COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. 2, No. 4, pp. 577-610

Commun. Comput. Phys.
August 2007

## Review Article

# The Unified Coordinate System in Computational Fluid Dynamics

W. H. Hui[*]

*Division of Mechanics, Research Centre for Applied Sciences, Academia Sinica, Nankang, Taipei 115, Taiwan.*

Received 17 May 2006; Accepted (in revised version) 10 November 2006

Communicated by Chi-Wang Shu

Available online 8 January 2007

**Abstract.** A fundamental issue in CFD is the role of coordinates and, in particular, the search for "optimal" coordinates. This paper reviews and generalizes the recently developed unified coordinate system (UC). For one-dimensional flow, UC uses a material coordinate and thus coincides with Lagrangian system. For two-dimensional flow it uses a material coordinate, with the other coordinate determined so as to preserve mesh othorgonality (or the Jacobian), whereas for three-dimensional flow it uses two material coordinates, with the third one determined so as to preserve mesh skewness (or the Jacobian). The unified coordinate system combines the advantages of both Eulerian and the Lagrangian system and beyond. Specifically, the followings are shown in this paper. (a) For 1-D flow, Lagrangian system plus shock-adaptive Godunov scheme is superior to Eulerian system. (b) The governing equations in any moving multi-dimensional coordinates can be written as a system of closed conservation partial differential equations (PDE) by appending the time evolution equations – called geometric conservation laws – of the coefficients of the transformation (from Cartesian to the moving coordinates) to the physical conservation laws; consequently, effects of coordinate movement on the flow are fully accounted for. (c) The system of Lagrangian gas dynamics equations is written in conservation PDE form, thus providing a foundation for developing Lagrangian schemes as moving mesh schemes. (d) The Lagrangian system of gas dynamics equations in two- and three-dimension are shown to be only weakly hyperbolic, in direct contrast to the Eulerian system which is fully hyperbolic; hence the two systems are not equivalent to each other. (e) The unified coordinate system possesses the advantages of the Lagrangian system in that contact discontinuities (including material interfaces and free surfaces) are resolved sharply. (f) In using the UC, there is no need to generate a body-fitted mesh prior to computing flow past a body; the mesh is automatically generated by the flow. Numerical examples are given to confirm these properties. Relations of the UC approach with the Arbitrary-Lagrangian-Eulerian (ALE) approach and with various moving coordinates approaches are also clarified.

[*]Corresponding author. *Email address:* `whhui@ust.hk` (W. H. Hui)

**PACS (2006)**: 47.11.-j, 47.40.-x

**Key words**: Unified coordinates, Eulerian coordinates, Lagrangian coordinates, contact discontinuities, automatic mesh generation, moving mesh, conservation form.

## Contents

## 1 Introduction

Computational Fluid Dynamics (CFD) uses large scale numerical computation to solve problems of fluid flow. It has been known since its onset that the numerical solution to a given flow depends on the relation between the flow and the coordinates (mesh) used to compute it. Each of the two well-known coordinate systems for describing fluid flow – Eulerian and Lagrangian – has advantages as well as drawbacks. Eulerian method is relatively simple, but its drawbacks are: (a) it smears contact discontinuities badly, and (b) it needs generating a body-fitted mesh prior to computing flow past a body. Lagrangian method, by contrast, resolves contact discontinuities (including material interfaces and free surfaces) sharply, but it too has drawbacks: (a) the gas dynamics equations could not be written in conservation partial differential equations (PDE) form, rendering numerical computation complicated, and (b) it breaks down due to cell deformation.

The objective of this paper is to review and generalize the recently developed unified coordinate system (UC) mostly by the author and his collaborators [1–16]. To put it in perspective we shall first comment on the relative merits of the existing coordinate systems, mainly, Eulerian, Lagrangian, Arbitrary-Lagrangian-Eulerian (ALE), and the moving mesh (coordinate).

### 1.1 Theoretical issues

For more than 200 years, two coordinate systems have existed for describing fluid flow: Eulerian system is fixed in space, whereas Lagrangian system follows the fluid. An immediate question is “are they equivalent to each other theoretically?” This question must have been asked by numerous researchers in fluid mechanics, and the answer presumably was positive. Surprisingly, the first mathematical proof of equivalency, meaning the existence of a one-to-one map between the two sets of weak solutions obtained by using the two systems, was given as late as 1987 by Wagner [17] and holds only for one-dimensional flow (Note: in the presence of a vacuum, the definition of weak solution for the Lagrangian equations must be strengthened to admit test functions which are discontinuous at the vacuum). For two- and three-dimensional flows, Hui et al [1, 2] showed that they are not equivalent to each other theoretically (see Section 5.4).

### 1.2 Computational issues

Computationally, Eulerian and Lagrangian systems are not equivalent even for one-dimensional flow. For 1-D flow, we shall show [18,19] in Section 2 that Lagrangian system plus shock-adaptive Godunov scheme [6, 20] is superior to the Eulerian one.

The situation in 2-D and 3-D flows is more complicated: Eulerian and Lagrangian system each has advantages and drawbacks. In general, Eulerian method is relatively simple, because the gas dynamics equations can be written in conservation PDE form, which provides the theoretical foundation for shock-capturing computation. However, it has two drawbacks: (a) it smears contact discontinuities badly, and (b) it needs generating a body-fitted mesh prior to computing flow past a body, but mesh generation is tedious, time-consuming and requires specialized training.

Lagrangian method, by contrast, resolves contact discontinuities (including material interfaces and free surfaces) sharply, because they coincide with Lagrangian coordinates. It, too, has two drawbacks: (a) it may break down due to cell deformation, because a Lagrangian computational cell is literally a fluid particle with finite – though small – size and hence deforms with the fluid and (b) the gas dynamics equations could not be written in conservation partial differential equations (PDE) form, rendering numerical computation complicated. In this regard, we note that as late as 1999 Serre [21] stated "Writing the equations of gas dynamics in Lagrangian coordinates is very complicated if (dimension) D$\geq$2". Interesting enough, it was shown [1] in the same year that with the unified coordinates it is easy to derive the Lagrangian gas dynamics equations in conservation PDE form (see Section 5.4).

Early efforts to combine the advantages of both Lagrangian and Eulerian systems have resulted in the famous Particle-in-Cell method [22, 23] and the Marker-and-Cell method [24–27]. The highly original idea of Harlow developed in the Particle-in-Cell method of separating a computational cycle into a Lagrangian phase plus a convective, or remap/rezone, phase has been widely used in many hydrodynamic computer codes, in particular, in the celebrated Arbitrary-Lagrangian-Eulaerian (ALE) code [28–33]. While ALE has achieved greatly in resolving contact discontinuities, an important phase of its computation – the remap/rezone phase – requires the intervention of the user [32], although new ideas have recently been proposed [33] to avoid it (more will be said about ALE in Sections 3.3 and 3.4).

Several moving mesh methods have also been developed [34–40], which require satisfying one or more space (or geometric) conservation equations, which are either derived mathematically or given intuitively, in addition to the physical conservation laws. While these are necessary condition(s), it is not certain whether the effects due to coordinate movement on the flow are fully taken into account (see Section 5.3 for more comments).

The unified coordinate system [1–16] has been developed against the above background. For two-dimensional flow UC uses a material coordinate and the other coordinate is determined so as to preserve mesh othorgonality (or the Jacobian), whereas for three-dimensional flow it uses two material coordinates, with the third one determined so as to preserve mesh skewness (or the Jacobian). In the one-dimensional case, it coincides with the Lagrangian system (see Section 6, Remark 6.3). It will be shown in this paper that UC has advantages of both Eulerian and Lagrangian system and beyond. Its relation with ALE and moving mesh methods will also be clarified.

This paper is organized as follows. In Section 2, we show that for 1-D flow, Lagrangian system with shock-adaptive Godunov scheme is superior to Eulerian system. Section 3 comments further on the current methods for computing multi-dimensional flow based on various coordinate systems, pointing out their relative merits and shortcomings. Section 4 states the requirements for an "optimal" coordinate system. Section 5 derives the conservation PDE form equations in arbitrary coordinates. Section 6 discusses the properties of the UC system. Section 7 describes the computation procedure. Typical results for 2-D computation are given in Section 8 and, finally, conclusions are given in Section 9. For simplicity, we consider inviscid flow of a $\gamma$-law perfect gas, but comments and examples on viscous flow computation will be given in Section 8.

## 2 One-dimensional flow

For one-dimensional flow, it is shown in Section 6 that unified coordinate system coincides with the Lagrangian one. We shall now show that Lagrangian coordinates plus shock-adaptive Godunov scheme [6,20] is superior to Eulerian coordinates.

We begin with the Euler equations of gas dynamics in Cartesian coordinates $(t,x)$ for a $\gamma$-law gas

$$\begin{aligned}
&\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho e\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ u(\rho e+p)\end{pmatrix}=0,\\
&e=\frac{1}{2}u^2+\frac{1}{\gamma-1}\frac{p}{\rho}.
\end{aligned}\tag{2.1}$$

Here $u$ is velocity, $p$ pressure, $\rho$ density and $e$ specific total energy. System (2.1) is hyperbolic and in conservation form. We transform (2.1) to the Lagrangian coordinates $(\lambda,\xi)$

by

$$\left\{\begin{array}{l} dt = d\lambda, \\ dx = u d\lambda + \dfrac{1}{\rho} d\xi, \end{array}\right. \tag{2.2}$$

to get

$$\frac{\partial}{\partial \lambda}\begin{pmatrix} 1/\rho \\ u \\ e \end{pmatrix} + \frac{\partial}{\partial \xi}\begin{pmatrix} -u \\ p \\ up \end{pmatrix} = 0. \tag{2.3}$$

System (2.3) is also hyperbolic and in conservation form, and Wagner [17] showed that solutions to (2.1) and to (2.3) are equivalent theoretically. However, they give different numerical solutions, as demonstrated by Example 1 in Figs. 1a and 1b [18, 19] for a Riemann problem (taken from [41]), whose solution consists of two shocks and a contact discontinuity in between. It is seen that while shock resolutions are similar, Lagrangian computation resolves the contact much better than Eulerian. This is because a contact line is a material line and hence coincides with Lagrangian coordinate, whereas it does not coincide with Eulerian coordinate. In passing, we note with interest that in their seminal papers, von Neumann [42] and Godunov [43] both use Lagrangian system (2.3), instead of Eulerian system (2.1).

The error near the contact discontinuity (Fig. 1b) in Lagrangian computation is associated with the well-known "wall over-heating" phenomena first discovered by von Neumann [44], but can be remedied using the shock-adaptive Godunov scheme [19, 20], instead of the classical Godunov scheme. In the shock-adaptive Godunov scheme shocks are fitted, using the Riemann solution with no extra cost, so we can replace the energy conservation equation by the entropy conservation equation which holds in the smooth flow region. Accordingly, (2.3) becomes

$$\frac{\partial}{\partial \lambda}\begin{pmatrix} 1/\rho \\ u \\ p/\rho^{\gamma} \end{pmatrix} + \frac{\partial}{\partial \xi}\begin{pmatrix} -u \\ p \\ 0 \end{pmatrix} = 0. \tag{2.4}$$

Computations using (2.4) resolve both shock and contact discontinuities sharply, as shown in Fig. 1c. We therefore conclude that Lagrangian (or UC) computation with shock-adaptive Godunov scheme is superior to Eulerian computation for 1-D flow, and is most robust, accurate and efficient. Indeed, all other known difficulties of Eulerian method, namely, start-up errors, slow moving shocks, low-density flow, sonic-point glitch, and wall-overheating, etc., are also avoided or cured [19].

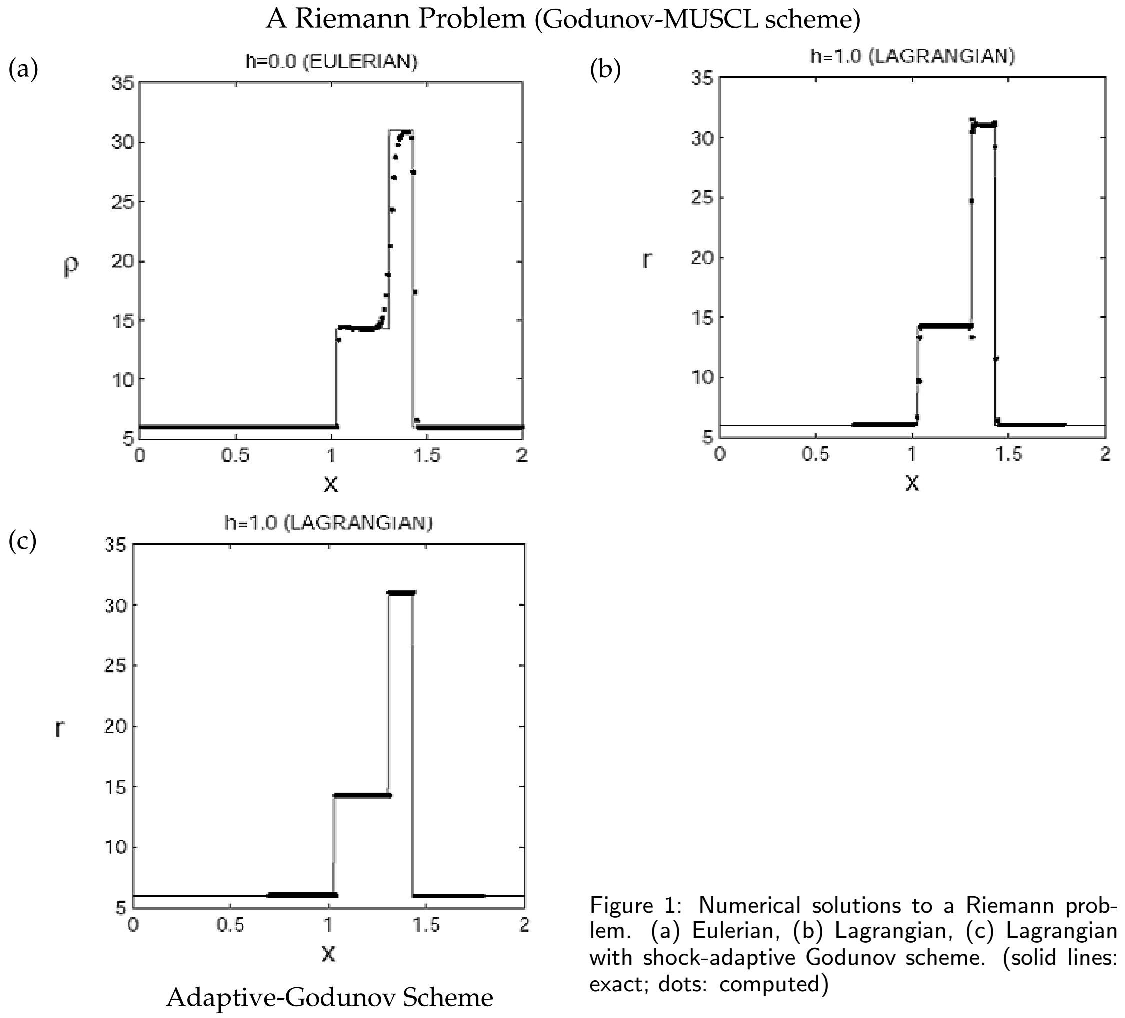


Figure 1: Numerical solutions to a Riemann problem. (a) Eulerian, (b) Lagrangian, (c) Lagrangian with shock-adaptive Godunov scheme. (solid lines: exact; dots: computed)

## 3 Multi-dimensional flow – Comments on current computational methods

In this section, we make further comments on existing methods for computing multi-dimensional flow based on different coordinates. A conservation law in arbitrary coordinates written in integral form is

$$\frac{d}{dt}\int_{\Omega_{\vec{Q}(t)}} U d\Omega = -\int_{\partial\Omega_{\vec{Q}(t)}} F\cdot\vec{n} dS, \tag{3.1}$$

where $t$ is a time variable, $U$ the conserved variable, $F(U)$ the flux, $\Omega$ an arbitrary control volume moving with arbitrary velocity $\vec{Q}$, and $\vec{n}$ the unit normal vector on the surface of

$\Omega$. For gas dynamics, (3.1) is given in details as

$$\left\{\begin{aligned}&\frac{d}{dt}\int\limits_{\Omega_{\vec{Q}(t)}}\rho\, d\Omega=-\int\limits_{\partial\Omega}\rho\left(\vec{q}-\vec{Q}\right)\cdot\vec{n}dS,\\&\frac{d}{dt}\int\limits_{\Omega_{\vec{Q}(t)}}\rho\, q_j\, d\Omega=-\int\limits_{\partial\Omega}\left[\rho\, q_j\left(\vec{q}-\vec{Q}\right)+p\vec{j}\right]\cdot\vec{n}dS,\quad j=1,2,3\\&\frac{d}{dt}\int\limits_{\Omega_{\vec{Q}(t)}}\rho\, e\, d\Omega=-\int\limits_{\partial\Omega}\left[\rho\, e\left(\vec{q}-\vec{Q}\right)+p\vec{q}\right]\cdot\vec{n}dS.\end{aligned}\right.\tag{3.2}$$

Here $\vec{q}$ is fluid velocity and $e=\frac{1}{2}\vec{q}^{\,2}+i(p,\rho)$, $i$ being the specific internal energy. The essence of CFD is to solve (3.2) for every small cell $\Omega$.

### 3.1 Eulerian computation

In this case $\vec{Q}$ = 0 and hence cells $\Omega$ are independent of time $t$. Assuming the flow variables to have continuous first derivatives, application of Gauss divergence theorem to (3.1) yields

$$\frac{\partial U}{\partial t}+\nabla\cdot F(U)=0.\tag{3.3}$$

Eq. (3.3) is in conservation PDE form and thus provides a foundation for shock-capturing methods [64,65]. This is the most important advantage of Eulerian coordinates. However, Eulerian computation has two drawbacks:

- Contact discontinuities (including material interfaces and free surfaces) are badly smeared, because they do not coincide with coordinate surfaces; typically contact smearing increases with number of computational time steps.
- For flow past a body, it is necessary to generate a body-fitted mesh prior to flow computation; mesh generation remains a tedious and time-consuming process after more than three decades of research.

### 3.2 Lagrangian computation

In this case $\vec{Q}=\vec{q}$, hence cells $\Omega$ move and deform literally with the fluid. The most important advantage of Lagrangian computation is that it can resolve contact discontinuities (including material surfaces and free surfaces) sharply, because the latters coincide with Lagrangian coordinates. However, Lagrangian computation, too, has two drawbacks:

- Due to cell movement and deformation-both are unknown a priori-the integral equation (3.2) could not be easily written in conservation PDE form (1-D flow is a fortuitous exception, see Eq. (2.3) above). This has serious consequences. To begin with, lack of conservation PDE form has prevented any Lagrangian scheme to be a scheme on a moving mesh in Eulerian space; the design of such a scheme has great historical importance as well as practical application, as pointed out in [45,46]. Furthermore, without conservation form PDE, additional computation procedure is needed to tract the movement of the cells in using (3.2). This is usually done by using staggered meshes in order to reduce the error (due to interpolation) for the velocity which controls the mesh movement. But switching between meshes requires interpolations of flow variables and geometries, producing numerical diffusion.

- Computation may break down due to cell deformation. This is because a Lagrangian computational cell is literally a fluid particle with finite size, no matter how small, and hence deforms with the fluid. A great effort in Lagrangian computation is to prevent it from breaking down by using special treatments; ALE method represents such a successful treatment.

### 3.3 The Arbitrary-Lagrangian-Eulerian (ALE) computation [28–33]

In this method a computational cycle consists of a computation phase in Lagrangian space followed by a remap/rezone phase to the Eulerian or ALE space. In the Lagrangian phase, it has the advantages as well as the drawbacks of the Lagrangian computation mentioned above, namely, it resolves contact discontinuities sharply but, due to the lack of conservation PDE form, it uses staggered meshes and hence produces numerical diffusion arising from switching between the meshes. Moreover, numerical diffusion is also introduced in the remap/rezone phase of the computation, because it too requires interpolations of the flow variables and of geometries. Indeed, it was demonstrated by Hall [47] that rezoning results in the type of errors encountered in Eulerian solutions and that with continuous remapping/rezoning, ALE computation is equivalent to Eulerian computation. The remap/rezone phase is, however, needed in order to prevent computational breakdown.

### 3.4 The unified coordinates computation

Details of the unified coordinate approach will be given in Sections 6 and 7 later. Here in order to compare with the ALE approach, we summarize its main features. These are: the governing equations are written in conservation PDE form and the mesh velocity is chosen such that, for 2-D flow, one of its coordinates coincides with material coordinate (in the 3-D case, two coordinates are material). The former property shares with the advantage of Eulerian method, whereas the latter shares with the advantage of the Lagrangian method.

The UC approach and the ALE approach share the same spirit that they both combine the best features of Lagrangian and Eulerian approach [31, p. 198] and that the coordinates move at an arbitrary velocity [32, p. 239]. However, the strategies are quite different. In the ALE approach, "the general strategy is to perform a Lagrangian step and to follow it with a remap step that maps the solution in the distorted Lagrangian mesh onto the spatially fixed Eulerian mesh or the ALE mesh" [32, p. 236], [31, p. 198]. This is usually done by employing a staggered mesh. Furthermore, the rezoning strategies are not generally prescribed; instead, "rezoning requires the intervention of the user, ..., and the success of the method depends heavily on the skill and patience of the user" [32, p. 322]. In the UC approach, since the governing equations are in conservation PDE form the computation is done in one step as if it were an Eulerian computation, but without numerical diffusion across contact discontinuities (see examples in Section 8), thus avoiding the numerical diffusion arising from the use of staggered mesh and from remapping/rezoning. The possible computational breakdown is prevented by requiring the mesh to be orthogonal (in 2-D), or to be skewness-preserving (in 3-D).

# 4 The "optimal coordinate system"

Can we have a coordinate system that combines the advantages of Eulerian and Lagrangian, while avoiding their drawbacks? Such a system would be "optimal" in some sense (whether or not a system is optimal depends on the criteria, which are necessarily subjective). Specifically, we want the system to possess the following properties for compressible flow computation:

- Conservation PDE form exists, as in Eulerian;
- Contact discontinuities are sharply resolved, as in Lagrangian;
- Mesh can be automatically generated to fit given body shapes;
- Mesh is orthogonal;
- Mesh is uniform; and more.

We regard the first three properties as most important and shall first investigate, in the next section, the existence of conservation PDE form of the governing equations in arbitrary coordinates.

# 5 Derivation of conservation form PDE

## 5.1 The transformation

We introduce arbitrary coordinates, $(\lambda,\xi,\eta,\zeta)$, via a transformation from Cartesian $(t,x,y,z)$ as follows:

$$\begin{aligned} dt &= d\lambda, \\ dx &= Ud\lambda + Ad\xi + Ld\eta + Pd\varsigma, \\ dy &= Vd\lambda + Bd\xi + Md\eta + Qd\varsigma, \\ dz &= Wd\lambda + Cd\xi + Nd\eta + Rd\varsigma. \end{aligned} \tag{5.1}$$

From (5.1), we get

$$\frac{D_{\vec{Q}}}{Dt}\begin{pmatrix}\xi\\ \eta\\ \varsigma\end{pmatrix}=0, \tag{5.2}$$

where $\frac{D_{\vec{Q}}}{Dt}\equiv\frac{\partial}{\partial t}+\vec{Q}\cdot\nabla_{\vec{x}}$. So the coordinates $(\xi,\eta,\zeta)$, and hence the computational cells, move with velocity $\vec{Q}=(U,V,W)$.

There are two special cases: Eulerian when $\vec{Q}=0$ and Lagrangian when $\vec{Q}=\vec{q}$. In the general case, we have a coordinate system with three degrees of freedom: $U$, $V$ and $W$ are arbitrary. On the other hand, the nine coefficients $A,B,\ldots,R$ in the transformation are not arbitrary, but must satisfy a set of compatibility conditions for $dx$, $dy$ and $dz$ to be total differentials. These conditions are:

$$\begin{aligned}
&\frac{\partial A}{\partial\lambda}=\frac{\partial U}{\partial\xi}, && \frac{\partial L}{\partial\lambda}=\frac{\partial U}{\partial\eta}, && \frac{\partial P}{\partial\lambda}=\frac{\partial U}{\partial\varsigma},\\
&\frac{\partial B}{\partial\lambda}=\frac{\partial V}{\partial\xi}, && \frac{\partial M}{\partial\lambda}=\frac{\partial V}{\partial\eta}, && \frac{\partial Q}{\partial\lambda}=\frac{\partial V}{\partial\varsigma},\\
&\frac{\partial C}{\partial\lambda}=\frac{\partial W}{\partial\xi}, && \frac{\partial N}{\partial\lambda}=\frac{\partial W}{\partial\eta}, && \frac{\partial R}{\partial\lambda}=\frac{\partial W}{\partial\varsigma},
\end{aligned}\tag{5.3a}$$

$$\begin{aligned}
&\frac{\partial A}{\partial\eta}=\frac{\partial L}{\partial\xi}, && \frac{\partial A}{\partial\varsigma}=\frac{\partial P}{\partial\xi}, && \frac{\partial L}{\partial\varsigma}=\frac{\partial P}{\partial\eta},\\
&\frac{\partial B}{\partial\eta}=\frac{\partial M}{\partial\xi}, && \frac{\partial B}{\partial\varsigma}=\frac{\partial Q}{\partial\xi}, && \frac{\partial M}{\partial\varsigma}=\frac{\partial Q}{\partial\eta},\\
&\frac{\partial C}{\partial\eta}=\frac{\partial N}{\partial\xi}, && \frac{\partial C}{\partial\varsigma}=\frac{\partial R}{\partial\xi}, && \frac{\partial N}{\partial\varsigma}=\frac{\partial R}{\partial\eta}.
\end{aligned}\tag{5.3b}$$

We note that of the eighteen conditions in (5.3), only nine of them are independent. We shall take the first nine conditions, (5.3a), which all involve $\lambda$-derivative and are called time-evolution, to be the independent conditions; the remaining nine, called differential constraints, then hold for all time provided they hold initially.

### 5.2 Derivation of governing equations in conservation PDE form

Consider conservation of mass equation, which can be written as

$$\begin{aligned}
0 &= \frac{\partial\rho}{\partial t}+\frac{\partial(\rho u_j)}{\partial x_j} \quad (j=1,2,3)\\
&= \frac{\partial(\rho u_\alpha)}{\partial x_\alpha} \quad (\alpha=0,1,2,3),(x_0=t,u_0=1)\\
&= \oint_{\partial\Omega}\rho u_\alpha \hat{d}x_\alpha
\end{aligned}\tag{5.4}$$

after using Gauss divergence theorem. Here $x_0=t$, $u_0=1$, $\Omega$ is any control volume, and the summation convention has been used. We define

$$\left\{\begin{aligned} \hat{d}x_0 &= \phantom{-}dx_1dx_2dx_3,\\ \hat{d}x_1 &= -dx_2dx_3dx_0,\\ \hat{d}x_2 &= \phantom{-}dx_3dx_0dx_1,\\ \hat{d}x_3 &= -dx_0dx_1dx_2,\end{aligned}\right. \qquad \text{and} \qquad \left\{\begin{aligned} \hat{x}_0 &= (x_1,x_2,x_3),\\ \hat{x}_1 &= (x_2,x_3,x_0),\\ \hat{x}_2 &= (x_3,x_0,x_1),\\ \hat{x}_3 &= (x_0,x_1,x_2).\end{aligned}\right.$$

$\hat{d}\xi_\beta$ and $\hat{\xi}_\beta$ are defined similarly. From the transformation we get

$$\hat{d}x_\alpha = U_{\alpha\beta}\hat{d}\xi_\beta, \tag{5.5}$$

where $\xi_0=\lambda$, $\xi_1=\xi$, $\xi_2=\eta$, $\xi_3=\varsigma$, and

$$U_{\alpha\beta}=\frac{\partial\hat{x}_\alpha}{\partial\hat{\xi}_\beta}. \tag{5.6}$$

Hence,

$$0=\oint_{\partial\Omega}\rho u_\alpha \hat{d}x_\alpha=\oint_{\partial\Omega}\rho u_\alpha U_{\alpha\beta}\hat{d}\xi_\beta.$$

Using Gauss divergence theorem, we get

$$\frac{\partial K_\beta}{\partial\xi_\beta}=0, \tag{5.7}$$

where $K_\beta=\rho u_\alpha U_{\alpha\beta}$. Eq. (5.7) is the mass equation written in conservation form in the arbitrary coordinate system. We can similarly derive the momentum and energy equations in conservation PDE form. To summarize, we have

$$\left\{\begin{array}{l} \dfrac{\partial(\rho u_\alpha)}{\partial x_\alpha}=0 \quad (\alpha=0,1,2,3)\\ \dfrac{\partial(\rho u_j u_\alpha)}{\partial x_\alpha}+\dfrac{\partial p}{\partial x_j}=0 \quad (j=1,2,3)\\ \dfrac{\partial(\rho u_\alpha H)}{\partial x_\alpha}-\dfrac{\partial p}{\partial x_0}=0 \end{array}\right. \tag{5.8}$$

in the Cartesian coordinates, and

$$\left\{\begin{array}{l} \dfrac{\partial K_\beta}{\partial\xi_\beta}=0 \quad (\beta=0,1,2,3)\\ \dfrac{\partial\left(K_\beta u_j+pU_{j\beta}\right)}{\partial\xi_\beta}=0\\ \dfrac{\partial\left(K_\beta H-pU_{0\beta}\right)}{\partial\xi_\beta}=0 \end{array}\right. \tag{5.9}$$

in arbitrary coordinates. Here $H=e+p/\rho$.

The physical laws are now written in conservation PDE form, (5.9), in arbitrary coordinates, including the Lagrangian. However, (5.9) is not a closed system, because it contains (through $U_{\alpha\beta}$) new unknowns: the coefficients $A,B,\ldots,R$ in the transformation (5.1). (Note: this explains why, as pointed out earlier, in Lagrangian coordinates, the physical laws alone cannot be written in closed conservation PDE form; the 1-D case, Eq. (2.3), is just a fortuitous exception). Although unknown, these coefficients are nevertheless related to the mesh velocity $(U,V,W)$ via the compatibility conditions (5.3a). To have a system of PDE that is closed and in conservation form it is, therefore, necessary and sufficient to append the time evolution equations, (5.3a), to the physical conservation laws, (5.9). Eq. (5.3a) will be called geometric conservation laws (GCL). In this regard, Eulerian coordinates are a degenerate case in that the geometric conservation laws reduce to triviality, and the physical conservation laws alone suffice as a closed system.

We conclude that Eqs. (5.9) and (5.3a) form a closed system of fourteen conservation PDE, containing fourteen unknowns: $\rho,p,\vec{q},A,B,\ldots,R$ (the mesh velocities $(U,V,W)$ in these equations will be determined in Section 6). These are:

$$\begin{aligned}
&\frac{\partial(\rho J)}{\partial\lambda}+\frac{\partial(\rho X)}{\partial\xi}+\frac{\partial(\rho Y)}{\partial\eta}+\frac{\partial(\rho Z)}{\partial\varsigma}=0,\\
&\frac{\partial(\rho Jq_j)}{\partial\lambda}+\frac{\partial(\rho Xq_j+p\vec{J}_1)}{\partial\xi}+\frac{\partial(\rho Yq_j+p\vec{J}_2)}{\partial\eta}+\frac{\partial(\rho Zq_j+p\vec{J}_3)}{\partial\varsigma}=0, \quad (j=1,2,3)\\
&\frac{\partial(\rho Je)}{\partial\lambda}+\frac{\partial(\rho Xe+p\vec{J}_1\cdot\vec{q})}{\partial\xi}+\frac{\partial(\rho Ye+p\vec{J}_2\cdot\vec{q})}{\partial\eta}+\frac{\partial(\rho Ze+p\vec{J}_3\cdot\vec{q})}{\partial\varsigma}=0,\\
&\frac{\partial A}{\partial\lambda}=\frac{\partial U}{\partial\xi},\quad \frac{\partial L}{\partial\lambda}=\frac{\partial U}{\partial\eta},\quad \frac{\partial P}{\partial\lambda}=\frac{\partial U}{\partial\varsigma},\\
&\frac{\partial B}{\partial\lambda}=\frac{\partial V}{\partial\xi},\quad \frac{\partial M}{\partial\lambda}=\frac{\partial V}{\partial\eta},\quad \frac{\partial Q}{\partial\lambda}=\frac{\partial V}{\partial\varsigma},\\
&\frac{\partial C}{\partial\lambda}=\frac{\partial W}{\partial\xi},\quad \frac{\partial N}{\partial\lambda}=\frac{\partial W}{\partial\eta},\quad \frac{\partial R}{\partial\lambda}=\frac{\partial W}{\partial\varsigma}.
\end{aligned}\tag{5.10}$$

Here $X=(\vec{q}-\vec{Q})\cdot\vec{J}_1$, $Y=(\vec{q}-\vec{Q})\cdot\vec{J}_2$, $Z=(\vec{q}-\vec{Q})\cdot\vec{J}_3$ and

$$\vec{J}_1=\begin{pmatrix} MR-NQ\\ NP-LR\\ LQ-MP\end{pmatrix},\quad \vec{J}_2=\begin{pmatrix} CQ-BR\\ AR-CP\\ BP-AQ\end{pmatrix},\quad \vec{J}_3=\begin{pmatrix} BN-CM\\ CL-AN\\ AM-BL\end{pmatrix},\quad J=\begin{vmatrix} ALP\\ BMQ\\ CNR\end{vmatrix}.$$

We remark that the 14-equation system in arbitrary coordinates, (5.10), appears to be much larger than the 5-equation system in Eulerian coordinates, (5.8), but the eigenvalues corresponding to the additional nine equations – the geometric conservation laws – are found to be equal to zero (multiplicity 9), giving rise only to new linearly degenerated waves and no new nonlinear waves (shocks, etc.). Details of these eigenvalues and their corresponding eigenvectors are given in [1] for 2-D flow and in reference #2 of [2] for 3-D flow. Therefore, system (5.10) does not generate spurious nonlinear waves that are not in the original system (5.8). Computationally (see Section 7.2 for details), the physical conservation laws are first solved, keeping the geometric variables $(A,B,\ldots,R)$ fixed, then the mesh velocity $(U,V,W)$ are computed, and finally the geometric variables are updated using the geometric conservation laws alone. In this way, even the additional linearly degenerated waves do not appear (see [1–3, 48, 54]). It is also interesting to observe that in the special case of Lagrangian coordinates when $\vec{Q}=\vec{q}$, the linearly degenerated wave corresponding to the zero eigenvalue is not new but is already in the original system.

As to computing time requirement, the bulk of it is spent on solving the physical conservation laws, which are the same as in Eulerian computation. Additional time used in updating the geometric conservation laws adds about 3% to the total, while additional time used to compute the mesh velocity $(U,V,W)$ adds another 5-10%, depending on the problems at hand.

## 5.3 Comments on closed system of governing equations in moving coordinates

It is well known [34–37] that when moving coordinates are used an additional conservation equation, namely the space conservation law (SCL) (or geometric conservation law), has to be satisfied; otherwise erroneous solutions are obtained and numerical instability may occur. The emphasis in [34–37] is to use the SCL to ensure that an initially spatially uniform steady flow remains uniform. The insufficiency of a single SCL for unsteady flow has since been pointed out and additional necessary conditions suggested [38–40]. In this regard, we note that our geometric conservation laws, (5.3a), are the sufficient and necessary conditions governing the coordinate movement. Consequently, effects of arbitrarily moving coordinates (hence moving mesh) on steady or unsteady, compressible or incompressible, flow are correctly and fully accounted for through the use of these GCL in the closed system of governing equations, (5.10). These geometric conservation laws are also very easy to apply as seen in the examples below; Example 8.7 in particular.

We further remark that the compatibility conditions (5.3b) are not appended to (5.10) for two reasons: (a) they would make the system over-determined and, (b) they can be derived from (5.3a), provided (5.3b) are satisfied initially, which can be ensured easily at the initialization of the computation. Nevertheless, we stress that if (5.3b) were not satisfied, transformation (5.1) would not be single-valued.

## 5.4 Lagrangian gas dynamics equations

For simplicity we consider the 2-D case, when (5.1) simplifies to

$$\left\{\begin{array}{l} dt=d\lambda, \\ dx=Ud\lambda+Ad\xi+Ld\eta, \\ dy=Vd\lambda+Bd\xi+Md\eta. \end{array}\right. \tag{5.11}$$

The governing equations (5.10) also reduce to

$$\frac{\partial E}{\partial \lambda}+\frac{\partial F}{\partial \xi}+\frac{\partial G}{\partial \eta}=0, \tag{5.12}$$

where

$$E=\begin{pmatrix} \rho J \\ \rho J u \\ \rho J v \\ \rho J e \\ A \\ B \\ L \\ M \end{pmatrix}, \quad F=\begin{pmatrix} \rho X \\ \rho X u+pM \\ \rho X v-pL \\ \rho X e+p(uM-vL) \\ -U \\ -V \\ 0 \\ 0 \end{pmatrix}, \quad G=\begin{pmatrix} \rho Y \\ \rho Y u-pB \\ \rho Y v+pA \\ \rho Y e+p(vA-uB) \\ 0 \\ 0 \\ -U \\ -V \end{pmatrix}, \tag{5.13}$$

with $J=AM-BL$, $X=(u-U)M-(v-V)L$ and $Y=(v-V)A-(u-U)B$. The first four equations in (5.13) are the physical conservation laws and the last four the geometric conservation laws.

In the special case of Lagrangian coordinates, (5.12) simplifies to

$$\frac{\partial E}{\partial \lambda}+\frac{\partial F}{\partial \xi}+\frac{\partial G}{\partial \eta}=0, \tag{5.14}$$

where

$$E=\begin{pmatrix} \rho J \\ \rho J u \\ \rho J v \\ \rho J e \\ A \\ B \\ L \\ M \end{pmatrix}, \quad F=\begin{pmatrix} 0 \\ pM \\ -pL \\ p(uM-vL) \\ -u \\ -v \\ 0 \\ 0 \end{pmatrix}, \quad G=\begin{pmatrix} 0 \\ -pB \\ pA \\ p(vA-uB) \\ 0 \\ 0 \\ -u \\ -v \end{pmatrix}. \tag{5.15}$$

**Remark 5.1.** This system of Lagrangian gas dynamics equations was given in conservation PDE form for the first time for 2-D flow in [1], and for 3-D flow in [2]. It is this conservation form PDE that provides a foundation for designing schemes that are moving mesh schemes in Eulerian space. Indeed, the first such scheme has just been proposed by Despres and Mazeran [45,46], who ingeniously incorporate the remaining compatibility conditions

$$\frac{\partial A}{\partial \eta}=\frac{\partial L}{\partial \xi}, \quad \frac{\partial B}{\partial \eta}=\frac{\partial M}{\partial \xi}, \tag{5.16}$$

which they call free divergence constraints, to re-write Eq. (5.14) into a canonical form.

**Remark 5.2.** Whereas system (5.12) is hyperbolic like the Eulerian case, the Lagrangian system (5.14) is found in [1] to be only weakly hyperbolic, meaning that while all eight eigenvalues are real, there does not exist a complete set of eight linearly independent eigenvectors. This finding is also confirmed in [45]†. It also holds for 3-D flow [2], and implies that Lagarangian system of gas dynamics is not equivalent to Eulerian (for the special case of steady flow, this finding was first reported in [7]). In particular, the Cauchy problem for weakly hyperbolic systems is well posed only in a weaker sense, compared with hyperbolic systems. This seems surprising, in view of the fact that Eulerian system of gas dynamics equations is long known to be hyperbolic. However, it has been convincingly demonstrated [45] that Lagrangian re-formulation of any Eulerian hyperbolic system will generically lead to a weakly hyperbolic system; the 1-D case, Eq. (2.3), being just an exception – it happens to be hyperbolic.

## 6 Properties of the unified coordinates

We shall call the system of coordinates $(\lambda,\xi,\eta,\zeta)$ in (5.1) unified in the sense that it unifies the Eulerian system when $\vec{Q}=0$ and the Lagrangian when $\vec{Q}=\vec{q}$, and also in the sense that the system of governing equations, (5.10), unites the geometrical conservation laws with the physical ones. We shall now prescribe conditions to be satisfied by the mesh velocity $\vec{Q}$ so as to give the unified coordinates useful properties.

Consider the 2-D case first. In this case there are two arbitrary functions, $U$ and $V$, and we can prescribe two requirements:

**(A)** Coordinate $\eta$ shall be a material coordinate, meaning

$$\frac{D_{\vec{q}}\eta}{Dt}=0. \tag{6.1}$$

Together with $\frac{D_{\vec{Q}}\eta}{Dt}=0$, we get

$$(v-V)A=(u-U)B. \tag{6.2}$$

(Eq. (6.2) may be used to simplify (5.12)).

Observations:

- Contact lines, being material lines, must coincide with coordinate lines and, therefore, can be resolved sharply.
- As the body surface is a material line, condition (6.2) guarantees that the mesh in UC is automatically a body-fitted mesh at all time. This provides the basis for automatic mesh generation (see Section 7.2).

†Depending on how the differential constraints are used in the governing equations, one can get different numbers of linearly independent eigenvectors associated with the zero eigenvalue. However, in all cases there does not exist a complete set of eight linearly independent eigenvectors and the system is only weakly hyperbolic.

- A material interface (including a free surface) corresponds to $\eta$=const and thus can be resolved sharply.
- $\eta(x,y,t)$ is a level set function, hence there is no need to introduce an extra level set function when using the level set method.

**(B)** Mesh angles, and hence mesh orthogonality, shall be preserved during the $\lambda$-marching computation. This means

$$\frac{\partial}{\partial\lambda}\cos^{-1}\left[\frac{\nabla\xi\cdot\nabla\eta}{|\nabla\xi||\nabla\eta|}\right]=\frac{\partial}{\partial\lambda}\cos^{-1}\left[\frac{AL+BM}{\sqrt{A^2+B^2}\sqrt{L^2+M^2}}\right]=0, \tag{6.3}$$

which, after eliminating $V$ from (6.2) and using the geometric conservation laws of (5.12), yields an ODE for $U$

$$\frac{\partial U}{\partial\eta}+P(\eta;\lambda,\xi)U=Q(\eta;\lambda,\xi), \tag{6.4a}$$

$$U(\eta;\lambda,\xi)=U_0(\lambda,\xi) \quad \text{at } \eta=\eta_0, \tag{6.4b}$$

where $U_0(\lambda,\xi)$ is arbitrary, and

$$\begin{aligned}
P(\eta;\lambda,\xi)&=\frac{S^2}{T^2J}\left(A\frac{\partial B}{\partial\xi}-B\frac{\partial A}{\partial\xi}\right)-\frac{L}{AJ}\left(A\frac{\partial B}{\partial\eta}-B\frac{\partial A}{\partial\eta}\right),\\
Q(\eta;\lambda,\xi)&=\frac{S^2A}{T^2J}\left(B\frac{\partial u}{\partial\xi}-A\frac{\partial v}{\partial\xi}\right)+\frac{L}{J}\left(A\frac{\partial v}{\partial\eta}-B\frac{\partial u}{\partial\eta}\right)+uP(\eta;\lambda,\xi),\\
S^2&=L^2+M^2, \quad T^2=A^2+B^2.
\end{aligned}$$

We note that the mesh-angle preserving condition is not unique, but can be replaced by any reasonable condition, e.g. preserving the Jacobian $J$, which would be particularly suitable for incompressible flow computation.

**Remark 6.1.** Condition (6.1) alone retains the advantages of Lagrangian coordinates: sharp resolution of contacts. If we further require

$$\frac{D_{\vec{q}}\xi}{Dt}=0, \tag{6.5}$$

then coordinates $(\xi,\eta)$ are Lagrangian and computation may break down due to large cell deformation. This can be prevented by the mesh-angle preserving condition (6.3), or by the Jacobian-preserving condition. The UC system may be regarded as a generalization of the Lagrangian system in that we retain only one condition of Lagrangianess, (6.1), while abandon the other one, (6.5), in favor of a mesh-angle preserving condition, or Jacobian preserving condition. The same may be said about the 3-D case (see Remark. 6.2 below)

**Remark 6.2.** In the case of 3-D flow, there are three arbitrary functions $U$, $V$ and $W$, so we should prescribe three requirements: we want $\eta$ and $\varsigma$ to be material coordinates, meaning

$$\frac{D_{\vec{q}}\eta}{Dt}=0 \tag{6.6}$$

and

$$\frac{D_{\vec{q}}\varsigma}{Dt}=0. \tag{6.7}$$

Eq. (6.6) and the second equation of (5.2) combine to give

$$(\vec{q}-\vec{Q})\cdot\vec{J}_2=0. \tag{6.8}$$

Similarly, Eq. (6.7) and the third equation of (5.2) combine to give

$$(\vec{q}-\vec{Q})\cdot\vec{J}_3=0. \tag{6.9}$$

(Eqs. (6.8) & (6.9) may be used to simplify (5.10)). As the third condition, we require that the mesh skewness

$$\kappa=\frac{|\vec{A}|\cdot|\vec{L}|\cdot|\vec{P}|}{\vec{A}\times\vec{L}\cdot\vec{P}}-1=\begin{cases} 0, & \text{orthogonal} \\ \infty, & \text{degenerate}\end{cases} \tag{6.10}$$

be preserved during the $\lambda$-marching computation [2]. Here $\vec{A}=(A,B,C)^T$, $\vec{L}=(L,M,N)^T$, and $\vec{P}=(P,Q,R)^T$. This condition is

$$\frac{\partial\kappa}{\partial\lambda}=0. \tag{6.11}$$

It might seem a possible alternative (to the three conditions (6.6), (6.7) and (6.11)) to use (6.6) and to preserve the two mesh angles during the $\lambda$-marching computation. This would then have the desirable effect that mesh orthogonality can also be preserved for 3-D flow (as in the 2-D case) when the mesh is initially orthogonal. However, this is impossible in general, because it would contradict a theorem of [49] that for steady flow past a body, orthogonal body-fitted mesh is possible if and only if the flow belongs to a special class called complex-lamellar [50], for which $\vec{q}\cdot\nabla\times\vec{q}\equiv 0$. We also note that as in the 2-D case, the condition of skewness-preserving can be replaced by that of Jacobian-preserving.

**Remark 6.3.** For 1-D flow, condition (6.1) alone means the unified coordinate coincides with Lagrangian coordinate.

**Remark 6.4.** In the special case of steady flow, Eqs. (6.6) and (6.7) become $\vec{q}\cdot\nabla\eta=0$ and $\vec{q}\cdot\nabla\varsigma$=0. Hence, fluid velocity vector $\vec{q}$ lies in the direction of the intersection line between the surfaces $\eta(x,y,z)$=const and $\varsigma(x,y,z)$=const. At the same time, the second and the third equations of (5.2) also become $\vec{Q}\cdot\nabla\eta=0$ and $\vec{Q}\cdot\nabla\varsigma=0$, and hence the mesh velocity vector $\vec{Q}$ must lie along the same intersection line. Therefore

$$\vec{Q}=h\vec{q}, \tag{6.12}$$

$h$ being an arbitrary function. This means the unified coordinates move in the direction of the fluid flow but not with its speed, in contrast to the Lagrangian coordinates which move with the fluid in both its direction and speed. This special case was studied in details in [1,2].

# 7 Computation procedure for 2-D flow

## 7.1 Special case: Steady supersonic flow

In this simpler case, the inviscid flow equations are

$$\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vH\end{pmatrix}=0. \tag{7.1}$$

Transformation (5.1) simplifies to

$$\begin{cases} dx=hud\lambda+Ad\xi,\\ dy=hvd\lambda+Bd\xi,\end{cases} \tag{7.2}$$

under which, Eq. (7.1) becomes

$$\frac{\partial}{\partial\lambda}\begin{pmatrix}K\\ H\\ Ku+pB\\ Kv-pA\\ A\\ B\end{pmatrix}+\frac{\partial}{\partial\xi}h\begin{pmatrix}0\\ 0\\ -pv\\ pu\\ -u\\ -v\end{pmatrix}=0, \tag{7.3}$$

where

$$K=\rho(Bu-Av),\quad e=\frac{1}{2}(u^2+v^2)+\frac{1}{\gamma-1}\frac{p}{\rho},\quad H=e+\frac{p}{\rho}.$$

The free function $h$ is chosen [51] to ensure mesh orthogonality:

$$Au+Bv=0. \tag{7.4}$$

Similar to the 1-D case we use shock-adaptive Godunov scheme [6,20], where the shock is fitted, using the Riemann solution with no extra cost. We can then replace the energy conservation equation by the entropy conservation equation, which holds in smooth flow region. Thus [19,51]

$$\frac{\partial}{\partial\lambda}\begin{pmatrix}K\\ H\\ p/\rho^\gamma\\ Ku+pB\\ Kv+\frac{pvB}{u}\end{pmatrix}+\frac{\partial}{\partial\xi}h\begin{pmatrix}0\\ 0\\ 0\\ -pv\\ pu\end{pmatrix}=0. \tag{7.5}$$

Eq. (7.5) is solved by space-marching in $\lambda$ [52]. To summarize, with the use of the unified coordinates, we have reduced, computationally, the problem of 2-D steady supersonic flow to that of 1-D unsteady flow, with $\lambda$ taking the place of time variable. For 2-D steady supersonic flow the space marching method based on UC is most robust, accurate and efficient, as shown by Examples 8.1 and 8.2.

We note that in Example 8.2, the zero normal velocity condition must be applied at the airfoil surface. As the surface is a streamline, it corresponds to a coordinate line and, therefore, it is very easy to satisfy the boundary condition there. This is usually done by solving a boundary Riemann problem there, with the flow in the fictitious cell on the other side of the surface forming a mirror reflection with respect to the body surface.

## 7.2 General case

In the general case when the flow is unsteady (Examples 8.5 and 8.7) or is steady but has a subsonic region (Examples 8.3 and 8.4), we compute the time-dependent equations, (5.12), by time marching in $\lambda$. The computation procedure for uniform flow past a body is illustrated by a Mach 0.8 flow past a NACA 0012 airfoil (Example 8.3 below) as follows:

**Initialization stage – automatic generation of body-fitted mesh in a computational window**

Given the grid sizes, $\Delta x$ and $\Delta y$, and the number of cells, $M \times N$, in the window.

1. Begin with a column of $N$ orthogonal cells, representing the given uniform flow in the $x$-direction (Fig. 2a). This gives the initial values of $(A,B,L,M)=(1,0,0,1)$. We also take $(U,V)=(u,v)$ initially.
2. Compute the solution to (5.12) by marching in time $\lambda$ using dimensional splitting: splitting into two 1-D systems in $\lambda\xi$ and $\lambda\eta$. Each of them is solved using the standard Godunov/MUSCL scheme with the minmod limiter. (Details are as follows. To update the solution from time $n$ to time $n+1$: (a) Solve the first four equations (the physical conservation laws) of (5.12) for $(\rho,p,u,v)$, keeping $A,B,L,M,U$ and $V$ fixed at time $n$ level. (b) Use this updated values to solve (6.4) for $U$ with $U_0(\lambda,\xi)$=const at the surface, and then solve (6.2) for $V$ at time $n+1$. (c) Use these updated values of $U$ and $V$ to update $(A,B,L,M)$ at time $n+1$ by integrating the geometric conservation laws). In all outer boundaries of the computational region at every time step, we apply the characteristic boundary conditions.
   After one time step $\Delta\lambda$, this column of cells moves to the right by $U\Delta\lambda$.
3. After several time steps when the initial column of cells has moved to the right by a distance equal to $\Delta x$, add one new column of cells on the left that is identical to the initial column.
4. Repeat this process of adding cell columns on the left of the computational region until the leading column of cells meets the body surface and also further down (Fig. 2b) we impose the boundary condition of zero normal velocity on the body surface, by solving a boundary Riemann problem as explained in Section 7.1.

5. Continue this process until after the columns of cells cover the whole body surface and further downstream, when we have $M$ columns of cells in the window (Fig. 2c).

This completes the initialization stage, and we now have an airfoil-fitted mesh and a flow field around it in the window of $M \times N$ cells. The flow-generated mesh (Fig. 2d) is body-fitted and orthogonal as predicted. It is also fairly uniform in the $x$-direction, because in solving (6.4) we have specified uniform data for $U$ at the surface. The flow field computed (Fig. 2e) so far is, however, only a very rough approximation to the correct one, partly because it has not reached the steady state and partly because the downstream boundary condition used in the transient times, e.g., in Fig. 2b, are obviously incorrect as the computational regions at those times have not yet reached the full window.

To compute further, one could use the body-fitted orthogonal mesh generated so far to perform an Eulerian computation with the associated flow field as an initial solution. This can be easily done by putting $U=V=0$ (thus without solving (6.4) & (6.2)) in the subsequent iterations towards a steady state. In this way, the UC approach plays the role of mesh generation for Eulerian computation.

An alternative and better way is to continue the UC computation in the

**Main stage – iteration with flow-adjusted mesh**

6. To iterate the computation towards a steady state, whenever we add a new column of cells on the left we also simultaneously delete the right-most column of cells from the computation window, thus keeping the window in the same size. To improve the solution, we may also use the information of the flow field at every time step, e.g., the surface pressure gradient, to adjust the initial data of $U$ at the body surface, $\eta=\eta_0$, in solving (6.4) so that the mesh is refined in regions of high pressure gradient (Fig. 2f). We note that the flow-adjusted refined mesh remains orthogonal. It also looks similar to those obtained by mesh re-distribution in Euelrian computation. But there are differences: mesh re-distribution typically requires generating another mesh by solving an elliptic equation. In doing so, conservation properties must be satisfied in the interpolations of geometry and flow variables between the two meshes. These issues do not arise in our flow-adjusted mesh approach as we need only one mesh; the only modification is to specify the initial data for $U$ at $\eta=\eta_0$ in solving (6.4) by using the information of the pressure gradient.

We note that in this example the flow is symmetric and, consequently, the mesh on the upper surface and that on the lower surface will arrive at the trailing edge simultaneously to re-connect for further computation downstream. If the flow is not symmetric, as in Examples 8.4 and 8.5 below, we choose the initial conditions, (6.4b), on the upper and lower surfaces such that the two meshes arrive at the trailing edge simultaneously to connect together for further computation downstream. This is easy to do: since $U_0(\lambda,\xi)$ is the $x$-component of the mesh velocity at the surface we only need to require that it takes the same value at the same position $x$ along the surfaces, until the meshes on the upper and lower surfaces reach the trailing edge.

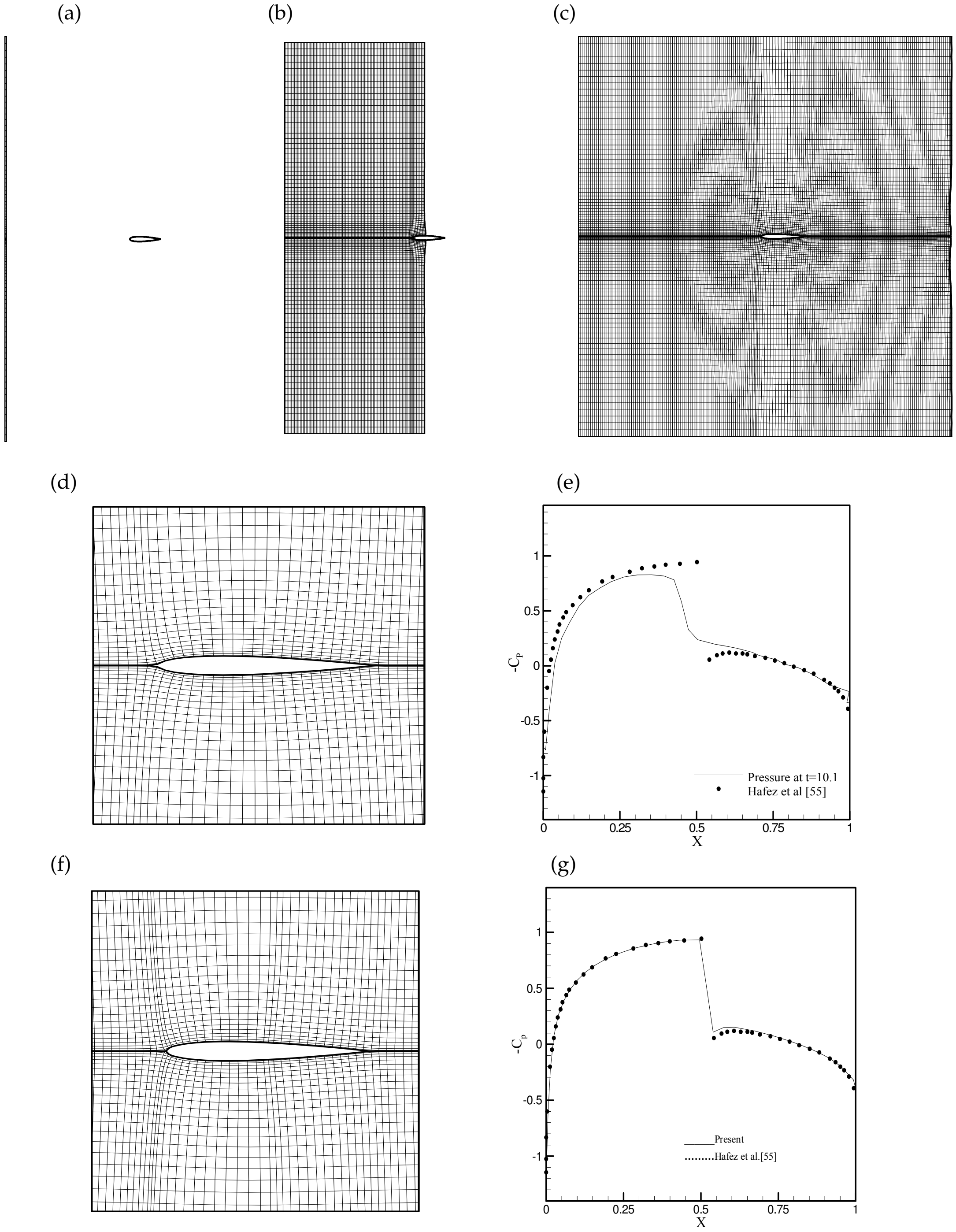
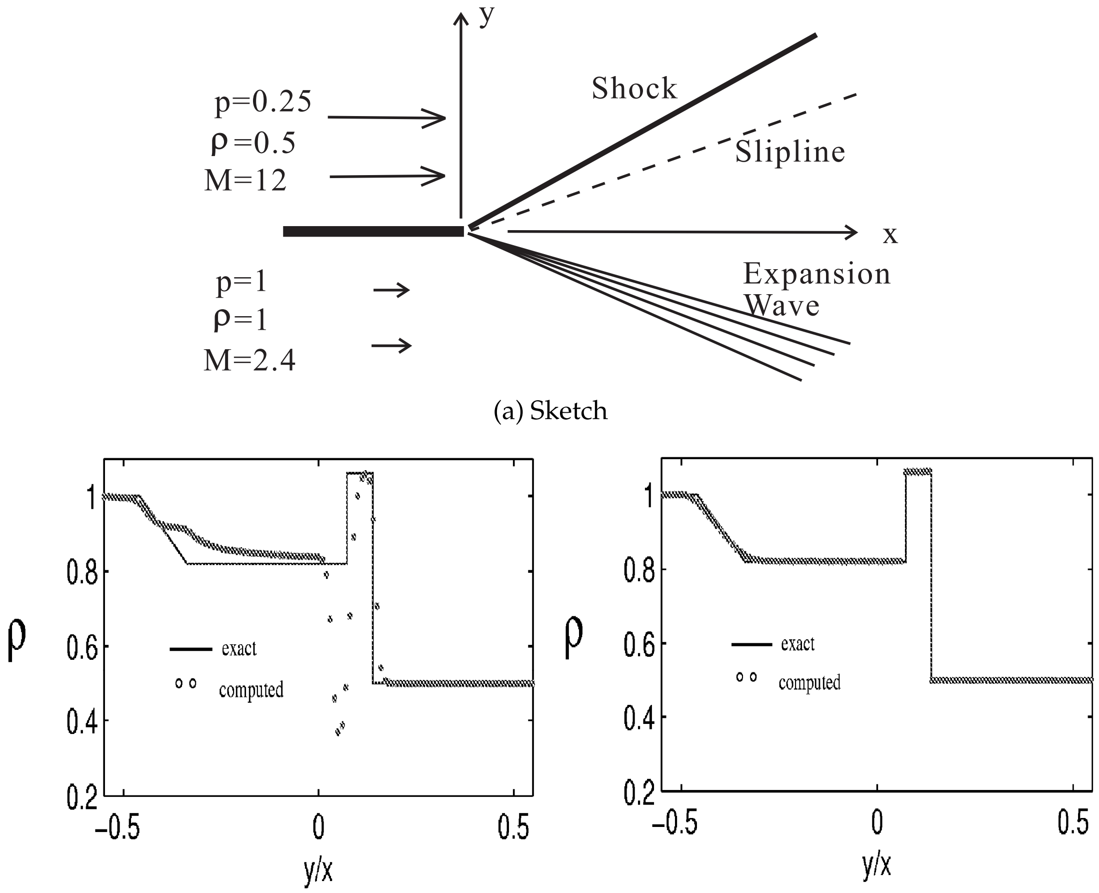


Figure 2: Mach 0.8 flow past a NACA 0012 airfoil at zero angle of attack. (a)-(c) flow-generated meshes, (d) close view of preliminary mesh, (e) preliminary surface pressure, (f) close view of final mesh, (g) final surface pressure.



(a) Sketch

(b) Eulerian Computation (Godunov-MUSCL)

(c) UC Computation

Figure 3: Numerical solutions to a 2-D steady Riemann problem. (a) sketch, (b) density, Eulerian computation, (c) density, UC computation with shock-adaptive Godunov scheme.

# 8 Typical results of 2-D computation

In all examples below $\gamma$=1.4 is used, unless otherwise stated.

### Steady supersonic flow

**Example 8.1.** This is a 2-D steady Riemann problem (see Fig. 3a for a sketch of the flow), and the flow is computed as explained in Section 7.1. Eulerian computation (Fig. 3b) gives very poor resolution of the contact discontinuity, which also affects the accuracy of the smooth flow, ie, the Prandtl-Meyer expansion fan. We note that gas kinetic BGK scheme produces similar results [53, Fig. 3.6]. The UC computation [19, 51] gives sharp resolution of both shock and contact discontinuities (Fig. 3c).

**Example 8.2.** This is a supersonic flow of $M_\infty$=1.745 past a diamond-shape airfoil with semi-apex angle of 7.5° placed at an angle of attack of 10° (Fig. 4a). The space-marching computation method was explained in Section 7.1. The flow-generated meshes [52] at different $\lambda$ are plotted in Figs. 4b-e, and the computed surface Mach numbers are plotted in Fig. 4f. Since the supersonic flows on the upper and lower surface are independent of each other, they are computed separately in this example. The computation took very little time (1.8s on a P4, 2.8GHz PC machine), yet the computed results are identical to the exact solution.

In Eulerian computation, a body-fitted mesh has to be generated prior to computing the flow. We also note that in this example, a space marching method in Eulerian coordinate system fails as the $x$-component (in the free stream direction) of the flow behind the shock is subsonic. A time marching method has to be employed whose computing time is more than three orders of magnitude longer (2,393s, same machine). Yet the computed shock and the Prandtl-Meyer expansions are smeared (Fig. 4g).

### Steady flow with a subsonic region

**Example 8.3.** Fig. 2 shows a sample computation [54] for a Mach 0.8 steady flow past a NACA 0012 airfoil at zero angle of attack. Note the orthogonality of the flow-generated meshes (Figs. 2d, 2f). Note also that despite the coarse mesh used, our result (Fig. 2g) is in good agreement with [55], which uses the potential flow approximation.

**Example 8.4.** Fig. 5 shows a sample computation [54] for a Mach 2.2 steady flow of air-SF6 past a NACA 0012 airfoil at an angle of attack of 8°. It is seen that without any special treatment, the contact line between the two fluids is sharply resolved.

### Unsteady flow

**Example 8.5.** This is an unsteady supersonic flow, $M_\infty$=3.0, past a diamond-shape airfoil with 10° vertex angle which is oscillating about its vertex according to $\theta = 2^\circ \sin 30\pi t$, where $\theta$ is the instantaneous pitching angle. The computed flow-generated meshes [48] at different times are plotted in Fig. 6. It is seen that at all times of the oscillation, the flow-generated meshes are body-fitted and are almost orthogonal, although the mesh on the expansion side of the airfoil is coarsened.

### Viscous flow

For viscous flow computation, the viscous terms in the Navier-Stokes equations are discretized centrally to model the physics of diffusion, but otherwise the computation procedure is similar to that of inviscid flow.

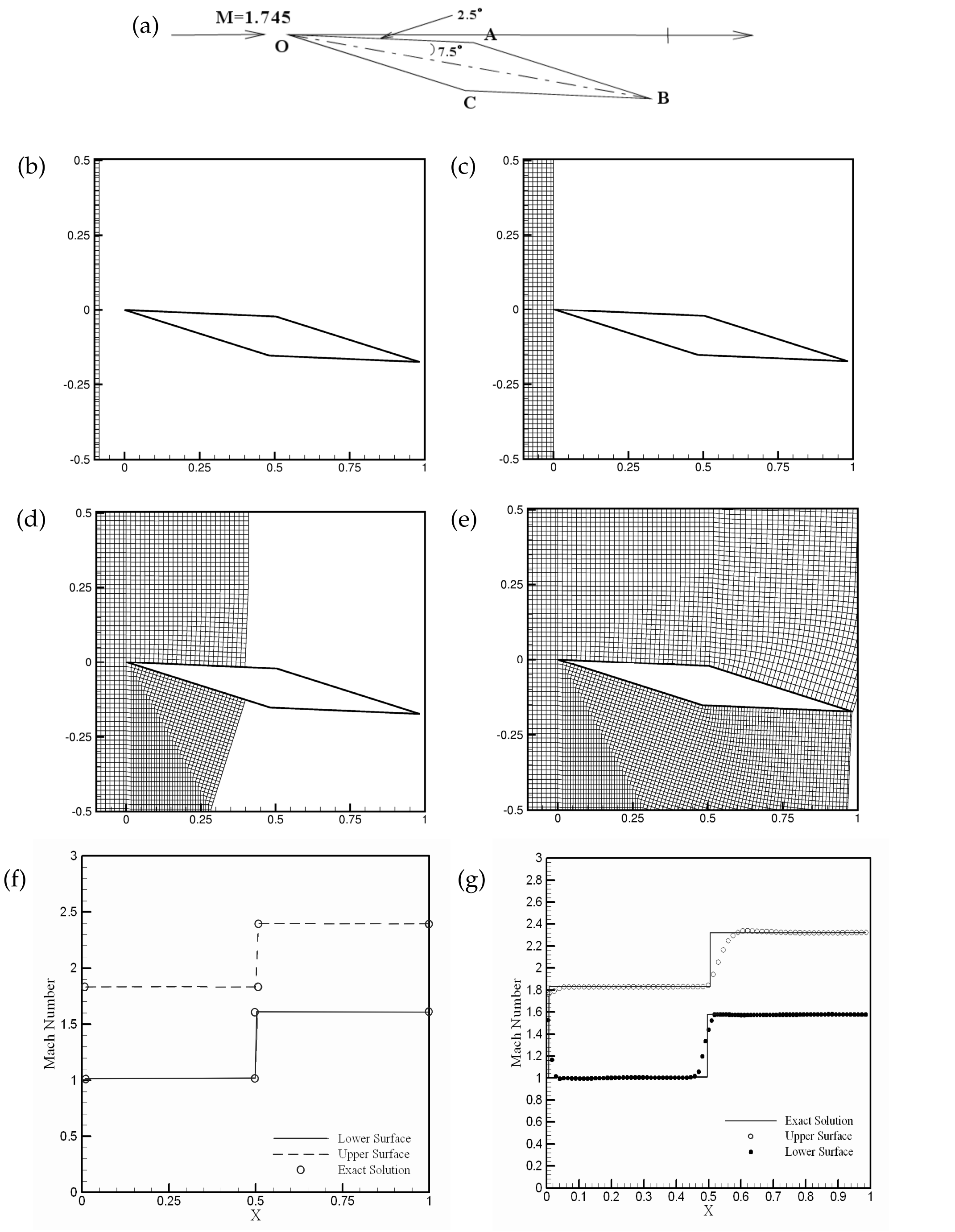


Figure 4: Steady supersonic flow past a diamond-shape airfoil by space-marching method. (a) sketch, (b) - (e) flow-generated meshes at different $\lambda$, (f) computed surface Mach number, 120 cells, computing time: 1.8s (P4, 2.8 GHz) (g) Eulerian computation (5th-order WENO scheme), 100×200 cells, computing time: 2,395s.

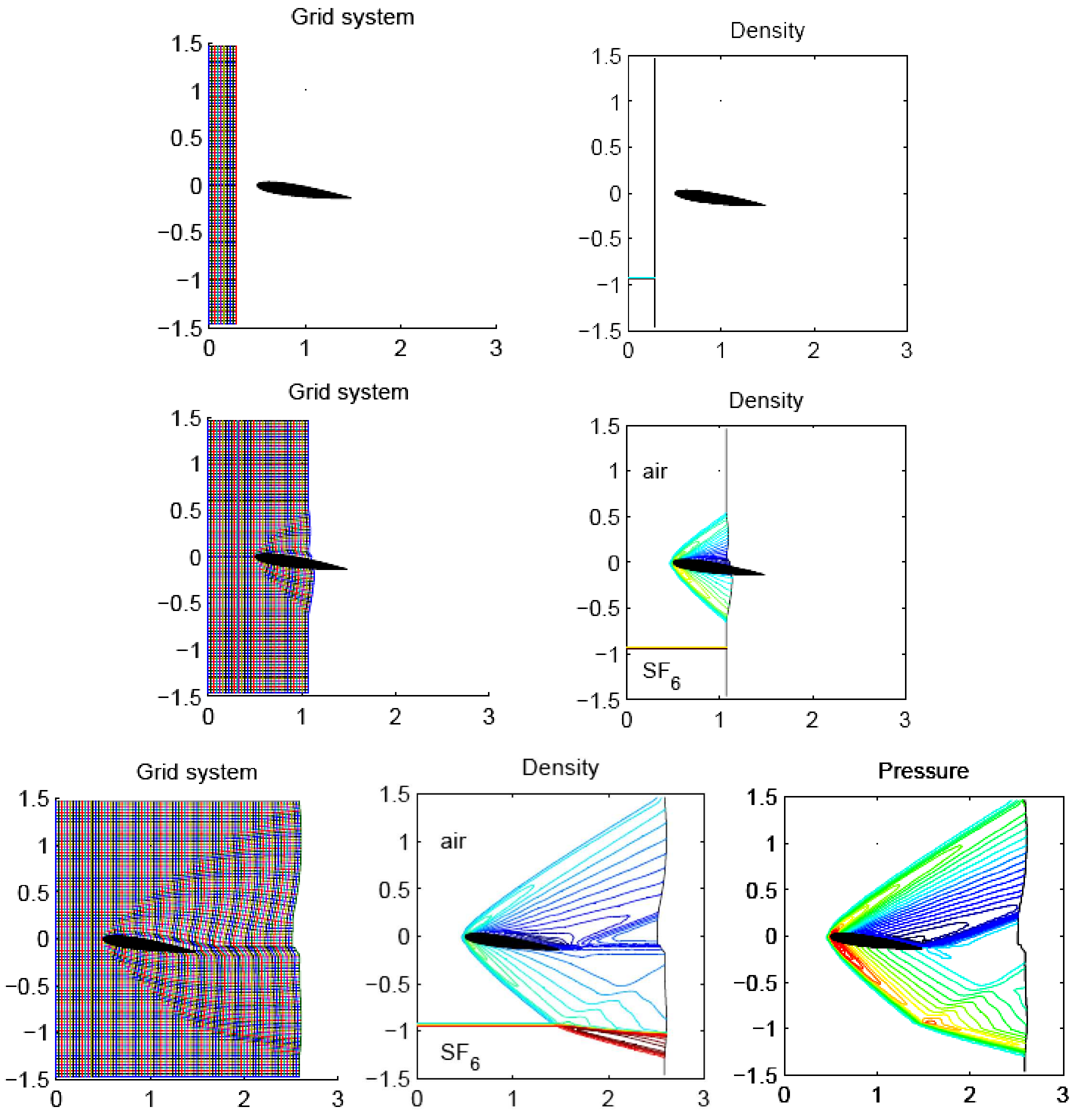


Figure 5: Mach 2.2 flow of air-SF6 over a NACA 0012 airfoil at 8° angle of attack. Flow-generated meshes and density (and pressure) contours at different times.

**Example 8.6.** This is a problem of shock/shock interaction in viscous flow [10]. Two oblique shocks are generated by two wedges in a channel, causing deflections of the flow by an angle of 5° (lower wedge) and 15° (upper wedge). The upstream Mach number is $M_\infty$=4.

Both inviscid and viscous computations are presented. For the viscous flow case, the Reynolds number is $R_e$=10 and Prandtl number $P_r$=0.72. Fig. 7 presents the inviscid computation, where it is seen that Eulerian computation smears the contact lines completely,

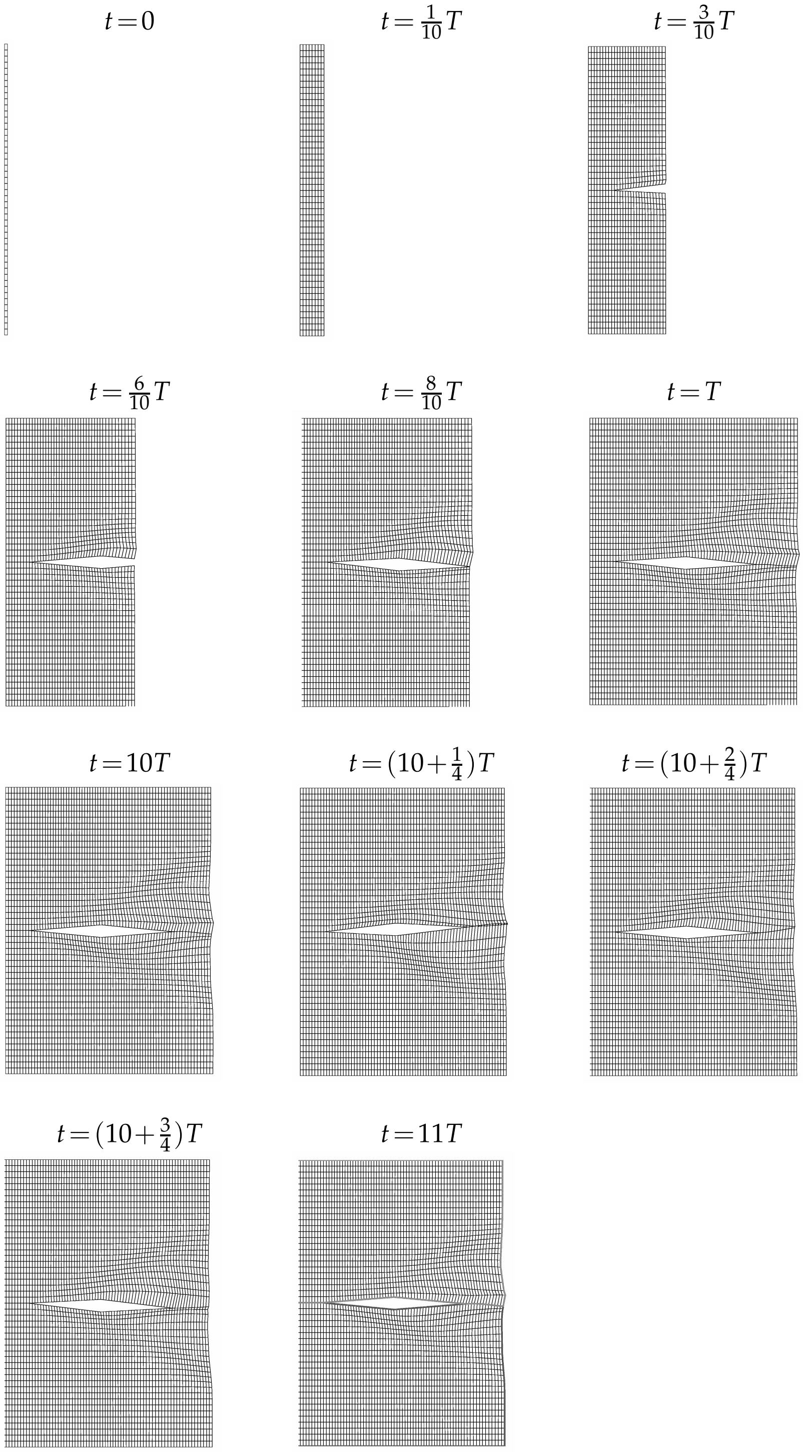


Figure 6: Flow-generated meshes for oscillating diamond-shape airfoil. Apex angle=10°, pitching motion about the apex: $\theta(t)=2^\circ\sin 2\pi t/T$, the period of oscillation $T=2\pi/30$, free stream Mach number $M$=3.0.

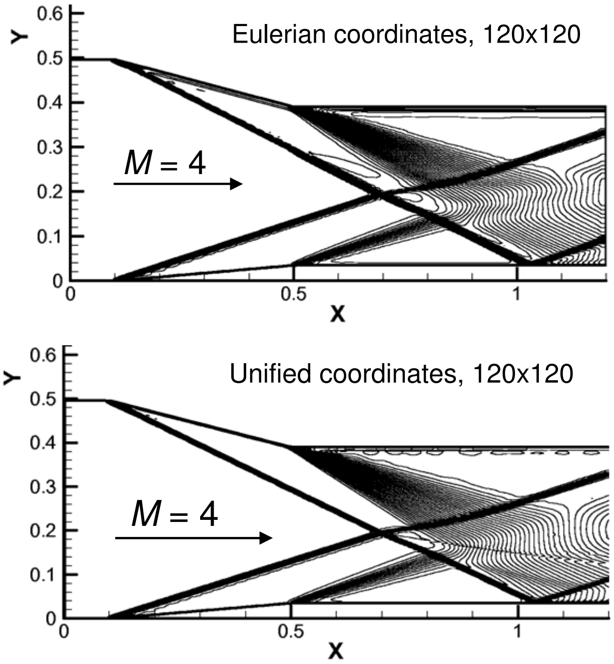


Figure 7: Computed Mach contours in shock-shock interaction: Inviscid flow.

whilst UC computation resolves it sharply. Fig. 8a shows the viscous flow computation and compares with Eulerian computation (Fig. 8b). Both use a mesh of 120×120 and the two results are similar. But Eulerian computation produces two straight leading shocks, similar to the inviscid case, whereas UC computation produces curved leading shocks, due to the displacement effect of viscosity. It is interesting to notice that with a finer mesh of 240×240, Eulerian computation also produces curved leading shocks (see Fig. 8c), similar to the UC results.

**Example 8.7.** This is the problem of unsteady incompressible aerodynamics of a freely falling plate. There is rich dynamic behavior, such as fluttering and tumbling (Fig. 10a). Numerical and experimental studies have been presented [56] and, more recently, Jin and Xu [57] have elegantly used the unified coordinates formulation, together with a gas-kinetic BGK solver [58], to obtain very accurate results. In [57] the computational mesh is fixed rigidly with the plate (Fig. 9), hence the mesh velocity is determined according to the translational and rotational velocity of the plate which, in turn, is determined by the unsteady aerodynamic forces and the gravitational force according to Newton's laws of motion. In UC formulation, (5.12), the effects of the mesh movement are fully accounted for, while the effects of viscosity are fully included in the gas-kinetic BGK solver. Computed trajectories of the falling rectangular plate are plotted in Fig. 10 and are seen to agree well with the experimental measurements in [56]. Some computed vorticity fields of the same motion are also given in Fig. 11.

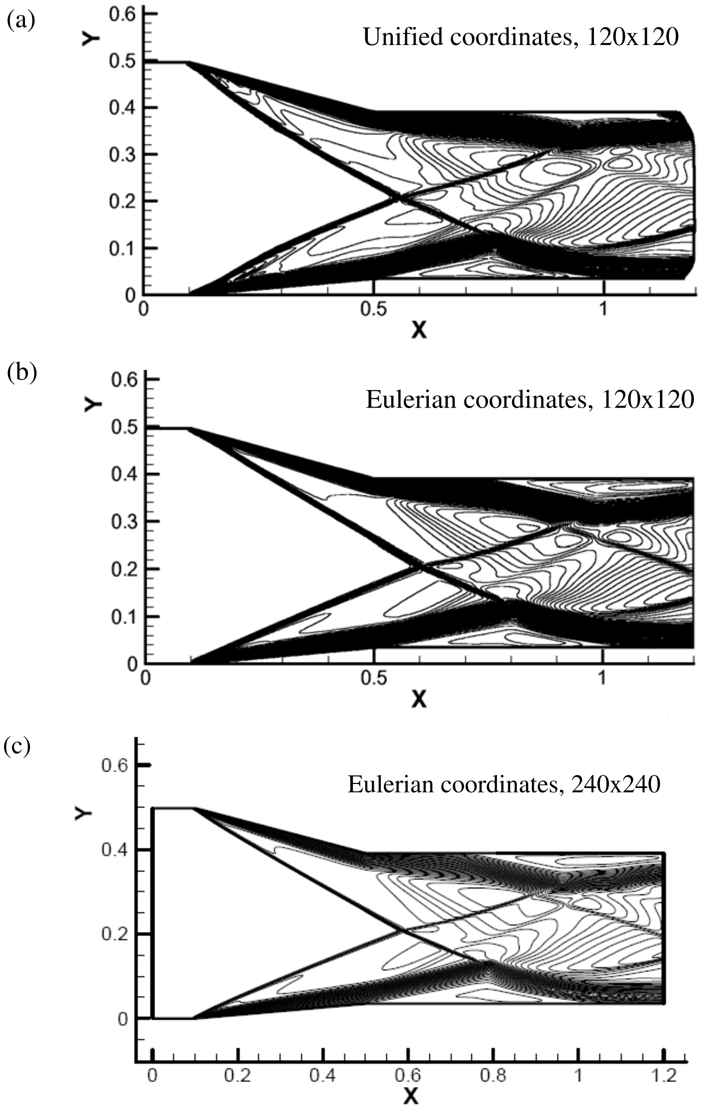


Figure 8: Computed Mach contours in shock-shock interaction: $M=4$, $Re=10^4$, $P_r=0.72$.

We remark that the well-established method of mesh adaptation [59, 60] can also be easily applied in the UC computations in the above examples to further improve their accuracies, and this has been clearly demonstrated in [53].

## Other possible applications

As the unified coordinate system resolves material interfaces and free surfaces sharply, it can be applied with advantage to problems involving free surfaces (such as water waves)

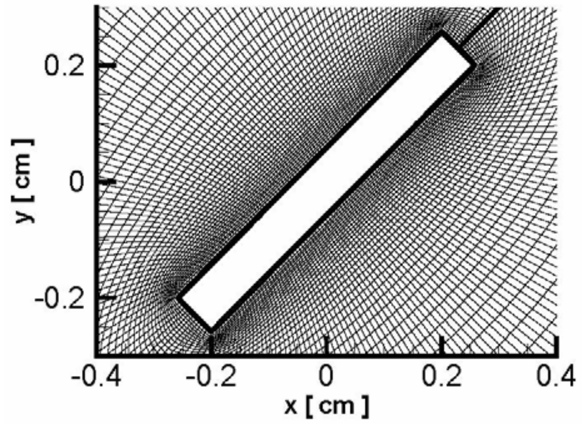


Figure 9: Fixed computational mesh around a rectangular plate.

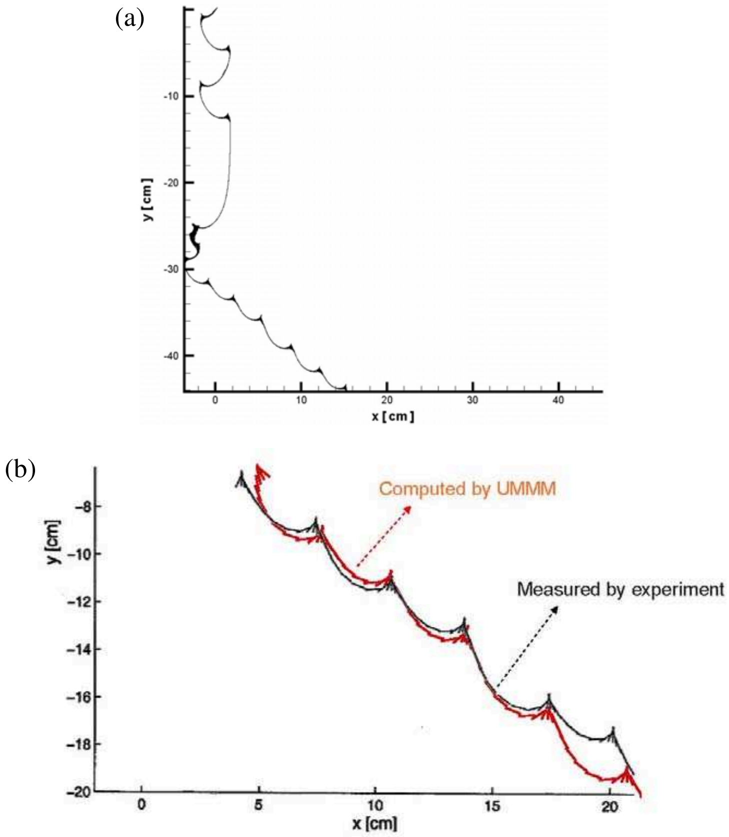


Figure 10: Trajectories of the falling rectangular plate. (a) Complete trajectory, (b) The tumbling phase: experiment (black) and computational (red).

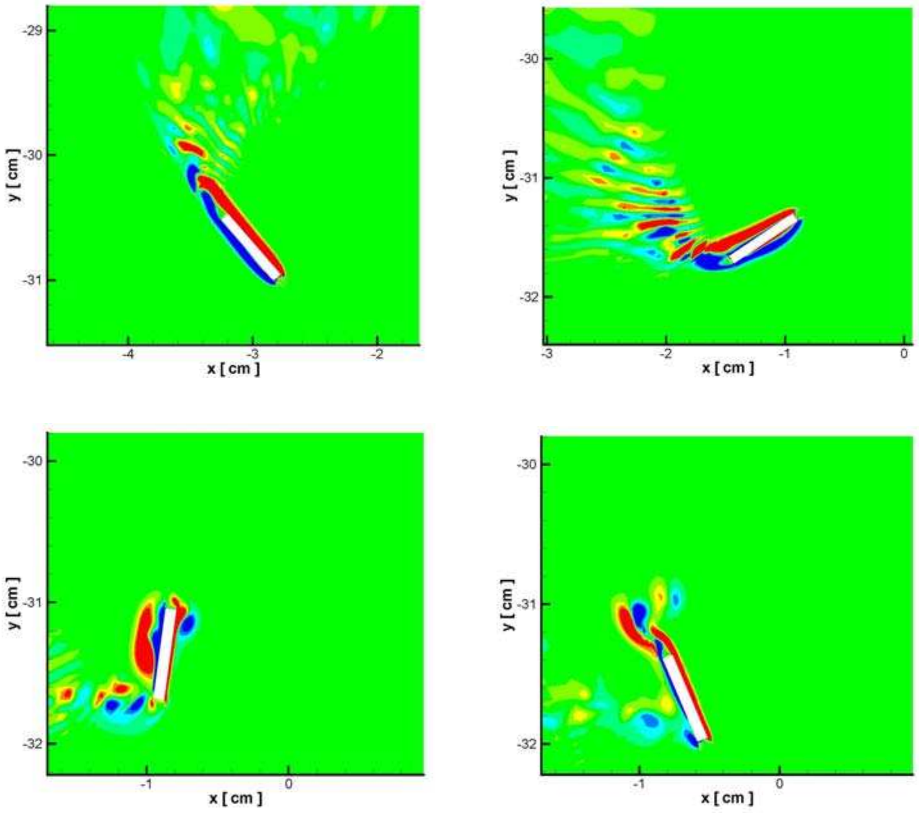


Figure 11: Computed vorticity fields of a falling rectangular plate at four instants during a full rotation.

and those of fluid/solid interactions, for which Eulerian methods would encounter some difficulties of unknown boundary and of small cut-cells [61,62]. Successes of application of the UC system to problems of this type have just been reported in [63], which also stresses the importance of the combined physical and geometrical conservation laws in the accurate computation of viscous flows using gas kinetic schemes. Other applications may include debris flows and blood flows in arteries.

## 9 Conclusions

A unified coordinate system, $(\lambda,\xi,\eta,\zeta)$, moving at velocity $\vec{Q}=(U,V,W)$ has been constructed in such a way that $\eta$ and $\zeta$ are material coordinates, while $\lambda$ is a time variable. The system has been shown to possess the following properties: (a) It unifies Eulerian and Lagrangian system. It also unites the physical and geometrical conservation laws to form a closed system; consequently, the effects of arbitrarily moving mesh on the flow are fully accounted for. (b) This system of governing equations is written in conservation

PDE form, making it as easy to compute as in Eulerian coordinates; in particular, it provides a foundation for designing Lagrangian shock-capturing schemes that are moving mesh schemes in Eulerian space. (c) The Lagrangian system of gas dynamics equations in two- and three-dimension are shown to be only weakly hyperbolic, in direct contrast to the Eulerian system which is hyperbolic; they are thus not equivalent to each other. (d) It resolves contact discontinuities (including material interfaces and free surfaces) as sharply as in Lagrangian system. (e) It provides a basis for automatic mesh generation; a body-fitted mesh is generated, automatically and simultaneously, by the flow being computed. (f) The unified mesh is orthogonal for 2-D flow and is schewness-preserving for 3-D flow. Several examples of 2-D flows are given to confirm these properties.

## Acknowledgments

This research was partially supported by a grant (HKUST6138/01P) from the Research Grants Council of Hong Kong. The author is indebted to Drs. C Jin and K Xu for the use of Figs. 9-11 from their unpublished paper [57], he also thanks the two referees for their valuable comments/suggestions, which have led to improvements of the paper.

### References

[1] W. H. Hui, P. Y. Li and Z. W. Li, A unified coordinate system for solving the two-dimensional Euler equations, J. Comput. Phys., 153 (1999), 596.

[2] W. H. Hui and S. Kudriakov, A unified coordinate system for solving the three-dimensional Euler equations, J. Comput. Phys., 172 (2001), 235.

[3] W. H. Hui and S. Kudriakov, Computation of shallow water waves using the unified coordinates, SIAM J. Sci. Comput., 23 (2002), 1615.

[4] C. Y. Loh and W. H. Hui, A new Lagrangian method for steady supersonic flow computation Part I: Godunov scheme, J. Comput. Phys., 89 (1990), 207.

[5] W. H. Hui and C. Y. Loh, A new Lagrangian method for steady supersonic flow computation, Part II: Slip-line resolution, J. Comput. Phys., 103 (1992), 450.

[6] W. H. Hui and C. Y. Loh, A new Lagrangian method for steady supersonic flow computation, Part III: Strong shocks, J. Comput. Phys., 103 (1992), 465.

[7] W. H. Hui and Y. C. Zhao, A generalized Lagrangian method for solving the Euler equations, in: A. Donato and F. Oliveri (Eds.), Nonlinear Hyperbolic Problems: Theoretical, Applied, and Computational Aspects, Vieweg, 1993, pp. 336.

[8] W. H. Hui and G. P. Zhao, Capturing contact discontinuities using the unified coordinates, in: K. J. Bathe (Ed.), Proceedings of the 2nd MIT Conference on Computational Fluid and Solid Mechanics, vol. 2, 2003, pp. 2000-2005.

[9] Z. N. Wu, A note on the unified coordinate system for computing shock waves, J. Comput. Phys., 180 (2002), 110.

[10] W. H. Hui, Z. N. Wu and B. Gao, Preliminary extension of the unified coordinate approach to computation of viscous flows, J. Sci. Comput., 30 (2007), 301. (See also B. Gao and Z. N. Wu, Physically related coordinate system for compressible flow, Mod. Phys. Lett. B, 19 (2005), 1455.)

[11] C. Y. Loh and M. S. Liou, A new Lagrangian method for solving the 2-D steady flow equations for real gas, J. Comput. Phys., 104 (1993), 150.

[12] C. Y. Loh and M. S. Liou, A new Lagrangian method for three-dimensional steady supersonic flows, J. Comput. Phys., 113 (1994), 224.

[13] M. S. Liou, An extended Lagrangian method, J. Comput. Phys., 118 (1995), 294.

[14] C. Y. Loh, M. S. Liou and W. H. Hui, An investigation of random choice method for 3-D steady supersonic flow, Int. J. Numer. Meth. Fluids, 29 (1999), 97.

[15] C. Y. Loh and W. H. Hui, A new Lagrangian method for time-dependent inviscid flow computation, SIAM J. Sci. Comput., 22 (2001), 330.

[16] P. Jia, S. Jiang and G. P. Zhao, Two-dimensional compressible multi-material flow calculations in a unified coordinate system, Comput. Fluids, 35 (2006), 168.

[17] D. H. Wagner, Equivalence of Euler and Lagrangian equations of gas dynamics for weak solutions, J. Differ. Equations, 68 (1987), 118.

[18] W. H. Hui and S. Koudriakov, Role of coordinates in the computation of discontinuities in one-dimensional flow, Comput. Fluid Dyn. J., 8 (2000), 495.

[19] W. H. Hui and S. Koudriakov, On contact overheating and other computational difficulties of shock-capturing methods, Comput. Fluid Dyn. J., 10 (2001), 192.

[20] C. Y. Lepage and W. H. Hui, A shock-adaptive Godunov scheme based on the generalized Lagrangian formulation, J. Comput. Phys., 122 (1995), 291.

[21] D. Serre, Systems of Conservation Laws, Cambridge University Press, 1999.

[22] F. H. Harlow, A Machine calculation method for hydrodynamic problems, Technical Report LAMS-1956, Los Alamos Scientific Laboratory, 1955.

[23] A. A. Amsden, The Particle-in-cell methods for the calculation of the dynamics of compressible fluids, Technical Report LA-3466, Los Alamos Scientific Laboratory, 1966.

[24] F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids, 8 (1965), 2182.

[25] F. H. Harlow, J. P. Shannon and J. E. Welch, The MAC method: A computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces, Technical Report LA-3425, Los Alamos Scientific Laboratory, 1965.

[26] A. J. Chorin, Numerical solution of Navier-Stokes equations, Math. Comput., 22 (1968), 745.

[27] F. H. Harlow and A. A. Amsden, A numerical fluid dynamics calculation for all flow speeds, J. Comput. Phys., 8 (1971), 197.

[28] J. G. Trulio, Theory and structure of the AFTON codes, Technical Report No. AFWL-TR-66-19, Air Force Weapons Laboratory, Kirtland, 1966.

[29] C. W. Hirt, A. A. Amsden and J. L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys., 14 (1974), 227.

[30] W. Pracht, Calculating three-dimensional fluid flows at all speeds with an Eulerian-Lagrangian computing mesh, J. Comput. Phys., 17 (1975), 132.

[31] L. G. Margolin, Introduction to "an arbitrary Lagrangian-Eulerian computing method for all flow speeds", J. Comput. Phys., 135 (1997), 198.

[32] D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Meth. Appl. Mech. Engrg., 99 (1992), 235.

[33] P. Knupp, L. G. Margolin and M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys., 176 (2002), 93.

[34] J. G. Trulio and K. R. Trigger, Numerical solution of the one-dimensional hydrodynamic equations in an arbitrary time-dependent coordinate system, Report UCLR-6522, University of California Lawrence Radiation Laboratory, 1961.

[35] P. D. Thomas and C. K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J., 17 (1979), 1030.

[36] I. Demirdzic and M. Peric, Space conservation law in finite volume calculations of fluid flow, Int. J. Numer. Meth. Fluids, 8 (1998), 1037.

[37] H. Guilard and C. Farhat, On the significance of geometric conservation law for flow computations on moving meshes, Comput. Method. Appl. Mech. Engrg., 190 (2000), 1467.

[38] S. A. Morton, R. B. Melville and M. R. Visbal, Accuracy and coupling issues of aeroelastic Navier-Stokes solutions of deforming meshes, AIAA Paper 97-1085, 1997.

[39] Y. Tamura and K. Fujii, Conservation law for moving and transformed grids, AIAA Paper 93-3365-CP, 1993.

[40] Y. Srinivas and E. Gutheil, Modification of conservation laws for use on moving grids, J. Comput. Phys., 2007, in press.

[41] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, Berlin, 1997.

[42] J. von Neumann and R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys., 21 (1950), 232.

[43] S. K. Godunov, Difference method of numerical computations of discontinuous solutions in hydrodynamic equations, Math. Sbornik, 47 (1959), 271.

[44] R. E. Peierls, Theory on von Neumann's method of treating shocks, Technical Report LA-332, Los Alamos Scientific Laboratory, 1945.

[45] B. Despres and C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Ration. Mech. Anal., 178 (2005), 327.

[46] B. Despres and C. Mazeran, Symmetrization of Lagrangian gas dynamics in dimension two and multidimensional solvers, Comptes Rendus (Mecanique), 331 (2003), 475.

[47] M. S. Hall, A comparison of the first and second order rezoned and Lagrangian Godunov solutions, J. Comput. Phys., 90 (1990), 458.

[48] W. H. Hui, G. P. Zhao, J. J. Hu and Y. Zheng, Flow-generated-grids – gridless computation using the unified coordinates, in: Proceedings of the 9th International Conference on Numerical Grid Generation, San Jose, 11-18 June, 2005, pp. 111-121.

[49] W. H. Hui and Y. He, Hyperbolicity and optimal coordinates of the three-dimensional steady Euler equations, SIAM J. Appl. Math., 57 (1997), 893.

[50] C. Rogers, W. K. Schief and W. H. Hui, On complex-lamellar motion of a Prim gas, J. Math. Anal. Appl., 266 (2002), 55.

[51] W. H. Hui and D. L. Chu, Optimal grid for the steady Euler equations, Comput. Fluid Dyn. J., 4 (1996), 403.

[52] W. H. Hui and J. J. Hu, Space-marching gridless computation of steady supersonic/hypersonic flow, Int. J. Comput. Fluid Dyn., 20 (2006), 55.

[53] C. Jin, Gas kinetic moving mesh methods for viscous flow simulations, PhD thesis, Hong Kong University of Science and Technology, 2006.

[54] W. H. Hui, J. J. Hu and K. M. Shyue, Role of coordinates in computational fluid dynamics, in: Proceedings of the 14th Canadian CFD Conference, Kingston, 16-18 July, 2006. To be published in the Int. J. Comput. Fluid Dyn., 2007.

[55] M. M. Hafez, S. Osher and W. Whitlow, Improved finite different schemes for transonic potential calculations, AIAA Paper 84-0092, AIAA 22nd Aerospace Sciences Meeting, 1984.

[56] A. Andersen, U. Pesavento and Z. J. Wang, Unisteady aerodynamics of fluttering and tumbling plates, J. Fluid Mech., 541 (2005), 65.

[57] C. Jin and K. Xu, Numerical study for unsteady aerodynamics of freely falling plates, in

preparation.

[58] K. Xu, A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method, J. Comput. Phys., 171 (2001), 289.

[59] H. Z. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function, Commun. Comput. Phys., 1 (2006), 656.

[60] H. Z. Tang and T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal., 41 (2003), 487.

[61] S. M. Murman, M. J. Aftosmis and M. J. Berger, Implicit approaches for moving boundaries in a 3-D Cartesian method, AIAA-2003-1119, 2003.

[62] M. Arienti, P. Hung, E. Morano and J. E. Shepherd, A level set approach to Eulerian-Lagrangian coupling, J. Comput. Phys., 185 (2003), 213.

[63] C. Jin and K. Xu, A unified moving grid gas-kinetic method in Eulerian space for viscous flow computation, J. Comput. Phys., 222 (2007), 155.

[64] B. van Leer, Upwind and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes, Commun. Comput. Phys., 1 (2006), 192.

[65] R. J. LeVeque, Finite Volume Methods for Hyperbolic Conservation Laws, Cambridge University Press, 2002.