Comparison between Two HLL-type Riemann Solvers for Strong Shock Calculation on ALE Framework

Zhijun Shen, Wei Yan

***Abstract*—This paper investigates solution behaviors under the strong shock interaction for moving mesh schemes based on the one-dimensional HLL-type Riemann solvers. Numerical experiments show that some schemes which updates the flow parameters directly on the moving mesh without using interpolation, may suffer from severe instability such as grid distortion. But the HLL solver can be free from such failings. It is well known that the numerical schemes on a fixed grid that can capture contact discontinuity accurately usually suffer from some disastrous carbuncle phenomenon, while others, such as the HLL scheme, are free from this kind of shock instability. Due to such high similarity, a research to combine the grid deformation and the numerical shock instability is developed. Furthermore, in order to unveil the cause of producing these failings, a HLL-type solver (denoted as the HLLS) with shear wave is constructed for inviscid, compressible gas flows and implemented into the arbitrary Lagrangian Eulerian (ALE) method. With the vorticity viscosity vanishing, the ALE HLLS scheme produce some numerical instability phenomena. This result indicates that viscosity associated to the shear wave may be unique reason to cause the instability or nonphysical deformation.**

***Index Terms*—Riemann solver, HLL, ALE, instability.**

I. Introduction

In many computational fluid dynamics (CFD) applications, boundaries of the physical domain of the flow might move in time. When moving boundaries experience large displacements, or when they undergo large deformations, the Lagrangian or the arbitrary Lagrangian Eulerian method (ALE) are useful tools to solve the flow problems on a moving and possibly deforming grid.

However, some Lagrangian and ALE methods may lead to collapse of a calculation under the interaction of the strong shock [1] [5] [7] [10]. In [10], we found that the nonphysical solution behavior depends on the shock strength and the grid aspect ratio, which is very similar to numerical shock instability (carbuncle) phenomenon in Eulerian methods [9].

The numerical shock instability was first reported in 1988 by Peery and Imlay [8] as they computed the supersonic flow field around a blunt body using Roe's scheme. Since the instability mechanism may ruin all efforts to compute grid-aligned shock waves using low-dissipative upwind schemes, from then on, several attempts have been made to understand and cure the phenomenon.

Manuscript received December 29, 2014. This project was supported by the National Natural Science Foundation of China (11471048), the Foundation of CAEP (2014A0202010), the Foundation of National Key Laboratory of Science and Technology Computation Physics.

Z. J. Shen and W. Yan are with the National Key Laboratory of Science and Technology on Computational Physics, Institute of Applied Physics and Computational Mathematics, P. O. Box 8009-26, Beijing 100088, China. The emails are shen_zhijun@iapcm.ac.cn, wyanmath01@sina.com.

Quirk [9] noticed that some schemes possessing the property of the good capturing of contact discontinuity show carbuncle phenomena while others free from carbuncle phenomena cannot capture contact discontinuity accurately, such as the HLL scheme [3]. Liou [4] observed that all the tested numerical functions that suffer from the shock instability have a term depending on pressure difference in the numerical mass flux while those free from the shock instability are independent of pressure difference in the numerical mass flux. Based on the numerical analysis and experiment, he considers that "the root of the multidimensional shock instability, which is manifested by the odd-even decoupling and carbuncle phenomena, is the existence of a pressure term in the mass flux". Xu and Li's analysis in [14] is in a qualitative agreement with Liou's conjecture in the sense that the pressure term in the numerical mass flux triggers the shock instability. One work of ours in [11] regarded that only the eigenvalue associated to the vorticity mode which is responsible for the instability. The conclusion is consistent with [6] but includes more elaborate analysis. However, since the stability analysis is linear, we can not conform whether it truly reflects nonlinear solution behavior.

This paper focuses on numerical method of moving grid and its stability. We construct a HLL-type Riemann solver by just restoring shear wave of the classical HLL approximate solver and denote it as HLLS. In order to restore the shear wave, we have to calculate the flow velocity in the star region of the HLL solver and split the HLL solver as a convective term and a work term. Such splitting formulation is rather interesting and to our knowledge is given here for the first time. The mass flux of the HLLS scheme is the same as the HLL one, therefore they do not depend on the pressure term for any Mach number. But the numerical performance on the shock instability is different from the HLL scheme. By comparing the numerical solutions using the HLLS solver and HLL solver, we draw some conclusions as follows: 1. the HLL approximate Riemann solver does not induce any instability whereas the HLLS triggers an instability. 2. Lious conjecture in [4] may be not correct, since the HLLS solver provides a counterexample. 2. The consequent in [11] should also apply to nonlinear situation. That means the dissipation diminishing on shear wave might be the unique reason to cause instability.

The outline of this paper is as follows. In section 2 the governing equations in the context of an ALE formulation and their space discretization are described. The one-dimensional HLL and HLLC approximate Riemann solvers are presented in section 3. The corresponding numerical results on both fixed and moving meshes are shown in section 4 to provide a

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

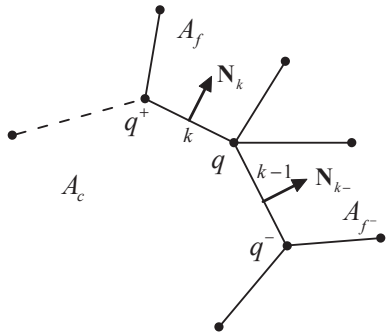

Fig. 1. Notations on a generic polygonal grid.

clear evidence of the robustness of this new scheme. Finally, the conclusions are summarized in section 5.

II. Governing equations and space discretization

The governing equations for inviscid flow in two-dimensions are as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = 0, \tag{1}$$

where the state vector and flux vectors are

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \qquad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho Eu + pu \end{bmatrix},$$

$$\mathbf{G}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho Ev + pv \end{bmatrix},$$

where ρ, p, E are the fluid density, pressure and total energy respectively, and $\mathbf{u} = (u, v)$ is the fluid velocity. The equation of state is in the form

$$p = (\gamma - 1)\rho e = (\gamma - 1)\rho(E - \frac{1}{2}(u^2 + v^2)),$$

where γ is the specific heat ratio.

It is convenient, for the subsequent discretization, to recast the system of equations in the following moving control volume formulation [2]:

$$\frac{d}{dt}\int_\Omega \mathbf{U} dxdy + \int_{\partial\Omega} [(\mathbf{F}, \mathbf{G}) \cdot \mathbf{N} - (\mathbf{w} \cdot \mathbf{N})\mathbf{U}] dl = 0 \tag{2}$$

where $\mathbf{w}$ is the moving velocity of the control volume Ω, and $\mathbf{N}$ is the unit outward normal direction on the boundary of Ω. If $\mathbf{w} = \mathbf{u}$, the system reduces to a Lagrangian formulation, and if $\mathbf{w} = 0$, it has Eulerian form.

Let's consider a generic grid at the discrete level. Each cell V_c of the mesh is assigned a unique index c. We use the index f to denote a generic neighboring cell V_f which has a common edge denoted as k with cell V_c, refer to Fig.1. The physical quantities such as the density ρ_c, pressure p_c, velocity $\mathbf{u}_c$, energy E_c, e_c are defined in the cell center of V_c. The moving velocity $\mathbf{w}_q$ of the grid is defined on the node q. In Fig. 1, $\mathbf{N}_k = ((n_x)_k, (n_y)_k)$ is the unit normal vector of the edge k. According to the normal direction of an edge between two neighboring cells, sometimes we use denotation L and R to express the left state and right state of the edge.

The above equations (2) are discretized by utilizing the idea of the Godunov's scheme,

$$\mathbf{U}_c^{n+1} = \frac{|V_c^n|\mathbf{U}_c^n}{|V_c^{n+1}|} - \frac{\Delta t}{|V_c^{n+1}|}\sum_k L_k((\mathbf{F}_k, \mathbf{G}_k) \cdot \mathbf{N}_k - w_k \mathbf{U}_k)$$

where the state vector with the superscript n indicates it is at the time t^n. $|V_c^n|$ and $|V_c^{n+1}|$ denote the volume of cell V_c at time t^n and t^{n+1} respectively, and $\Delta t = t^{n+1} - t^n$. w_k is the normal moving velocity of the cell edge k, which is defined at the center of the cell edge,

$$w_k = 0.5(\mathbf{w}_q + \mathbf{w}_{q+}) \cdot \mathbf{N}_k.$$

L_k is the length of the edge k. In this paper, all quantities appearing in flux functions $\mathbf{U}_k, \mathbf{F}_k, \mathbf{G}_k$ are set at time t^n for simplicity.

According to the rotation invariance of the Eulerian equations, there is

$$(\mathbf{F}_k, \mathbf{G}_k) \cdot \mathbf{N}_k = (\mathcal{T}_k)^{-1}\mathbf{F}(\mathcal{T}_k \mathbf{U}_k), \tag{3}$$

where the $\mathcal{T}_k$ is the rotation matrix and $(\mathcal{T}_k)^{-1}$ is its inverse, namely

$$\mathcal{T}_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & (n_x)_k & (n_y)_k & 0 \\ 0 & -(n_y)_k & (n_x)_k & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

In this transformation, the above numerical scheme can be written as

$$\mathbf{U}_c^{n+1} = \frac{|V_c^n}{|V_c^{n+1}|}|\mathbf{U}_c^n - \frac{\Delta t}{|V_c^{n+1}|}\sum_k L_k(\mathcal{T}_k)^{-1}\mathbf{F}^k, \tag{5}$$

where $\mathbf{F}^k = \mathbf{F}(\mathcal{T}_k\mathbf{U}_k) - w_k\mathcal{T}_k\mathbf{U}_k$. The update of the grid is

$$\mathbf{x}_q^{n+1} = \mathbf{x}_q^n + \Delta t \mathbf{w}_q, \tag{6}$$

where $\mathbf{x}_q$ is the nodal coordinate. $\mathbf{w}_q$ is provided by a specific grid strategy.

III. The one dimensional approximate Riemann solver

The numerical flux $\mathbf{F}^k$ between the cell V_c and V_f can be obtained from many methods. The most popular way is to solve a one-dimensional Riemann problem along the normal direction of the edge k:

$$\begin{aligned} &\mathbf{U}_t + \mathbf{F}_x = 0, \\ IVS: \qquad &\mathbf{U}(x,0) = \begin{cases} \mathbf{U}_L = \mathcal{T}_k\mathbf{U}_c^n, x < 0, \\ \mathbf{U}_R = \mathcal{T}_k\mathbf{U}_f^n, x > 0, \end{cases} \end{aligned} \tag{7}$$

and the numerical flux can be expressed by

$$\mathbf{F}^k(w, \mathbf{U}_L, \mathbf{U}_R) = \mathbf{F}^w - w\mathbf{U}^w, \tag{8}$$

where $w = 0.5(\mathbf{w}_q + \mathbf{w}_{q+}) \cdot \mathbf{N}_k$ is the normal projection of the moving velocity defined on the cell edge k, and $\mathbf{U}^w$ and $\mathbf{F}^w$ are state and flux functions along the grid velocity $x/t = w$ respectively.

It is well known that a wide variety of Riemann solvers have been devised to solve the above Riemann problem. In this paper we will consider the classical HLL solver and its modified scheme: the HLLS solver.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

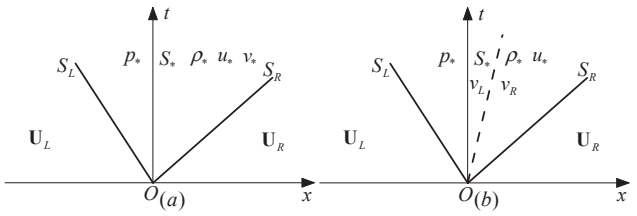

Fig. 2. Structures of the approximate solutions of the HLL-type Riemann problem for the x-split two dimensional Euler equations. (a) HLL solver, (b) HLLS solver.

A. The HLL approximate solver

Harten, Lax and van Leer put forward the following approximate Riemann solver [3]. The wave structure is illustrated in Fig. 2(a). The solution in the star region consists of a single state $\mathbf{U}_*$ separated from data states by two waves of speeds S_L and S_R. The state vector is

$$\mathbf{U}_{HLL}(w,\mathbf{U}_L,\mathbf{U}_R)=\begin{cases}\mathbf{U}_L, & \text{if} \quad w\le S_L,\\ \mathbf{U}_*, & \text{if} \quad S_L<w\le S_R,\\ \mathbf{U}_R, & \text{if} \quad w>S_R,\end{cases} \tag{9}$$

where term $\mathbf{U}_*$ represents the average intermediate state between two waves, which propagates with speeds S_L and S_R. $w = x/t$ is the position to calculate the flux. The corresponding flux is defined as

$$\mathbf{F}_{HLL}(w,\mathbf{U}_L,\mathbf{U}_R)=\begin{cases}\mathbf{F}_L, & \text{if} \quad w\le S_L,\\ \mathbf{F}_*, & \text{if} \quad S_L<w\le S_R,\\ \mathbf{F}_R, & \text{if} \quad w>S_R.\end{cases} \tag{10}$$

The average intermediate state $\mathbf{U}_*$ and flux $\mathbf{F}_*$ can be found from the Rankine- Hugoniot jump conditions across each wave.

$$\mathbf{F}_*-\mathbf{F}_L = S_L(\mathbf{U}_*-\mathbf{U}_L), \tag{11}$$
$$\mathbf{F}_*-\mathbf{F}_R = S_R(\mathbf{U}_*-\mathbf{U}_R). \tag{12}$$

One seldom concerns about the concrete physical quantities in the star region of the HLL solver. Different from the exact Riemann solver and HLLC solver, the flow velocity S_* in the HLL solver has distinct value with the state velocity u_*. In fact, the numerical flux can be split as a convective term and a pressure term,

$$\mathbf{U}_*=\begin{cases}\rho_*,\\ \rho_*u_*,\\ \rho_*v_*,\\ \rho_*E_*,\end{cases} \qquad \mathbf{F}_*=\begin{cases}\rho_*S_*,\\ \rho_*u_*S_*+p_*,\\ \rho_*v_*S_*+0_*,\\ \rho_*E_*S_*+(pS)_*.\end{cases} \tag{13}$$

where the physical quantities ρ_*, u_*, v_*, E_* represent the flow states in the star region, and $S_*, p_*, (pS)_*$ are the transport velocity, pressure and work term. 0_* is viscosity of the shear wave. The HLL formulation in (13) is quite interesting and to the authors knowledge it has not been given in the literature elsewhere. After some simple algebra, we have

$$\begin{aligned}\rho_* &= \frac{\alpha_R-\alpha_L}{S_R-S_L},\\ u_* &= \frac{\alpha_R u_R-\alpha_L u_L+p_L-p_R}{\alpha_R-\alpha_L},\\ v_* &= \frac{\alpha_R v_R-\alpha_L v_L}{\alpha_R-\alpha_L},\\ E_* &= \frac{\alpha_R E_R-\alpha_L E_L+p_L u_L-p_R u_R}{\alpha_R-\alpha_L}.\\ S_* &= \frac{\alpha_R S_L-\alpha_L S_R}{\alpha_R-\alpha_L},\\ p_* &= \frac{\alpha_R p_L-\alpha_L p_R-\alpha_L\alpha_R(u_L-u_R)}{\alpha_R-\alpha_L},\\ 0_* &= \frac{-\alpha_L\alpha_R(v_L-v_R)}{\alpha_R-\alpha_L},\\ (pS)_* &= \frac{\alpha_R(pu)_L-\alpha_L(pu)_R-\alpha_L\alpha_R(E_L-E_R)}{\alpha_R-\alpha_L},\end{aligned}$$

where $\alpha_L=\rho_L(S_L-u_L), \alpha_R=\rho_R(S_R-u_R)$.

The numerical interface flux is

$$\mathbf{F}^k_{HLL}=\begin{cases}\mathbf{F}_L-w\mathbf{U}_L, & \text{if} \quad w\le S_L,\\ \mathbf{F}_*-w\mathbf{U}_*, & \text{if} \quad S_L<w\le S_R,\\ \mathbf{F}_R-w\mathbf{U}_R, & \text{if} \quad w>S_R.\end{cases}$$

When $w = 0$, we denote the classical HLL flux $\mathbf{F}^k_{HLL}$ as the Eulerian HLL solver, when $w \neq 0$, we call it the ALE HLL solver for the sake of distinction.

B. The HLL approximate solver with shear wave

A shortcoming of the HLL scheme is exposed by contact discontinuities, shear waves and material interfaces. These waves are associated with the multiple eigenvalue $\lambda_2 = \lambda_3 = S_*$. Note that all that matters is the average across the wave structure, without regard for the spatial variations of the solution of the Riemann problem in the star region. The work of restoring the missing waves has done by Toro, Spruce and Speares [13], which all of the missing middle waves are put back into the structure of the approximate Riemann solver. However, it is a well-known fact that the modified solver, the HLLC approximate Riemann solver, has to suffer from the carbuncle phenomenon.

Now we only restore a shear wave in the star region of the HLL solver. The wave structure is illustrated in Fig. 2(b), in which the velocities in the tangent direction will remain invariance between the shear wave. The state vector is

$$\mathbf{U}_{L,*}=\begin{cases}\rho_*,\\ \rho_*u_*,\\ \rho_*v_L,\\ \rho_*E_*,\end{cases} \qquad \mathbf{U}_{R,*}=\begin{cases}\rho_*,\\ \rho_*u_*,\\ \rho_*v_R,\\ \rho_*E_*,\end{cases} \tag{14}$$

The corresponding fluxes are

$$\mathbf{F}_{L,*}=\begin{cases}\rho_*S_*,\\ \rho_*u_*S_*+p_*,\\ \rho_*v_LS_*,\\ \rho_*E_*S_*+(pS)_*,\end{cases}$$
$$\mathbf{F}_{R,*}=\begin{cases}\rho_*S_*,\\ \rho_*u_*S_*+p_*,\\ \rho_*v_RS_*,\\ \rho_*E_*S_*+(pS)_*.\end{cases} \tag{15}$$

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Notice that the flow states in the star region and numerical fluxes satisfy the Rankine-Hugoniot jump conditions across each wave. That is

$$\begin{aligned} \mathbf{F}_{L,*} - \mathbf{F}_L &= S_L(\mathbf{U}_{L,*} - \mathbf{U}_L), && (16)\\ \mathbf{F}_{R,*} - \mathbf{F}_{L,*} &= S_*(\mathbf{U}_{R,*} - \mathbf{U}_{L,*}), && (17)\\ \mathbf{F}_R - \mathbf{F}_{R,*} &= S_R(\mathbf{U}_R - \mathbf{U}_{R,*}). && (18) \end{aligned}$$

The resulting numerical interface flux is

$$\mathbf{F}^k_{HLLS} = \begin{cases} \mathbf{F}_L - w\mathbf{U}_L, & \text{if } w \le S_L,\\ \mathbf{F}_{L,*} - w\mathbf{U}_{L,*}, & \text{if } S_L < w \le S_*,\\ \mathbf{F}_{R,*} - w\mathbf{U}_{R,*}, & \text{if } S_* < w \le S_R,\\ \mathbf{F}_R - w\mathbf{U}_R, & \text{if } w > S_R. \end{cases}$$

We denote the approximate Riemann solver as HLLS, where S stands for shear wave. Corresponding to whether w is identical zero or not, the HLLS method is categorized as the Eulerian HLLS or ALE HLLS one.

In order to complete the HLLS Riemann solver, S_R and S_L must be estimated appropriately. There are many choices to obtain them [12]. Here we use the simplest waves-speed estimation,

$$\begin{aligned} S_L &= \min(u_L - c_L, u_R - c_R),\\ S_R &= \max(u_L + c_L, u_R + c_R), && (19) \end{aligned}$$

where c_L and c_R are the sound speeds of the left and the right states.

Due to the satisfaction of the jump condition, the scheme is capable of capture shock rather well. When the flow is in an one dimensional case, the HLLS solver degenerates to the HLL one. Therefore some good properties of the HLL solver are preserved by the HLLS solver, such as be free of low frequency fluctuations in case of slowly moving shock, and others.

C. A discussion to the shock instability

The unique difference between the HLL scheme and HLLS scheme is in the shear wave. One has viscosity on the shear wave and the other does not. The mass fluxes of two schemes are same, which can be rewritten as follows in Eulerian framework:

$$\begin{aligned} F^{(\rho,k)} &= \rho_* S_*\\ &= \frac{1}{2}[(\rho u)_L + (\rho u)_R - D^{(\rho)}\Delta\rho + D^{(u)}\Delta u + D^{(p)}\Delta p], \end{aligned}$$

where dissipation term is expanded in terms of primitive variables, $\Delta = (\cdot)_R - (\cdot)_L$.

Both ρ_* and S_* do not contain the pressure term, thus we have $D^{(p)} = 0$. That means that the HLLS scheme has no pressure contribution to the numerical mass flux. We know that the HLL scheme is in agreement with Liou's conjecture that 'the condition, $D^{(p)} = 0$ for $\forall M$, is sufficient for a scheme to prevent the shock instability from occurring'.

Unfortunately the Eulerian HLLS method and ALE HLLS method are unstable under the interaction of the strong shock.

IV. Numerical experiments

We will present two shock examples to compare the solutions between the schemes using the HLLS and HLL solvers respectively. One is the famous bow shock in the blunt body problem on a fixed grid [8], the second is the stern Saltzmann problem on a moving mesh [1].

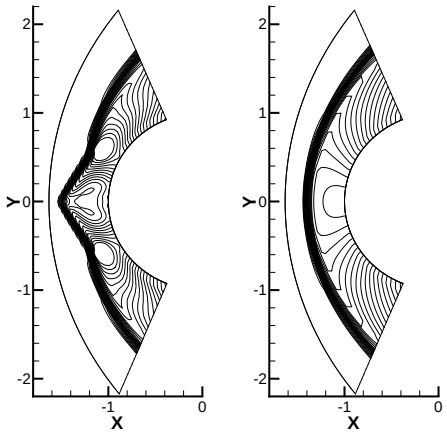

Fig. 3. The density contours for a hypersonic flow over a cylinder. Thirty equally spaced contour lines from $\rho = 1.2$ to $\rho = 6$, left: HLLS, right: HLL.

A. Mach 20 hypersonic flow over a cylinder

This is a well-known test to examine the catastrophic carbuncle failings of upwind schemes. We need to simulate a inviscid flow with Mach number $M_a = 20$ around a circular cylinder. In this test problem, the specific heat ratio of the gas is $\gamma = 1.4$, and the initial inflow has states $(\rho, u, v, p) = (1, M_a\sqrt{\gamma}, 0, 1)$. The computing domain is displayed in Fig. 3 and 20×400 structure grids and the first order accurate scheme are used. The density contours at $t = 2$ by the HLLS and HLL solvers are illustrated in Fig.3. The carbuncle phenomenon appears when using the HLLS scheme and disappears when using the HLL scheme.

B. Saltzmann problem

The Saltzmann test problem [1] is usually used to verify the robustness of a numerical scheme when the mesh is not aligned with the fluid flow. The initial mesh is generated by following formulae

$$\begin{aligned} x_{i,j} &= (i-1)h_x + [0.1 - (j-1)h_y]\sin(\pi(i-1)h_x),\\ y_{i,j} &= (j-1)h_y, \end{aligned}$$

where $h_x = h_y = 0.01$, $1 \le i \le 101, 1 \le j \le 11$.

The initial state is $(\rho_0, p_0, u_0, v_0, \gamma) = (1, 0, 0, 0, 5/3)$. The left boundary condition at $x = 0$ is a pistol with velocity $u = 1$. On all the other boundary, we set up wall conditions.

To move the grid in an approxiamte Lagrangian manner, we adopt a grid moving strategy in CAVEAT code [1]. This algorithm requires that the projection of velocity at a vertex into the normal direction of an edge (sharing the vertex) should be equal to the contact velocity of a one-dimensional Riemann solution on that edge. Refer to Fig. 4. Due to the fact that the problem is often overdetermined at a vertex, the vertex velocity $\mathbf{w}_q$ can be obtained by minimizing the following quadratic functional for each vertex q

$$Func(\mathbf{w}_q) = \sum_{k \in \mathcal{K}(q)} L_k(\mathbf{w}_q \cdot \mathbf{N}_k - S_{k,*})^2, \qquad (20)$$

where $\mathbf{N}_k$ is the unit normal direction of edge k, and $S_{k,*}$ is the contact velocity S_* along the norm direction of k. $\mathcal{K}(q)$ is the number of the edges sharing a vertex q.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

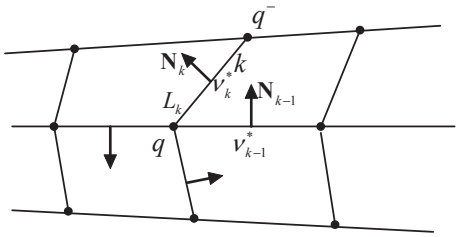

Fig. 4. Velocities associated with a cell vertex.

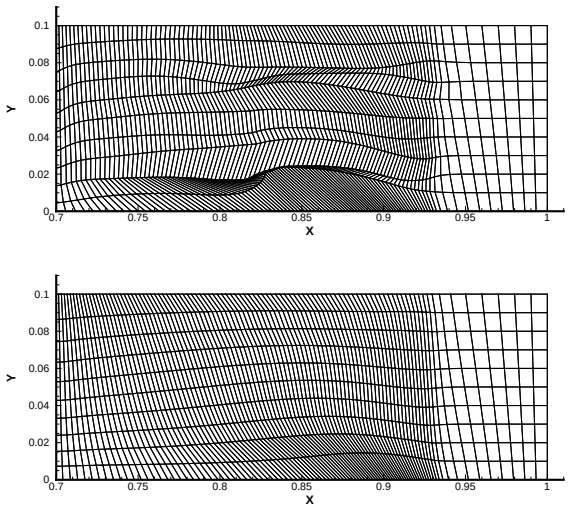

Fig. 5. The grids of the Saltzmann problem at $t = 0.7$ using the HLLS and HLL approximate Riemann solvers, top: HLLS, bottom: HLL.

After given the nodal velocity $\mathbf{w}_q$ to move the grid, the ALE calculation can be performed. The exact solution is a planar shock wave that moves at speed $4/3$ from left to right. The propagation of the shock wave at $t = 0.7$ is displayed in Fig. 5. The ALE HLL preserves one-dimensional solution very well, whereas the ALE HLLS produces severe grid distortion.

V. Conclusion

When there exists strong shock, the instability mechanism appearing in the ALE calculation on a moving mesh has close relation with numerical shock instability in the Eulerian method.

In this paper, we pay attention to what is the factor most likely to lead to the occurrence of unstable phenomenon. We construct a HLLS scheme, which is a HLL-type Riemann solver with a shear wave, and compare its solution behavior with the classical HLL solver. The HLL method is stable while the HLLS scheme is unstable although it has same mass flux with the HLL one. The concrete performance is that the HLLS solver produces the carbuncle phenomenon in the Eulerian calculation and leads to a severe grid deformation in the ALE computation. The result indicates that Liou's conjecture for the root of the multidimensional shock instability may not be correct in a general case. The unique factor to cause instability may be viscosity vanishing associated to the shear wave.

References

[1] J. K. Dukowicz and B. Meltz, Vorticity Errors in Multidimensional Lagrangian Codes, J. Comput. Phys. 99(1992) 115-134.
[2] S. Galera, P. H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (2010) 5755-5787.
[3] A. Harten, P. D. Lax and B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983), 35-61.
[4] M. S. Liou, Mass Flux schemes and connection to shock instability, J. Comput. Phys. 160 (2000), 623-648.
[5] H. Luo, J. D. Baum, R. Lohner, On the computation of multi-material flows using ALE formulation, J. Comput. Phys. 194 (2004) 304-328.
[6] J. M. Moschetta, J. Gressier, J. C. Robinet, G. Casalis, The carbuncle phenomenon: a genuine Euler instability?, in: E.F. Toro (Ed.), Godunov Methods, Theory and Applications, Kluwer Academic/Plenum Publishers, 1995, pp. 639C645.
[7] R. Pakmor, A. Bauer and V. Springel, Magnetohydrodynamics on an unstructured moving grid, Mon. Not. R. Astron. Soc. 418(2)(2011) 1392-1401.
[8] K.M. Peery and S.T. Imlay, Blunt body flow simulations, AIAA Paper 88-2924, 1988.
[9] J. Quirk, A contribution to the Great Riemann Solver Debate, Int. J. Numer. Meth. Fluid. 18 (1994), 555-574.
[10] Z. J. Shen, W. Yan, G. W. Yuan, A robust and contact resolving Riemann solver on unstructured mesh, Part II, ALE method, J. Comput. Phys. 268(2014) 456-484.
[11] Z. J. Shen, W. Yan, and G. W. Yuan, A Stability Analysis of Hybrid Schemes to Cure Shock Instability, Commun.Comput. Phys. 15(2014) 1320-1342.
[12] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, Berlin, 1997.
[13] E. F. Toro,M. Spruce and W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock Wave 4 (1994) 25-34.
[14] K. Xu, Z. Li, Dissipative mechanism in Godunov-type schemes, Int. J. Numer. Methods Fluids, 37 (2001), 1-22.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)