Computers & Fluids 54 (2012) 105–117

Contents lists available at SciVerse ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Solving the Euler and Navier–Stokes equations by the AMR–CESE method

Chaowei Jiang [a,*], Shuxin Cui [b], Xueshang Feng [a]

[a] SIGMA Weather Group, State Key Laboratory for Space Weather, Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Jet Propulsion, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 26 December 2010
Received in revised form 28 September 2011
Accepted 7 October 2011
Available online 19 October 2011

Keywords:
Adaptive mesh refinement
The CESE method
Shock capture
Boundary layer

ABSTRACT

The application of the AMR–CESE method for solving the Euler and Navier–Stokes equations is presented. The method is a combination of the space–time conservation element and solution element (CESE) method and the adaptive mesh refinement (AMR) technique. Its implementation is based on modification of the original CESE method and utilization of the framework of a parallel-AMR package PARAMESH to manage the block-AMR grid system. Furthermore, a variable time step algorithm is introduced to realize adaptivity of the method in both space and time. A test suite of standard problems for Euler and Navier–Stokes flows are calculated and the results show high resolution, high efficiency and versatility of the method for both shock capture and resolving boundary layer.

1. Introduction

Adaptive mesh refinement (AMR) is an attractive avenue in modern CFD as a compromise between high resolution requirement and computational resource limitation. By automatically adapting the computational mesh to the solution of the governing partial differential equations (PDEs), methods based on AMR can assign more mesh points for regions demanding high resolution (e.g., high gradient regions) and at the same time, give fewer mesh points to other less interesting regions (low gradient regions), thereby providing the required resolution while minimizing memory requirements and CPU time. AMR technique has been developing rapidly since the pioneering work by Berger and Colella [1], and combined with the parallel computation technique, it is becoming a standard tool for treating problems with multi-orders of spatial or temporal scales. Up to now, there have been a number of software infrastructures developed to support the parallel-AMR implementation of PDEs. These include AmrLib/BoxLib [2], Chombo [3], DAGH [4], GrACE [5], SAMRAI [6] and PARAMESH [7].

PARAMESH, a package of Fortran 90 subroutines, is designed to provide an application developer with an easy route to extend an existing serial code which uses a logically Cartesian structured mesh into a parallel code with AMR [7]. The package builds a hierarchy of sub-grids (i.e., blocks) to cover the computational domain, with spatial resolution varying to satisfy the demands of the application. Each grid block has the same logically Cartesian mesh with different spatial scale. Between the blocks, only jumps of a factor of two in resolution are allowed. These sub-grid blocks form the nodes of a tree data structure (quad-tree in 2D or oct-tree in 3D), which is convenient to organize the blocks and their relationships. The adaptation of the solution is realized by dynamically dividing (refining) and coarsening appropriate blocks when running. The package is fully parallelized on the share-memory parallel computer using the message passing interface (MPI) library. Due to the high performance and usability of this publicly available toolkit, it has been utilized for management of the AMR grid system and parallelization in many codes, including the AMR–MHDFCT, ATHENA and FLASH in the astrophysics community [7–9], and available on the website http://www.physics.drexel.edu/olson/paramesh-doc/Users_manual/amr.html. In the framework of PARAMESH, what the user need to provide includes: (1) a runtime parameter file containing the basic configurations; (2) a main control program; (3) a subroutine of refinement criteria to test the refinement or de-refinement of the blocks; (4) boundary conditions and (5) the final but the most important one, a solver working on one single block.

With the help of PARAMESH, the AMR–CESE method as its name implies, is an implementation of the space–time conservation element and solution element (CESE) method on the AMR grid [10]. The CESE method, originated by Chang [11,12], is a relatively novel but successful scheme for conservation laws. Different from the traditional numerical methods, the key principle of the CESE method is treating space and time as one entity in calculating flux balance. It is a genuine multidimensional and high-resolution scheme without any type of dimensional splitting and Riemann solver. In the CESE method, the first-order spatial derivatives are also considered as variables to be solved. By designing the solution element (SE) and conservation element (CE), the CESE method allows the

* Corresponding author.
E-mail address: cwjiang@spaceweather.ac.cn (C. Jiang).

0045-7930/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.10.006

physical parameters to have smooth profiles inside a SE, while between SEs or in CEs, they may be discontinuous. Thus, the CESE method can capture sharp discontinuity within a few grid points, and in addition, the introduced numerical damping effect in this method is controllable. The CESE method has been applied to many fields ranging from hydrodynamics to magnetohydrodynamics including the viscous and invisid flows and the capabilities are well-established [12–16]. Combining such an excellent scheme with the AMR technique is valuable and of significance for many applications considering both of their merits. In our previous work, the AMR–CESE method is proposed and applied to magnetohydrodynamics and the advantages have emerged [10,17].

In this paper, we report solving the conservation laws of the Euler equations and Navier–Stokes (NS) equations using the AMR–CESE method. This is a further application and examination of the method and serves an extension of our previous work [10] with some more issues addressed. The paper is organized as follows. Firstly a brief summary of our realization of the AMR–CESE method is given in Section 2 after presenting the governing equations to be solved. Then we provides a suite of numerical tests for both the Euler and NS equations in Section 3 with relevant discusses. Finally we offer concluding remarks in Section 4.

2. The AMR–CESE method

2.1. The governing equations

We aim at solving the time-dependent Euler and Navier–Stokes (NS) equations. In conservation form, the non-dimensionalized equations can be written as

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}-\frac{\partial \mathbf{F}_v}{\partial x}-\frac{\partial \mathbf{G}_v}{\partial y}=\mathbf{0} \tag{1}$$

where the flow variables ($\mathbf{U}$) and inviscid fluxes ($\mathbf{F},\mathbf{G}$) are

$$\mathbf{U}=(\rho,\rho\mathbf{v},E)^T=(\rho,\rho v_x,\rho v_y,E)^T, \tag{2}$$

$$\mathbf{F}=\begin{pmatrix}\rho v_x\\ \rho v_x^2+p\\ \rho v_x v_y\\ (E+p)v_x\end{pmatrix};\quad \mathbf{G}=\begin{pmatrix}\rho v_y\\ \rho v_y v_x\\ \rho v_y^2+p\\ (E+p)v_y\end{pmatrix}. \tag{3}$$

The viscous fluxes ($\mathbf{F}_v,\mathbf{G}_v$) are set to be zero for Euler equations. For NS equations they are

$$\mathbf{F}_v=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ v_x\tau_{xx}+v_y\tau_{xy}-q_x\end{pmatrix};\quad \mathbf{G}_v=\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ v_x\tau_{xy}+v_y\tau_{yy}-q_y\end{pmatrix}, \tag{4}$$

respectively. Here ρ, p, $\mathbf{v}$ are density, pressure and velocity with components (v_x,v_y). The specific total energy is $E=p/(\gamma-1)+\rho\left(v_x^2+v_y^2\right)/2$ and γ is the ratio of the specific heats. The stress components in Eq. (4) are as follows,

$$\begin{aligned}
\tau_{xx}&=\frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial v_x}{\partial x}-\frac{2}{3}\frac{\partial v_y}{\partial y}\right),\\
\tau_{yy}&=\frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial v_y}{\partial y}-\frac{2}{3}\frac{\partial v_x}{\partial x}\right),\\
\tau_{xy}&=\frac{\mu}{Re}\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right),\\
q_x&=-\frac{\mu}{Re}\frac{1}{(\gamma-1)M^2Pr}\frac{\partial T}{\partial x},\\
q_y&=-\frac{\mu}{Re}\frac{1}{(\gamma-1)M^2Pr}\frac{\partial T}{\partial y}
\end{aligned} \tag{5}$$

where μ is the viscosity coefficient, Re is the Reynolds number, Pr is the Prandtl number and M is the freestream Mach number. T is the temperature and given by the equation of state $\gamma M^2 p=\rho T$. According to the Sutherland's law, the viscosity coefficient is

$$\mu=T^{1.5}\frac{1+T_s/T_0}{T+T_s/T_0} \tag{6}$$

with the Sutherland temperature $T_s=110$ K and the reference temperature $T_0=298$ K. Since in the CESE method the gradients of flow variables ($\mathbf{U}_x,\mathbf{U}_y$) are also included as the solution variables besides $\mathbf{U}$, the viscous fluxes can be expressed as functions of $\mathbf{U}$, $\mathbf{U}_x$, $\mathbf{U}_y$.

2.2. The improved CESE method

Two modifications are brought to the original CESE method here for the purpose of application on the block-AMR grid system. As the first one, we rearrange the space–time solution points in an uniform rectangular mesh as shown in Fig. 1. The spatial domain is divided into rectangular meshes evenly, with i, j and n as the indices for x, y and t, respectively. Here, the projections of the solution points onto the x–y plane are directly given on the mesh nodes. The mesh nodes marked by filled circle such as A, B, C and D belong to solution points at time level $n-1/2$ and the open-circle points like G belong to the time level of n, with the time interval of $\Delta t/2$. This is more convenient than the original method where the CESE solution points must be calculated after setting the grids. The reason is that here for any rectangular, e.g., ABCD, its centroid is explicitly the center point, e.g., G. We note that such improvement is also adopted in [18,19], and we have extended it to the three-dimensional case in a more general way [10].

The modification can further show its advantages when defining the solution element and conservation element and advancing the solutions (Fig. 2). The solution element, for instance, of G′ is defined as the union of three planes A′B′C′D′, W″E″EW, S″N″NS, which intersect at G′ and are perpendicular to each other. Similarly, the conservation element used for solution advancing at G′ is the cuboid of ABCDA′B′C′D′ since G′ is the centroid of A′B′C′D′. The CESE method advances the solution at G′ by balancing the total space–time fluxes leaving through the surface of the CE, using a general version of the Gauss's divergence law in the three-dimensional Euclidean space E_3 of (x,y,t),

$$\oint_{S(V)}\mathbf{h}_m\cdot d\mathbf{s}=\int_V\nabla\cdot\mathbf{h}_m dV=0. \tag{7}$$

Here $m=1,\ 2,\ 3,\ 4$ denotes the components, $\mathbf{h}_m=(F_m-F_{vm},G_m-G_{vm},U_m)$ is the space–time flux vector, $S(V)$ is

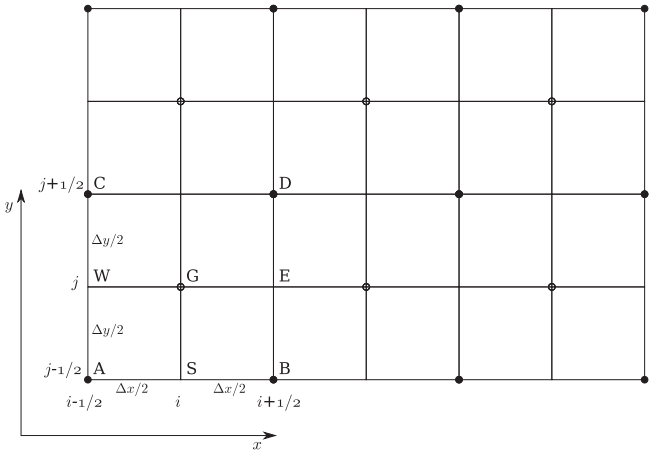

Fig. 1. The CESE scheme in 2D; the projection of the mesh points in E_3 onto the x–y plane.

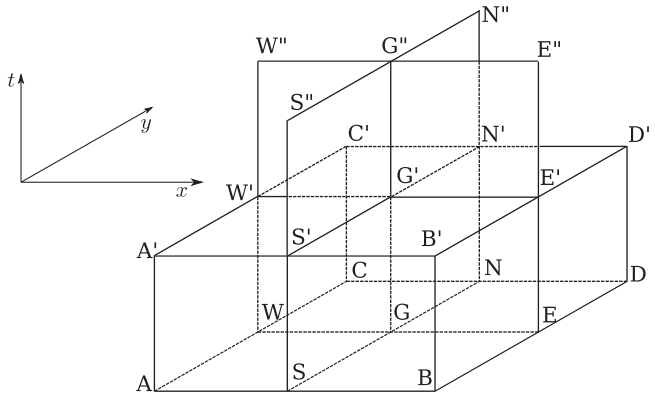

Fig. 2. The CESE scheme in 2D; definitions of CE and SE.

the boundary of the space–time volume V in E_3, and $\mathbf{h}_m \cdot d\mathbf{s}$ is the space–time flux leaving through the surface element $d\mathbf{s}$.

Now, unlike the original method, note that the surfaces of the CE are parallel to the coordinate planes. Consequently, the integration of fluxes leaving through the surface of the CE is greatly simplified, as the normal vector of each face is along the corresponding axis and only one component of the spacetime flux vector is needed on each face. This will reduce the CPU time significantly as majority of the computation time in the CESE method is consumed by the flux integration. The simplification stands out especially for the three-dimensional case [10].

After setuping the above basic ingredients of the CESE method, the CESE advancing routine for flow variables $\mathbf{U}$ can be derived straightforwardly, and we simply present the final formula here

$$\begin{aligned}(U_m)_{i,j}^{n} = & \frac{1}{\Delta x \Delta y} \sum_{p=-1,1;q=-1,1} \\ & \times \frac{\Delta x \Delta y}{4}\left[(U_m)_{ip,jq}^{n-1/2} + (U_{mx})_{ip,jq}^{n-1/2}\left(-p\frac{\Delta x}{4}\right) + (U_{my})_{ip,jq}^{n-1/2}\left(-q\frac{\Delta y}{4}\right)\right] \\ & - p \\ & \times \frac{\Delta y \Delta t}{4}\left[(F_m - F_{vm})_{ip,jq}^{n-1/2} + (F_{my})_{ip,jq}^{n-1/2}\left(-q\frac{\Delta y}{4}\right) + (F_{mt})_{ip,jq}^{n-1/2}\left(\frac{\Delta t}{4}\right)\right] \\ & - q \\ & \times \frac{\Delta x \Delta t}{4}\left[(G_m - G_{vm})_{ip,jq}^{n-1/2} + (G_{mx})_{ip,jq}^{n-1/2}\left(-p\frac{\Delta x}{4}\right) + (G_{mt})_{ip,jq}^{n-1/2}\left(\frac{\Delta t}{4}\right)\right], \end{aligned} \tag{8}$$

where $ip = i + p/2$, $jq = j + q/2$ for short and the subscripts x, y and t denote the corresponding derivatives as usual in descriptions of the CESE method [13,14,10]. For computations of the spatial gradients of the flow variables, $\mathbf{U}_x$ and $\mathbf{U}_y$, the similar procedures of finite-difference and reweighting of an α type are used. In the present work for solving the Euler equations, we set $\alpha = 2$ and for NS equations, we set $\alpha = 0$, which is equivalent to central-difference when calculating the spatial gradients.

The second modification is a minor one by merging two half timesteps into one full timestep with the variables of time level n reserved only. That is because for the case of the grid system with an AMR algorithm, variables on new child blocks generated by refinement are prolonged from their underlying parent blocks using spatial interpolation. On the other hand if a block is coarsened, variables on the relevant child-blocks are restricted back to their parent. Such operations (prolongation and restriction) must be done on data of the same time level. This requirement can be realized by hiding the intermediate time level $n + 1/2$ in the subroutine of CESE solver, which only provides the interface for the solution of time level n with integer numbers. Note that in this way, the scheme can be regarded as a nine-point (3×3) stencil with diagonal points in two-dimension, and no space–time staggering exists.

2.3. AMR implementation

2.3.1. The basic configuration

With the CESE solver well prepared, we integrate it into the framework of the PARAMESH code. In details for the basic configuration:

(1) we specify all the solution variables $(\mathbf{U}, \mathbf{U}_x, \mathbf{U}_y)$ at cell center, with the number of variables `nvar` $= 4 \times 3 = 12$ defined in PARAMESH.
(2) The blocks are set to consist of 8×8 cells, typically, to enable easy load balance and flexible AMR. One layer of guard cells containing diagonal elements is specified for the nine points stencil. The guard cells are used for communications between the neighboring blocks.
(3) The linear interpolation is used when prolonging the data from parent blocks to their new born child blocks and average is used when restricting data on child blocks back to their parents. The same interpolation is used when filling guard cells at block boundaries next to less refined neighbor blocks. The prolongation and restriction are carried out also on the variable gradients $(\mathbf{U}_x, \mathbf{U}_y)$ which make our scheme much different from the AMR implementations on finite volume scheme.

2.3.2. Refinement criteria

Then a set of refinement criteria is needed for the program to decide which blocks required to be refined or de-refined (i.e., coarsened) when running. Although several approaches are possible, in this work, we use multiple physics-based refinement criteria [20–23]. They are the curl and divergence of velocity and the gradients of density and pressure, giving the refinement criteria computed for each cell as

$$\begin{aligned} \chi_d &= \sqrt{V}\frac{|\nabla \cdot \mathbf{v}|}{|\mathbf{v}| + \epsilon a}, \quad \chi_c = \sqrt{V}\frac{|\nabla \times \mathbf{v}|}{|\mathbf{v}| + \epsilon a}, \\ \chi_\rho &= \sqrt{V}\frac{|\nabla \rho|}{\rho}, \quad \chi_p = \sqrt{V}\frac{|\nabla p|}{p}. \end{aligned} \tag{9}$$

Here V is the cell volume, $a = \gamma\sqrt{p/\rho}$ is the sound speed and a factor $\epsilon \ll 1$ is introduced to get rid of the concern when $\mathbf{v}$ is equal to zero. These criteria can detect all distinctive features in the flow, like the shock, the shear layer, the vortex and the contact discontinuity [23]. Different combination of these criteria can be used for particular purpose. It is worth noting that the quantities in Eq. (9) can be directly obtained by using the solution gradients $(\mathbf{U}_x, \mathbf{U}_y)$ without any finite-difference.

For a given combination of these criteria, the program tests each block in the following way. If any of the maxima of the criteria in the combination is greater than the threshold for refinement, the block is flagged to be refined, while if all of the maxima of these criteria are less than the threshold for coarsening, the block is flagged to be coarsened. In order to set the thresholds, the standard deviation about zero is computed for each χ [21]:

$$\sigma_R = \sqrt{\frac{\sum_{i=1}^{N}(\chi_R^2)_i}{N}}, \tag{10}$$

with i pertaining to the indices of all of cells in the computation domain and R denoting the subscripts in Eq. (9). Multiplying each σ by different, properly chosen factors gives the thresholds.

2.3.3. Variable timestep

By now, the implementation of the CESE method on AMR grid is established for the uniform-timestep scheme that using a uniform timestep Δt globally in each advancing step. However, if defining the block local timestep Δt_b by using the local grid size ΔL_b, a fixed Courant number CFL and the global maximal wave speed v_{max} as

$$\Delta t_b = \mathrm{CFL}\frac{\Delta L_b}{v_{max}}, \tag{11}$$

it varies significantly due to the intrinsic variation in resolution of the AMR grid. In such situation, using an uniform finest timestep for the CESE method can produce excessive numerical diffusion for the region with relatively low resolution. Since the numerical diffusion of the CESE method is sensitive to the ratio of the timestep to the grid size (or the local timestep), especially when this ratio is smaller than 0.1 [24,25]. That is very common in the AMR grid with typically four or five levels of refinement used and the resulting numerical dissipation is in particular unfavorable for the viscous flows with high Reynolds number.

On an ordinary non-uniform grid the sensitivity of the CESE method to the timestep can be partially overcome by a Courant number insensitive (CNIS) scheme [24]. A more effective and suitable method on the block-AMR grid is a variable timestep algorithm, which is using the local timesteps directly on the blocks. This kind of method can reduce the numerical diffusion more thoroughly than the CNIS scheme, and furthermore, save the computation resource effectively by avoiding the unnecessary fine-steps of solution advancing on the coarser blocks. Similar procedures of using local time stepping in the CESE method can also be found in [26,27].

Here we give a short description of our variable timestep algorithm [10]. We group the blocks by their local timesteps with each group of blocks having the same timestep. Actually, the group can also be defined by the refinement level. In Fig. 3, for instance with three refinement levels used, the blocks are divided into three groups with timesteps of Δt_1, Δt_2 and Δt_3, respectively, and $\Delta t_1 = 2\Delta t_2 = 4\Delta t_3$. Correspondingly, the groups are name as `group 1`, `group 2` and `group 3`. The figure shows one complete advancing step before the modification of the grid by the refine-derefine procedure. The color arrows represent one step of solution advance of a given group, and the groups with the same color can be advanced simultaneously. The black arrows denote the necessity of time interpolation of solution when guard cell filling. That is at the interface of timestep jump, linear time interpolation from the larger-timestep block is used to set guard cell values when needed on the block with the finer timestep. The grid will not be modified until all blocks have been advanced through the biggest timestep Δt_1.

2.3.4. Global conservation

In principle, the global conservation should be preserved in solving the conservation laws, which needs some additional considerations on AMR grid (e.g., using flux correction for finite-volume methods as used by Berger and Colella [1] and conservative prolongation). In the present implementation of CESE on AMR grid, the global conservation cannot be fulfilled strictly because of two non-conservative operators. One is the linear prolongation and the other one is the guard cell filling at the interfaces of blocks with different space–time resolutions. Even so, our previous MHD results are satisfactory by using these simple operators [10]. We also note that in the work of Henshaw and Schwendeman [28] where the AMR algorithm is used combining with the overlapping grid method (which is another factor for non-conservation), good results are obtained without considering the global conservation. Actually in the present circumstances the strictly global conservation does not mean that importance as it seems for the following reasons.

It is true that conservativeness is critical for discontinuity-capture schemes to obtain the correct weak solution, since the Rankine-Hugoniot relations must be satisfied crossing the discontinuity. However, the discontinuity is local and the conservation satisfied locally near the discontinuity is adequate. Provided with a set of good refinement criteria, the AMR grid system can always enclose the discontinuities within bands of blocks at the same refinement level, usually the highest level, with enough buffers (see Fig. 4 for example). In this way, it seems that the blocks close to the discontinuities are driven to move in the same manners as moving of the discontinuities (actually it is not the block itself but the mesh resolution that moves), and the discontinuities themselves never pass or straddle the resolution interfaces throughout the computation. In other words, for a given discontinuity once formed, it only 'feels' the mesh of the same resolution without 'knowing' the existence of the dynamically, varying-resolution grid system. This avoids the two non-conservative operators in the bands of blocks that encompass the discontinuities, since child blocks are created or destroyed only at the edges of the buffer zone. Then the CESE solver can advance the solutions crossing the discontinuity just like in the uniform grid and thus the local

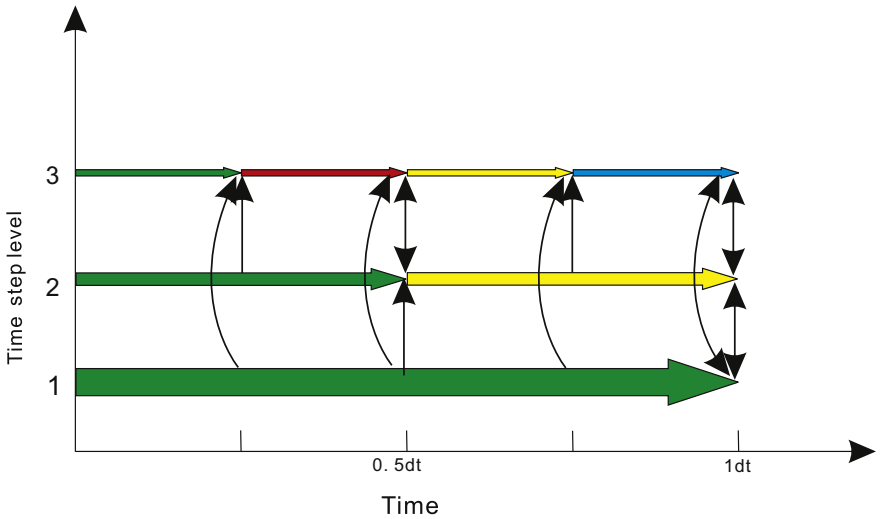

Fig. 3. The scheme of variable timestep: an example.

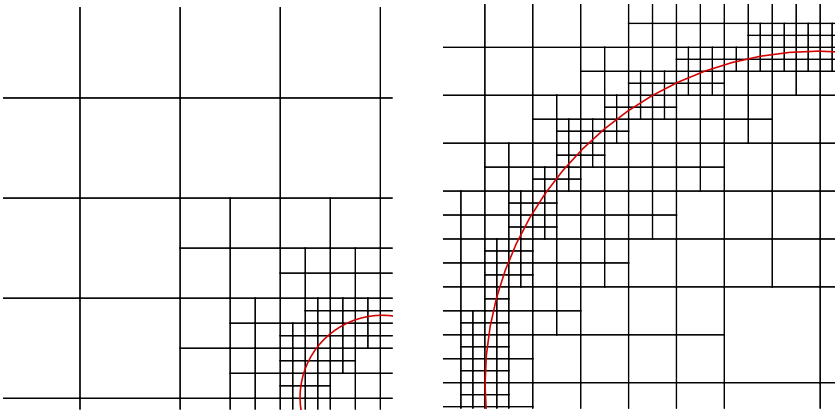

Fig. 4. An example for the discontinuity capture by the AMR grid. The curved line represents the discontinuity and each cell here represents one grid block that consists of 8×8 cells. This grid is extracted from the two-dimensional blast wave problem (Section 3.2): left shows the initial shock and right shows the shock at the end of the computation. Note that the grid resolution evolves in the same manners as the shock moves and well enclosing of the shock by blocks at the highest refinement level.

conservation is still preserved. In this respect, what is more important is the design of good refinement criteria that can direct the variation of the grid resolution effectively and pave the way in advance for the formation and movement of discontinuities.

2.4. Boundary conditions

At last for the sake of completeness, we address all the boundary conditions used for the following numerical tests (Section 3). There are four types including the inflow condition, the outflow condition, the slip and non-slip wall conditions. Since the computational domain consists of cells and the solutions are represented at the cell centers, the boundary conditions are specified by directly assigning values at the guard cells that abut the boundary faces. This is a great simplification of the boundary procedure from the original method where one should consider the space–time staggering nature of the points at the boundaries. Denote the solutions at these guard cells as $\left(\mathbf{U}^0, \mathbf{U}^0_x, \mathbf{U}^0_y\right)$ and solutions at their immediately neighboring cells inside the domain as $\left(\mathbf{U}^1, \mathbf{U}^1_x, \mathbf{U}^1_y\right)$. Using the lower y-boundary as an example, the boundary conditions are given as follows.

(1) For the the inflow condition, $\mathbf{U}^0$ is assigned by the given inflow values and the gradients are set to be $\mathbf{U}^0_x = \mathbf{U}^0_y = \mathbf{0}$.
(2) For the outflow condition, $\mathbf{U}^0 = \mathbf{U}^1, \mathbf{U}^0_x = \mathbf{U}^1_x$ and $\mathbf{U}^0_y = \mathbf{0}$.
(3) For the slip wall condition, also named as the reflecting boundary condition, the values are given first in the form of the primitive values as

$$\begin{aligned}
&(\rho, v_x, v_y, p)^0 = (\rho, v_x, -v_y, p)^1,\\
&\left(\frac{\partial\rho}{\partial x}, \frac{\partial v_x}{\partial x}, \frac{\partial v_y}{\partial x}, \frac{\partial p}{\partial x}\right)^0 = \left(\frac{\partial\rho}{\partial x}, \frac{\partial v_x}{\partial x}, -\frac{\partial v_y}{\partial x}, \frac{\partial p}{\partial x}\right)^1,\\
&\left(\frac{\partial\rho}{\partial y}, \frac{\partial v_x}{\partial y}, \frac{\partial v_y}{\partial y}, \frac{\partial p}{\partial y}\right)^0 = \left(-\frac{\partial\rho}{\partial y}, -\frac{\partial v_x}{\partial y}, \frac{\partial v_y}{\partial y}, -\frac{\partial p}{\partial y}\right)^1,
\end{aligned} \tag{12}$$

and then transformed to the solution variables.

(4) For the non-slip wall condition with the wall speed of $v_x = v_d$,

$$\begin{aligned}
&(\rho, v_x, v_y, p)^0 = (\rho, 2v_d - v_x, -v_y, p)^1,\\
&\left(\frac{\partial\rho}{\partial x}, \frac{\partial v_x}{\partial x}, \frac{\partial v_y}{\partial x}, \frac{\partial p}{\partial x}\right)^0 = \left(\frac{\partial\rho}{\partial x}, -\frac{\partial v_x}{\partial x}, -\frac{\partial v_y}{\partial x}, \frac{\partial p}{\partial x}\right)^1,\\
&\left(\frac{\partial\rho}{\partial y}, \frac{\partial v_x}{\partial y}, \frac{\partial v_y}{\partial y}, \frac{\partial p}{\partial y}\right)^0 = \left(-\frac{\partial\rho}{\partial y}, \frac{\partial v_x}{\partial y}, \frac{\partial v_y}{\partial y}, -\frac{\partial p}{\partial y}\right)^1.
\end{aligned} \tag{13}$$

For other faces, all these types of boundary conditions can be derived in the similar way.

3. Numerical results

To validate the AMR–CESE method and demonstrate its capabilities, a suite of test benchmarks are calculated. The problems are chosen with the first three for Euler equations and the rest for NS equations. The specific heat ratio $\gamma = 1.4$ and the Prandtl number $Pr = 0.72$ are given in the tests when needed. We use the full set of the refinement criteria in Eq. (9) for these problems if without special explanation.

3.1. Sod shock-tube problem

The Sod shock-tube problem [29] is commonly used for code validation. The initial values are given in a one-dimensional computational domain ($x \in [-1, 1]$) as

$$\mathbf{v} = \mathbf{0}, (\rho, p) = \begin{cases} (1, 1) & \text{if } x < 0;\\ (0.125, 0.1) & \text{if } x > 0, \end{cases} \tag{14}$$

Four levels of refinements are used with the grid size of the base level $\Delta x = 2/256$. Solution at $t = 0.5$ is plotted in Fig. 5 along with the analytic solution. The refinement levels are also displayed to show the distribution of the mesh points. The mesh points are well clustered at the contact discontinuity and the right shock which is resolved with only two points at the highest level of refinement. No numerical oscillation is detected although the simple linear interpolation is used to reconstruct the solution on newborn blocks.

To assess to what extent the global conservation constraint is broken, we plot in Fig. 6 the evolutions of the relative deviations of the global conservation, which is calculated by

$$\delta_U(t) = \frac{\int_D U(x,t)\mathrm{d}x - \int_D U(x,0)\mathrm{d}x + (F_R - F_L)t}{\int_D U(x,0)\mathrm{d}x}. \tag{15}$$

Here D is the total computational domain, U represents the conservative quantities and F_L and F_R denote the fluxes at the left and right boundaries. It shows that the errors increase as the grid evolved but within very low values ($\sim 10^{-3}$) that can hardly affect the discontinue capture.

3.2. Adiabatic blast wave

In this problem, we examine the method to capture a strong two-dimensional shock. That is a blast wave initialized by an over-pressured circular region in the center of a static flow [30]. Here the same parameters given in [30] are used. The computational

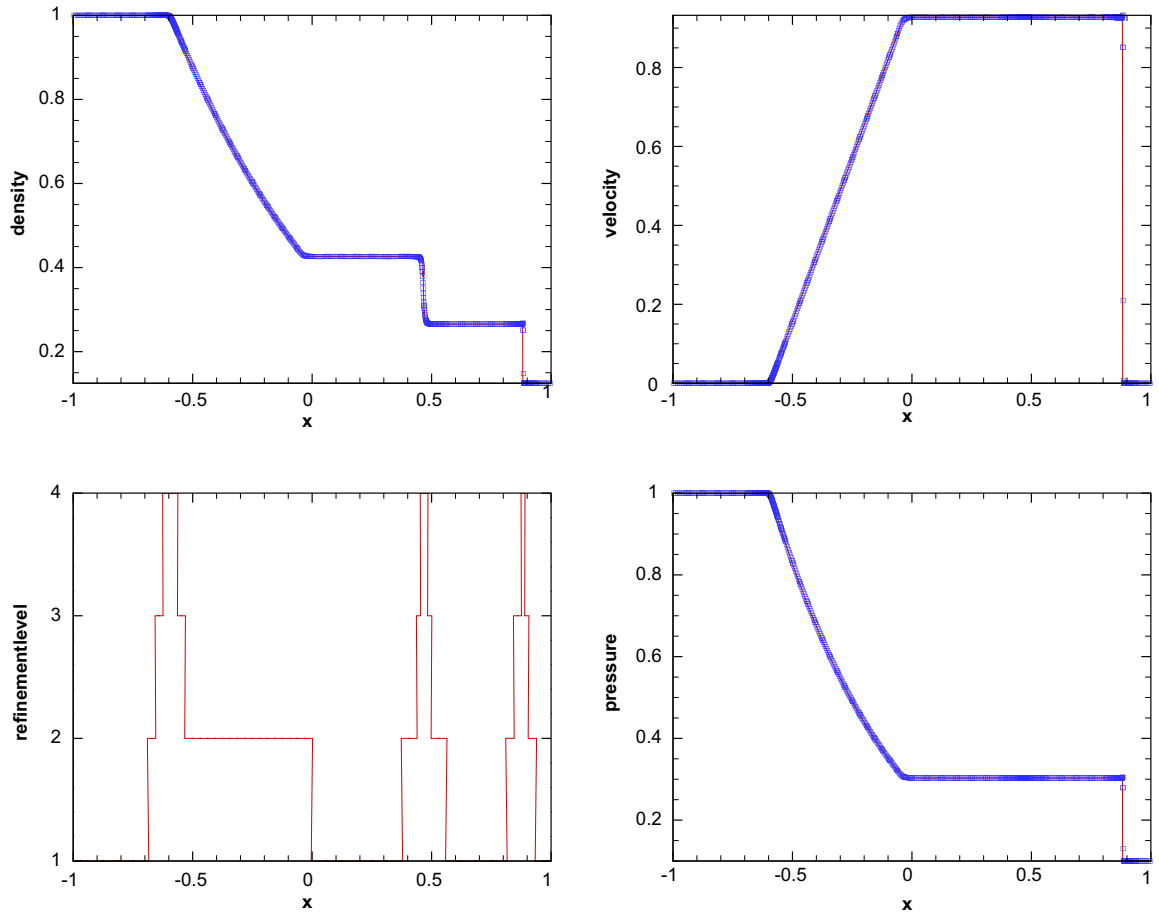

Fig. 5. Sod shock-tube problem: the profiles of density, velocity, pressure and refinement levels at $t = 0.5$. The solid line is the analytic solution.

domain is a unit square with the center at $(0,0)$. The initial condition is specified as

$$\rho = 1, \mathbf{v} = \mathbf{0}, p = \begin{cases} 10 & \text{if } \sqrt{x^2 + y^2} < 0.1; \\ 0.1 & \text{if } \sqrt{x^2 + y^2} \geqslant 0.1, \end{cases} \tag{16}$$

Four levels of refinements is allowed with coarsest mesh size of $\Delta x = \Delta y = 1/64$, and the initial grid is refined according to the pressure profile. Fig. 7 shows the density and the grid blocks at time $t = 0.2$. Good grid adaption and highly symmetry can be observed. In Fig. 8 the variables along two lines ($y = 0$ and $x = y$) through the

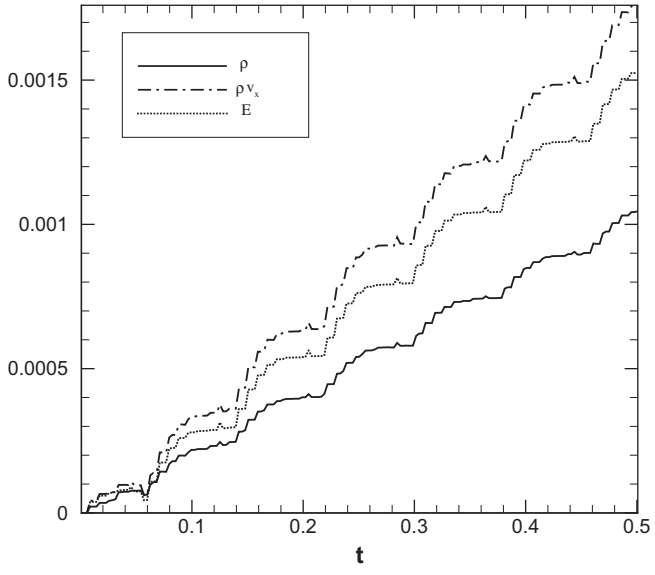

Fig. 6. Sod shock-tube problem: the evolutions of the non-conservative errors with time.

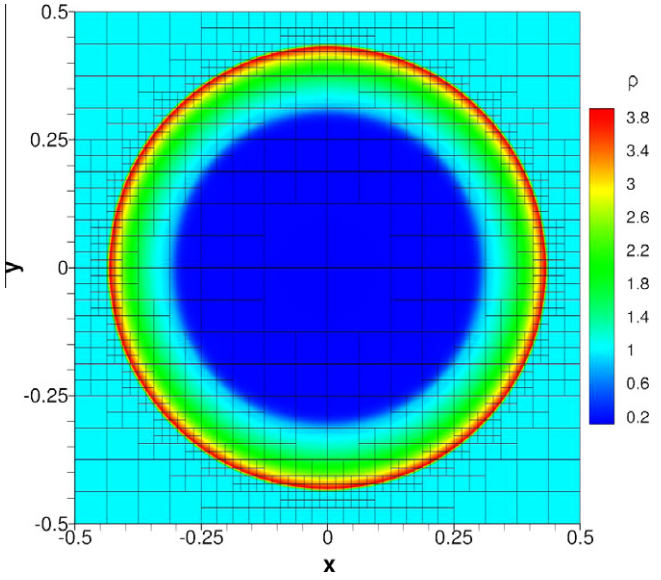

Fig. 7. Adiabatic blast wave: density at $t = 0.2$ and the corresponding grid blocks.

domain center are compared with the reference solution obtained by the original, well-established CESE method on an uniform 256^2 grid. The profiles are consistent with results in [30] and it further demonstrates the good symmetry and the correct capture of the discontinuities. Check of the global conservation also gives the relative errors within rather small values ($\sim 5 \times 10^{-3}$). All these confirm the AMR–CESE method for the following, more complicated problems.

3.3. Wind tunnel with a step

To further demonstrate the shock-capture capability of the combination of the CESE and AMR methods, the classic problem of a wind tunnel containing a forward facing step is calculated. This problem has been employed by many researchers for code validation and comparison since it was first proposed by Emery [31–33]. Due to the multiple shock reflections and interactions in the problem, it is a severe test to examine the code for capturing such complicated flow features.

The problem is initialized with an uniform supersonic flow with Mach 3 in a wind tunnel of $(x,y) \in [0,3] \times [0,1]$, as shown in Fig. 9. Between $x = 0.6$ and $x = 3$ along the x-axis is a step with height of 0.2. Inflow condition is given at the left boundary ($x = 0$) with values of

$$(\rho, \mathbf{v}, p) = (1.4, 3, 0, 1) \tag{17}$$

which is also served as the initial condition. Outflow condition is used for the right-hand x-boundary. All the other faces are treated

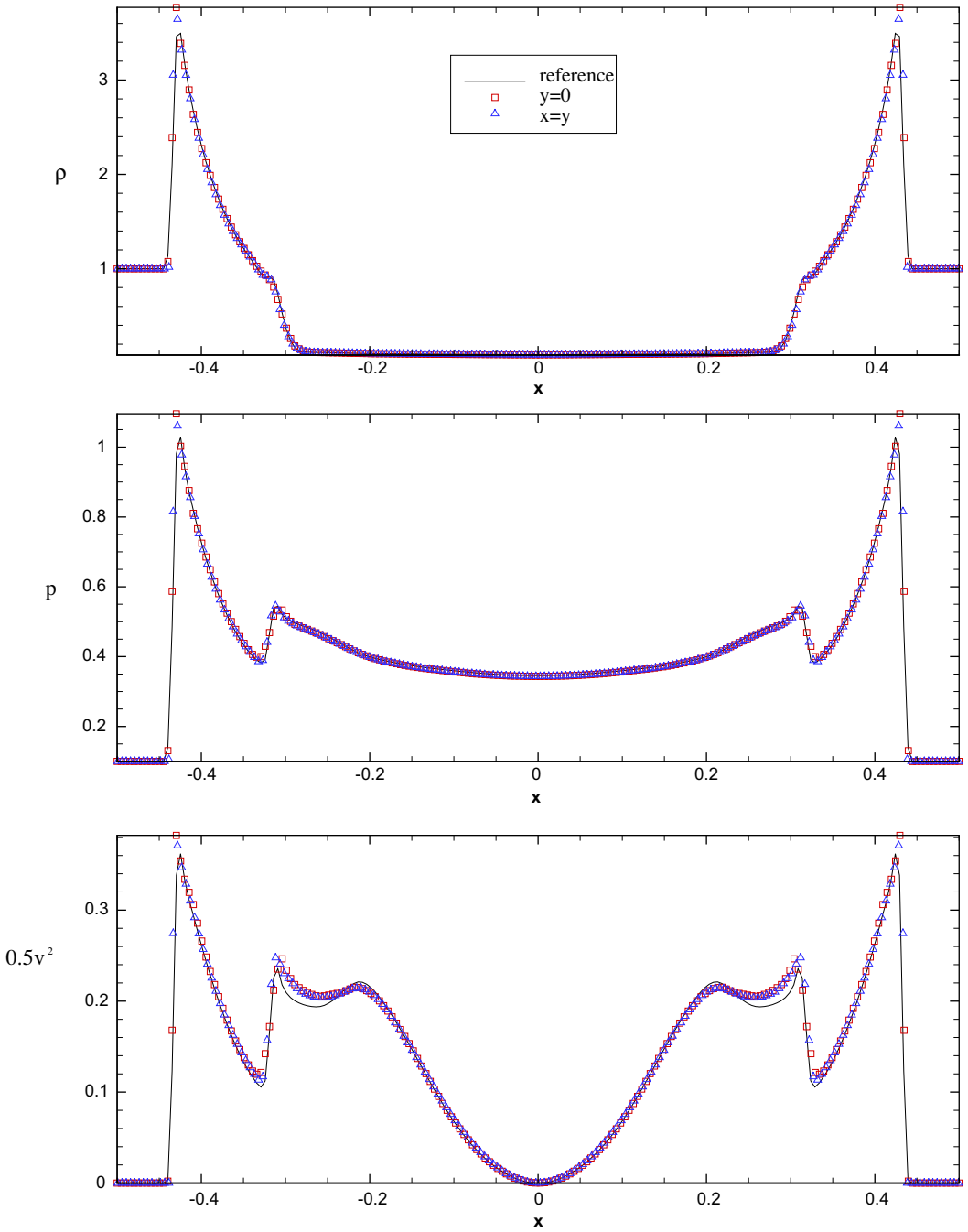

Fig. 8. Adiabatic blast wave: profiles of density ρ, pressure p and specific kinetic energy $0.5v^2$ along two lines through the blast center. The reference values is computed by the original CESE method on an uniform 256^2 grid.

with the reflecting boundary condition simply with no special treatment at the step corner. The computational domain is uniformly divided into blocks initially and four refinement levels are used.

With evolving of the solution, a curved shock forms in front of the step immediately, and as the shock moves, it reaches the top wall and is reflected. The reflection point moves left and then a Mach stem forms when the incident shock angle exceeds the maximum value for regular reflection (40° for $\gamma = 1.4$). Meanwhile, the reflected shock goes downward and reaches the top of the step, where it is reflected back again. Then this secondly reflected shock propagates upward and is reflected by the top wall to form a third shock. Besides, the over-expansion flow at the step corner impinges on the step top and results in an individual weak shock which successively intersects with the first and second reflected shocks. After then, the flow evolves slowly and converges to a steady state.

The evolution of density from $t = 0$ to $t = 4$, when the most interesting flow develops [32], is shown in Fig. 9 with the time interval of 1. As can be seen, the main structures of our results (i.e., the Mach stem, expansion fan at the step corner and the interaction between the reflected shocks with rarefaction waves) are highly consistent with those in Woodward and Colella [32] and are much more crisply reproduced by the adaptive grid system shown in the right column of Fig. 9. For many schemes the individual shock originated from the step corner is hard to be captured because of its weakness and it is likely to be dissipated by the diffusion effect before interaction with the reflected shock. Our method does not suffer from this numerical effect and the profile of such shock is distinct. In this sequence of side-by-side figures, one can vividly observe the evolution of the grid resolution responding to the moving of the shocks, which shows the capability of the present refinement criteria.

Note that the artificial Mach stem at the top of the step, which usually arises in low-resolution runs without a special treatment of the step corner, almost disappears in our results. Another numerical effect, the Kelvin–Helmholtz instability of the contact discontinuity along the top wall as described in Woodward and Colella [32], is also well controlled in our method. The same problem is calculated by another AMR code Flash [34], and results published online show that such instability turns out to be more visible on the high-resolution grid (http://www.flash.uchicago.edu/). On the contrary, a detailed comparison can show that in our results the instability is a little weaker than that of Woodward and Colella [32], even though our resolution is higher.

A detailed comparison of the AMR–CESE results with a reference solution is given in Fig. 10 by plotting the distribution of the density along the horizontal centerline ($y = 0.5$). Here the reference solution is calculated by the original CESE method on an uniform grid at the highest resolution, i.e., an uniform grid with 960×320 points excluding the step. As can be seen from the figures, both the locations and the jumpings of all the discontinuities are highly in

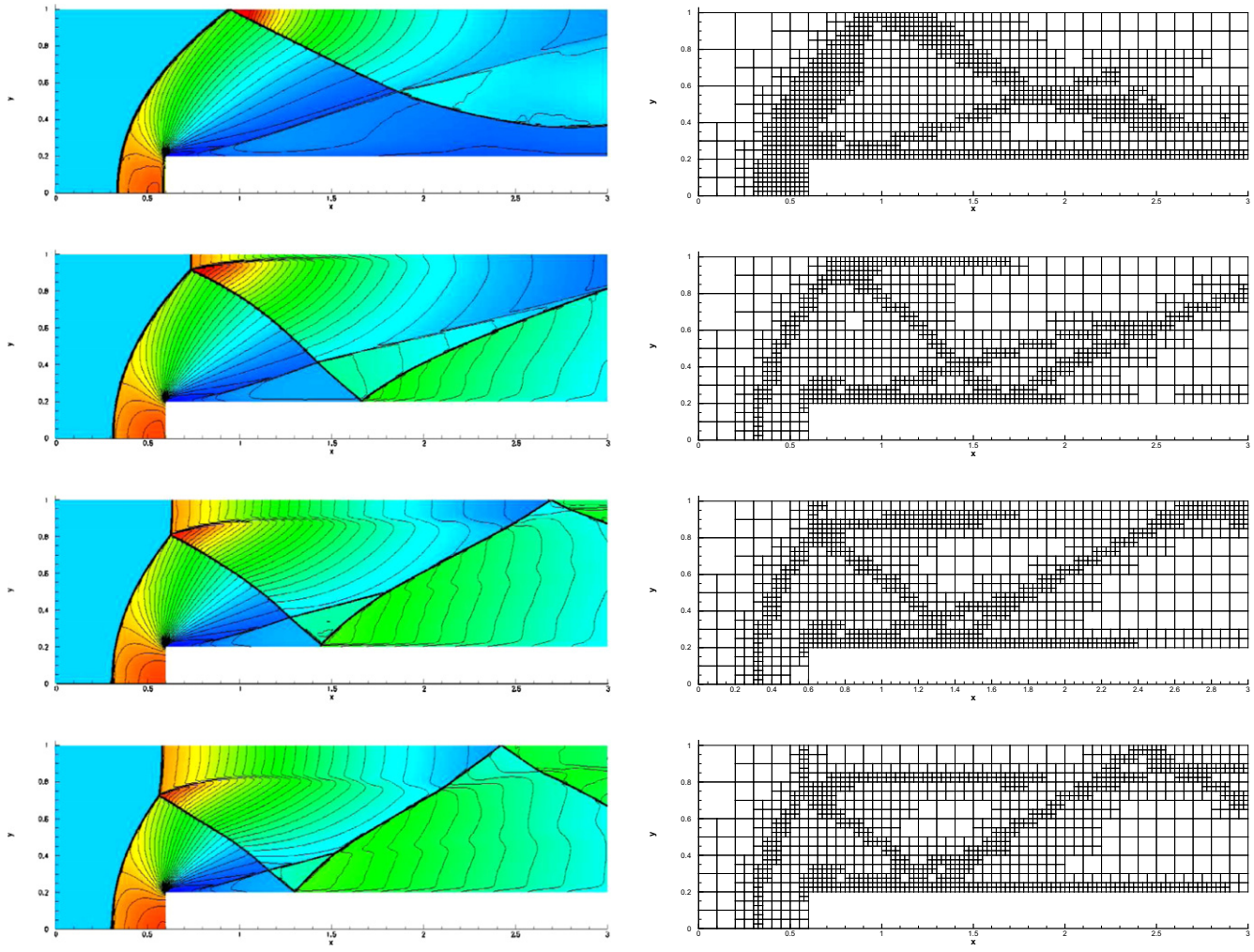

Fig. 9. Wind tunnel with a step: density at $t = 1$ to $t = 4$ and the grid blocks at the corresponding time. 30 contour lines are shown from 0.3 to 6.5.

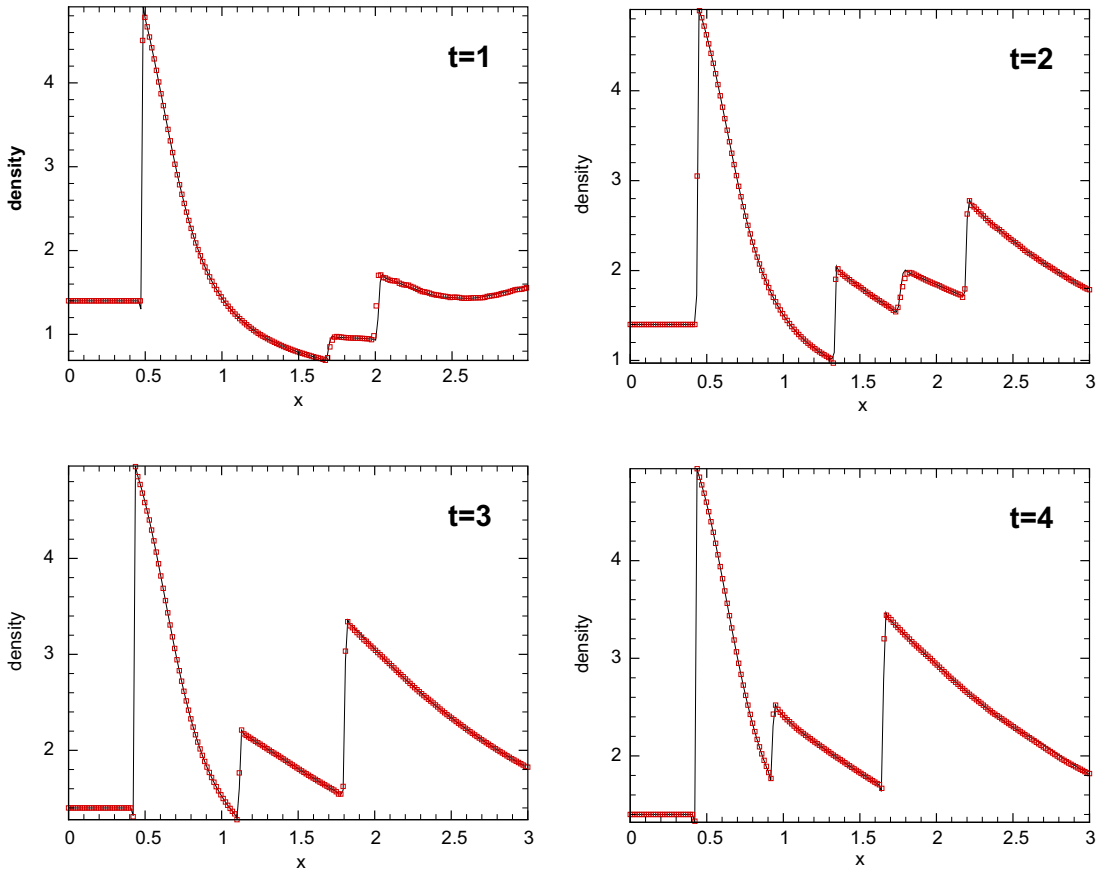

Fig. 10. Wind tunnel with a step: profiles of density along the centerline $y = 0.5$ at $t = 1$ to $t = 4$. The AMR–CESE solution is denoted by the squares, which are interpolated from the AMR grid to uniformly 200 points at the line. The solid line is the reference solution computed by the original CESE method with an uniform grid at the highest resolution of the AMR grid.

agreement with the reference solution at each time in the evolution. With the same condition of computer resource (i.e., same number of processors on the same platform), the AMR test costs about one third of CPU time used by the uniform fixed-mesh test. The computation of this problem represents a typical success of the AMR–CESE method by achieving the same high resolution at relatively very low cost.

3.4. Laminar flow over flat plate

The first problem for NS equations is a laminar flow over a flat plate. This classical problem with the Blasius solution [35] represents the simplest of flow problems yet brings out some fundamental aspects of viscous flow and the boundary layer theory. It provides a basic test since accurate prediction of the boundary layer is critical to development of a viscous flow solver.

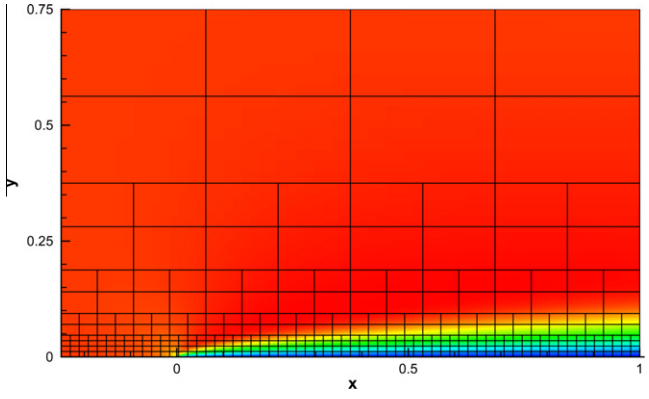

Fig. 11. Laminar flow over flat plate: the grid blocks and color-coded image of the velocity v_x. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

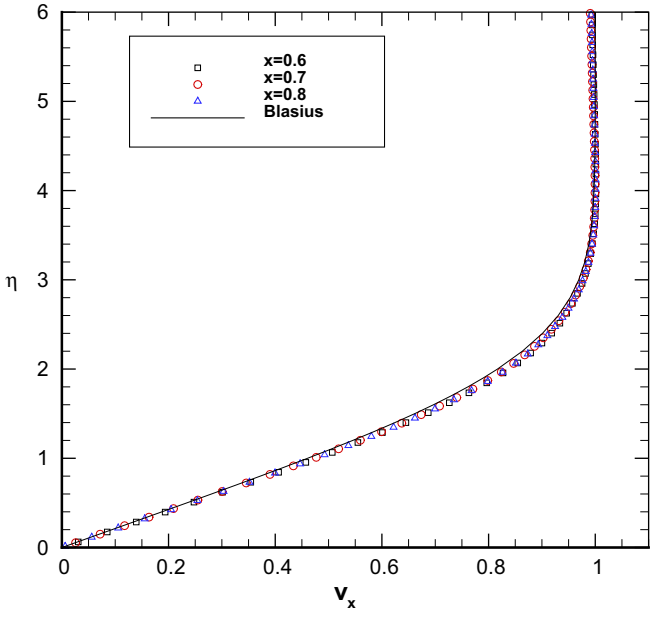

Fig. 12. Laminar flow over flat plate: similarity variable η vs. velocity v_x. Three different values 0.6, 0.7 and 0.8 for x are plotted.

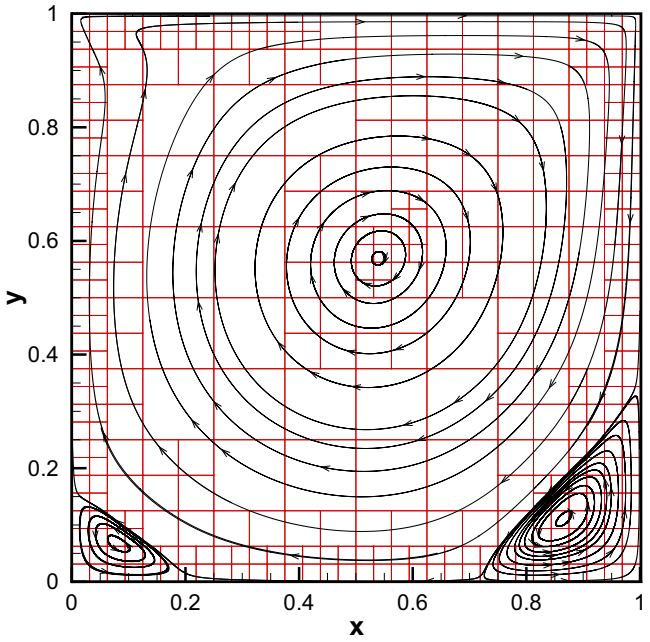

Fig. 13. Driven cavity flow problem: streamline and the grid blocks (each cell here represents one grid block).

As shown in Fig. 11, the computational domain is $(x,y) \in [-0.25,1] \times [0,1]$, where the plate is represented by a non-slip wall at the bottom boundary with $0 < x \leqslant 1$. The subsonic inflow is specified at the left boundary ($x = -0.25$) and a fixed pressure outflow is given at the right boundary ($x = 1$). On the bottom boundary ahead of the plate, a slip wall condition is applied to capture the effect of leading edge flow. At the upper boundary non-reflective boundary condition is used to model the farfield flow. The free-stream Mach number is $M = 0.1$ and the Reynolds number is chosen $Re = 1000$. The inflow and the initial conditions are

$$(\rho, \mathbf{v}, p) = \left(1, 1, 0, \frac{1}{\gamma M^2}\right). \tag{18}$$

Usually a stretched (compressed) grid is adopted to resolve the boundary layer [36], which needs special consideration of the grid stretching function. With the aid of AMR grid technique, the task of grid generation becomes very simple. The computational grid is produced by first uniformly dividing the domain into 4×4 blocks and then refining the blocks touching the plate using four more refinement levels (Fig. 11). In such way, the distance from the plate to the first grid point is $1/2^9$ of the domain height, which is sufficient for resolving the boundary layer. Since the feature of flow above the boundary layer does not change significantly in the run, the grid is kept static with the main objective to resolve the boundary layer.

Fig. 11 also shows the converged velocity v_x. In Fig. 12 the result is compared with the Blasius solution [35] by plotting the variation of velocity v_x with the similarity variable η given by

$$\eta = y\sqrt{\frac{\rho Re}{2x\mu}}. \tag{19}$$

It shows that our result are in good consistency with the Blasius solution.

3.5. Two-dimensional lid-driven cavity

Because of the existence of several complex flow phenomena (corner eddies, longitudinal vortices, primary vortices, and interactions between vortices) in it, the driven cavity flow has been taken as a classical benchmark for evaluation of the numerical schemes, especially the incompressible viscous solvers [37,38,25,39,40]. Here it is calculated to examine the method at the incompressible limit and the capability of modeling the viscosity effect, since in this problem the viscosity plays a main role and the flow is driven by the friction force.

As usual, the computational domain is $(x,y) \in [0,1] \times [0,1]$ and the initial conditions are given as

$$\rho = 1, p = \frac{1}{\gamma M^2}, \mathbf{v} = \mathbf{0}. \tag{20}$$

The Mach number and the Reynolds number is set as $M = 0.1$ and $Re = 1000$. The top boundary ($y = 1$) is a moving lid with speed of $v_x = 1$, and other three boundaries are stationary walls. All the boundary conditions are given by the non-slip wall condition. Initially, the domain is uniformly divided into 8×8 blocks. When running three levels of refinements are allowed such that the mesh size of the middle level is $\Delta x = \Delta y = 1/128$. The velocity divergence $\nabla \cdot \mathbf{v}$ is excluded from the refinement criteria because of the incompressible condition, i.e., $\nabla \cdot \mathbf{v} = 0$.

The solution reaches the steady state at $t \sim 30$. Fig. 13 shows the streamline and the final grid. The complexity of the flow in the cavity are well represented by the grid system with the finest blocks clustering near the three vortexes. For a detailed comparison with

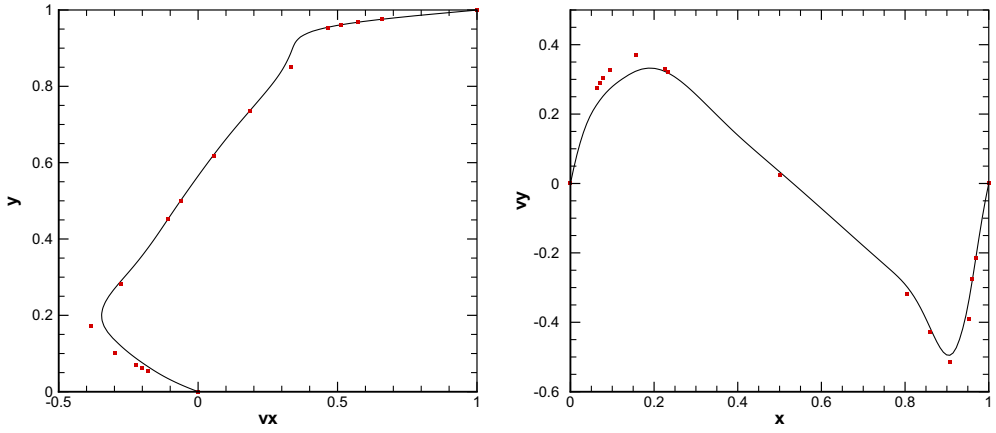

Fig. 14. Driven cavity flow problem: distributions of velocity component v_x along the vertical centerline (left) and v_y along the horizontal centerline (right). The lines are our result and the squares denote Ghia's result.

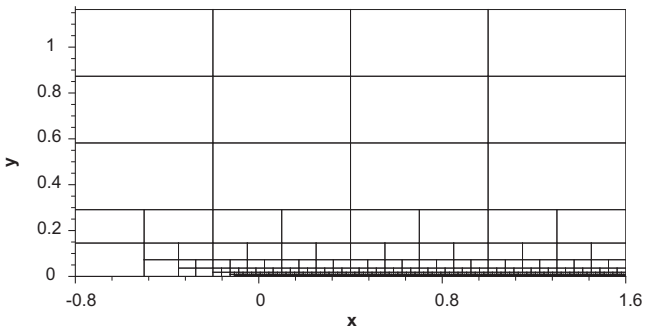

Fig. 15. Shock/boundary layer interaction problem: the initial grid.

the result reported in Ghia et al. [37] with an uniform 128×128 grid, we plot the velocity distributions along the vertical and horizontal centerlines in Fig. 14. As can be seen, there is a slight deviation between the two results for $0.1 \leqslant x \leqslant 0.2$ and $0.1 \leqslant y \leqslant 0.2$. This is mainly because the coarsest blocks are in such regions, where the grid resolution is half of that in Ghia et al. [37]. Even so, the solution of the rest region match very well with Ghia's and the main features of the flow (i.e., the locations of the vortexes) are well reproduced.

3.6. Shock wave boundary layer interaction

The shock/boundary layer interaction problem in [41] is often used as a benchmark for solvers that designed for the full Navier–Stokes equations [38,25], and is a particularly good test for our purpose due to the coexistence of shocks and boundary layer. Following Zhang et al. [25], the computational domain is $(x,y) \in [0,2.4] \times [0,1.164]$ (Fig. 15). The left and the upper boundary conditions are specified to introduce an oblique shock impinging on the solid wall on the bottom boundary ($x \geqslant 0$). Boundary layer separation will occur at the shock impinging if the shock is strong and the incident shock angle is large enough. Reproducing such flow features requires clustered cells near the solid wall with grid size $\Delta y \leqslant 1 \times 10^{-3}$ and the advantages of AMR can be seen to resolve the boundary layer and capture the shocks simultaneously.

In the present test, the shock incident angle is set to be 32.6° and the inflow from the left and upper boundaries are given as

$$(\rho_1, \mathbf{v}_1, p_1) = (1.00000, 1.00000, 0.00000, 0.17857) \tag{21}$$

and

$$(\rho_2, \mathbf{v}_2, p_2) = (1.13074, 0.96644, -0.05248, 0.21213) \tag{22}$$

respectively, for the desired shock angle. The free-stream (the left inflow) Mach number is $M = 2.0$ and the Reynolds number

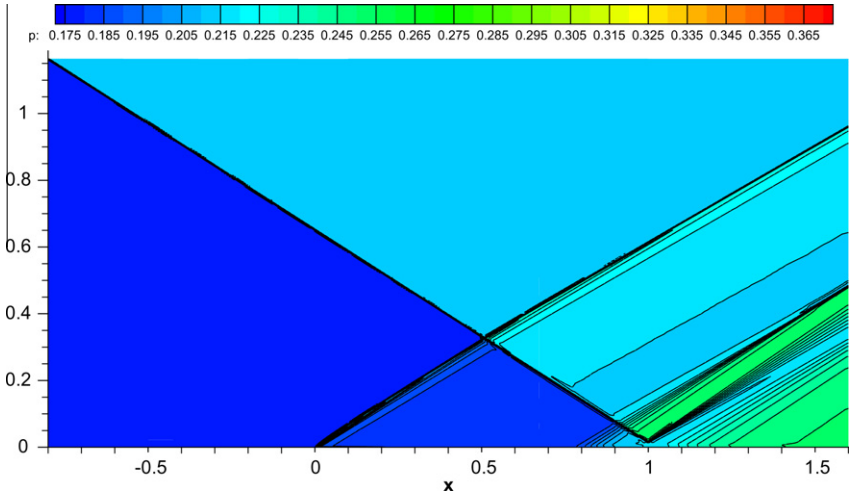

Fig. 16. Shock/boundary layer interaction problem: pressure contours.

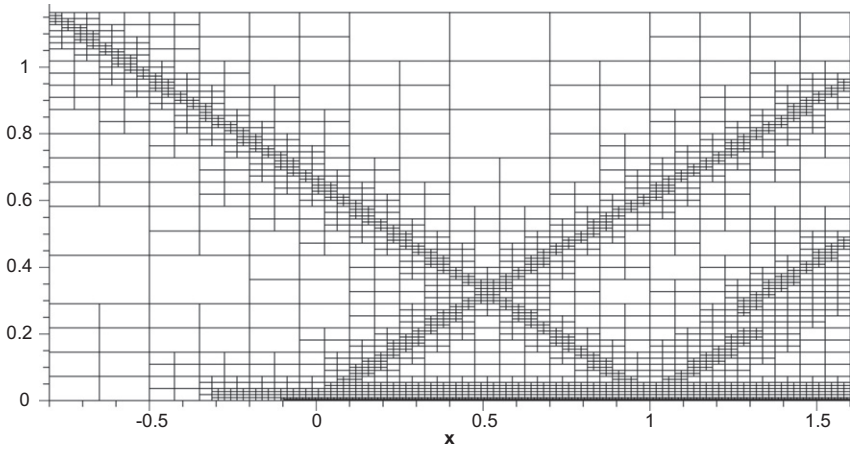

Fig. 17. Shock/boundary layer interaction problem: the converged grid.

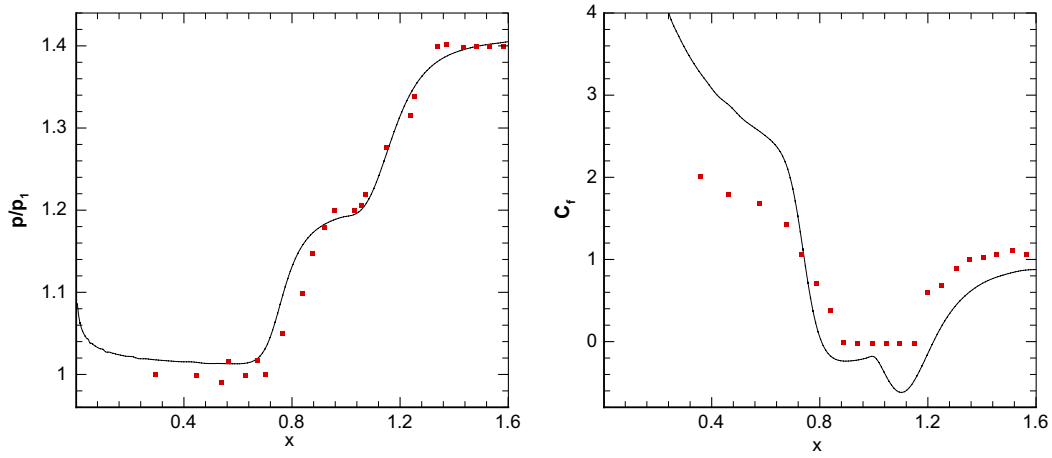

Fig. 18. Shock/boundary layer interaction problem: pressure (left) and skin friction (right) distribution along the solid wall. The line is the present result and the squares denote the experiment data.

$Re = 2.95 \times 10^5$. On the bottom boundary, similar to the laminar flow over a flat plate problem (Section 3.4), slip wall condition is prescribed for $x < 0$ and non-slip wall condition for $x \geqslant 0$. Finally non-reflective boundary condition is used on the right boundary as a supersonic outlet. The initial conditions are given by the same values of the left inflow. The initial grid is generated in the similar manner as in Section 3.4. It consists of 8×8 blocks of the basic refinement level and the blocks touching the solid wall are refined with six more levels (see the bottom of the Fig. 15). Thus the distance from first grid point to the wall is nearly 2.4×10^{-4} of the domain height. Four levels of refinements are used for the grid system to capture the shocks dynamically. Specially, blocks abutting the bottom wall are restricted from being coarsened to keep high-resolution there.

Fig. 16 shows the pressure contours. The boundary layer separation can be visibly observed at the shock impinging point ($x = 1$). The incident and reflective shocks are crisply captured and well characterized by the converged grid structure shown in Fig. 17. As there are seven levels of blocks used in the test, the mesh size varies $2^6(64)$ times across the computational domain and thus the difference in local timesteps is 64 times. It is such case that brings to light the effects of the variable-timestep algorithm, since it can reduce the numerical diffusion effectively for shock capture and save the computational resource significantly compared to the uniform-timestep scheme.

The pressure and the friction coefficient along the solid wall are plotted and compared with the corresponding experimental data [41] in the Fig. 18, respectively, which show that our results are in agreement with the experimental data. The present results, except the friction distribution, are also comparable with those reported in Zhang et al. [38,25], where the similar CESE method is used, but on an unstructured static grid. We note that a special treatment of the non-slip wall is used by Zhang et al. [38,25], which can characterize the shear stress by the wall more accurately. We expect to improve our result in the future using this special boundary treatment.

4. Conclusions

In this paper, we report an extension of the AMR–CESE method for Euler and Navier–Stokes equations. The method is an implementation of the CESE method on AMR grid with the help of PARAMESH. For the compatibility of integrating the CESE solver within the PARAMESH framework, improvements are introduced. They includes a variation of the the original CESE grid, merging of the intermediate advancing steps and the variable timestep algorithm between grid blocks. The AMR–CESE method retains all the salient advantages of the original CESE method, and owns new merits including (1) simplified definitions of the CE/SE, (2) the speedup of the CESE solver and (3) solution-adaptivity in both space and time which is consistent with the key principle of the CESE method. Even on an extremely non-uniform mesh, the method can control the numerical dissipation effectively without the CNIS scheme, due to the adoption of the variable timestep algorithm. A suite of numerical benchmarks including one and two-dimensional problems for the Euler and Navier–Stokes equations are calculated. Comparison with the experimental or previously reported numerical results demonstrates the validation of our method. The method shows capabilities of both capturing discontinuity with high resolution in the supersonic flows and high-performance resolving the boundary layer effect in the viscous flows.

Acknowledgments

The work is jointly supported by the National Natural Science Foundation of China (41031066, 40921063, 40874091, 40890162, 40904050, 40874077, and 40536029), the 973 Project under Grant 2006CB806304, and the Specialized Research Fund for State Key Laboratories. The numerical calculation has been completed on our SIGMA Cluster computing system. The PARAMESH software used in this work was developed at the NASA Goddard Space Flight Center and Drexel University under NASA's HPCC and ESTO/CT Projects and under Grant NNG04GP79G from the NASA/AISR project. Special thanks go to our anonymous reviewers for helpful suggestions for the improvement of the paper.

References

[1] Berger MJ, Colella P. Local adaptive mesh refinement for shock hydrodynamics. J Comput Phys 1989;82:64–84.
[2] Rendleman C, Beckner V, Lijewski M, Crutchfield W, Bell J. Parallelization of structured hierarchical adaptive mesh refinement algorithms. Comput Visual Sci 2000;3:147–57.
[3] Colella et al. P. Chombo, software package for AMR applications. Technical report, Lawrence Berkely National Laboratory; 2007.
[4] Parashar M, Browne J. Distributed adaptive grid hierarchy. Technical report, Rutgers University; 2007.
[5] Parashar M. GrACE grid adaptive computational engine. Technical report, Rutgers University; 2007.
[6] Garaizar X, Hornung R, Kohn S. Structured adaptive mesh refinement applications infracture. Technical report, Lawrence Livermore National Laboratory; 1999.

[7] MacNeice P, Olson KM, Mobarry C, de Fainchtein R, Packer C. PARAMESH: a parallel adaptive mesh refinement community toolkit. Comput Phys Commun 2000;126:330–54.
[8] Olson K, MacNeice P. An over of the paramesh amr software and some of its applications. In: Proceedings of the Chicago workshop on adaptive mesh refinement methods – theory and applications. Berlin: Springer; 2005.
[9] Olson K. Paramesh: a parallel adaptive grid tool. In: Parallel computational fluid dynamics. Elsevier; 2006.
[10] Jiang CW, Feng XS, Zhang J, Zhong DK. Amr simulations of magnetohydrodynamic problems by the cese method in curvilinear coordinates. Solar Phys 2010;267:463–91.
[11] Chang S. The method of space-time conservation element and solution element: a new approach for solving the Navier–Stokes and Euler equations. J Comput Phys 1995;119:295–324.
[12] Chang S, Wang X, Chow C. The space-time conservation element and solution element method: a new high-resolution and genuinely multidimensional paradigm for solving conservation laws. J Comput Phys 1999;156:89–136.
[13] Zhang Z, Yu STJ, Chang S. A space-time conservation element and solution element method for solving the two- and three-dimensional unsteady Euler equations using quadrilateral and hexahedral meshes. J Comput Phys 2002;175:168–99.
[14] Feng X, Hu Y, Wei F. Modeling the resistive MHD by the cese method. Solar Phys 2006;235:235–57.
[15] Feng X, Zhou Y, Wu ST. A novel numerical implementation for solar wind modeling by the modified conservation element/soluti on element method. Astrophys J 2007;655:1110–26.
[16] Zhang M, John Yu S, Henry Lin S, Chang S, Blankson I. Solving the MHD equations by the space time conservation element and solution element method. J Comput Phys 2006;214:599–617.
[17] Jiang C, Feng X, Fan Y, Xiang C. Reconstruction of the coronal magnetic field using the CESE-MHD method. Astrophys J 2011;727. 101+.
[18] Wang J, Liu K, Zhang D. An improved CE/SE scheme for multi-material elastic–plastic flows and its applications. Comput Fluids 2009;38:544–51.
[19] Wang G, Zhang D, Liu K, Wang J. An improved CE/SE scheme for numerical simulation of gaseous and two-phase detonations. Comput Fluids 2010;39:168–77.
[20] Powell KG, Roe PL, Quirk J. Adaptive-mesh algorithms for computational fluid dynamics, in: Algorithmic trends in computational fluid dynamics. The Institute for Computer Applications in Science and Engineering (ICASE)/LaRC Workshop, NASA Langley Research Center, Hampton, VA, US, September 15–17; 1991. p. 303–337 [A95-91915].
[21] Dezeeuw D, Powell KG. An adaptively refined Cartesian mesh solver for the Euler equations. J Comput Phys 1993;104:56–68.
[22] Powell KG, Roe PL, Linde TJ, Gombosi TI, de Zeeuw DL. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. J Comput Phys 1999;154:284–309.
[23] Groth CPT, Northrup SA. Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh 2005.
[24] Chang SC. Courant number insensitive CE/SE schemes. AIAA Paper 2002-3890; 2002.
[25] Zhang MJ, Yu STJ, Chang SC. Solving the Navier–Stokes equations by the CESE method. AIAA Paper 2004;75; 2004.
[26] Chang S-C, Wu Y, Yang V, Wang X-Y. Local time-stepping procedures for the space-time conservation element and solution element method. Int J Comput Fluid Dynam 2005;19:359–80.
[27] Yen JC. Demonstration of a multi-dimensional time-accuate local time stepping CESE method. In: AIAA aeroacoustics conference; 2011. p. 5–8.
[28] Henshaw WD, Schwendeman DW. Parallel computation of three-dimensional flows using overlapping grids with adaptive mesh refinement. J Comput Phys 2008;227:7469–502.
[29] Sod GA. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J Comput Phys 1978;27:1–31.
[30] Skinner MA, Ostriker EC. The athena astrophysical magnetohydrodynamics code in cylindrical geometry. Astrophys J Suppl Ser 2010;188:290.
[31] Emery AE. J Comput Phys 1968;2:306.
[32] Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. J Comput Phys 1984;54:115–73.
[33] Giannakouros J, Karniadakis GE. A spectral element-FCT method for the compressible Euler equations. J Comput Phys 1994;115:65–85.
[34] Linde T. MHD simulations with the FLASH code. APS Meeting Abstracts; 2002. F3005+.
[35] White F. Fluid mechanics. 4th ed. New York: McGraw-Hill; 1999.
[36] Anderson JD. Computational fluid dynamic: the basics with applications, vol. 168. New York: McGraw-Hill; 1995.
[37] Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. J Comput Phys 1982;48:387–411.
[38] Zhang ZC, Yu STJ, Wang XY, Chang SC, Himansu A, Jorgenson PCE. The CE/SE method for Navier–Stokes equations using unstructured meshes for flows at all speeds. AIAA Paper; 2000. p. 393.
[39] Guo Y, Hsu A, Wu J, Yang Z, Oyediran A. Extension of CE/SE method to 2D viscous flows. Comput Fluids 2004;33:1349–61.
[40] Venkatachari BS. Development and validation of a transient viscous flow solver based on a space-time CE/SE framework. Master's thesis, The University of Alabama; 2005.
[41] Hakkinen RJ, Greber I, Trilling L, Abarbanel SS. The interaction of an oblique shock wave with a laminar boundary layer. NASA Memo 2-18-59W; 1959.