test problem in Figure 7 is obtained using the 2nd-order-in-space IMPICE method. Comparing to the numerical solution of this problem in Figure 6, the solution using the 2nd-order-in-space IMPICE method is less diffusive and more accurate.

The spatial error norms and the orders of accuracy for the test cases in Table I using the 2nd-order-in-space IMPICE method are shown in Table V. When calculating the spatial errors, we use the converged numerical solutions of these test problems as described in Section 7.2 for the exact solutions. The result in Table V shows that the 2nd-order-in-space IMPICE method does reduce the spatial errors, increase the orders of accuracy in both conserved and primitive variables. However,

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

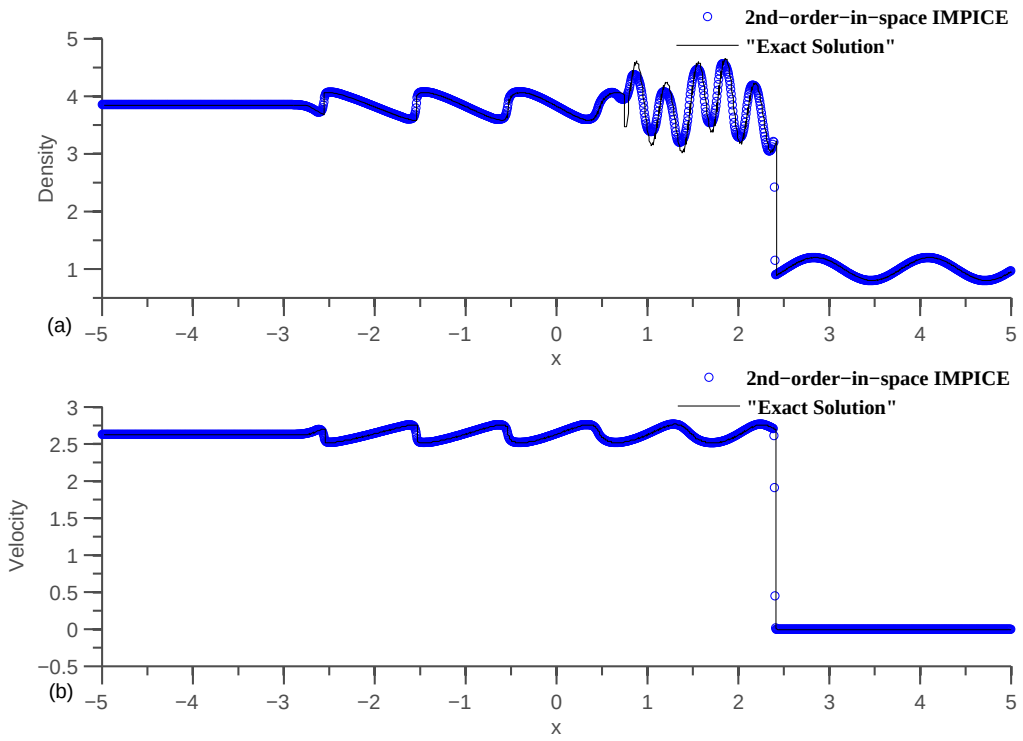

Figure 7. The 2nd-order-in-space IMPICE numerical solution for Shu and Osher test problem with N=1600(cells) and $C_{cfl} = 0.2$: (a) density and (b) velocity.

the orders of spatial accuracy are not close to second-order as expected, but degenerate into first-order and below. The observation concurs with those of Greenough and Rider [8] in that when discontinuities are present high-order methods may not always deliver the expected advantages and may reduce their order of accuracy to first order. In addition, Berzins [3] shows how unless there is sufficient resolution in terms of meshpoints in a front then the positivity requirement will tend to favor the use of lower order methods.

We also would like to estimate the spatial error in the numerical solutions of the Shu and Osher test problem. Since the analytic solution to the Shu and Osher test problem is not readily available, a highly resolved numerical solution is used to estimate the integral term in equation (67) when calculating spatial errors. The highly resolved numerical solution is generated from running the 2nd-order-in-space IMPICE method with $N = 25,600$(cells) and $C_{cfl} = 0.2$. The exact cell average in equation (67), $\frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(x,t_n)dx$, is the numerical integration obtained from the highly resolved solution while the exact solution of time integration, $\boldsymbol{U_j}\left[t_n; t_0, U_j^0\right]$, is the converged numerical solutions as discussed earlier in Section 7.2. The spatial error norms and the orders of accuracy of the Shu and Osher test problem are shown in Table VI. As shown in Table VI, there is also a degeneration in the orders of accuracy for the result of the Shu and Osher test problem when the mesh size N is below 1600 and an improvement in its orders when the mesh size N is above 1600. This result is consistent with that of Greenough and Rider [8].

The numerical results of the 2nd-order-in-space IMPICE for inviscid and viscous Burgers' problem are included in Appendices A and B. The spatial error norms and the orders of accuracy for

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Table V. Spatial Error using the 2nd-order-in-space IMPICE: L_1-norms and the order of accuracy m of the conserved and primitive variables for the test cases in Table I. The exact solutions $\boldsymbol{U_j}\left[T_{end}; t_0, U_j^0\right]$ are the converged numerical solutions as described in Section 7.2

	$es^{\rho}(T_{end})$		$es^{\rho u}(T_{end})$		$es^{\rho E}(T_{end})$		$es^{u}(T_{end})$		$es^{p}(T_{end})$	
N	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m
100	7.10E-003	—	5.97E-003	—	1.50E-002	—	1.25E-002	—	5.72E-003	—
200	4.10E-003	0.79	3.51E-003	0.77	7.37E-003	1.03	6.02E-003	1.05	2.81E-003	1.03
P1 400	2.23E-003	0.88	1.92E-003	0.87	3.70E-003	0.99	2.85E-003	1.08	1.36E-003	1.05
800	1.21E-003	0.87	1.05E-003	0.87	1.87E-003	0.99	1.31E-003	1.12	6.50E-004	1.07
1600	7.21E-004	0.75	6.13E-004	0.78	9.33E-004	1.00	5.83E-004	1.17	3.14E-004	1.05
100	9.92E-003	—	9.39E-003	—	2.07E-002	—	1.50E-002	—	7.39E-003	—
200	5.49E-003	0.85	5.34E-003	0.81	1.14E-002	0.86	7.28E-003	1.04	3.72E-003	0.99
P2 400	3.39E-003	0.69	3.48E-003	0.62	6.18E-003	0.88	3.51E-003	1.05	1.85E-003	1.00
800	1.94E-003	0.80	2.01E-003	0.79	3.25E-003	0.93	1.63E-003	1.11	9.11E-004	1.03
1600	1.15E-003	0.75	1.26E-003	0.68	1.94E-003	0.75	9.91E-004	0.71	5.15E-004	0.82
100	8.56E-002	—	1.71E+000	—	5.34E+001	—	3.38E-001	—	8.03E+000	—
200	6.35E-002	0.43	1.32E+000	0.37	2.70E+001	0.98	2.36E-001	0.52	4.47E+000	0.85
P3 400	4.01E-002	0.66	8.39E-001	0.65	1.83E+001	0.56	1.36E-001	0.79	2.64E+000	0.76
800	2.57E-002	0.64	5.05E-001	0.73	1.08E+001	0.76	6.97E-002	0.97	1.39E+000	0.92
1600	1.76E-002	0.54	3.47E-001	0.54	4.66E+000	1.21	3.50E-002	0.99	7.01E-001	0.99
100	3.57E-001	—	3.61E+000	—	4.97E+001	—	1.17E-001	—	1.33E+001	—
200	2.80E-001	0.35	2.89E+000	0.32	3.62E+001	0.45	7.85E-002	0.57	9.76E+000	0.45
P4 400	1.74E-001	0.69	1.83E+000	0.66	2.25E+001	0.69	4.12E-002	0.93	5.91E+000	0.72
800	1.11E-001	0.64	1.22E+000	0.58	1.67E+001	0.43	2.87E-002	0.52	4.27E+000	0.47
1600	7.12E-002	0.64	7.97E-001	0.61	1.01E+001	0.72	1.81E-002	0.66	2.62E+000	0.70
100	1.63E-002	—	2.78E-002	—	9.54E-002	—	1.90E-002	—	1.99E-002	—
200	1.25E-002	0.38	2.15E-002	0.37	4.14E-002	1.20	1.02E-002	0.89	1.08E-002	0.88
P5 400	7.55E-003	0.73	1.23E-002	0.80	2.30E-002	0.85	4.82E-003	1.09	5.10E-003	1.09
800	4.63E-003	0.70	7.66E-003	0.69	1.48E-002	0.63	2.55E-003	0.92	2.72E-003	0.91
1600	2.86E-003	0.70	4.56E-003	0.75	8.10E-003	0.87	1.20E-003	1.08	1.28E-003	1.08

Table VI. Spatial Error using the 2nd-order-in-space IMPICE: L_1-norms and the order of accuracy m of the conserved and primitive variables for Shu and Osher test problem. The exact solutions $\boldsymbol{U_j}\left[T_{end}; t_0, U_j^0\right]$ are the converged numerical solutions.

	$es^{\rho}(T_{end})$		$es^{\rho u}(T_{end})$		$es^{\rho E}(T_{end})$		$es^{u}(T_{end})$		$es^{p}(T_{end})$	
N	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m
200	9.41E-01	—	1.95E+00	—	4.85E+00	—	2.45E-01	—	1.44E+00	—
400	7.30E-01	0.37	1.65E+00	0.25	3.28E+00	0.57	1.19E-01	1.05	7.03E-01	1.04
800	4.60E-01	0.67	1.08E+00	0.61	1.99E+00	0.72	6.04E-02	0.98	3.42E-01	1.04
1600	2.03E-01	1.18	4.79E-01	1.18	8.95E-01	1.15	2.85E-02	1.08	1.58E-01	1.11
3200	8.12E-02	1.32	1.87E-01	1.36	3.50E-01	1.36	1.22E-02	1.23	6.71E-02	1.24

the numerical solutions of the inviscid Burgers' problem obtained from using the 2nd-order-in-space IMPICE method in Table VII show that the orders of convergence are not very close to second-order. However, the orders of convergence for the numerical solutions of the viscous Burgers' problem using the 2nd-order-in-space IMPICE method as shown in Table VIII are close to second-order for the cases of $\epsilon = 0.05$ and $\epsilon = 0.01$ as the solutions for these cases are smooth as shown in Figure 9. But the orders degenerate into first-order for the case of $\epsilon = 0.0001$. This is due to the development of the smooth steep front that appears close to a discontinuity in the solution of the viscous Burgers' problem when the viscosity ϵ becomes small.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

10. CONCLUSIONS

We have presented IMPICE, an improved Production ICE method, that uses a conservative scheme, slope limiters and a simple approximate Riemann solver for improving the numerical solutions and eliminating existing oscillations of Production ICE which is an implementation of the ICE method in Uintah Computational Framework to simulate fluid flows. The IMPICE method with a linear spatial and temporal discretization is expected to be first-order accuracy in time and space. However, for the cases with existing discontinuities in their solutions, the order of accuracy in space is less than 1 as shown in Section 7. As it is important to have the method of higher-order of accuracy in both time and space, we have presented the non-linear spatial and temporal discretization of the IMPICE method. These are the method of temporal extrapolation, and the higher-order advection. While the method of temporal extrapolation successfully raises the order of accuracy to 2nd-order-in-time, a less-than-expected order of accuracy in space is obtained from using the higher-order advection for the problems with discontinuities.

It has shown that the IMPICE method is capable of capturing shocks and contact surfaces, and the higher-order IMPICE method is even able to capture the detailed features and structures of the flow with shock-turbulence interaction in Shu-Osher problem. As IMPICE has been successfully applied to complex 1D problems, we are currently working on the extension of the method to the 3D case.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Harman at University of Utah for many helpful discussions. This work was supported by the U.S. Department of Energy through the Center for the Simulation of Accidental Fires and Explosions (C-SAFE) under grant W-7405-ENG-48. The center would like to acknowledge the help of Prof. Kashiwa of Los Alamos in developing the Production ICE code. We would also like to thank the referees for their helpful and constructive remarks that have helped to improve the paper greatly.

A. IMPICE METHOD FOR INVISCID BURGERS' EQUATION

Consider the one-dimensional Burgers' equation in the inviscid limit

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \tag{94}$$

with $f(u) = \frac{1}{2}u^2$ and initial data $u(x,0) = u_0(x)$, where $u_0(x)$ is a given function, $x \in \mathbb{R}$ and $t > 0$. The solution, $u(x/t; u^{n(L)}_{j+\frac{1}{2}}, u^{n(R)}_{j+\frac{1}{2}})$, to the Riemann problem of the Burgers' equation (94) with initial data

$$u(x,t_n) = \begin{cases} u^{n(L)}_{j+\frac{1}{2}} & if(x < x_{j+\frac{1}{2}}) \\ u^{n(R)}_{j+\frac{1}{2}} & if(x > x_{j+\frac{1}{2}}), \end{cases} \tag{95}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

at $x/t = 0$ is used in the IMPICE method. The approximate solution $u(0; u^{n(L)}_{j+\frac{1}{2}}, u^{n(R)}_{j+\frac{1}{2}})$ is given by

$$u(0; u^{n(L)}_{j+\frac{1}{2}}, u^{n(R)}_{j+\frac{1}{2}}) = \begin{cases} u^{n(L)}_{j+\frac{1}{2}} & if \quad (0 < u^{n(L)}_{j+\frac{1}{2}} < u^{n(R)}_{j+\frac{1}{2}}) \\ 0 & if \quad (u^{n(L)}_{j+\frac{1}{2}} \leq 0 \leq u^{n(R)}_{j+\frac{1}{2}}) \\ u^{n(R)}_{j+\frac{1}{2}} & if \quad (u^{n(L)}_{j+\frac{1}{2}} < u^{n(R)}_{j+\frac{1}{2}} < 0) \\ u^{n(L)}_{j+\frac{1}{2}} & if \quad (u^{n(R)}_{j+\frac{1}{2}} < u^{n(L)}_{j+\frac{1}{2}} \quad and \quad S > 0) \\ u^{n(R)}_{j+\frac{1}{2}} & if \quad (u^{n(R)}_{j+\frac{1}{2}} < u^{n(L)}_{j+\frac{1}{2}} \quad and \quad S < 0), \end{cases} \tag{96}$$

where $S = (u^{n(L)}_{j+\frac{1}{2}} + u^{n(R)}_{j+\frac{1}{2}})/2$.

With the same spatial and temporal discretizations as in Section 2 and known cell averages at time t_n, the steps to obtain cell averages at time t_{n+1} are as follows.

A.1. IMPICE Method Description

The Primary Phase: At face center, a data reconstruction is done as follows

$$u^{n(L)}_{j+\frac{1}{2}} = u^n_j + \frac{\Delta x}{2}\bar{\Delta}u^n_j, \quad u^{n(R)}_{j+\frac{1}{2}} = u^n_{j+1} - \frac{\Delta x}{2}\bar{\Delta}u^n_{j+1}, \tag{97}$$

where $\bar{\Delta}u^n_j$ is the limited slope of u. The face-centered velocity, $u^n_{j+\frac{1}{2}}$, at t_n is determined using the approximate solution of Riemann problem where $u^n_{j+\frac{1}{2}} = u(0; u^{n(L)}_{j+\frac{1}{2}}, u^{n(R)}_{j+\frac{1}{2}})$. The equation of velocity evolution,

$$u_t + uu_x = 0, \tag{98}$$

is written in Lagrangian form as

$$\frac{Du}{Dt} = 0. \tag{99}$$

The face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, is then given by

$$u^*_{j+\frac{1}{2}} = u^n_{j+\frac{1}{2}}. \tag{100}$$

In order to apply the Lagrangian and Eulerian phases, we rewrite equation (94) as

$$(\rho u)_t + \left(\rho u^2\right)_x = \frac{1}{2}\left(u^2\right)_x \tag{101}$$

where ρ is a constant and equal to 1.

The Lagrangian Phase:

$$V^L_j(\rho u)^L_j = V^n_j(\rho u)^n_j + \Delta t\left(\frac{1}{2}\left(u^*_{j+\frac{1}{2}}\right)^2 - \frac{1}{2}\left(u^*_{j-\frac{1}{2}}\right)^2\right), \tag{102}$$

where $V^n_j = \Delta x$ and $V^L_j = \Delta x + \Delta t(u^*_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}})$. As ρ is a constant and equal to 1, the above equation may be rewritten as follows

$$V^L_j u^L_j = V^n_j u^n_j + \Delta t\left(\frac{1}{2}\left(u^*_{j+\frac{1}{2}}\right)^2 - \frac{1}{2}\left(u^*_{j-\frac{1}{2}}\right)^2\right) \tag{103}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

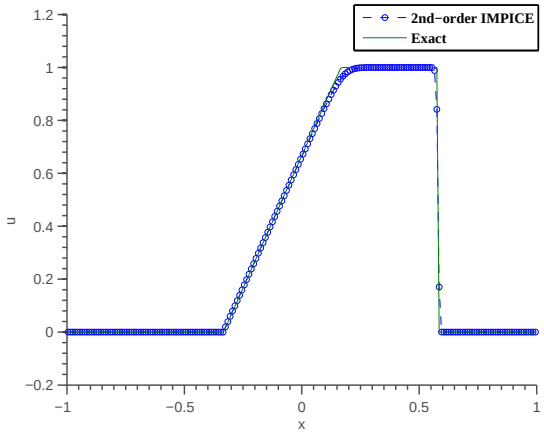

Figure 8. The 2nd-order (2nd-order-in-space and 2nd-order-in-time) IMPICE numerical solutions for the inviscid Burgers' problem at $T_{end} = 0.5$ and on the spatial domain $[-1.0, 1.0]$ with N=200(cells) and $C_{cfl} = 0.2$.

The Eulerian Phase:

$$V_j^{n+1}(\rho u)_j^{n+1} = V_j^L(\rho u)_j^L - \Delta t\left(u^*_{j+\frac{1}{2}}\langle \rho u\rangle_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}}\langle \rho u\rangle_{j-\frac{1}{2}}\right) \tag{104}$$

where $V_j^{n+1} = \Delta x$. As ρ is a constant and equal to 1, the above equation may be rewritten as follows

$$V_j^{n+1}u_j^{n+1} = V_j^L u_j^L - \Delta t\left(u^*_{j+\frac{1}{2}}\langle u\rangle_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}}\langle u\rangle_{j-\frac{1}{2}}\right). \tag{105}$$

For first-order advection, $\langle u\rangle_{j+\frac{1}{2}}$ is approximated using (25) and for second-order advection, it is approximated using (90).

A.2. Numerical Results and Accuracy in Space and Time

The initial condition used is given by:

$$u_0(x) = \begin{cases} 1.0 & if|x| < \frac{1}{3} \\ 0.0 & otherwise. \end{cases} \tag{106}$$

The numerical solution of the inviscid Burgers' problem using the 2nd-order-in-space and 2nd-order-in-time IMPICE method is shown in Figure 8. The spatial and temporal error norms and the orders of accuracy for this problem are summarized in Table VII. For temporal errors, the orders of accuracy are as expected whereas the orders of accuracy are around 1.0 for the 1st-order method and very close to 2.0 for the 2nd-order method. However, there is a degeneration in the spatial orders of accuracy as happened for the above test problems.

B. IMPICE METHOD FOR VISCOUS BURGERS' EQUATION

The viscous form of Burgers' Equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \epsilon\frac{\partial^2 u}{\partial x^2}, \tag{107}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

Table VII. Spatial and Temporal Errors: L_1-norms and the order of accuracy for the inviscid Burgers' problem at $T_{end} = 0.5$ on the spatial domain $[-1.0, 1.0]$. The temporal errors are calculated for the grid using N=200(cells) and the time-integrated exact solutions are the converged numerical solutions.

$et^u(T_{end})$					$es^u(T_{end})$				
	1st-order		2nd-order			1st-order		2nd-order	
C_{cfl}	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	N	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m
0.2	3.53E-03	—	6.89E-05	—	100	5.14E-02	—	1.86E-02	—
0.1	1.75E-03	1.02	1.67E-05	2.05	200	2.52E-02	1.03	6.08E-03	1.61
0.05	8.67E-04	1.01	4.22E-06	1.98	400	1.62E-02	0.64	3.27E-03	0.90
0.025	4.31E-04	1.01	1.05E-06	2.01	800	8.48E-03	0.93	1.52E-03	1.11
0.0125	2.15E-04	1.01	2.61E-07	2.01	1600	5.27E-03	0.69	8.16E-04	0.90

with $f(u) = \frac{1}{2}u^2$ and initial data $u(x,0) = u_0(x)$, where $u_0(x)$ is a given function and ϵ is a constant, $x \in \mathbb{R}$ and $t > 0$.

With the same spatial and temporal discretization as in Section 2 and known cell averages at time t_n, the steps to obtain cell averages at time t_{n+1} are as follows.

B.1. IMPICE Method Description

The Primary Phase: The equation of velocity evolution,

$$u_t + uu_x = \epsilon u_{xx}, \tag{108}$$

is written in Lagrangian form as

$$\frac{Du}{Dt} = \epsilon u_{xx}. \tag{109}$$

The face-centered fluxing velocity, $u^*_{j+\frac{1}{2}}$, is approximated using an explicit scheme in the Lagrangian frame as:

$$u^*_{j+\frac{1}{2}} = u^n_{j+\frac{1}{2}} + \frac{\Delta t}{2}\left[\epsilon\frac{u^n_{j+\frac{3}{2}} - 2u^n_{j+\frac{1}{2}} + u^n_{j-\frac{1}{2}}}{\Delta x^2}\right], \tag{110}$$

where the calculation of $u^n_{j+\frac{1}{2}}$ has already been discussed in Appendix A. In order to apply the Lagrangian and Eulerian phases, we rewrite equation (107) as

$$(\rho u)_t + \left(\rho u^2\right)_x = \left[\frac{1}{2}u^2 + \epsilon u_x\right]_x \tag{111}$$

where ρ is a constant and equal to 1.

The Lagrangian Phase: The discrete form of the Lagrangian part of equation (111) is as follows

$$V^L_j(\rho u)^L_j = V^n_j(\rho u)^L_j + \Delta t\left[\left(\frac{1}{2}\left(u^*_{j+\frac{1}{2}}\right)^2 + \epsilon\frac{u^*_{j+\frac{3}{2}} - u^*_{j-\frac{1}{2}}}{2\Delta x}\right) - \left(\frac{1}{2}\left(u^*_{j-\frac{1}{2}}\right)^2 + \epsilon\frac{u^*_{j+\frac{1}{2}} - u^*_{j-\frac{3}{2}}}{2\Delta x}\right)\right] \tag{112}$$

where $V^n_j = \Delta x$ and $V^n_j = \Delta x + \Delta t(u^*_{j+\frac{1}{2}} - u^*_{j-\frac{1}{2}})$ and ρ is a constant and equal to 1.

The Eulerian Phase: The Eulerian Phase for the viscous Burgers' Equation is the same as the Eulerian Phase for the inviscid Burgers' problem in Appendix A.1.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

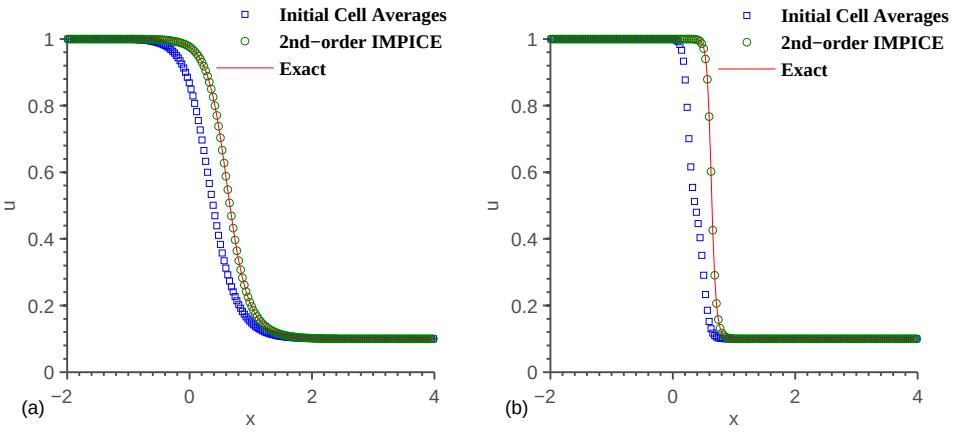

Figure 9. The 2nd-order (2nd-order-in-space and 2nd-order-in-time) IMPICE numerical solutions for the viscous Burgers' problem at $T_{end} = 0.5$ from the initial cell averages with N=200(cells) and $C_{cfl} = 0.2$: (a)$\epsilon = 0.05$ and (b)$\epsilon = 0.01$

B.2. Numerical Results and Accuracy in Space and Time

The initial condition for the viscous Burgers' problem satisfies the below analytical solution

$$u(x,t) = \begin{cases} (0.5 + 0.1e^{\frac{c}{\epsilon}} + e^{\frac{-a}{\epsilon}})/(1 + e^{\frac{c}{\epsilon}} + e^{\frac{-a}{\epsilon}}) & if \quad (a > 0) \quad and \quad (a > b) \\ (0.1 + 0.5e^{\frac{-c}{\epsilon}} + e^{\frac{-b}{\epsilon}})/(1 + e^{\frac{-c}{\epsilon}} + e^{\frac{-b}{\epsilon}}) & if \quad (b > 0) \quad and \quad (b > a) \\ (1 + 0.5e^{\frac{a}{\epsilon}} + 0.1e^{\frac{b}{\epsilon}})/(1 + e^{\frac{a}{\epsilon}} + e^{\frac{b}{\epsilon}}) & otherwise \end{cases} \tag{113}$$

where

$$a = \frac{x - 0.25 - 0.75t}{4}, \quad b = \frac{0.9x - 0.325 - 0.495t}{2}, \quad and \quad c = \frac{0.8x - 0.4 - 0.24t}{4}. \tag{114}$$

The numerical solutions of the viscous Burgers' problem on the spatial domain $[-2.0, 4.0]$ using the 2nd-order-in-space and 2nd-order-in-time IMPICE method for various values of ϵ are shown in Figures 9 and 10. The spatial and temporal error norms and orders of accuracy for the viscous Burgers' problem with these values of ϵ are summarized in Table VIII. The orders of accuracy for temporal errors are consistently around 1.0 for 1st-order method and around 2.0 for 2nd-order method. The convergence rates of spatial errors for the viscous Burgers' problem improve for larger values of ϵ, and get close to 1 for the 1st-order method and 2 for the 2nd-order method. However, there is a degeneration in accuracies for small ϵ. When $\epsilon = 0.0001$, the order is below 1 for the 1st-order method and approaching 1 for the 2nd-order method. This is due to the development of the steep front that appears close to a discontinuity in the numerical solution of the viscous Burgers' problem when ϵ approaches 0.

C. IMPICE METHOD VERSUS CELL-CENTERED ICE METHOD

The results in Figures 11–15 show the improvement obtained from the application of slope limiters in the data resconstruction of the Riemann problem. In Figures 11–15, the numerical results of the IMPICE method are compared against the numerical results of the cell-centered ICE method, which is implemented using Kashiwa *et al* [21] and chooses to conserve mass, linear momentum and total energy. We use first-order advection for both of these methods. As seen in 11–15, the IMPICE method helps to eliminate the non-physical oscillations in the implementation of the cell-centered ICE method.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

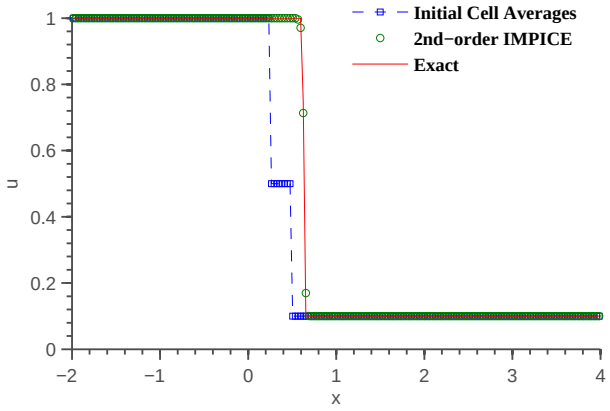

Figure 10. The 2nd-order (2nd-order-in-space and 2nd-order-in-time) IMPICE numerical solutions for the viscous Burgers' problem at $T_{end} = 0.5$ from the initial cell averages with N=200(cells) and $C_{cfl} = 0.2$ and $\epsilon = 0.0001$

Table VIII. Spatial and Temporal Errors: L_1-norms and the order of accuracy for the viscous Burgers' problem at $T_{end} = 0.5$ on the spatial domain $[-2.0, 4.0]$. The temporal errors are calculated for the grid using N=200(cells) and the time-integrated exact solutions are the converged numerical solutions.

		$et^u(T_{end})$					$es^u(T_{end})$			
		1st-order		2nd-order			1st-order		2nd-order	
ϵ	C_{cfl}	$\|.\|_{L_1}$	n	$\|.\|_{L_1}$	n	N	$\|.\|_{L_1}$	m	$\|.\|_{L_1}$	m
	0.2	1.04E-03	—	5.13E-06	—	100	1.73E-02	—	5.01E-04	—
	0.1	5.17E-04	1.00	1.28E-06	2.00	200	9.00E-03	0.94	1.43E-04	1.81
0.05	0.05	2.58E-04	1.00	3.21E-07	2.00	400	4.60E-03	0.97	4.11E-05	1.80
	0.025	1.29E-04	1.00	8.01E-08	2.00	800	2.32E-03	0.98	1.24E-05	1.73
	0.0125	6.41E-05	1.01	2.00E-08	2.00	1600	1.17E-03	0.99	3.53E-06	1.80
	0.2	2.18E-03	—	2.48E-05	—	100	3.54E-02	—	4.43E-03	—
	0.1	1.08E-03	1.01	6.17E-06	2.01	200	2.04E-02	0.80	1.08E-03	2.04
0.01	0.05	5.38E-04	1.01	1.54E-06	2.00	400	1.10E-02	0.89	3.15E-04	1.77
	0.025	2.68E-04	1.01	3.84E-07	2.00	800	5.78E-03	0.93	8.50E-05	1.89
	0.0125	1.33E-04	1.01	9.61E-08	2.00	1600	2.97E-03	0.96	2.23E-05	1.93
	0.2	2.91E-03	—	5.17E-05	—	100	5.78E-02	—	2.38E-02	—
	0.1	1.43E-03	1.02	1.27E-05	2.03	200	2.36E-02	1.29	1.20E-02	0.99
0.0001	0.05	7.11E-04	1.01	3.16E-06	2.00	400	1.46E-02	0.69	6.52E-03	0.88
	0.025	3.54E-04	1.01	7.84E-07	2.01	800	8.28E-03	0.82	3.26E-03	1.00
	0.0125	1.76E-04	1.01	1.96E-07	2.00	1600	4.94E-03	0.75	1.46E-03	1.16

D. DIFFERENT CALCULATIONS OF THE FACE-CENTERED PRESSURE, $p^*_{j+\frac{1}{2}}$

When discussing how to calculate the face-centered pressure, $p^*_{j+\frac{1}{2}}$, in the implementation of the IMPICE method in Section 5, we mentioned that there were two other ways to calculate this quantity in Kashiwa *et al* [21]. We will present in this section the proposed methods of [21] and see how these methods will change the results if implemented in the IMPICE method.

The following derivation is extracted from Kashiwa *et al* [21]. The first step in calculating the face-centered pressure $p^*_{j+\frac{1}{2}}$ is to differentiate the momentum equation. Taking the partial derivative of (7) in space, the obtained equation is:

$$(u_t + uu_x)_x = -\left(\frac{p_x}{\rho}\right)_x. \tag{115}$$

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

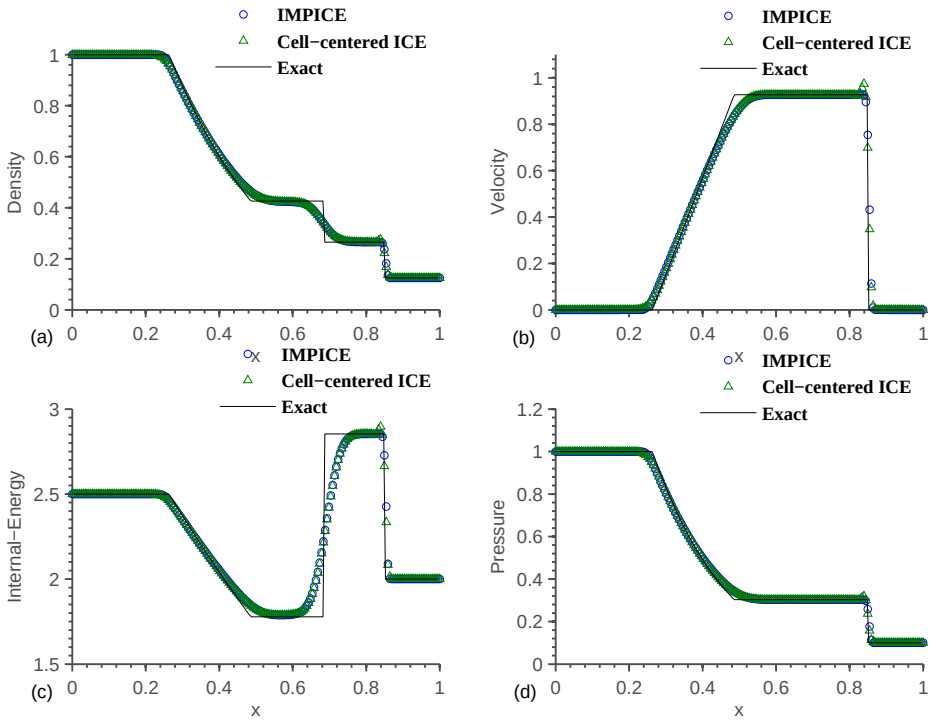

Figure 11. Cell-centered ICE and IMPICE numerical solutions for **P1** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

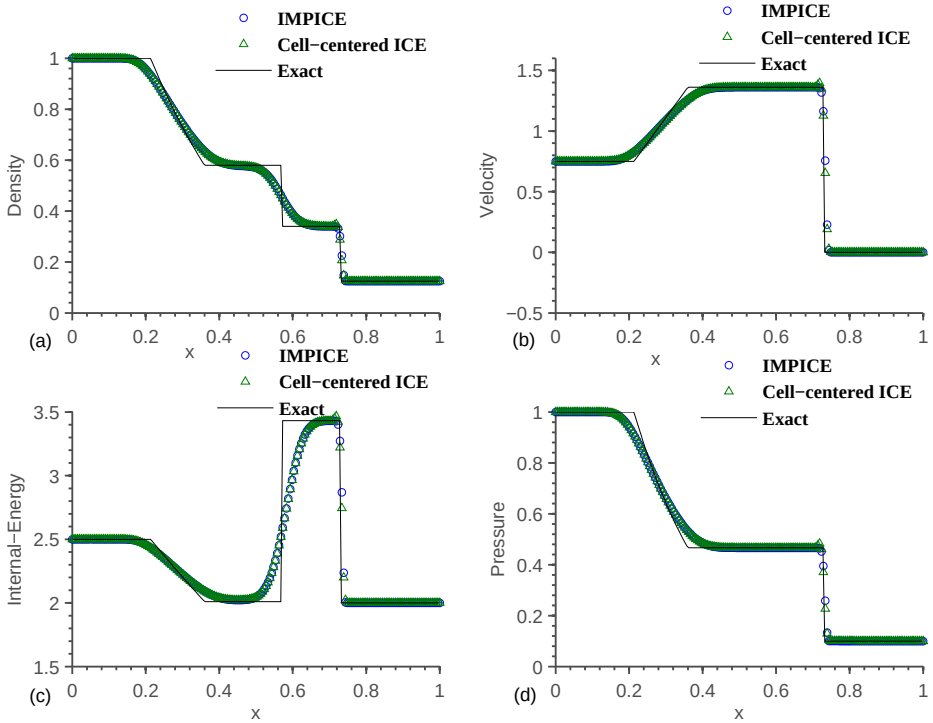

Figure 12. Cell-centered ICE and IMPICE numerical solutions for **P2** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

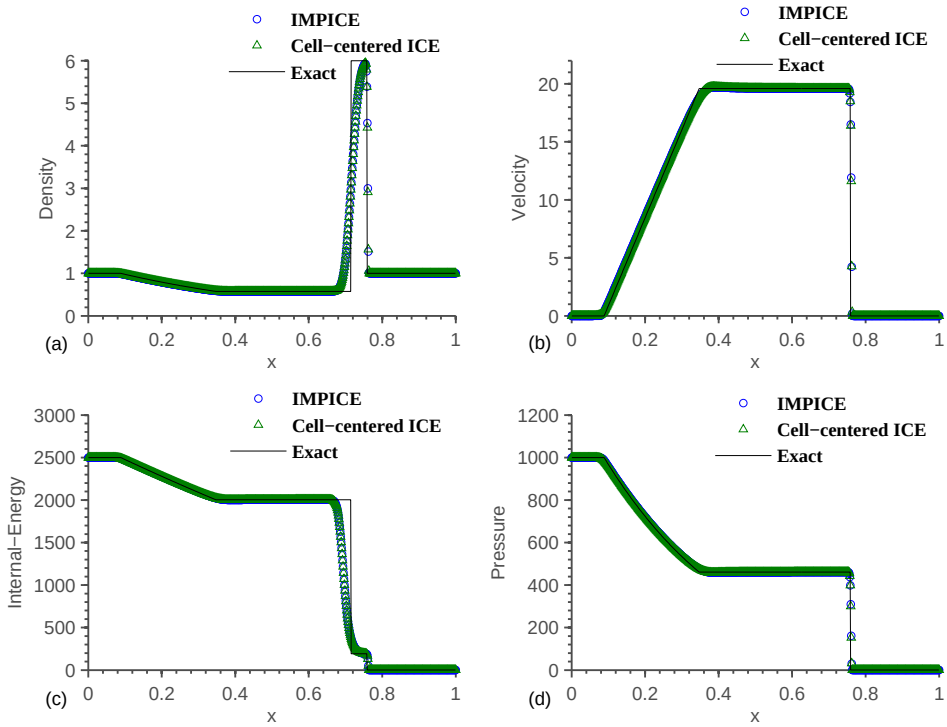

Figure 13. Cell-centered ICE and IMPICE numerical solutions for **P3** test problem with N=800(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

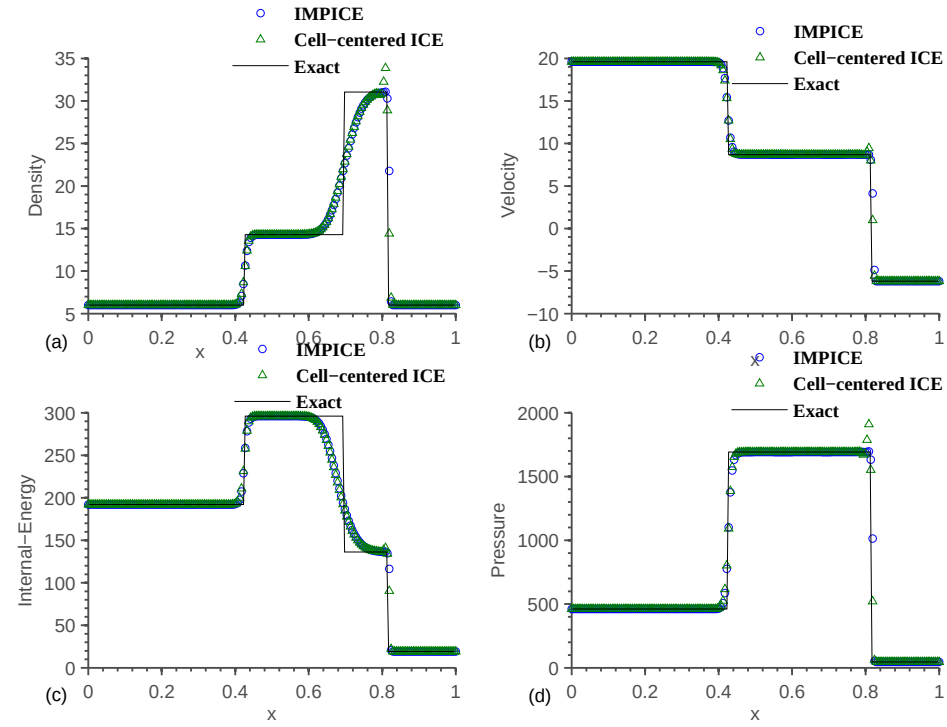

Figure 14. Cell-centered ICE and IMPICE numerical solutions for **P4** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

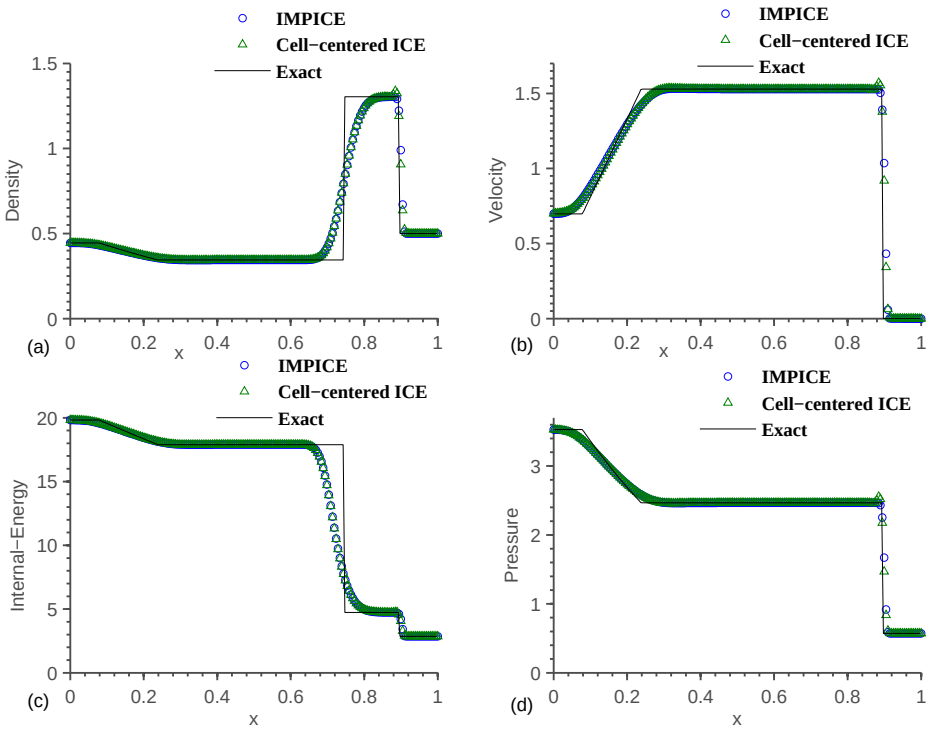

Figure 15. Cell-centered ICE and IMPICE numerical solutions for **P5** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

The time dependent quantity u_{tx} is eliminated using the partial time derivative of the pressure equation (14) which is given by

$$(p_t + up_x)_t = -\left(c^2\rho u_x\right)_t . \tag{116}$$

From these equations Kashiwa *et al* [21] state without derivation that linearization produces

$$u_{xt} = -\frac{u}{c^2\rho}\left(\frac{Dp}{Dt}\right)_x , \tag{117}$$

and so derive the potential equation for face-centered pressure:

$$\left(\frac{p_x}{\rho}\right)_x = \frac{u}{c^2\rho}\left(\frac{Dp}{Dt}\right)_x - (uu_x)_x . \tag{118}$$

One discrete form of this is

$$\frac{1}{\Delta x^2}\left[\frac{p_{j+1}^{n+\frac{1}{2}} - p_{j+\frac{1}{2}}^*}{\rho_{j+1}^n} - \frac{p_{j+\frac{1}{2}}^* - p_j^{n+\frac{1}{2}}}{\rho_j^n}\right] = \left(\frac{u}{c^2\rho}\right)_{j+\frac{1}{2}}^n \frac{1}{\Delta t\Delta x}\left(\delta p_{j+1}^n - \delta p_j^n\right)$$
$$-\frac{1}{\Delta x^2}\left[u_{j+1}^n(u_{j+1}^n - u_{j+\frac{1}{2}}^*) - u_j^n(u_{j+\frac{1}{2}}^* - u_j^n)\right] .$$

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

The face-centered pressure is then defined by

$$\begin{aligned} p^*_{j+\frac{1}{2}} = & \left(\frac{\rho^n_j p^{n+\frac{1}{2}}_{j+1} + \rho^n_{j+1} p^{n+\frac{1}{2}}_j}{\rho^n_{j+1} + \rho^n_j} \right) \\ & + \frac{\Delta x}{\Delta t} \left(\frac{u}{c^2 \rho} \right)^n_{j+\frac{1}{2}} \left(\frac{\rho^n_{j+1} \rho^n_j}{\rho^n_{j+1} + \rho^n_j} \right) \left(\delta p^n_{j+1} - \delta p^n_j \right) \\ & - \left(\frac{\rho^n_{j+1} \rho^n_j}{\rho^n_{j+1} + \rho^n_j} \right) \left[u^n_{j+1} (u^n_{j+1} - u^*_{j+\frac{1}{2}}) - u^n_j (u^*_{j+\frac{1}{2}} - u^n_j) \right]. \end{aligned} \tag{119}$$

The above equation is used to estimate the face-centered pressure , $p^*_{j+\frac{1}{2}}$, that will be used in the Lagrangian phase. It is recognized in [21] that the second term in equation (119) is important in high-speed problems and the third term looks somewhat like a bulk viscosity. These terms help to remove numerical noise, but introduces a diffusive effect in the method. A limited version of (119) is given by

$$\begin{aligned} p^*_{j+\frac{1}{2}} \quad = \quad & \left(\frac{\rho^n_j p^{n+\frac{1}{2}}_{j+1} + \rho^n_{j+1} p^{n+\frac{1}{2}}_j}{\rho^n_{j+1} + \rho^n_j} \right) \\ & + \psi \frac{\Delta x}{\Delta t} \left(\frac{u}{c^2 \rho} \right)^n_{j+\frac{1}{2}} \left(\frac{\rho^n_{j+1} \rho^n_j}{\rho^n_{j+1} + \rho^n_j} \right) \left(\delta p^n_{j+1} - \delta p^n_j \right) \\ & - \psi \left(\frac{\rho^n_{j+1} \rho^n_j}{\rho^n_{j+1} + \rho^n_j} \right) \left[u^n_{j+1} (u^n_{j+1} - u^*_{j+\frac{1}{2}}) - u^n_j (u^*_{j+\frac{1}{2}} - u^n_j) \right], \end{aligned} \tag{120}$$

where ψ is a "limiter" that is designed such that $0 \leq \psi \leq 1$, with values tending towards zero if the velocity field is smooth to remove numerical noise in the velocity. The DIVU limiter is introduced by Kashiwa and Lee in [20] is used for the purpose of limiting the velocity field in calculating limited face-centered pressure $p^*_{j+\frac{1}{2}}$. The limiter is required at the cell interface and is a function of the face-centered velocity divergence $D^n_{j+\frac{1}{2}}$ and the face-centered velocity divergences on either side of the face, $D^{n(+)}_{j+\frac{1}{2}}$ and $D^{n(-)}_{j+\frac{1}{2}}$. We define these face-centered velocity divergences as

$$D^n_{j+\frac{1}{2}} = u^n_{j+1} - u^n_j; \quad D^{n(+)}_{j+\frac{1}{2}} = u^n_{j+2} - u^n_{j+1}; \quad D^{n(-)}_{j+\frac{1}{2}} = u^n_j - u^n_{j-1}. \tag{121}$$

Then the limiter is given by

$$\psi = \begin{cases} 1 - \max \left[0, min \left(\frac{D^n_{j+\frac{1}{2}}}{D^{n(-)}_{j+\frac{1}{2}}}, \frac{D^n_{j+\frac{1}{2}}}{D^{n(+)}_{j+\frac{1}{2}}}, \frac{D^{n(-)}_{j+\frac{1}{2}}}{D^n_{j+\frac{1}{2}}}, \frac{D^{n(+)}_{j+\frac{1}{2}}}{D^n_{j+\frac{1}{2}}} \right) \right] & if \quad D^n_{j+\frac{1}{2}} \leq 0. \\ 0 \qquad otherwise. \end{cases} \tag{122}$$

In order to make sure the calculated face-centered pressure, $p^*_{j+\frac{1}{2}}$, is bounded by the surrounding cell-centered pressures at $t_{n+\frac{1}{2}}$, $p^{n+\frac{1}{2}}_j$ and $p^{n+\frac{1}{2}}_{j+1}$, its calculated value is clamped with respect to $[p_{min}, p_{max}]$ where

$$p_{min} = \min(p^{n+\frac{1}{2}}_j, p^{n+\frac{1}{2}}_{j+1}); \quad p_{max} = \max(p^{n+\frac{1}{2}}_j, p^{n+\frac{1}{2}}_{j+1}). \tag{123}$$

This means the face-centered pressure, $p^*_{j+\frac{1}{2}}$, is set to p_{min} if $p^*_{j+\frac{1}{2}} < p_{min}$ and is set to p_{max} if $p^*_{j+\frac{1}{2}} > p_{max}$.

We compare the numerical results obtained from the IMPICE method and the pressure-limited IMPICE method (PL-IMPICE) for the test cases in Table I in Figures 16–19. The PL-IMPICE method uses the implementation of the IMPICE method in Section 5 except that the face-centered pressure, $p^*_{j+\frac{1}{2}}$, is

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

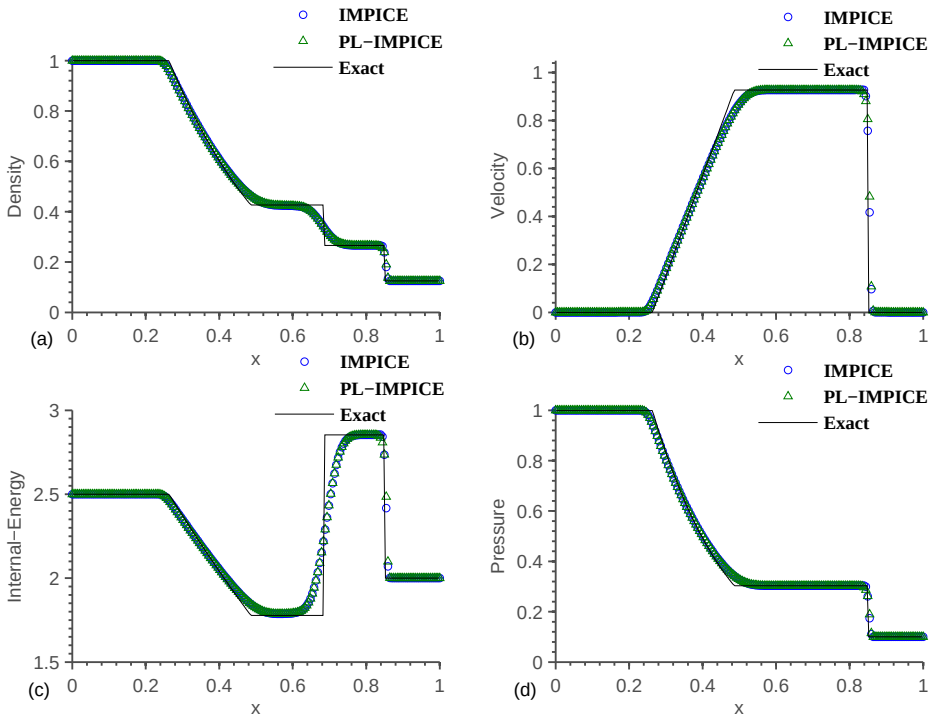

Figure 16. PL-IMPICE and IMPICE numerical solutions for **P1** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

calculated using the limited version in (120). As shown in Figures 16–19, there is a slight difference in the numerical solutions of these methods at the discontinuous regions. However, there are no non-physical oscillations presented in the numerical solutions of these two methods.

REFERENCES

1. Ben-Artzi M, Falcovitz J. A second order Godunov-Type Scheme for Compressible Fluid Dynamics. *J. Comput. Phys.* 1984; **55**:1-32.
2. Ben-Artzi M. Application of the Generalised Riemann Problem Method to 1-D Compressible Flows with Interfaces. *J. Comput. Phys.* 1986; **65**:170-178.
3. Berzins M. Nonlinear Data-Bounded Polynomial Approximations and their Applications in ENO Methods. *Numerical Algorithms* 2010; **55**(2):171-188
4. Casulli V, Greenspan D. Pressure Method for the Numerical Solution of Transient, Compressible Fluid Flows. *Int. J. Num. Meth. Fluids* 1984; **4**:1001-1012.
5. Colella P. A Direct Eulerian MUSCL Scheme for Gas Dynamics. *SIAM J. Sci. Stat. Comput.* 1985; **6**:104-117.
6. Davis SF. Simplified Second-order Godunov-Type Methods. *SIAM J. Sci. Stat. Comput.* 1988; **9**(3):445-473.
7. Gaskell PH, Lau AKC. Curvature-Compensated Convective Transport: SMART, a New Boundedness-Preserving Transport Algorithm. *Int. J. Num. Meth. Fluids* 1988; **8**(6):617-641.
8. Greenough JA, Rider WJ. A quantitative comparison of numerical methods for the compressible Euler equations: fifth-order WENO and piecewise-linear Godunov. *Journal of Computational Physics* 2004; **196**:259-281.
9. Guilkey JE, Harman T, Banerjee B. An Eulerian-Lagrangian Approach for Simulating Explosions of Energetic Devices. *Computers and Structures* 2007; **85**:660-674.
10. Guilkey JE, Harman T, Xia A, Kashiwa B, McMurtry PA. An Eulerian-Lagrangian Approach for Large Deformation Fluid Structure Interaction Problems, Part 1: Algorithm Development. *WIT Press* 2003; 143–156.
11. Harlow FH, Amsden AA. Numerical Calculation of Almost Incompressible Flow. *Journal of Computational Physics* 1968; **3**:80-93.

Copyright © 0000 John Wiley & Sons, Ltd.

Prepared using fldauth.cls
DOI: 10.1002/fld

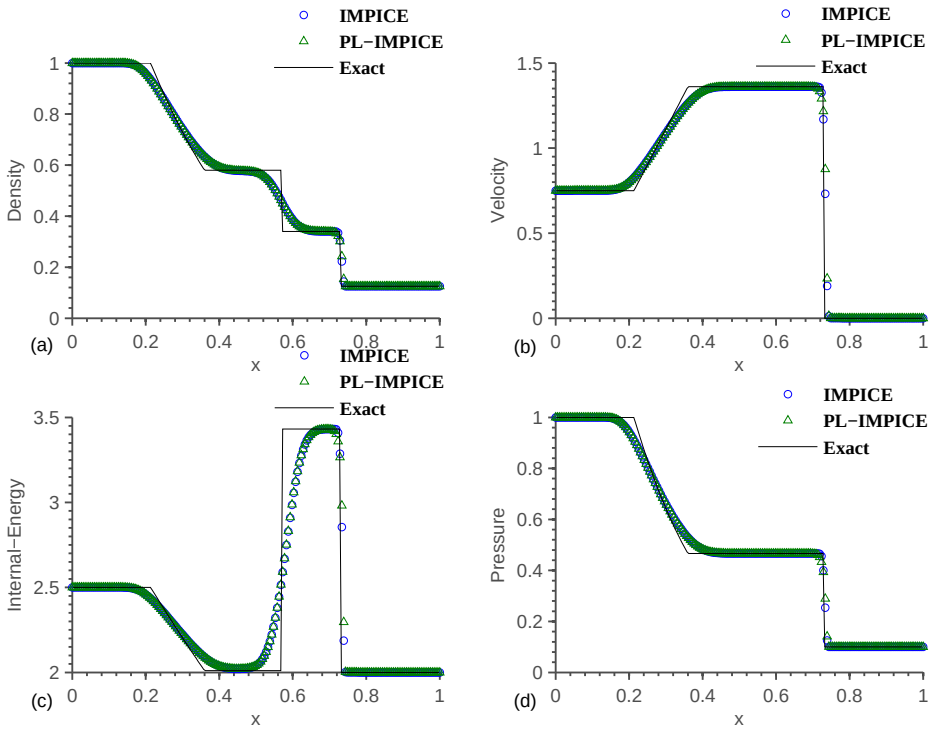

Figure 17. PL-IMPICE and IMPICE numerical solutions for **P2** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

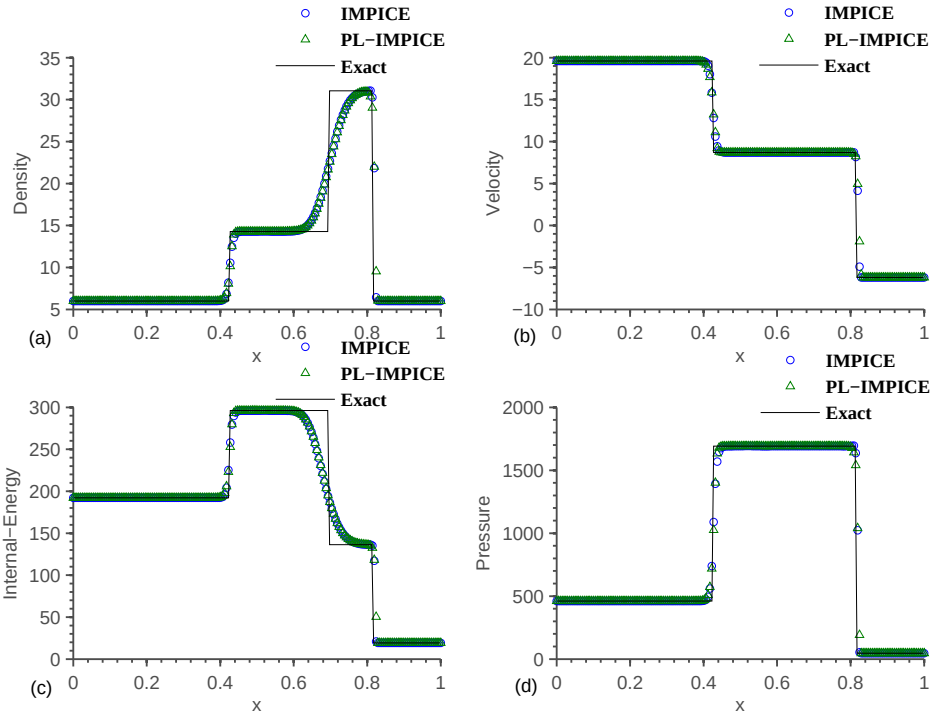

Figure 18. PL-IMPICE and IMPICE numerical solutions for **P4** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

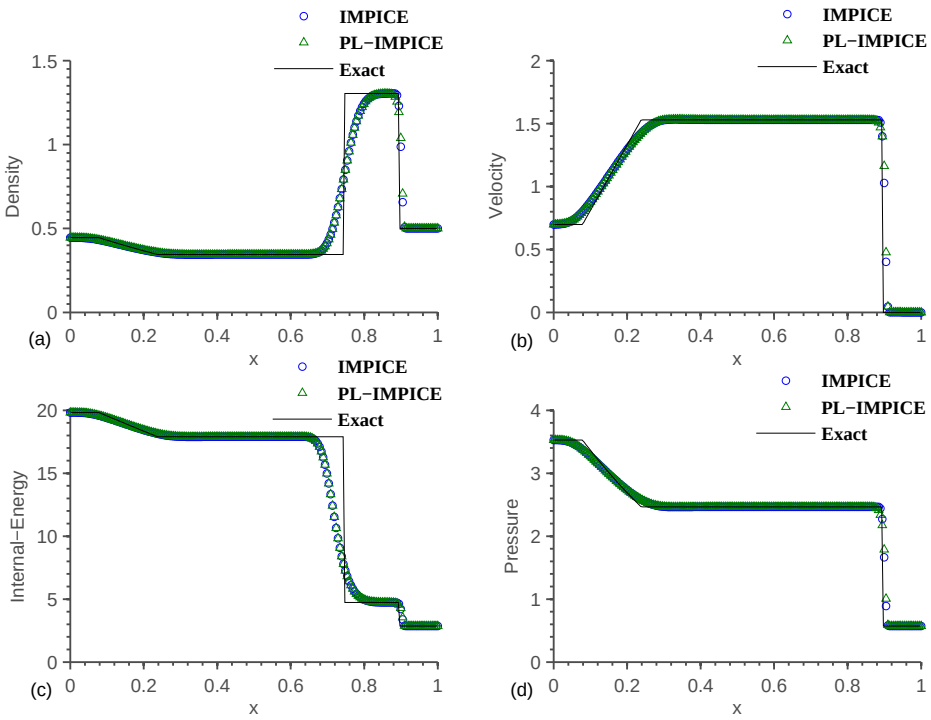

Figure 19. PL-IMPICE and IMPICE numerical solutions for **P5** test problem with N=200(cells) and $C_{cfl} = 0.2$: (a) density; (b) velocity; (d) internal-energy; and (c) pressure.

12. Harlow FH, Amsden AA. A Numerical Fluid Dynamics Calculation Method for All Flow Speeds. *Journal of Computational Physics* 1971; **8**:197-213.
13. Harlow FH, Amsden AA. Numerical Calculation of Multiple Fluid Flow. *Journal of Computational Physics* 1975; **17**:19-52.
14. Harman T, Guilkey JE, Kashiwa B, Schmidt J, McMurtry P. An Eulerian-Lagrangian Approach for Large Deformation Fluid Structure Interaction Problems, Part 2: Multi-Physics Simulations within a Modern Computational Framework. *WIT Press* 2003; **Fluid Structure Interaction II**:157-166.
15. Harten A, Lax PD, van Leer B. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Rev.* 1983; **25**:35-61.
16. Harten A, Osher S. Uniformly High-Order Accurate Nonoscillatory Schemes I. *SIAM J. Numer. Anal.* 1987; **24**:279-309.
17. Hou TY, LeFloch PG. Why Nonconservative Schemes Converge to Wrong solutions: Error Analysis. *Mathematics of Computation* 1994; **62**:497-530.
18. Issa RI, Gosman AD, Watkins AP. The Computation of Compressible and Incompressible Flow of Fluid with a Free Surface. *Phys. Fluids* 1965; **8**(12):2182-2189.
19. Issa RI. Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting. *Journal of Computational Physics* 1986; **62**:40-65.
20. Kashiwa BA, Lee WH. Comparisons between the Cell-Centered and Staggered Mesh Lagrangian Hydrodynamics. In *Advances in the Free Lagrange Method*, Trease HE, Fritz MJ, Crowley WP (eds), Springer Verlag, Berlin 1991; 277-288
21. Kashiwa BA, Padial NT, Rauenzahn RM, Vanderheyden WB. A Cell-Centered ICE Method for Multiphase Flow Simulations. *Proceedings ASME Symposium on Numerical Methods for Multiphase Flows, Lake Tahoe, NV, 19-23 June 1994*.
22. Kashiwa BA. A Multified Model and Method for Fluid-Structure Interaction Dynamics. *Los Alamos National Laboratory, Los Alamos* 2001; Technical Report LA-UR-01-1136.
23. Kwatra N, Su J, Gretarsson J, Fedkiw R. A method for avoiding the acoustic time step restriction in compressible flow. *Preprint submitted to JCP*.
24. Laney CB. Computational Gasdynamics. *Cambridge University Press* 1998; ISBN 0-521-62558-0.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld

25. Lax PD. Weak Solutions of Nonlinear Hyperbolic Equations and their Numerical Computation. *Commun. Pure Appl. Math.* 1954; **7**:159–193.
26. Lax PD, Wendroff B. Systems of Conservation Laws. *Communications in Pure and Applied Mathematics* 1960; **13**:217–237.
27. Livne OE. ICE Algorithm and the Davis Advection Scheme. *SCI Institute, University of Utah* 2006; Technical Report No. UUSCI-2006-006.
28. Livne OE. ICE Algorithm for the Shocktube Problem. *SCI Institute, University of Utah* 2006; Technical Report No. UUSCI-2006-007.
29. Martín MP, Taylor EM, Wu M, Weirs VG. A Bandwidth-optimized WENO Scheme for the Effective Direct Numerical Simulation of Compressible Turbulence. *Journal of Computational Physics* 2006; **220**:270–289.
30. Mehdizadeh Khalsaraei M. An Improvement on the Positivity Results for 2-stage Explicit Runge-Kutta Methods. *Journal of Computational and Applied Mathematics* 2010; **235**:137-143.
31. Reinelt D, Laurs A, Adomeit G. Ignition and Combustion of a packed bed in a Stagnation Point Flow. *Combustion and Flame* 1994; **99**(2):395-403.
32. Shu CW, Osher SJ. Efficient Implementation of Essentially Nonoscillatory Shock Capturing Schemes II. *J. Comp. Phys.* 1989; **83**:32-78.
33. Sweby PK. High Resolution Schemes using Flux-Limiters for Hyperbolic Conservation Laws. *SIAM J. Num. Anal.* 1984; **21**:995-1011.
34. Toro EF. Riemann Solvers and Numerical Methods for Fluids Dynamics: A Practical Introduction. *Springer, Third Edition* 2008; ISBN 978-3-540-25202-3.
35. Toro EF, Hidalgo A, Dumbser M. FORCE schemes on unstructed meshes I: Conservative Hyperbolic Systems. *Journal of Computational physics* 2009; **228**:3368-3389.
36. Van Albada GD, Van Leer B, Roberts WW. A Comparative Study of Computational Methods in Cosmic Gas Dynamics. *Astron. Astrophysics* 1982; **108**:76-
37. van der Heul DR, Vuik C, Wesseling. A Conservative Pressure-Correction Method for Flow at All Speeds. *Computers and Fluids* 2003; **32**:1113-1132.
38. VanderHeyden WB, Kashiwa BA. Compatible Fluxes for Van Leer Advection. *Journal of Computational Physics* 1998; **146**:1-28.
39. Van Leer B. Towards the Ultimate Conservative Difference Scheme II. Monotonicity and Conservation Combined in a Second Order Scheme. *J. Comp. Phys.* 1974; **14**:361-370.
40. Van Leer B. Towards the Ultimate Conservative Difference Scheme III. Upstream-Centered Finite-Difference Schemes for Ideal Compressible Flow. *J. Comp. Phys.* 1977; **23**:263-275.
41. Van Leer B. Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Godunov's Method. *J. Comp. Phys.* 1979; **32**:101-
42. Van Leer B. On the Relation Between the Upwind-Differencing Schemes of Godunov, Enguist-Osher and Roe. *SIAM J. Sci. Stat. Comput.* 1985; **5**(1):1-20.
43. Waterson NP, Deconinck H. A Unified Approach to the Design and Application of Bounded Higher-Order Convection Schemes. *In Numerical Methods in Laminar and Turbulent Flows, Proceedings of the Ninth International Conference* 1995; **9**(1):203-214.

Copyright © 0000 John Wiley & Sons, Ltd.
Prepared using fldauth.cls

DOI: 10.1002/fld