Gas-Kinetic Moving Mesh Methods for Viscous Flow Simulations

by

Changqiu Jin

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in Mathematics

July 2006, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Jin Chang

Gas-Kinetic Moving Mesh Methods for Viscous Flow Simulations

by

Changqiu Jin

This is to certify that I have examined the above doctor of philosophy thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Kun Xu

Prof. Kun Xu, Supervisor

WPL

Prof W.P. Li, Head of Department (Acting)

Department of Mathematics

30 September 2006

Acknowledgment

It is my pleasure to offer my gratitude and thanks to my advisor Kun Xu, who introduced me to the problem, guided and encouraged me throughout the course of this work. I would like to thank the Department of Mathematics for providing me with postgraduate studentship award so that I have the valuable opportunity to study here. I would like to thank my postgraduate fellow students in the Department of Mathematics, H.W. Liu, Q.L. He, Y. Xiao, L. Yang, who kindly shared their good and bad time with me and offered their unforgettable help in the past years.

I am indebted to Prof. W.H. Hui, Prof. X.P. Wang, Dr.Y. Xiang, Prof. H.Z. Tang, Prof. Z.W. Li, Dr. Q.B Li, Dr. X. He, Dr. J.Z, Jiang, Prof. M.L. Mao for their kindly teaching, friendship, and encouragement during the period of my postgraduate study.

I also owe special thanks to my teacher, Prof. S. Jiang, and my friend, Prof. G.P Zhao in China for their caring and encouraging.

Last but not the least, special thanks and deep gratitude to my parents, and especially to my husband Guoliang Hu for his forever love, large support, and deep caring. Without their love, support and help, this thesis would never be finished.

Contents

List of Figures

List of Tables

Gas-kinetic Moving Mesh Methods for Viscous Flow Simulations

Changqiu Jin

Department of Mathematics
The Hong Kong University of Science and Technology

Abstract

In this thesis, the gas-kinetic BGK scheme for two-dimensional viscous flow computation is extended to the moving meshes. Specifically, both the static adaptive grid method and dynamic unified moving mesh method have been developed. In the former one, the grid movement and adaptation are controlled by a monitor function which depends on the gradient of flow variables, such as density or velocity. The grid points in the adaptive grid method can be easily moved and concentrated to the regions with large density and velocity gradients, such as multi-material interface and the boundary layer. Therefore, the adaptive grid method is more accurate and efficient than the methods with stationary mesh points. In the dynamic moving mesh method, the gas-kinetic BGK equation is first reformulated under a unified coordinate transformation with grid velocity included. Then, a unified conservative gas-kinetic scheme is constructed for the viscous flow computation on moving meshes. Due to the coupling between the grid velocity and the overall solution algorithm, the Eulerian and Lagrangian methods become two limiting cases in the current gas-kinetic method. The moving grid method extends the applicable regime of the gas-kinetic scheme to the flows with free surface and moving boundaries, such as dam break problem and airfoil oscillations. In order to further increase the robustness of the moving grid scheme, the above two methods, i.e., static and dynamic ones, have been uniquely combined to move, redistribute, and remedy the distorted meshes in the fluid computations. Many numerical examples from incompressible low speed flow to the supersonic shock interaction are presented. The accuracy and robustness of the moving mesh have been fully demonstrated. In the unsteady aerodynamic flow application, the falling plate problems with the rich dynamic behavior, such as tumbling and fluttering, have been studied. Excellent agreement between the experimental measurements and the numerical computations has been obtained.

Chapter 1

Introduction

In this chapter, the computational fluid dynamic (CFD) methods, i.e., the Eulerian method, the Lagrangian method, arbitrary Lagrangian–Eulerian (ALE), and the unified coordinate system method, will be discussed.

1.1 Computational fluid dynamic (CFD) methods

The study of the computational fluid dynamic started from 1950s. Due to the important application background of CFD on science and engineering, such as aerodynamics, hydraulics, industrial manufacturing, as well as weapon designing, the fast developing trend on CFD has occurred from 1960s and considerable progress has been made over the past four decades [46]. Most applications of computational fluid dynamics are based on the solution of Euler or Navier–Stokes (N–S) equations. It is generally accepted that the Navier–Stokes equations describe all the properties of a continuous flow system and are sufficiently accurate for many applications. The Euler equations describe the inviscid flow, where the dissipative transport phenomena due to viscosity, mass diffusion, and thermal conductivity are neglected. In this sense, it should simulate physical flows in the limit of vanishing viscosities. In contrast, the Navier–Stokes equations describe the viscous flow, where the transport phenomena of friction, and thermal conduction, are included. But, there is still no mass diffusion term.

In the past decades, various numerical methods have been developed to solve Euler and N−S equations. The numerical methods fall in the three major categories: finite difference method, finite volume method, and finite element method. In all above numerical methods, the coordinates used in the numerical discretization play a significant role in accurate and robustness of the solutions. Broadly speaking, there are mainly two coordinate system have been used, which are *Eulerian system* method, and *Lagrangian system one*. Nevertheless, in the recent three decades, many CFD algorithm researchers have concentrated on the development of an *arbitrary Lagrangian − Eulerian* (ALE) method due to its wide application in multi-phase flows. The main consideration of ALE method is to combine the advantages of both Eulerian and lagrangian coordinate systems. Most recently, *a unified coordinate system* has been successfully constructed and implemented for the Euler equations by Hui. et al [31], where the Eulerian and Lagrangian approaches become two limiting cases of the unified coordinate system method.

The Eulerian description observes the flow motion at a fixed location and the flow passes by with certain velocity. This has a close relation to the conventional computation approach in that each fixed grid point can be thought of as an observing station, which is used to describe the flow motion locally. With this approach, the meshes are generated mainly based on the geometrical constraint of the studying problem, without specifically taking account of the fluid motion. Therefore, the grid lines are basically independent of the fluid's stream or path lines. Computationally, the Eulerian method has fixed computational cells in space, while fluid particles move across the cell interface freely. If there is any discontinuity in the flow, the location of the discontinuity may be in the interior of a cell, which cannot be resolved by the cell size. Therefore, the convective flux in the Eulerian method causes excessive numerical diffusion. For example, the slip lines and shock waves can be smeared badly. The numerical dissipation in the contact/slip or shear layer is increasing with the numerical time steps. Therefore, some special treatment techniques, such as artificial compression, sub–cell resolution, and re-initialization in both level set and volume of fluid methods, are employed [25, 26, 66, 73].

In spite of numerical diffusion resulting from approximation of convective terms, the Eulerian description does offer convenience and simplicity both conceptually and geometrically. For example, the grid can be constructed regularly, which can be simply based on geometry constraints and be independent of flow features. In order to overcome the associated numerical defects, in the past decades the CFD algorithm developers have been extensively searching better ways to deal with the convective terms. With the success of the Eulerian approach in almost all aspects of CFD applications, the inaccuracy due to numerical diffusion in the fluid interface problems becomes more obvious, especially in multi-dimensional problems. Since the convective terms do not explicitly appear in the Lagrangian formulation, the numerical mixing automatically disappears, which renders the Lagrangian approach more attractive in this aspect. However, due to its own technical difficulties in the Lagrangian approach, the research progress in this direction is very limited. In what follows, we shall detail the concept of the Lagrangian approach on the numerical point of view. The differences between the two descriptions of Eulerian and Lagrangian will become obvious.

The Lagrangian description follows the motion and properties of given fluid particles as they travel to different locations. In particular, since the particle path in steady flow coincides with the streamline, no fluid particles will across the streamline. In other words, while staying in contact, neighboring streams will not mix via convection. The Lagrangian coordinates are literally fluid particles. Numerically, there is no convective flux across cell interface and numerical diffusion is thus minimized. However, in comparison with the Eulerian fluid system, the gas dynamics equations in Lagrangian description cannot be written in a conservative partial differential equation (PDE) form. Also, the computational grids following fluid particles literally can result in severe grid deformation, causing inaccuracy and even breakdown of the computation. To prevent it from happening, the most well-known Lagrangian method in use at the present time is *the Arbitrary Lagrangian–Eulerian* Technique (ALE) [27, 58, 50], which implements continuous rezoning and re-mapping to the Eulerian smooth grid. Then, after remedying the mesh, the Lagrangian computation continues until re-meshing again.

In general, the motion of the grid can be chosen arbitrarily in flow simulations. The philosophy of the Arbitrary Lagrangian–Eulerian methodology is to exploit this degree of freedom to improve the robustness, accuracy and efficiency of the simulation. The main elements in an ALE simulation include an explicit Lagrangian phase in which mesh moves with fluid, followed by a rezone phase in which a new mesh is generated. Then, the solution obtained by the Lagrangian step is interpolated to the new mesh by a conservative scheme. In the flow region with shear deformation, the computational mesh can be easily tangled. Therefore, it is impractical to use purely Lagrangian method, and the rezoning mesh is necessary.

There are mainly two strategies to determine where the rezoning is needed. The first one is to directly perform some rezoning when the meshes have been tangling. Generally, this strategy requires global re-meshing because any small change in the grid cannot produce an untangled regular mesh with good geometrical quality. Global re-meshing may produce a mesh which is far from the original mesh, which will lead to significant error in the re-mesh and interpolation process. The second strategy is to start re-mesh before mesh tangling, which needs to perform rezoning regularly. An example of a sophisticated rezoning strategy suitable for this approach is the Reference Jacobian Matrix (RJ) method which is aimed at maintaining geometrical quality of the mesh while keeping it as close as possible to the Lagrangian mesh [37]. This strategy does not require any user intervention, which is very important for simplicity and rezoning is performed many times during the simulation. The second approach is superior to the previous one because it constantly maintains geometrical quality, which reduces the numerical error due to mesh distortion. At the same time, the later approach can preserve certain "natural adaptivity" of the Lagrangian method by keeping the rezoned mesh as close as possible to the Lagrangian one. It is true that an additional error will be introduced by frequent re-mapping process. Usually, the interpolation (remapping) of flow variables between different computational cells will result in loss of accuracy, which is manifested as numerical diffusion which ALE targets to avoid in the first place.

In order to keep the advantages of Lagrangian method in the crisp resolution of contact wave and interfaces while mostly avoiding the re-mapping process, recently a moving mesh method in *a unified coordinates system* for the inviscid Euler equations has been developed by Hui et. al [31]. In the unified coordinate system, with a prescribed grid velocity, the inviscid flow equations are written in a conservative form in the computational domain (λ, ξ, η), as well as the geometric conservation laws in controlling the mesh deformation. The most distinguishable merit in the unified coordinate method is that the fluid equations and geometric evolution equations are written in a combined conservative system, which is different from the transformed fluid equations alone in a generalized coordinate system [28]. Furthermore, due to the coupling of the fluid and geometry movement, for the first time the multidimensional Lagrangian gas dynamic equations have been written in a conservative form [31]. Theoretically, it has been pointed out that the multidimensional Lagrangian system in 2D and 3D is only weakly hyperbolic. Numerically, in the unified coordinate system the fluid and geometric variables can be updated simultaneously. In order to overcome the disadvantage in the purely Lagrangian method, in the unified coordinate system the grid velocity is set to be $h\mathbf{q}$, where $\mathbf{q}$ is the fluid velocity and h is a free parameter to be determined by other conditions, such as the mesh alignment with the slip surface, or grid angle maintenance during the mesh movement. Therefore, the grid velocity can be changed locally according to the value of h. The great achievement of the unified coordinate method is that the numerical diffusion across the slip line is reduced to a minimum level with the crisp capturing of contact discontinuity. Also, with the adaptation of shock front tracking method, both shock and contact waves can be accurately captured without any smearing [40]. However, in the cases with complicated flow movement, in order to avoid the severe mesh distortion, the constraint, such as keeping mesh orthogonality and grid angles, has to be imposed in the unified coordinate system all the time. As a result, in most cases, the constraint automatically enforces the mesh velocity being zero, such as in the gas implosion case inside a square. Otherwise, for flow problem with circulation, any mesh movement method with the grid speed being coupled with the fluid velocity will distort the mesh eventually and stop the computation. Also, in order to capture the slip line, the unified method is mainly focusing on the solution of the Euler equations. For the viscous flow, the detail formulation of the Navier-Stokes equations in a unified coordinate system are very complicated, see Appendix A. As an alternative, many moving mesh methods don't solve the viscous and heat conduction terms directly in the moving reference of frame. Instead, they are still discretized in the inertial reference of frame, then transformed into the generalized coordinate system. However, in the current thesis, numerically instead of solving directly the viscous macroscopic governing equations in a moving frame of reference, we are going to solve the gas-kinetic equation for the viscous solution. Instead of dissipative terms in the conventionally NS equations, the relaxation term appears in the kinetic equation, which makes it easy in obtaining viscous solution in a moving frame of reference. Therefore, based on the gas–kinetic BGK scheme, *the unified moving grid gas-kinetic method* proposed by the authors in [34], will be presented in this thesis. For the Eulerian formulation, the gas–kinetic BGK scheme has been well developed for the compressible Navier–Stokes solution. The current thesis will concentrate on its extension to adaptive and moving meshes.

1.2 Moving mesh methods (MMM) for CFD

In CFD, an important step is to generate grids, and/or to move mesh, according to the accuracy requirement or boundary condition. The commonly used method is the Eulerian one, where the grid is fixed in the computational process. For the Lagrangian method, ALE method, and unified coordinate method, the grid itself will be time-dependent. In the past several decades, the moving mesh method has been developed and applied in a wide range of physical and industrial fluid problems, i.e., reactive flow in a piston engine, flow around a pitching airfoil or moving bodies, and semiconductor device fabrication (e.g., modeling oxide flow, crystal growth, or phase change). Numerous examples demonstrate that the strategies of adaptive grid and moving mesh methodology can greatly reduce the errors and improve the computational efficiency in their applications.

Currently, there exist different classifications on the moving mesh methods. Firstly, we can distinguish the moving mesh methods by the strategy to find new positions of mesh nodes. In terms of this strategy, the moving mesh can be divided in two groups. The first group of methods is usually called *static method*, where the grid positions are generated directly at an instant of time according to the fluid variable gradients and this process is decoupled from the original fluid solver. Many important principles in this direction have been established, including the equi-distribution methods in 1D [48], minimizing the Winslow variable diffusion functional [84], the harmonic mapping methods [21], and the variational approach [74, 33, 12] etc. Usually the static method solves the elliptic equation and can be commonly interpreted as a variational method. In this thesis, the adaptive grid method for viscous flow computation presented in Chapter 3 belongs to the group. The second group is called *dynamic method*, where nodal velocities are defined to move the grid to a new location during a short time interval along with the fluid, body, or boundary movement. The fluid velocity is usually used in the determination of mesh movement and this process of the mesh movement is coupled with the governing PDEs. The examples of this category include the classical Lagrangian method [32, 81] with the grid velocity being equal to fluid velocity, and the unified coordinate system method [31], the moving finite element method [3], and the methods based on the geometric conservation law [10] etc. In this thesis, the unified moving grid gas-kinetic method will be constructed, where the grid velocity is a degree of freedom, which can be the fluid velocity, moving boundary velocity, and airfoil oscillating velocity etc. For the second group, besides the fluid equations in a moving frame, the new governing equations related to the mesh movement itself have to be solved as well. The nonlinear system is enlarged in this group. But, it does not require any explicit data interpolation between meshes. In summary, in the first group, the mesh movement equation is decoupled from the governing PDEs. It means that spatial and temporal discretizations of the PDEs can be done separately with the standard methods. After each time step, the solution needs to be interpolated to a new mesh. In certain sense, the above classification is similar to the one in [77], where two groups are categorized as interpolation−free and interpolation−based methods.

More specifically, the moving mesh method facus on a mapping from a regular domain in a computational space Ω_c to an irregularly shaped domain in physical space ω_c. The physical domain is covered with a computational mesh suitable for the solution by using finite difference/volume/element methods, and the computational space is fixed. In the static method, the key ingredients consist of mesh generating equations, error indicator and estimators, and interpolations.

Mesh equations determine a one−to−one mapping from a regular domain in a computational space to a real domain in physical space, i.e., the Euler−Lagrangian equations in the variational approach [74].

Error indicator and estimator is the basis in the determination of mesh position. A well−developed theory of a *posteriori* error estimate does exist for elliptic equations [62] and Burgers' equation [36], which can be adopted to approximate the computational error. However, this is not the case for hyperbolic equation. In most moving mesh methods, the frequently used error indicators are either monitor functions or weight functions. In general, a monitor function can be a matrix, and is used as coefficients in mesh equations defining the mesh movement. In chapter 3 of this thesis, a directional monitor functions is used,

$$M(x,t) = (1 + \alpha_l |u(x,t)|^2 + \beta_l |\nabla u(x,t)|^2)^{\gamma_l/2}, \qquad l = 1, \ldots, d$$

where $u(x,t)$ and $\nabla u(x,t)$ are a solution and its gradient respectively, α_l, β_l and γ_l are some positive constants to control the level of adaptivity. The unit number 1 is added to prevent possible mesh degeneracy in regions where function u is constant. Any reasonable error indicator can be used in u, and ∇u. Here d denotes number of spatial dimension. Based on the principle of equi-distribution, the monitor function is unchangeable on all grid points. However, in most cases, mesh points are concentrated in regions where the monitor function is large. Extensive discussion and numerical experiments with different error indicators and monitor functions can be found in [30]. On other hand, in the dynamic method, the theory is not well established. The key step here is to give the grid velocity, then under a coordinate transformation, the coupled governing PDEs are obtained. The choice of grid velocity depends on the fluid problems. Chapter 4 will present the details about the choice of grid velocity.

1.3 Objective of this thesis

The objective of this thesis is to develop a gas-kinetic scheme for the viscous solution on the moving meshes. More specifically, we will extend the gas–kinetic BGK-NS scheme to the moving meshes based on both static and dynamic methods stated above. Since the gas-kinetic BGK scheme has been well developed for the compressible viscous flow computation, the new moving mesh methods will strength the efficiency and accuracy of the gas-kinetic flow solver. In particular, based on the dynamic method, the gas-kinetic scheme can be used in the numerical simulations for the inviscid and viscous flows with free surface and moving boundaries, such as dam break problem and airfoil oscillations etc. As shown in the numerical study for unsteady aerodynamics of falling plates in chapter 5, the current gas-kinetic scheme can present accurate viscous solutions along with the mesh movement. Also, the coupling of the dynamic mesh with the mesh adaptation method enhances the accuracy of the current method greatly by concentrating more grid points in the regions with large density and velocity gradients. This is the first time that a moving mesh scheme based on the gas-kinetic equation is developed while the Eulerian and the Lagrangian approaches become two limiting cases.

Although the adaptive and moving mesh methods have been used in many areas of scientific computation, the methods and examples presented in this thesis are exclusively limited to the computational fluid dynamics (CFD). The thesis is organized as follow. Chapter 2 presents the basic kinetic theory and introduces the numerical method for the kinetic equation, such as kinetic flux vector splitting (KFVS) method and the gas-kinetic BGK scheme [86, 88, 89]. Chapter 3 describes the adaptive grid method for viscous flow computation, the so-called static mesh method. In Chapter 4, a unified moving mesh gas kinetic method for two dimensional viscous flows will be developed under the unified coordinate transformation and many numerical examples are presented. Moreover, chapter 5 will apply the unified moving mesh method to the unsteady aerodynamic flows of falling plates, where falling plates with the rich dynamic behavior will be simulated. The excellent match between the experimental measurement and the computations validates the current approach.

Chapter 2

Gas-kinetic BGK method for inviscid and viscous flows

There are two ways to describe flow motion. The first one is based on macroscopic quantities, such as mass, momentum and energy densities, as well as the physical law governing these quantities, such as the Euler, Navier-Stokes, Burnett, or high order approximate equations supplied by the equation of state. Another type of description comes from microscopic considerations, i.e., the gas kinetic theory. The fundamental quantity in this description is the particle distribution function $f(x_i, u_i, t)$, which gives the number density of molecules in the six-dimensional phase space $(x_i, u_i) = (x, y, z, u, v, w)$. The evolution equation for the gas-distribution function f is the Boltzmann equation. Physically, the gas kinetic equation provides more information about the gas flow and has larger applicability than the macroscopic counterpart.

The development of numerical schemes based on the gas-kinetic theory for compressible flow simulations began in the 1960's. The Chu's method [16], based on gas-kinetic Bhatnagar-Gross-Kook (BGK) model, with discretized particle velocity space, is one of the earliest kinetic methods used for shock tube calculations. Another kinetic scheme used in early 70's is the beam scheme [65], which approximates the gas distribution function as three delta functions in one-dimensional case and the transport is based on the collisionless Boltzmann equation. Numerically, the beam scheme is almost identical to the flux vector splitting (FVS) scheme of Steger-Warming [69]. Unfortunately, the beam scheme is only known in the astrophysical society in the 70's. In the past two decades, many researchers have contributed to gas-kinetic schemes. A partial list includes Reitz [61], pullin [59], Deshpande [18], Elizarova and Chetverushkin [22], and many others.

The Boltzmann equation with vanishing collision term is called collisionless Boltzmann equation. The flux evaluation of most kinetic schemes introduced above is based on the collisionless Boltzmann equation, the so-called Kinetic Flux Vector Splitting (KFVS) scheme. In the past decade, the gas-kinetic methods based on BGK Model have also been developed greatly to model the gas evolution process more precisely. The schemes of this class are named BGK-type schemes in order to distinguish them from other Boltzmann-type schemes based on the collisionless Boltzmann equation. The BGK-type schemes take into account the particle collisions in the whole gas evolution process or flux evaluation process at a cell interface within a time step, from which a time-dependent gas distribution function and the resulting numerical fluxes at the cell interface can be obtained. Due to the intrinsic connection between the BGK model and viscous governing equations, the BGK method gives Navier-Stokes solutions directly in smooth regions. In the discontinuous regions, the scheme provides a delicate dissipative mechanism to get a stable and crisp shock transition, and the scheme becomes a shock capturing method. Since the gas evolution process in the BGK scheme is a relaxation process from a non-equilibrium state to an equilibrium one, the entropy condition is always satisfied in the BGK method. In this chapter, the gas-kinetic BGK schemes developed in [86, 88] for Navier-Stokes equations will be introduced and compared it with other methods.

2.1 Boltzmann Equation and Bhatnagar-Gross-Kook (BGK) Model of the Boltzmann Equation

In the gas kinetic theory, the particle number density n, is approximated by a gas distribution function $f(x_i, u_i, t)$, where (x_i, t) is the location of any point in space and time, $u_i = (u, v, w)$ is particle velocity with three components in the x, y, and z directions. The relation between the macroscopic fluid density ρ and f can be expressed by the integral

$$\rho = \int\int\int f du dv dw, \tag{2.1}$$

in the particle velocity space. For molecules with internal degree of freedom, such as rotation and vibration, the distribution function f can take these internal motion into account as well through additional variables ξ_i, where $\xi_i = (\xi_1, \xi_2, \cdots, \xi_N)$ are the components of the internal particle velocity in N dimensions. For monotonic gas, the internal degree of freedom N is equal to 0. For diatomic gases, under the normal pressure and temperature, N is equal to 2 which accounts for two independent rotational degrees of freedom. Then, the ratio of the specific heat capacities C_v and C_p for the gas in equilibrium state is commonly denoted by γ,

$$\gamma = \frac{C_p}{C_v} = \frac{(N+3)+2}{N+3}.$$

In order to understand the internal variable ξ_i inside the gas distribution function, let's first write down the Maxwell-Boltzmann distribution g for the equilibrium state,

$$g = \rho(\frac{\lambda}{\pi})^{\frac{N+3}{2}} e^{-\lambda((u-U)^2+(v-V)^2+(w-W)^2+\xi_1^2+\cdots+\xi_N^2)}, \tag{2.2}$$

where ρ is the density, $U_i = (U, V, W)$, are the macroscopic velocity in the $x-$, $y-$ and $z-$directions, $\lambda = \frac{m}{2kT}$, where m is the molecular mass, k is the Boltzmann constant, and T is the temperature. In the above equation, the parameters λ, U_i and ρ which determine g uniquely are functions of space and time. Taking moments of equilibrium state g, the mass, momentum and energy densities at any point in space and time can be obtained. For example, the macroscopic and microscopic descriptions are related by

$$\begin{pmatrix} \rho \\ \rho U_i \\ \rho \epsilon \end{pmatrix} = \int g \begin{pmatrix} 1 \\ u_i \\ \frac{1}{2}(u_i^2 + \xi^2) \end{pmatrix} d\Xi,$$

where $d\Xi = dudvdwd\xi$, $d\xi = d\xi_1 d\xi_2 \dots d\xi_K$. If we re-define the internal variable ξ_i as a vector in K dimensions, in 3-Dimensional case we have

$$K = N = \frac{5 - 3\gamma}{\gamma - 1}.$$

For 1-Dimensional gas flow, the random particle motion in y and z directions can be included in the internal variable ξ of the molecules. As a result, the internal degree of freedom becomes $K = N + 2$. The distribution function g in the 1-Dimensional case goes to

$$\begin{aligned} g &= \rho(\frac{\lambda}{\pi})^{\frac{N+3}{2}} e^{-\lambda((u-U)^2+v^2+w^2+\xi_1^2+\dots+\xi_N^2)} \\ &= \rho(\frac{\lambda}{\pi})^{\frac{K+1}{2}} e^{-\lambda((u-U)^2+\xi^2)}. \end{aligned} \tag{2.3}$$

In 2-Dimensional flow calculations, the random particle motion in the z direction is included into the internal variable ξ, the total number of degrees of freedom $K = N+1$, and the equilibrium distribution function is

$$\begin{aligned} g &= \rho(\frac{\lambda}{\pi})^{\frac{N+3}{2}} e^{-\lambda((u-U)^2+(v-V)^2+w^2+\xi_1^2+\dots+\xi_N^2)} \\ &= \rho(\frac{\lambda}{\pi})^{\frac{K+2}{2}} e^{-\lambda((u-U)^2+(v-V)^2+\xi^2)}. \end{aligned} \tag{2.4}$$

Usually, we do not know the explicit form of the gas distribution function f in the highly non-equilibrium flow regions, such as that inside a strong shock wave. What we know is the dynamical evolution of the distribution function f, the so-called Boltzmann Equation,

$$f_t + u_i f_{x_i} + a_i f_{u_i} = Q(f, f). \tag{2.5}$$

Here f is the real gas distribution function, a_i is the external forcing term acting on the particle in i-th direction, and $Q(f, f)$ is the collision operator. From the physical constraints of the conservation of mass, momentum and energy during particle collisions, the following compatibility condition has to be satisfied,

$$\int \psi_\alpha Q(f,f) d\Xi = 0, \tag{2.6}$$

where $d\Xi = dudvdwd\xi_1 d\xi_2 \dots d\xi_K$ and $\psi_\alpha = (1, u, v, w, \frac{1}{2}(u^2+v^2+w^2+\xi^2))^T$. The gas kinetic theory suggests that the Navier-Stokes equations are valid if the length scale L of the flow is much larger than the mean free path $\bar{l}$ of the molecules, i.e.,

$$K_n = \frac{\bar{l}}{L} \ll 1,$$

where K_n is defined as the Knudsen number. In most compressible flow problems, we face the calculation of shock and boundary layers. For the shock waves and boundary layers, the characteristic length scales are different. For example, in a boundary layer, the significant length scale is the thickness of the boundary layer,

$$L \sim \frac{1}{Re^{1/2}}.$$

On the other hand, for a shock wave the thickness of the shock structure is the characteristic length scale,

$$L \sim \frac{1}{Re},$$

where Re are the Reynolds numbers, which can be very high for the boundary layer problems and close to 1 for the shock front.

Remarks:

1. In the Boltzmann equation (2.5), the advection term on the left hand side always drives f away from local equilibrium distribution; the collision term on the right hand side $Q(f,f)$ pushes f back to equilibrium. Although, $Q(f,f)$ does not change the local mass, momentum and energy, it does re-distribute particles in the phase space (u_i, ξ_i), subsequently change the transport coefficients of the particle system, e.g., viscosity and heat-conductivity. The real flow evolution is governed by the competition and balance between the convection and collision terms. The viscosity and heat conduction coefficients can be properly determined through the determination of particle collision time.

2. For any shock capturing method, the numerical shock region usually spans over a few mesh points. So, the mean free path of the 'numerical fluid' in these regions, which is proportional to the shock thickness, is on the order of the cell size, i.e. $\bar{l} \sim \triangle x$. As a result, the numerics amplifies the thickness of shock layer. The BGK method, presented in this thesis, could consistently capture the amplified numerical shock region from the controllable particle collision time τ, which also ranges from the physical one $\tau \sim \mu$ to the numerical one $\tau \sim \triangle t$, where μ is the viscosity coefficient and $\triangle t$ is the time step. The robustness and accuracy of the BGK method is mainly due to its ability to capture both equilibrium and nonequilibrium gas flow.

One of the main functions of the particle collision term is to drive the gas distribution function f back to the equilibrium state g which has one to one correspondence with the local mass, momentum and energy densities, $(\rho, \rho U_i)$ and $(\rho\epsilon)$. Due to the particle collision, during a time interval dt a fraction of dt/τ of molecules in a given small volume undergoes collision, where τ is the average time interval between successive particle collisions, the so-called particle collision time. The collision term in the BGK model alters the velocity-distribution function from f to g. With the assumption of the rate of changes df/dt of f due to collisions is $-(f-g)/\tau$, the Boltzmann equation without external forcing term becomes [6],

$$\frac{\partial f}{\partial t} + u_i \frac{\partial f}{\partial x_i} = -\frac{f-g}{\tau}. \tag{2.7}$$

This is so-called BGK model. At the same time, due to the mass, momentum and energy conservation in particle collisions, the collision term $(g-f)/\tau$ satisfies the compatibility condition,

$$\int \frac{g-f}{\tau} \psi d\Xi = 0. \tag{2.8}$$

Eq.(2.7) is a nonlinear equation since the distribution function g depends on ρ, ρU_i, $\rho\epsilon$, which is the integrals of the function f. The above BGK model is basically a relaxation model.

If τ is a local constant, Eq.(2.7) may be written in integral form [38],

$$f(x_i, t, u_i, \xi) = \frac{1}{\tau} \int_{t_0}^{t} g(x_i - u_i(t - t'), t', u_i, \xi) e^{-(t-t')/\tau} dt' + \\ e^{-(t-t_0)/\tau} f_0(x_i - u_i(t - t_0), t_0, u_i, \xi), \qquad (2.9)$$

where f_0 is the gas distribution function f at t_0, and g is the equilibrium state in (x_i, t). As a special case, we consider an initial gas distribution function $f(x_i, t_0)$ which does not depend on the spatial coordinates. The corresponding macroscopic mass ρ, momentum ρU_i, and energy $\rho\epsilon$ are constant in space. Then, the corresponding equilibrium state g will be a constant in both space and time. From Eq.(2.9), we have

$$f(x_i, t) = (1 - e^{-(t-t_0)/\tau}) g + e^{-(t-t_0)/\tau} f_0(x_i, t_0). \qquad (2.10)$$

The distribution function f tends to the equilibrium state g exponentially with a characteristic relaxation time τ. For example, the smaller the τ is, the faster the equilibrium will be attained. From this example, we can observe that the collision time is related to the system dissipation and the gas-kinetic description provides more information than the macroscopic one. Although, all macroscopic quantities are homogeneous and time independent, the particle distribution actually is a function of time. Consequently, the dissipative property of the gas system is also a function of time. The evolution from f to g is a process of increasing of entropy, which corresponds to an irreversible process.

The particle collision time is related to both viscosity and heat conduction coefficients, and these two parameters are related through the Prandtl number,

$$Pr = \frac{\eta C_p}{\kappa},$$

where η and κ are the coefficients of viscosity and thermal conductivity respectively. The Prandtl number is almost a constant for air, and the value is about 0.72 at the common temperature. From the BGK model, we can derive that the Prandtl number becomes fixed with the value $Pr = 1$. Numerically, we can easily fix this number through the modification of heat flux across a cell interface. Since in the continuum regime the behavior of the fluid depends very little on the nature of individual particles, the most important properties are: conservation, symmetry and dissipation. The BGK model satisfies all these requirements [41]. In the following, we are going to introduce a few shock capturing kinetic schemes.

2.2 Kinetic flux vector splitting (KFVS) method

It is well-known that the Euler equations can be derived from the Boltzmann equation with a local equilibrium distribution function. For an equilibrium state, f is equal to the maxwellian distribution function g, and the collision term $Q(f, f)$ goes to zero automatically, i.e., $Q(g, g) \equiv 0$. So, in the 1-D case, once $f = g$ holds, the Boltzmann equation becomes

$$f_t + uf_x = 0, \tag{2.11}$$

which is called collisionless Boltzmann equation. With the initial condition of the gas distribution function $f_0(x, 0)$ at time $t = 0$, the exact solution of the collisionless Boltzmann equation is

$$f = f_0(x - ut, t). \tag{2.12}$$

For example, for the same initial condition as the Riemann problem, two constant equilibrium states at $x \leq 0$ and $x > 0$ can be constructed around $x = 0$,

$$f_0 = g_l(1 - H(x)) + g_r H(x) = \begin{cases} g_l \quad , & x \leq 0, \\ g_r \quad , & x > 0, \end{cases} \tag{2.13}$$

where $H(x)$ is the Heaviside function. The two equilibrium states g_l and g_r have one to one correspondence with the macroscopic flow variables $(\rho_l, \rho_l U_l, \rho_r \epsilon_l, \quad \rho_r, \rho_r U_r, \rho_r \epsilon_r)$ on both sides and have the form

$$g_l = \rho_l (\frac{\lambda_l}{\pi})^{\frac{K+1}{2}} e^{-\lambda_l((u-U_l)^2+\xi^2)}, \quad g_r = \rho_r (\frac{\lambda_r}{\pi})^{\frac{K+1}{2}} e^{-\lambda_r((u-U_r)^2+\xi^2)},$$

respectively. The exact solution of the collisionless Boltzmann equation is

$$f(x, t) = f_0(x - ut) = g_l(1 - H(x - ut)) + g_r H(x - ut). \tag{2.14}$$

Since $t > 0$, the above equation can be reformulated as

$$f(x,t) = g_l(1 - H(\frac{x}{t} - u)) + g_r H(\frac{x}{t} - u) = f(\frac{x}{t}) = \begin{cases} g_l, & u \geq \frac{x}{t}, \\ g_r, & u < \frac{x}{t}, \end{cases} \tag{2.15}$$

which is similar to the Riemann solution of the Euler equations. The collisionless Boltzmann equation also provides a similarity solution for the case with two constant initial states. Although this similarity solution is commonly referred to as an approximate Riemann solution, it provides a different gas evolution picture, which can be demonstrated by the following example. Firstly, we calculate the exact solutions of Eq. (2.11) under the following initial condition

$$(\rho, \rho U, \rho\epsilon) = \begin{cases} (1.0, & -2.0, & 3.0), & x \leq 0, \\ (1.0, & 2.0, & 3.0), & x > 0, \end{cases} \tag{2.16}$$

from which two Maxwellians of g_l and g_r in Eq. (2.13)can be constructed. Based on the solution (2.15), the mass, momentum and energy densities at any point in space x and a fixed time t can be obtained by taking the moments of $f(x,t)$,

$$\begin{pmatrix} \rho(x,t) \\ \rho U(x,t) \\ \rho\epsilon(x,t) \end{pmatrix} = \int\int_{-\infty}^{\infty} f(x,t)\psi_\alpha du d\xi = \int\int_{\frac{x}{t}}^{\infty} g_l \psi_\alpha du d\xi + \int\int_{-\infty}^{\frac{x}{t}} g_r \psi_\alpha du d\xi$$

$$= \int\int_{\frac{x}{t}}^{\infty} \rho_l (\frac{\lambda_l}{\pi})^{\frac{K+1}{2}} e^{-\lambda_l((u-U_l)^2+\xi^2)} \psi_\alpha du d\xi + \int\int_{-\infty}^{\frac{x}{t}} \rho_r (\frac{\lambda_r}{\pi})^{\frac{K+1}{2}} e^{-\lambda_r((u-U_r)^2+\xi^2)} \psi_\alpha du d\xi$$

$$= \int\int_{0}^{\infty} \rho_l (\frac{\lambda_l}{\pi})^{\frac{K+1}{2}} e^{-\lambda_l((\tilde{u}-\tilde{U}_l)^2+\xi^2)} \psi_\alpha d\tilde{u} d\xi + \int\int_{-\infty}^{0} \rho_r (\frac{\lambda_r}{\pi})^{\frac{K+1}{2}} e^{-\lambda_r((\tilde{u}-\tilde{U}_r)^2+\xi^2)} \psi_\alpha d\tilde{u} d\xi$$

where $\tilde{u} = u - \frac{x}{t}$, $\tilde{U}_l = U_l - \frac{x}{t}$, $\tilde{U}_r = U_r - \frac{x}{t}$, and $\psi_\alpha = (1, \tilde{u} + \frac{x}{t}, \frac{1}{2}((\tilde{u} + \frac{x}{t})^2 + \xi^2))^T$.

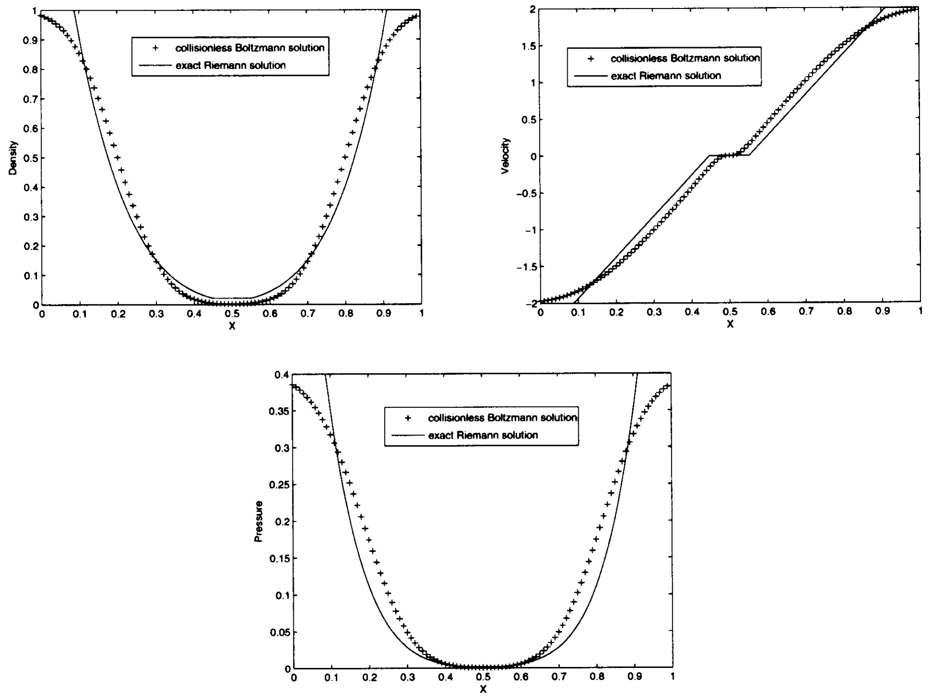


Figure 2.1: Exact Euler solutions (solid line) and collisionless Boltzmann solutions (2.17) ($*$ symbol) at time $t = 0.15$ units for the same initial condition

Then, the macroscopic variables in space and time can be obtained,

$$\begin{pmatrix} \rho(x,t) \\ \rho U(x,t) \\ \rho\epsilon(x,t) \end{pmatrix} = \rho_l \begin{pmatrix} \frac{1}{2}erfc(-\sqrt{\lambda_l}\tilde{U}_l) \\ (\frac{1}{2}\tilde{U}_l + \frac{1}{2}\frac{x}{t})erfc(-\sqrt{\lambda_l}\tilde{U}_l) + \frac{1}{2}\frac{e^{-\lambda_l\tilde{U}_l^2}}{\sqrt{\pi\lambda_l}} \\ (\frac{(\tilde{U}_l+\frac{x}{t})^2}{4} + \frac{(K+1)}{8\lambda_l})erfc(-\sqrt{\lambda_l}\tilde{U}_l) + (\frac{\tilde{U}_l}{4} + \frac{1}{2}\frac{x}{t})\frac{e^{-\lambda_l\tilde{U}_l^2}}{\sqrt{\pi\lambda_l}} \end{pmatrix}$$

$$+\rho_r \begin{pmatrix} \frac{1}{2}erfc(\sqrt{\lambda_r}\tilde{U}_r) \\ (\frac{1}{2}\tilde{U}_r + \frac{1}{2}\frac{x}{t})erfc(\sqrt{\lambda_r}\tilde{U}_l) - \frac{1}{2}\frac{e^{-\lambda_r\tilde{U}_r^2}}{\sqrt{\pi\lambda_r}} \\ (\frac{(\tilde{U}_r+\frac{x}{t})^2}{4} + \frac{(K+1)}{8\lambda_r})erfc(\sqrt{\lambda_r}\tilde{U}_r) - (\frac{\tilde{U}_r}{4} + \frac{1}{2}\frac{x}{t})\frac{e^{-\lambda_r\tilde{U}_r^2}}{\sqrt{\pi\lambda_r}} \end{pmatrix} \qquad (2.17)$$

The density, velocity and pressure distributions in the above equation at time $t = 0.15$ units are plotted in Fig. 2.1. The solid lines refer to the exact solution from the similarity solution of the Euler equations and the $\times$ symbol are the solutions from the collisionless Boltzmann equation (2.15). The above solution consists of two strong rarefaction waves and a trivial stationary contact discontinuity. The two rarefaction waves are quite smeared. This figure shows the difference between the exact Euler solution and the exact collisionless Boltzmann solution under the same initial condition. In other words, even with the initial equilibrium assumption, the dynamics of the collisionless Boltzmann equation is different from that of the Euler equations. The Eq.(2.11) is the correct description of the equilibrium flow motion if the flow remains in a local equilibrium state. Nevertheless, even with an initial local equilibrium state, the collisionless Boltzmann equation transports the gas distribution function away from its equilibrium initial condition. Physically, the mechanism for bringing the distribution function back to a maxwellian is the collision suffered by the gas molecules, the so-called collision term in the Boltzmann equation. Therefore, the collisionless Boltzmann equation without this mechanism, cannot keep the equilibrium state, and theoretically cannot be used as a Euler solver. The collisionless mechanism has intrinsically rooted in many flux vector splitting schemee.

Based on the collisionless Boltzmann solution (2.11), the scheme based on the solution (2.15) for the flux evaluation at a cell interface is so-called Kinetic Flux Vector Splitting (KFVS) scheme. Although the KFVS scheme ignores particle collisions in the gas evolution stage, it still gives a reasonable numerical solution, which is different from the exact solution of the collisionless Boltzmann equation. It is the numerics which adds pseudo-particle collisions in the scheme through the projection and reconstruction stages of a numerical method. In the following section, the first order and second KFVS schemes are presented.

2.2.1 1st-order KFVS

In 1-D case, each cell occupies a small space $x \in [x_{j-1/2}, x_{j+1/2}]$, where $j+1/2$ denotes the cell interface between cells j and $j+1$, and the cell center is located at x_j. $W_j = (\rho_j, \rho_j U_j, \rho_j \epsilon_j)$ is the initial conservative flow variables, i.e., the mass, momentum and energy densities inside each cell j.

An equilibrium state g_j has the following form

$$g_j = \rho_j(\frac{\lambda_j}{\pi})^{\frac{K+1}{2}} e^{-\lambda_j((u-U_j)^2+\xi^2)},$$

which can be uniquely determined from W. So, around a cell interface $x_{j+1/2}$, we have $g_l = g_j$ and $g_r = g_{j+1}$ on the left and right hand sides. The solution f of the collisionless Boltzmann equation at $x_{j+1/2}$ becomes

$$f(x_{j+1/2}, t) = g_l(1 - H(x - x_{j+1/2})) + g_r H(x - x_{j+1/2}) = \begin{cases} g_j \ , & u \leq 0, \\ g_{j+1}, & u > 0. \end{cases} \tag{2.18}$$

From the above distribution function, the numerical fluxes for the mass, momentum and energy across the cell interface can be constructed, which are

$$\begin{aligned} \mathcal{F}_{j+1/2} = \begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_{\rho U} \\ \mathcal{F}_{\rho\epsilon} \end{pmatrix}_{j+1/2} &= \int u\psi_\alpha f(x_{j+1/2}, t) du d\xi \\ &= \int_{u>0} \int u\psi_\alpha g_j du d\xi + \int_{u<0} \int u\psi_\alpha g_{j+1} du d\xi \end{aligned} \tag{2.19}$$

where $\psi_\alpha = (1, u, \frac{1}{2}(u^2 + \xi^2))^T$. Further, the fluxes can be expressed as

$$\begin{aligned} \begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_{\rho U} \\ \mathcal{F}_{\rho\epsilon} \end{pmatrix}_{j+1/2} &= \rho_j \begin{pmatrix} \frac{U_j}{2} erfc(-\sqrt{\lambda_j} U_j) + \frac{1}{2} \frac{e^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}} \\ (\frac{U_j^2}{2} + \frac{1}{4\lambda_j}) erfc(-\sqrt{\lambda_j} U_j) + \frac{U_j}{2} \frac{e^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}} \\ (\frac{U_j^3}{4} + \frac{K+3}{8\lambda_j} U_j) erfc(-\sqrt{\lambda_j} U_j) + (\frac{U_j^2}{4} + \frac{K+2}{8\lambda_j}) \frac{e^{-\lambda_j U_j^2}}{\sqrt{\pi\lambda_j}} \end{pmatrix} \\ &+ \rho_{j+1} \begin{pmatrix} \frac{U_{j+1}}{2} erfc(-\sqrt{\lambda_{j+1}} U_{j+1}) - \frac{1}{2} \frac{e^{-\lambda_{j+1} U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}} \\ (\frac{U_{j+1}^2}{2} + \frac{1}{4\lambda_{j+1}}) erfc(-\sqrt{\lambda_{j+1}} U_{j+1}) - \frac{U_{j+1}}{2} \frac{e^{-\lambda_{j+1} U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}} \\ (\frac{U_{j+1}^3}{4} + \frac{K+3}{8\lambda_{j+1}} U_{j+1}) erfc(-\sqrt{\lambda_{j+1}} U_{j+1}) - (\frac{U_{j+1}^2}{4} + \frac{K+2}{8\lambda_{j+1}}) \frac{e^{-\lambda_{j+1} U_{j+1}^2}}{\sqrt{\pi\lambda_{j+1}}} \end{pmatrix}, \end{aligned} \tag{2.20}$$

where the complementary error function is defined by

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.$$

Using the above numerical fluxes, a finite volume method updates the cell averaged flow variables $W_j^n = (\rho_j, \rho_j U_j, \rho_j \epsilon_j)^n$ inside each cell as

$$W_j^{n+1} = W_j^n + \frac{\Delta t}{\Delta x}(\mathcal{F}_{j-1/2} - \mathcal{F}_{j+1/2}), \tag{2.21}$$

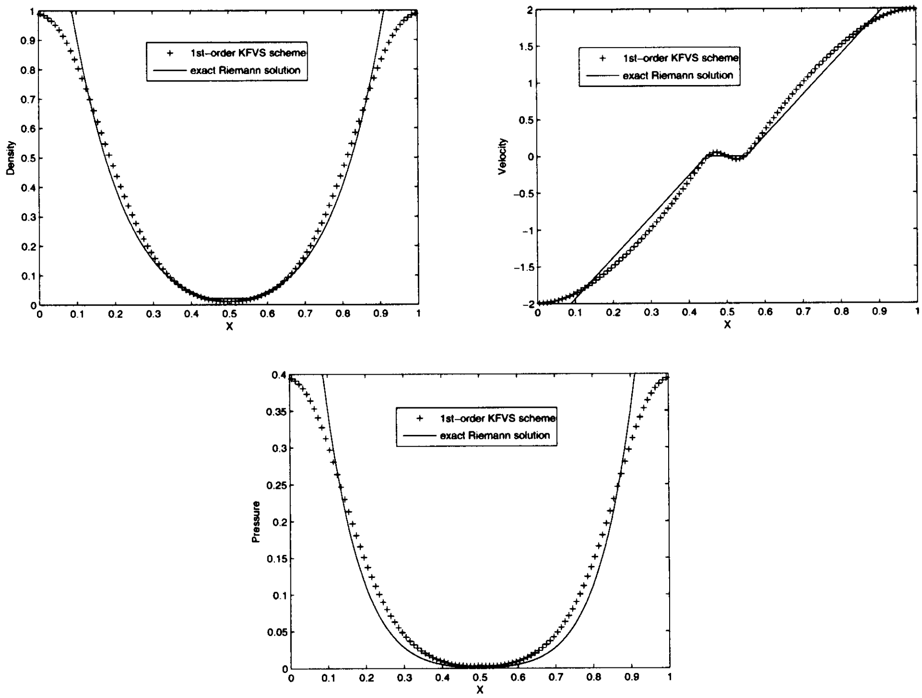


Figure 2.2: Exact Euler solutions (solid line) and numerical solutions (× symbol) using the 1st-order KFVS scheme at time $t = 0.15$ units.

where n is the time step number, Δt is time step, and Δx the cell size.

Let's apply the 1st-order KFVS scheme to the same test case above Eq.(2.16). In Fig. (2.2) the density, velocity and pressure distributions are plotted, where the solid lines are the exact Riemann solutions and the × symbol are the numerical solutions with 100 grid points using the 1st-order KFVS scheme. Obviously, we can observe the difference between the numerical solutions and the exact solutions from the collisionless Boltzmann equation by comparing Fig. 2.1 with Fig. 2.2. The difference is due to the preparation of Maxwellian distribution functions at the beginning of each time step, which is equivalent to adding pseudo-particle collisions into the collisionless Boltzmann method. The diffusivity in the KFVS scheme is not only from the truncation error of the numerical discretization, but also from the intrinsic nature of the governing equation. The numerical dissipation is proportional to the time step, or the pseudo-particle collision time.

2.2.2 2nd-order KFVS

In order to develop a high order scheme, we need first to reconstruct a high order initial condition. For example, the van Leer limiter [80],

$$L_j = (\mathrm{sign}(s_+) + \mathrm{sign}(s_-))\frac{|s_+||s_-|}{|s_+| + |s_-|} \tag{2.22}$$

can be used for the reconstruction of initial conservative variables inside the jth cell, where s_+ and s_- represent the possible slopes of conservative variables in the jth cell:

$$s_+ = \frac{W_{j+1} - W_j}{x_{j+1} - x_j} \quad , \quad s_- = \frac{W_j - W_{j-1}}{x_j - x_{j-1}}$$

with $W_l = W_j$ and $W_r = W_{j+1}$, $\frac{\partial W_l}{\partial x} = L_j$ and $\frac{\partial W_r}{\partial x} = L_{j+1}$ at the left and the right hand sides of a cell interface $x_{j+1/2}$. A second order accurate initial condition can be constructed,

$$\begin{pmatrix} W^l_{j+1/2} \\ W^r_{j+1/2} \end{pmatrix} = \begin{pmatrix} W_l + \frac{\partial W_l}{\partial x}(x_{j+1/2} - x_j) \\ W_r + \frac{\partial W_r}{\partial x}(x_{j+1/2} - x_{j+1}) \end{pmatrix}, \tag{2.23}$$

from which the equivalent initial gas distribution function f_0 around a cell interface $x_{j+1/2}$ is obtained (see the schematic figure Fig.2.3)

$$f_0(x) = \begin{cases} g_l(1 + a_l(x - x_{j+1/2})) & , \quad x - x_{j+1/2} \leq 0, \\ g_r(1 + a_r(x - x_{j+1/2})) & , \quad x - x_{j+1/2} \geq 0. \end{cases} \tag{2.24}$$

Based on the above initial condition, the solution f of the collisionless Boltzmann equation (2.11) at $x_{j+1/2}$ and time t becomes

$$f(x_{j+1/2}, t) = f_0(x_{j+1/2} - ut) = \begin{cases} g_l(1 + a_l(-ut)) & , \quad u \geq 0, \\ g_r(1 + a_r(-ut)) & , \quad u \leq 0, \end{cases} \tag{2.25}$$

where the terms a_l and a_r are based on the Taylor expansion of the Maxwellian distribution function and have the form

$$a^l = a^l_1 + a^l_2 u + a^l_3 \frac{1}{2}(u^2 + \xi^2) \quad , \quad a^r = a^r_1 + a^r_2 u + a^r_3 \frac{1}{2}(u^2 + \xi^2).$$

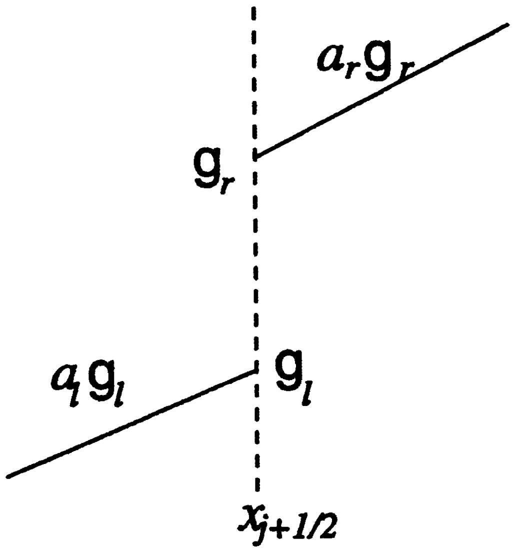


Figure 2.3: Initial gas distribution function for the 2nd–order KFVS scheme.

By using the relation between the microscopic gas distribution function and the macroscopic variables at $x_{j+1/2}$, we get

$$\int \psi_\alpha g_l du d\xi = W^l_{j+1/2}, \quad \int \psi_\alpha g_r du d\xi = W^r_{j+1/2},$$

$$\int \psi_\alpha g_l a_l du d\xi = \frac{W^l_{j+1/2} - W_l}{x_{j+1/2} - x_j}, \quad \int \psi_\alpha g_r a_r du d\xi = \frac{W_r - W^r_{j+1/2}}{x_{j+1} - x_{j+1/2}}. \tag{2.26}$$

Moreover, Eq.(2.26) on both sides of a cell interface can be simplified as

$$M \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \frac{1}{\rho} \begin{pmatrix} \frac{\partial \rho}{\partial x} \\ \frac{\partial(\rho U)}{\partial x} \\ \frac{\partial(\rho \epsilon)}{\partial x} \end{pmatrix}, \tag{2.27}$$

where the symmetric matrix M has the form

$$M = \begin{pmatrix} 1 & U & \frac{1}{2}(U^2 + \frac{K+1}{2\lambda}) \\ U & U^2 + \frac{1}{2\lambda} & \frac{1}{2}(U^3 + \frac{(K+3)U}{2\lambda}) \\ \frac{1}{2}(U^2 + \frac{K+1}{2\lambda}) & \frac{1}{2}(U^3 + \frac{(K+3)U}{2\lambda}) & \frac{1}{4}(U^4 + \frac{(K+3)U^2}{\lambda} + \frac{(K^2+4K+3)}{4\lambda^2}) \end{pmatrix}. \tag{2.28}$$

These derivatives of macroscopic variables on the right side of Eq.(2.26) can be calculated by (2.22). The solutions of Eq.(2.27) are

$$\begin{cases} a_3 = \frac{4\lambda^2}{K+1}(B - 2UA), \\ a_2 = 2\lambda(A - a_3 \frac{U}{2\lambda}), \\ a_1 = \frac{1}{\rho}\frac{\partial \rho}{\partial x} - a_2 U - a_3(\frac{U^2}{2} + \frac{K+1}{4\lambda}), \end{cases}$$

where

$$A = \frac{1}{\rho}\left(\frac{\partial(\rho U)}{\partial x} - U\frac{\partial \rho}{\partial x}\right) \quad , \quad B = \frac{1}{\rho}\left(2\frac{\partial(\rho \epsilon)}{\partial x} - (U^2 + \frac{K+1}{2\lambda})\frac{\partial \rho}{\partial x}\right).$$

On the other hand, the values of (a_1, a_2, a_3) can be also obtained directly by the Taylor expansion of the Maxwellian distribution function in terms of the macroscopic flow variables. From Eq.(2.25), the time–dependent numerical fluxes of the mass, momentum and energy can be written as,

$$\mathcal{F}_{j+1/2} = \begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_{\rho u} \\ \mathcal{F}_{\rho\epsilon} \end{pmatrix}_{j+1/2} = \int_{u>0}\int u \begin{pmatrix} 1 \\ u \\ \frac{1}{2}(u^2+\xi^2) \end{pmatrix} g_l(1 - a_l u t) du d\xi + \int_{u<0}\int u \begin{pmatrix} 1 \\ u \\ \frac{1}{2}(u^2+\xi^2) \end{pmatrix} g_r(1 - a_r u t) du d\xi. \qquad (2.29)$$

Then, the flow variables inside the jth cell can be updated through

$$W_j^{n+1} = W_j^n + \frac{1}{\Delta x}\int_0^{\Delta t} (\mathcal{F}_{j-1/2} - \mathcal{F}_{j+1/2}) dt. \qquad (2.30)$$

For the same test case shown above, the numerical results from the current 2nd-order KFVS scheme are shown in Fig. 2.4. Comparing Fig. 2.4 with Fig. 2.2, the 2nd-order KFVS scheme enhances the accuracy of the two strong rarefaction waves. But, the KFVS scheme usually gives a poorer result than those obtained from the Godunov or flux difference splitting (FDS) schemes. Even for the Euler solutions, the 2nd-order FDS scheme usually gives less dissipative results than those from the 2nd-order FVS scheme. But, the robustness of the FDS schemes can be inferior in comparison with FVS methods. For example, the Roe's scheme can be easily failed for the current test case.

Again, the gas evolution model in the KFVS scheme is based on the collisionless Boltzmann equation. Therefore, in the gas evolution stage of the KFVS scheme, the particles can transport freely. For example, gas in high temperature region is freely moving into low temperature region without suffering any particle collisions. As a result, the free penetration of particles strongly and easily smears any temperature gradients and removes the possible formation of contact discontinuity wave. This can be clearly seen in the Sod test case in [87] (on page 26-39). Similarly, the 'shock' from collisionless Boltzmann equation will also be smeared due to the free transport of particles across the "shock" front even though the shock has self-steepening mechanism. Numerically, the particles are not absolutely moving freely as described in the collisionless Boltzmann equation. The numerical particles do suffer some kind of collisions to reduce the dissipation or penetration to smear any gradient. In order to reduce the numerical dissipation, the real particle collision has to be added in the particle transport process, where the physical collision time τ is introduced into the solution f around each cell interface. In the following, the gas-kinetic BGK scheme will be introduced.

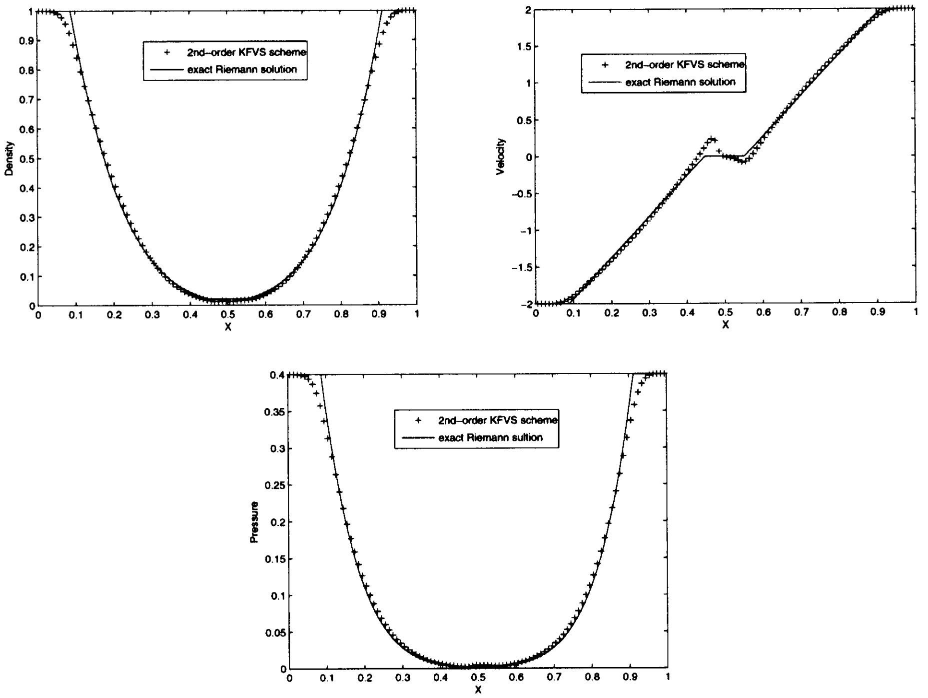


Figure 2.4: Exact Euler solutions (solid line) and numerical solutions (× symbol) using the 2nd-order KFVS scheme at time $t = 0.15$ units.

2.3 Gas-kinetic BGK method with the directional splitting fluxes

For the 2D BGK equation [6]

$$f_t + uf_x + vf_y = \frac{g-f}{\tau}, \tag{2.31}$$

we can use a directional splitting method to solve it. The BGK model in the $x-$direction can be written as

$$f_t + uf_x = \frac{g-f}{\tau}, \tag{2.32}$$

where f is the gas distribution function and g is the equilibrium state approached by f. Both f and g are functions of space (x, y), time t, particle velocities (u, v), and internal variable ξ. The particle collision time τ is related to the viscosity and heat conduction coefficients. The equilibrium state is a Maxwellian distribution,

$$g = \rho(\frac{\lambda}{\pi})^{\frac{K+2}{2}} e^{-\lambda((u-U)^2+(v-V)^2+\xi^2)},$$

where ρ is the density, U and V are the macroscopic velocities in the x and y directions, and λ is related to the gas temperature $m/2kT$. For a 2D flow, the particle motion in the $z-$ direction is included into the internal variable ξ, and the total number of degrees of freedom K in ξ is equal to $(5-3\gamma)/(\gamma-1)+1$. In the equilibrium state, ξ^2 is equal to $\xi^2 = \xi_1^2 + \xi_2^2 + ... + \xi_K^2$. The relation between mass ρ, momentum $(m = \rho U, n = \rho V)$, and energy densities $\rho\epsilon$ with the distribution function f is

$$w = \begin{pmatrix} \rho \\ m \\ n \\ \rho\epsilon \end{pmatrix} = \int \psi_\alpha f d\Xi, \quad \alpha = 1, 2, 3, 4, \tag{2.33}$$

where ψ_α is the component of the vector of moments

$$\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3, \psi_4)^T = (1, u, v, \frac{1}{2}(u^2 + v^2 + \xi^2))^T,$$

and $d\Xi = dudvd\xi$ is the volume element in the phase space with $d\xi = d\xi_1 d\xi_2 ... d\xi_K$. Since mass, momentum and energy are conserved during particle collisions, f and g satisfy the conservation constraint

$$\int (g-f)\psi_\alpha d\Xi = 0, \qquad \alpha = 1,2,3,4, \tag{2.34}$$

at any point in space and time.

For a local equilibrium state with $f = g$, the Euler equations can be obtained by taking the moments of ψ_α to Eq. (2.32). This yields

$$\int \psi_\alpha (g_t + ug_x) d\Xi = 0, \qquad \alpha = 1,2,3,4, \tag{2.35}$$

and the corresponding Euler equations in the $x-$direction are

$$\begin{pmatrix} \rho \\ \rho U \\ \rho V \\ \rho\epsilon \end{pmatrix}_t + \begin{pmatrix} \rho U \\ \rho U^2 + p \\ \rho UV \\ (\rho\epsilon + p)U \end{pmatrix}_x = 0, \tag{2.36}$$

where $\rho\epsilon = \frac{1}{2}\rho(U^2 + V^2 + \frac{K+2}{2\lambda})$ and $p = \rho/2\lambda$.

On the other hand, to the first order of τ, the Chapmann-Enskog expansion gives $f = g - \tau(g_t + ug_x)$. Taking moments of ψ again to the BGK equation with the new f, we get

$$\int \psi_\alpha (g_t + ug_x) d\Xi = \tau \int \psi (g_{tt} + 2ug_{xt} + u^2 g_{xx}) d\Xi, \tag{2.37}$$

from which the Navier-Stokes equations with a dynamic viscous coefficient $\mu = \tau p$ can be obtained,

$$\begin{pmatrix} \rho \\ \rho U \\ \rho V \\ \rho\epsilon \end{pmatrix}_t + \begin{pmatrix} \rho U \\ \rho U^2 + p \\ \rho UV \\ (\rho\epsilon + p)U \end{pmatrix}_x = \begin{pmatrix} 0 \\ S_{1x} \\ S_{2x} \\ S_{3x} \end{pmatrix}_x, \tag{2.38}$$

where

$$
\begin{aligned}
S_{1x} &= \tau p\left[2\frac{\partial U}{\partial x} - \frac{2}{K+2}\frac{\partial U}{\partial x}\right],\\
S_{2x} &= \tau p\frac{\partial V}{\partial x},\\
S_{3x} &= \tau p\left[2\frac{\partial U}{\partial x} + V\frac{\partial V}{\partial x} - \frac{2}{K+2}U\frac{\partial U}{\partial x} + \frac{K+4}{4}\frac{\partial}{\partial x}\left(\frac{1}{\lambda}\right)\right].
\end{aligned}
$$

The derivation from the BGK model in 3D case to the NS equations for a monatomic gas can be found in [87].

The general solution f of the BGK model at a cell interface $x_{i+1/2}$ and time t is

$$f(x_{i+1/2},t,u,v,\xi) = \frac{1}{\tau}\int_0^t g(x',t',u,v,\xi)e^{-(t-t')/\tau}dt' + e^{-t/\tau}f_0(x_{i+1/2}-ut), \tag{2.39}$$

where $x' = x_{i+1/2} - u(t-t')$ is the trajectory of a particle motion and f_0 is the initial gas distribution function f at the beginning of each time step ($t = 0$). Two unknowns g and f_0 must be specified in Eq.(2.39) in order to obtain the solution f.

The directional splitting BGK-NS method is described in [86], where only the physical variables in the normal direction of a cell interface are considered. The initial gas distribution function f_0 is assumed to have the form,

$$f_0 = \begin{cases} g^l[1 + a^l(x - x_{i+1/2}) - \tau(a^l u + A^l)] & , \quad x - x_{i+1/2} \le 0, \\ g^r[1 + a^r(x - x_{i+1/2}) - \tau(a^r u + A^r)] & , \quad x - x_{i+1/2} \ge 0. \end{cases} \tag{2.40}$$

The terms proportional to τ represent the non-equilibrium parts in the Chapman-Enskog expansion of the BGK model. The non-equilibrium parts have no direct contribution to the conservative flow variables, i.e.,

$$
\begin{aligned}
&\int (a^l u + A^l)\psi_\alpha g^l d\Xi = 0,\\
&\int (a^r u + A^r)\psi_\alpha g^l d\Xi = 0.
\end{aligned} \tag{2.41}
$$

After having f_0, the equilibrium state g around ($x = x_{i+1/2}, t = 0$) is constructed as

$$g = g_0\left[1 + (1 - \mathrm{H}[x - x_{i+1/2}])\bar{a}^l(x - x_{i+1/2}) + \mathrm{H}[x - x_{i+1/2}]\bar{a}^r(x - x_{i+1/2}) + \bar{A}t\right], \tag{2.42}$$

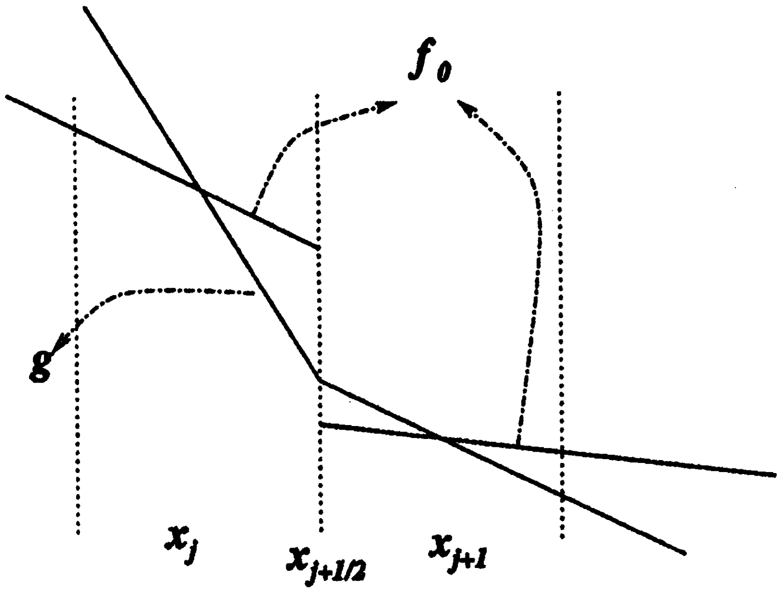


Figure 2.5: The spatial distribution of the initial state f_0 and the equilibrium state g at $t = 0$ for the 2nd-order BGK scheme.

where H$[x - x_{i+1/2}]$ is the Heaviside function defined by

$$H[x - x_{i+1/2}] = \begin{cases} 0 \quad , \quad x - x_{i+1/2} < 0, \\ 1 \quad , \quad x - x_{i+1/2} \geq 0. \end{cases}$$

Here g_0 is a local Maxwellian distribution function located at $x = x_{i+1/2}$. Even though g is continuous at $x = x_{i+1/2}$, it has different slopes at $x < x_{i+1/2}$ and $x > x_{i+1/2}$; see Fig.2.5. In both f_0 and g, $a^l, A^l, a^r, A^r, \bar{a}^l, \bar{a}^r$, and $\bar{A}$ are related to the derivatives of a Maxwellian in space and time. The dependence of $a^l, a^r, ..., \bar{A}$ on the particle velocities can be obtained from a Taylor expansion of a Maxwellian and has the following form,

$$\begin{aligned} a^l &= a_1^l + a_2^l + a_3^l v + a_4^l \frac{1}{2}(u^2 + v^2 + \xi^2) = a_\alpha^l \psi_\alpha, \\ A^l &= A_1^l + A_2^l + A_3^l v + A_4^l \frac{1}{2}(u^2 + v^2 + \xi^2) = A_\alpha^l \psi_\alpha, \\ \bar{A} &= \bar{A}_1 + \bar{A}_2 + \bar{A}_3 v + \bar{A}_4 \frac{1}{2}(u^2 + v^2 + \xi^2) = \bar{A}_\alpha \psi_\alpha, \end{aligned}$$

where $\alpha = 1, 2, 3, 4$, and all coefficients $a_1^l, a_2^l, \ldots, \bar{A}_4$ are local constants.

In the reconstruction step described in the section 2.2.2, the distributions $\bar{\rho}_j(x), \bar{m}_j(x)$, $\bar{n}_j(x)$, and $\bar{\rho}_j\bar{\epsilon}_j(x)$ are obtained inside each cell $x_{j-1/2} \leq x \leq x_{j+1/2}$. At the cell interface $x_{j+1/2}$, the left and right macroscopic states are

$$\bar{w}_j(x_{j+1/2}) = \begin{pmatrix} \bar{\rho}_j(x_{j+1/2}) \\ \bar{m}_j(x_{j+1/2}) \\ \bar{n}_j(x_{j+1/2}) \\ \bar{\rho}_j\bar{\epsilon}_j(x_{j+1/2}) \end{pmatrix}, \qquad \bar{w}_{j+1}(x_{j+1/2}) = \begin{pmatrix} \bar{\rho}_{j+1}(x_{j+1/2}) \\ \bar{m}_{j+1}(x_{j+1/2}) \\ \bar{n}_{j+1}(x_{j+1/2}) \\ \bar{\rho}_{j+1}\bar{\epsilon}_{j+1}(x_{j+1/2}) \end{pmatrix} \tag{2.43}$$

By using the relation between the gas distribution function f and the macroscopic variables Eq.(2.33), at $x_{j+1/2}$, we get

$$\int g^l \psi d\Xi = \bar{w}_j(x_{j+1/2}); \quad \int g^l a^l \psi d\Xi = \frac{\bar{w}_j(x_{j+1/2}) - \bar{w}_j(x_j)}{\Delta x^-}, \tag{2.44}$$

$$\int g^r \psi d\Xi = \bar{w}_{j+1}(x_{j+1/2}); \int g^r a^r \psi d\Xi = \frac{\bar{w}_{j+1}(x_{j+1}) - \bar{w}_{j+1}(x_{j+1/2})}{\Delta x^+}, \tag{2.45}$$

where $\Delta x^- = x_{j+1/2} - x_j$ and $\Delta x^+ = x_{j+1} - x_{j+1/2}$, and the Maxwellians are,

$$\begin{aligned} g^l &= \rho^l \left(\frac{\lambda^l}{\pi}\right)^{\frac{K+2}{2}} e^{-\lambda^l((u-U^l)^2+(v-V^l)^2+\xi^2)}, \\ g^r &= \rho^r \left(\frac{\lambda^r}{\pi}\right)^{\frac{K+2}{2}} e^{-\lambda^r((u-U^r)^2+(v-V^r)^2+\xi^2)}. \end{aligned}$$

All the parameters in g^l and g^r can be uniquely determined from Eq.(2.44) and (2.45), and

$$\begin{pmatrix} \rho^l \\ U^l \\ V^l \\ \lambda^l \end{pmatrix} = \begin{pmatrix} \bar{\rho}_j(x_{j+1/2}) \\ \bar{m}_j(x_{j+1/2})/\bar{\rho}_j(x_{j+1/2}) \\ \bar{n}_j(x_{j+1/2})/\bar{\rho}_j(x_{j+1/2}) \\ \lambda^l \end{pmatrix}, \quad \begin{pmatrix} \rho^r \\ U^r \\ V^r \\ \lambda^r \end{pmatrix} = \begin{pmatrix} \bar{\rho}_{j+1}(x_{j+1/2}) \\ \bar{m}_{j+1}(x_{j+1/2})/\bar{\rho}_{j+1}(x_{j+1/2}) \\ \bar{n}_{j+1}(x_{j+1/2})/\bar{\rho}_{j+1}(x_{j+1/2}) \\ \lambda^r \end{pmatrix},$$

where

$$\lambda_l = \frac{(K+2)\bar{\rho}_j(x_{j+1/2})}{4\left(\bar{\rho}_j\bar{\epsilon}_j(x_{j+1/2}) - \frac{1}{2}(\bar{m}_j^2(x_{j+1/2}) + \bar{n}_j^2(x_{j+1/2}))/\bar{\rho}_j(x_{j+1/2})\right)},$$

and

$$\lambda_r = \frac{(K+2)\bar{\rho}_{j+1}(x_{j+1/2})}{4\left(\bar{\rho}_{j+1}\bar{\epsilon}_{j+1}(x_{j+1/2}) - \frac{1}{2}(\bar{m}_{j+1}^2(x_{j+1/2}) + \bar{n}_{j+1}^2(x_{j+1/2}))/\bar{\rho}_{j+1}(x_{j+1/2})\right)}.$$

The slope a^l in Eq.(2.44) can be computed from,

$$\frac{\bar{w}_j(x_{j+1/2}) - \bar{w}_j(x_j)}{\rho^l \triangle x^+} = M^l_{\alpha\beta} \begin{pmatrix} a^l_1 \\ a^l_2 \\ a^l_3 \\ a^l_4 \end{pmatrix} = M^l_{\alpha\beta} a^l_\beta \quad , \tag{2.46}$$

where

$$M^l_{\alpha\beta} = \frac{1}{\rho^l} \int g^l \psi_\alpha \psi_\beta d\Xi.$$

The above matrix and the direct evaluation of the solution from the above equation are similar to the one in section 2.2.2. For g^r, the matrix $M^r_{\alpha\beta} = \frac{1}{\rho^r} \int g^r \psi_\alpha \psi_\beta d\Xi$ has the same structure as $M^l_{\alpha\beta}$, and the slope a^r in Eq.(2.45) can be obtained similarly. Further A^l and A^r can be determined from Eq.(2.41), which are

$$\begin{aligned} M^l_{\alpha\beta} A^l_\beta &= -\frac{1}{\rho^l} \int g^l a^l u \psi_\alpha d\Xi, \\ M^r_{\alpha\beta} A^r_\beta &= -\frac{1}{\rho^r} \int g^r a^r u \psi_\alpha d\Xi. \end{aligned} \tag{2.47}$$

Since $M^l_{\alpha\beta}$, $M^r_{\alpha\beta}$, and the right-hand sides of the above equations are known, the parameters A^l and A^r can be obtained subsequently using the previous method in Chapter 2.2.2.

In the equilibrium state g, g_0 at the cell interface is defined by

$$g_0 = \rho_0 \left(\frac{\lambda_0}{\pi} \right)^{\frac{K+2}{2}} e^{-\lambda_0((u-U_0)^2 + (v-V_0)^2 + \xi^2)}.$$

The corresponding values of ρ_0, U_0, V_0, and λ_0 in g_0 can be determined as follows. Taking the limit $t \rightarrow 0$ in Eq.(2.39) and substituting its solution into Eq.(2.34), the conservation constraint at $(x = x_{j+1/2}, t = 0)$ gives

$$\int g_0 \psi d\Xi = w_0 = \int_{u>0} \int g^l \psi d\Xi + \int_{u>0} \int g^r \psi d\Xi, \tag{2.48}$$

where $w_0 = (\rho_0, m_0, n_0, \rho_0 \epsilon_0)^T$ is the macroscopic conservative variables located at the cell interface at time $t = 0$. After having w_0, λ_0 in g_0 can be found from

$$\lambda_0 = (K+2)\rho_0 \bigg/ \left(4\left(\rho_0\epsilon_0 - \frac{1}{2}(m_0^2 + n_0^2)/\rho_0\right)\right)$$

Then, $\bar{a}^l$ and $\bar{a}^r$ of g in Eq.(2.39) can be calculated from

$$\frac{w_0 - \bar{w}_j(x_j)}{\rho_0 \Delta x^-} = \bar{M}^0_{\alpha\beta} \begin{pmatrix} \bar{a}^l_1 \\ \bar{a}^l_2 \\ \bar{a}^l_3 \\ \bar{a}^l_4 \end{pmatrix} = \bar{M}^0_{\alpha\beta}\bar{a}^l_\beta \quad , \tag{2.49}$$

and

$$\frac{\bar{w}_j(x_{j+1}) - w_0}{\rho_0 \Delta x^+} = \bar{M}^0_{\alpha\beta} \begin{pmatrix} \bar{a}^r_1 \\ \bar{a}^r_2 \\ \bar{a}^r_3 \\ \bar{a}^r_4 \end{pmatrix} = \bar{M}^0_{\alpha\beta}\bar{a}^r_\beta \quad , \tag{2.50}$$

where the matrix $\frac{1}{\rho_0}\bar{M}^0_{\alpha\beta} = \int g_0 \psi_\alpha \psi_\beta d\Xi$ is known; see Fig.2.5. Hence, $(\bar{a}^l_1, \bar{a}^l_2, \bar{a}^l_3, \bar{a}^l_4)^T$ and $(\bar{a}^r_1, \bar{a}^r_2, \bar{a}^r_3, \bar{a}^r_4)^T$ can be found following the same procedure as a^l and a^r.

Up to now, we have determined all parameters in the initial gas distribution function f_0 and the equilibrium state g at the beginning of each time step $t = 0$, the gas distribution function f at a cell interface can be expressed as

$$\begin{aligned} f(x_{i+1/2}, t, u, v, \xi) &= (1 - e^{-t/\tau}) g_0 \\ &\quad + (\tau(-1 + e^{-t/\tau}) + t e^{-t/\tau})(\bar{a}^l \mathrm{H}[u] + \bar{a}^r(1 - \mathrm{H}[u])) u g_0 \\ &\quad + \tau(t/\tau - 1 + e^{-t/\tau})\bar{A} g_0 + e^{-t/\tau}((1 - (t+\tau) u a^l)\mathrm{H}[u] g^l \\ &\quad + (1 - u(t+\tau)a^r)(1 - \mathrm{H}[u]) g^r) \\ &\quad + e^{-t/\tau}(-\tau A^l \mathrm{H}[u] g^l - \tau A^r (1 - \mathrm{H}[u]) g^r). \end{aligned} \tag{2.51}$$

The only unknown left in the above expression is $\bar{A}$. Since both f and g contain $\bar{A}$, the integration of the conservation constraint Eq.(2.34) at $x_{i+1/2}$ over the whole time step Δt gives

$$\int_0^{\Delta t} \int (g - f)\psi d\Xi = 0,$$

which goes to

$$\begin{aligned}
\bar{M}^0_{\alpha\beta}\bar{A}_\beta &\equiv \frac{1}{\rho_0}(\partial\rho/\partial t, \partial m/\partial t, \partial n/\partial t, \partial E/\partial t)^T \\
&= \frac{1}{\rho_0}\int[\gamma_1 g_0 + \gamma_2 u(\bar{a}^l H[u] + \bar{a}^r(1-H[u]))g_0 + \gamma_3(H[u]g^l + (1-H[u])g^r) \\
&\quad +\gamma_4 u(a^l H[u]g^l + a^r(1-H[u])g^r) + \gamma_5((a_l u + A^l)H[u]g^l \\
&\quad +(a^r u + A^r)(1-H[u])g^r)]\psi_\alpha d\Xi, \qquad (2.52)
\end{aligned}$$

where

$$\begin{aligned}
\gamma_0 &= \Delta t - \tau(1-e^{-\Delta t/\tau}), \\
\gamma_1 &= -(1-e^{-\Delta t/\tau})/\gamma_0, \\
\gamma_2 &= (-\Delta t + 2\tau(1-e^{-\Delta t/\tau}) - \Delta t e^{-\Delta t/\tau})/\gamma_0, \\
\gamma_3 &= (1-e^{-\Delta t/\tau})/\gamma_0, \\
\gamma_4 &= (\Delta t e^{-\Delta t/\tau} - \tau(1-e^{-\Delta t/\tau}))/\gamma_0, \\
\gamma_5 &= -\tau(1-e^{-\Delta t/\tau})/\gamma_0,
\end{aligned}$$

Finally, the time-dependent numerical fluxes in the normal-direction across the cell interface can be computed by

$$\mathcal{F}_{i+1/2} = \begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_m \\ \mathcal{F}_n \\ \mathcal{F}_{\rho\epsilon} \end{pmatrix}_{i+1/2} = \int u \begin{pmatrix} 1 \\ u \\ v \\ \frac{1}{2}(u^2+v^2+\xi^2) \end{pmatrix} f(x_{i+1/2}, t, u, v, \xi)d\Xi, \qquad (2.53)$$

where $f(x_{i+1/2}, t, u, v, \xi)$ is given by Eq.(2.37). By integrating the above equation to the whole time step, we can get the total mass, momentum and energy transport in an explicit scheme. Once we get the fluxes, the flow variables inside each cell can be updated through

$$W_j^{n+1} = W_j^n + \frac{1}{\Delta x}\int_0^{\Delta t}(\mathcal{F}_{j-1/2} - \mathcal{F}_{j+1/2})dt. \qquad (2.54)$$

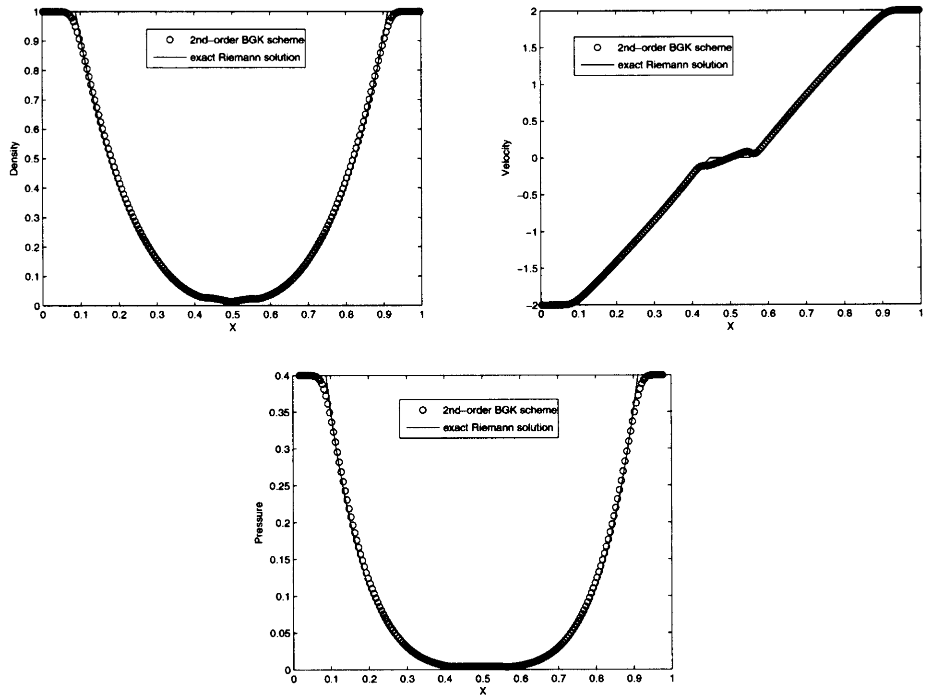


Figure 2.6: Exact Euler solutions (solid line) and numerical solutions (× symbol) using the 2nd-order gas-kinetic BGK scheme at time $t = 0.15$ units.

Usually, the above gas-kinetic BGK scheme for the Navier-Stokes equations is called the directional splitting BGK-NS method, which is also referred in the following Chapters.

The Sjögreen test case has been calculated by the exact solution from collisionless Boltzmann equation (2.17), and the 1st and 2nd KFVS schemes respectively. The results from the 2nd-order BGK scheme are shown in Fig.2.6, which are closer to the exact Riemann solution by comparing it with Fig.2.1, Fig.2.2 and Fig.2.4. Note that the BGK scheme solves the Navier-Stokes equations instead of the Euler ones. As realized, the gas-kinetic BGK scheme has advantages for the capturing of strong rarefaction waves.

For the simulation of viscous and heat conducting flow, a subject of debate on the BGK scheme is that the scheme has a fixed Prandtl number with unity. This is a well-known result for the BGK model. In order to change the above Prandtl number to any realistic value, different methods to resolve this problem have been proposed. A simple method is presented in [86]. In the BGK method, we have obtained explicitly the time-dependent gas distribution function f at the cell interface Eq. (2.51). Therefore, the time-dependent heat flux can be evaluated precisely,

$$q = \frac{1}{2}\int (u-U)((u-U)^2 + (v-V)^2 + \xi^2) f d\Xi, \tag{2.55}$$

where the average velocities U and V are defined by

$$U = \int u f d\Xi \Big/ \int f d\Xi \,, \quad V = \int v f d\Xi \Big/ \int f d\Xi \,.$$

Then, the easiest way to fix the Prandtl number for the BGK scheme is to modify the energy flux by subtracting the above heat flux (2.55) and adding another one with a correct Prandtl number,

$$\mathcal{F}_{\rho\epsilon}^{new} = \mathcal{F}_{\rho\epsilon} + \left(\frac{1}{Pr} - 1\right) q, \tag{2.56}$$

where $\mathcal{F}_{\rho\epsilon}$ is the energy flux in Eq.(2.53).

2.4 Multidimensional gas-kinetic BGK scheme for viscous flows

In recent years, the gas-kinetic scheme based on the Bhatnagar-Gross-Krook (BGK) model for the Navier-Stokes equations has been developed by implementing a multidimensional particle propagation mechanism in the flux evaluation, where the gradients of flow variables in both normal and tangential directions of a cell interface are explicitly included. The Multidimensional gas-kinetic BGK scheme is to pave the way to extend the current approach directly to the flow computation with unstructured mesh, where the flow gradients in both parallel and perpendicular directions around a cell interface can be explicitly taken into account in a viscous flow simulation. In [52], the Multidimensional gas-kinetic BGK scheme has been applied on unstructured mesh for 3D computation of viscous flows. An important issue in the design and development of aerospace vehicles is the accurate calculation of various types of flow phenomena on aerodynamic performance and aero-thermal loads. Evaluation of aerodynamic heating during reentry flight is one of the key issues, where complicated flow phenomena, i.e., shock boundary layer interaction, flow separation, and viscous/inviscid interaction, will be encountered. To the current stage, the computation of the interaction between a shock wave and a separated region in the hypersonic flow is still a very challenging problem in computational fluid dynamics. In terms of heat transfer, significant differences between the computational results and experiments indicate that further investigation and design of more accurate viscous flow solvers are needed to understand these phenomena. The severe heating rates produced by the viscous/inviscid interactions and by the shock/shock interactions can cause catastrophic failure for the vehicles in the hypersonic flight. Over the past 15 years there have been concerted efforts both in Europe and America to validate the Navier-Stokes and DSMC-based methods in the description of complex hypersonic flows. Extensive codes validation have been conducted. A number of experiments specifically designed with simple model configuration in the laminar hyper velocity flow have been constructed and continuously being used to examine complex flow phenomena, such as the viscous/inviscid interactions.

Based on the gas-kinetic theory, the Navier-Stokes equations can be derived from the Boltzmann equation using the Chapman-Enskog expansion. In the gas-kinetic representation, all flow variables become the moments of a single particle distribution function. Since a gas distribution function is used to describe both equilibrium and non-equilibrium states, the inviscid and viscous fluxes are obtained simultaneously in the gas-kinetic scheme. However, in the traditional upwinding schemes for the Navier-Stokes solutions, an operator splitting method is commonly adopted, where the Riemann solver or equivalent flux evaluation based on two constant states is used for the inviscid part and the central differences for the viscous and heat conduction parts. Theoretically, the use of different flow distributions for the inviscid and viscous parts is artificial, which may introduce numerical error in the hypersonic viscous flow computation with strong coupling of the inviscid and viscous flow interactions. For example, the dissipative characteristic of upwind schemes in the regions with sharp gradients in the boundary layer may trigger unsteady mechanism to prevent from obtaining steady state solution. Even though high-order discretization can be introduced for the inviscid and viscous parts separately, an operator splitting error due to the different initial condition (or equivalently kinematic dissipation) can be hardly eliminated, especially in the cases with severe coupling between transport and dissipative heating. On the contrary, for the gas-kinetic BGK scheme, both inviscid and viscous parts are recovered in a single gas distribution function f, which is based on the same initial condition. As pointed out earlier [86], the gas-kinetic description is capable of giving a more complete description of the non-equilibrium flow. Even though the non-equilibrium parts in a gas-distribution function have no direct contribution to the macroscopic mass, momentum and energy, but they do contribute to the higher moments, such as the fluxes. In [88], the multidimensional gas-kinetic BGK scheme and its application to the hypersonic viscous flow are presented.

For a multi-dimensional scheme, the general solution f of the BGK model at a cell interface $x_{i+1/2}, y_j$ and time t is

$$f(x_{i+1/2}, y_j, t, u, v, \xi) = \frac{1}{\tau}\int_0^t g(x', y', t', u, v, \xi)e^{-(t-t')/\tau}dt' + e^{-t/\tau}f_0(x_{i+1/2} - ut, y_j - vt), \tag{2.57}$$

where $x' = x_{i+1/2} - u(t-t')$, $y' = y_j - v(t-t')$ are the trajectory of a particle motion and f_0 is the initial gas distribution function f at the beginning of each time step ($t = 0$). Two unknowns g and f_0 must be specified in Eq.(2.57) in order to obtain the solution f. Based on the Chapman-Enskog expansion of the BGK model Eq.(2.31), the gas distribution function up to the Navier-Stokes order at the point ($x_{i+1/2} = 0, y_j = 0$) and time ($t = 0$) has the form [55],

$$f_{NS} = g - \tau(g_t + ug_x + vg_y),$$

where $\phi_1 = -\tau(g_t + ug_x + vg_y)$ has to satisfy the compatibility condition $\int \psi_\alpha \phi_1 d\Xi = 0$.

To the 2nd-order accuracy, the gas distribution function can be approximated as

$$f_{NS} = g + g_x(x - x_{i+1/2}) + g_y(y - y_j) - \tau(g_t + ug_x + vg_y).$$

Therefore, in the multidimensional gas-kinetic scheme with the initial discontinuous macroscopic variables at the left and right hand sides of a cell interface the initial gas distribution function f_0 has the form,

$$f_0 = \begin{cases} g^l(1 + a^l(x - x_{i+1/2}) + b^l y - \tau(a^l + b^l + A^l)) \quad , & x - x_{i+1/2} < 0, \\ g^r(1 + a^r(x - x_{i+1/2}) + b^r y - \tau(a^r + b^r + A^r)), & x - x_{i+1/2} \geq 0, \end{cases} \tag{2.58}$$

where $g^l a^l, g^r a^r, g^l A^l$ and $g^r A^r$ are related to the spatial and temporal derivatives of the equilibrium states. Different from directional splitting method, $g^l b^l$ and $g^r b^r$ correspond to the gradients in the tangential direction along the cell interface. The terms proportional to τ represent the non-equilibrium parts in the Chapman-Enskog expansion, and have no direct contribution to the conservative flow variables, i.e.,

$$\begin{aligned} &\int (a^l u + b^l v + A^l)\psi_\alpha g^l d\Xi = 0, \\ &\int (a^r u + b^r v + A^r)\psi_\alpha g^r d\Xi = 0, \end{aligned} \tag{2.59}$$

which are the exact equations to determine A^l and A^r. In the above f_0, g^l, g^r, a^l ,and a^r have the same definition as the corresponding parameters in the previous direction splitting BGK-NS method.

After having f_0, the equilibrium state g around $(x = x_{i+1/2}, y = y_j, t = 0)$ is constructed as

$$g = g_0(1+(1-H[x-x_{i+1/2}])\bar{a}^l(x-x_{i+1/2})+H[x-x_{i+1/2}]\bar{a}^r(x-x_{i+1/2})+\bar{b}y+\bar{A}t), \tag{2.60}$$

where $\bar{b}$ is the additional term related to the flow variation in the tangential direction, and g_0 is a local Maxwellian distribution function located at $(x = x_{i+1/2}, y = y_j)$. In both f_0 and g, $a^l, A^l, a^r, A^r, \bar{a}^l, \bar{a}^r, \bar{b}$, and $\bar{A}$ are related to the derivatives of a Maxwellian in space and time. Similar to the previous BGK-NS method, a^l and

a^r can be uniquely determined. With the definition of Maxwellian distributions

$$g^l = \rho^l \left(\frac{\lambda^l}{\pi}\right)^{\frac{K+2}{2}} e^{-\lambda^l((u-U^l)^2+(v-V^l)^2+\xi^2)},$$

and

$$g^r = \rho^r \left(\frac{\lambda^r}{\pi}\right)^{\frac{K+2}{2}} e^{-\lambda^r((u-U^r)^2+(v-V^r)^2+\xi^2)},$$

in the tangential direction b^l and b^r can be obtained from

$$\int g^l b^l \psi_\alpha d\Xi = \vec{t} \cdot \nabla w^l; \quad \int g^r b^r \psi_\alpha d\Xi = \vec{t} \cdot \nabla w^r,$$

where $\vec{t}$ is the unit vector in the tangential direction along the cell interface, and ∇w^l and ∇w^r are gradients of macroscopic variables $w = (\rho, \rho u, \rho v, \rho\epsilon)^T$ on the left and right hand sides of a cell interface. After determining the terms a^l, b^l, a^r, and b^r, A^l and A^r in f_0 can be found from Eq.(2.59), which are

$$\begin{aligned} M^l_{\alpha\beta} A^l_\beta &= -\frac{1}{\rho^l} \int (a^l u + b^l v) \psi_\alpha g^l d\Xi, \\ M^r_{\alpha\beta} A^r_\beta &= -\frac{1}{\rho^r} \int (a^r u + b^r v) \psi_\alpha g^r d\Xi, \end{aligned} \tag{2.61}$$

where $M^l_{\alpha\beta} = \int g^l \psi_\alpha \psi_\beta d\Xi / \rho^l$, and $M^r_{\alpha\beta} = \int g^r \psi_\alpha \psi_\beta d\Xi / \rho^r$.

For the equilibrium state g in Eq.(2.60), g_0 is still defined by Eq.(2.48), and $\bar{a}^l$ and $\bar{a}^r$ are also evaluated by Eq.(2.49) and (2.50) respectively. The term $\bar{b}$ is calculated from

$$\int g_0 \bar{b} \psi d\Xi = \int_{u>0} g^l b^l \psi d\Xi + \int_{u<0} g^r b^r \psi d\Xi. \tag{2.62}$$

Now, we have determined all parameters in the initial gas distribution function f_0 and equilibrium state g at the beginning of each time step $t = 0$. After substituting Eqs.(2.58) and (2.60) into Eq.(2.57), the gas distribution f at a cell interface can be expressed as,

$$\begin{aligned}
f(x_{i+1/2}, y_j, t, u, v, \xi) \;=\;& (1 - e^{-t/\tau})g_0 \\
&+ (\tau(-1 + e^{-t/\tau}) + te^{-t/\tau})(\bar{a}^l \mathrm{H}[u] + \bar{a}^r(1 - \mathrm{H}[u]) + \bar{b}v)ug_0 \\
&+ \tau(t/\tau - 1 + e^{-t/\tau})\bar{A}g_0 + e^{-t/\tau}((1 - (t + \tau)(ua^l + vb^l))\mathrm{H}[u]g^l \\
&+ (1 - (t + \tau)(ua^r + vb^r)(1 - \mathrm{H}[u])g^r) \\
&+ e^{-t/\tau}(-\tau A^l \mathrm{H}[u]g^l - \tau A^r(1 - \mathrm{H}[u])g^r). \qquad (2.63)
\end{aligned}$$

The only unknown left in the above expression is $\bar{A}$, which is contained in both f (Eq.(2.63)) and g (Eq.(2.60)). The integration of the conservation constraint at $x_{i+1/2}, y_j$ over the whole time step $\triangle t$ gives

$$\int_0^{\triangle t} \int (g - f)\psi_\alpha dt d\Xi = 0,$$

from which $\bar{A}$ can be obtained. The multidimensional BGK scheme is similar to the direction splitting gas-kinetic scheme except additional terms b^l, b^r, and $\bar{b}$ related to the flow variations in the tangential direction.

Finally, the time-dependent numerical fluxes in the normal-direction across the cell interface can be computed by

$$\mathcal{F}_{i+1/2,j} == \int u \begin{pmatrix} 1 \\ u \\ v \\ \frac{1}{2}(u^2 + v^2 + \xi^2) \end{pmatrix} f(x_{i+1/2}, y_j, t, u, v, \xi) d\Xi, \qquad (2.64)$$

where

$$\mathcal{F}_{i+1/2,j} = \begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_m \\ \mathcal{F}_n \\ \mathcal{F}_{\rho\epsilon} \end{pmatrix}_{i+1/2,j} .$$

The flow variables $W_{i,j} = (\rho, (\rho u), (\rho v), (\rho\epsilon))^T_{i,j}$ inside cell (i,j) can be updated by

$$W^{n+1}_{i,j} = W^n_{i,j} + \int_0^{\Delta t} \left(\frac{1}{\Delta x}(\mathcal{F}_{i-1/2,j} - \mathcal{F}_{i+1/2,j}) + \frac{1}{\Delta y}(\mathcal{F}_{i,j-1/2,j} - \mathcal{F}_{i,j+1/2,j}) \right) dt. \tag{2.65}$$

In summary, the above multidimensional gas-kinetic BGK scheme includes both normal and tangential slopes across a cell interface in the flux calculation. Theoretically, the above method can enhance the accuracy of the BGK-NS scheme although it still remains second-order accuracy. Numerically, it has been shown, such as in [42], that the multi-dimensional scheme can capture some physical phenomena which are absent from the directional splitting method, such as the thermal creeping flow. The advantages of the multidimensional gas-kinetic BGK scheme are the following:

1.The scheme is able to capture the characteristics flow more accurately with curved boundary or nonuniform temperature, such as the inverted velocity distribution in rarefied cylindrical Couette flow and the weak flow field induced by the temperature gradient of a body;

2. It can also give better results in high-speed micro-channel flow and power-law fluid flow between concentric rotating cylinders, when compared with direct simulation Monte Carlo and analytic solutions.

2.5 Multidimensional BGK scheme for nearly incompressible viscous flows

As stated in the above two sections, the gas-kinetic BGK scheme is focusing on the high speed compressible flows described by the Navier-Stokes equations. For some benchmark cases with the low Mach number, the above multidimensional gas-kinetic BGK scheme can be much simplified to simulate the approximate incompressible isothermal viscous flows. For the incompressible flow, there is no discontinuity. For a smooth flow, the distribution function f_0 in Eq.(2.58) can be simplified as $g^l = g^r$, $a^l = a^r$, and $b^l = b^r$. Consequently, Eq.(2.62) gives $g_0 = g^l = g^r$. Similarly, $\bar{a}^l = \bar{a}^r = a^l = a^r$ and $\bar{b} = b^l = b^r$ can be derived. The incompressible flow can be simulated using BGK scheme at low Mach number. This is equivalent to the artificial compressibility method. In this situation, the flow variables $w = (\rho, \rho u, \rho v)$, where the flow energy $\rho\epsilon$ can be ignored, is not necessary to be divided into left and right states with a discontinuity at each cell interface. Here, with the definition $\partial g/\partial x = ag$, $\partial g/\partial y = bg$, and $\partial g/\partial t = Ag$, we have

$$\begin{aligned} a &= a_1 + a_2 u + a_3 v, \\ b &= b_1 + b_2 u + b_3 v, \\ A &= A_1 + A_2 u + A_3 v. \end{aligned}$$

For the isothermal N-S equations, $\lambda = m/2kT$ is a constant, where k, T, and m are the Boltzmann constant, temperature, and molecular mass. With a given Mach number M, λ can be determined by

$$\lambda = \frac{M^2}{2\mathbf{U}_\infty^2},$$

where $\mathbf{U}_\infty = (U_\infty, V_\infty)$ is the upstream velocity, and $\mathbf{U}_\infty^2 = U_\infty^2 + V_\infty^2$. In the 2D case, in order to accurately capture the incompressible flow behavior the gas is limited to move only in a plane without any random motion in the third direction. Then, the internal degree of freedom K is assumed to be equal to 0, while $\gamma = 2$ is obtained with the relation of $\gamma = (K+4)/(K+2)$. Therefore, the local Maxwellian equilibrium distribution function at the point $\mathbf{x}_0 = (x_0, y_0)$ with the particle velocity $\mathbf{u} = (u, v)$ is defined by

$$g(\mathbf{x}_0, \mathbf{u}, 0) = \rho(\frac{\lambda}{2\pi})e^{-\lambda((u-U)^2+(v-V)^2)}. \tag{2.66}$$

The initial condition $f_0(\mathbf{x}, \mathbf{u}, t = 0)$ at the beginning of each time-step around $\mathbf{x}$ is assumed to be a distribution function truncated to the Navier-Stokes order,

$$f_0(\mathbf{x}, \mathbf{u}, 0) = g(\mathbf{x}_0, \mathbf{u}, 0) + f^{(1)}(\mathbf{x}_0, \mathbf{u}, 0) + (\mathbf{x} - \mathbf{x}_0) \cdot \nabla g.$$

For the BGK model, the non-equilibrium state $f^{(1)}$ is $f^{(1)} = -\tau D g$ with $D = \partial_t + \mathbf{u} \cdot \nabla$. The specific form becomes

$$f^{(1)} = -\tau(\partial g/\partial t + \mathbf{u} \cdot \nabla g) = -\tau(au + bv + A)g.$$

Therefore, f_0 at time $t = 0$ can be written as

$$f_0(\mathbf{x}, \mathbf{u}, 0) = g\left(1 - \tau(au + bv + A) + a(x - x_0) + b(y - y_0)\right). \tag{2.67}$$

The equilibrium distribution in space and time around $(\mathbf{x}_0, t = 0)$ can be obtained from the expansion of an equilibrium state

$$\begin{aligned} g(\mathbf{x}, \mathbf{u}, t) &= g(\mathbf{x}, \mathbf{u}, 0) + (\partial g/\partial x)(x - x_0) + (\partial g/\partial x)(y - y_0) + (\partial g/\partial t)t \\ &= g(1 + a(x - x_0) + b(y - y_0) + At). \end{aligned}$$

Substituting both g and f_0 into the general integral solution (2.57) in the above section, the distribution function f at a cell interface $(x = x_0)$ becomes

$$f(\mathbf{x}_0, \mathbf{u}, t) = g\left(1 - \tau(au + bv + A) + At\right). \tag{2.68}$$

In $f(\mathbf{x}_0, \mathbf{u}, t)$, the coefficients a_1, a_2, a_3 and b_1, b_2, b_3 can be found from the relation

$$\int\int \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} agdudv = \begin{pmatrix} \partial\rho/\partial x \\ \partial(\rho u)/\partial x \\ \partial(\rho v)/\partial x \end{pmatrix}, \quad \int\int \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} bgdudv = \begin{pmatrix} \partial\rho/\partial y \\ \partial(\rho u)/\partial y \\ \partial(\rho v)/\partial y \end{pmatrix}.$$

An algorithm similar to the one in previous Chapter 2.2.2 can be used. The interpolation of macroscopic variables $\rho, \rho u, \rho v$, and their derivatives in space around $\mathbf{x}_0$ is presented in Appendix A of [89]. After determining a and b, the term A is obtained from the compatibility condition

$$\int\int \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} f^{(1)}dudv = \int\int \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} (-\tau(au + bv + A)g)dudv = 0,$$

which gives

$$\int\int \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} Agdudv = -\int\int \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} (au + bv)gdudv.$$

Let's define (x_i, y_j) as the center point of the space element $\Omega_{i,j}$ with four cell interfaces $(x_{i-1/2}, y_j)$, $(x_{i+1/2}, y_j)$, $(x_i, y_{j-1/2})$, and $(x_i, y_{j+1/2})$. Therefore, the fluxes across the cell interface $(x_{i+1/2}, y_j)$ in the x-direction can be evaluated by

$$\mathcal{F}_{i+1/2,j} = \begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_{\rho u} \\ \mathcal{F}_{\rho v} \end{pmatrix}_{i+1/2,j} = \int\int u \begin{pmatrix} 1 \\ u \\ v \end{pmatrix} f(\mathbf{x}_{i,j}, \mathbf{u}, t)dudv.$$

Similarly, $\mathcal{F}_{i,j+1/2}$, the fluxes at the cell interface $(x_i, y_{j+1/2})$ in the y direction, can be constructed. Therefore, the flow variables are updated by

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{\Omega_{i,j}}^{n+1} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \end{pmatrix}_{\Omega_{i,j}}^{n} + \int_0^{\Delta t} \left(\frac{(\mathcal{F}_{i-1/2,j} - \mathcal{F}_{i+1/2,j})}{\Delta x} + \frac{(\mathcal{F}_{i,j-1/2} - \mathcal{F}_{i,j+1/2})}{\Delta y} \right) dt,$$

which presents a accurate Navier-Stokes flow solver in the incompressible limit.

Chapter 3

Adaptive grid method for two dimensional viscous flows

In the past two decades, various adaptive grid methods have been developed for the numerical solutions in the physical and engineering applications, such as fluid dynamics, hydraulics, combustion, heat transfer, and material science. Physical problems in these areas, such as shock waves, boundary layers, and detonative waves, may require extremely fine mesh to resolve accurately large solution variations. The use of well-refined uniform mesh becomes computationally prohibitive when dealing with systems in multi-dimensions. Adaptive grid methods not only produce a high grid-point density in large gradient regions to improve the accuracy of numerical solution, but also decrease the cost of numerical calculation in comparison with the uniform grid. Currently, existing adaptive grid methods can be summarized as follows: the variational approach of Winslow [83], Brackbill et al. [8, 9], Ren and Wang [63], and Tang and Tang [74]; finite element methods of Millers [53], and Davis Flaherty et al. [17]; moving mesh PDEs of Russell et al. [11, 70], Li and Petzold [44], and Ceniceros and Hou [13]; and moving mesh methods based on harmonic mapping of Dvinsky [21] and Li et al. [43]; and many others.

Many existing adaptive methods are mainly targeting on the inviscid flow problems. For the viscous solution, a few finite element adaptive methods have been developed for the incompressible Navier-Stokes equations [19, 56]. For the compressible flow applications, the accurate capturing of the physical structures, such as the viscous and heat conducting boundary layer and the separation bubble around an airfoil, are critically important. In these regions, large velocity and temperature gradients may exist. In order to capture these phenomena in an efficient way, an adaptive moving mesh method is an optimum choice. Therefore, it is very important to study and develop adaptive grid method for compressible viscous flow simulations.

As we have introduced in Chapter 2, in the gas-kinetic theory the Navier-Stokes equations can be derived from the Boltzmann equation using the Chapman-Enskog expansion [14]; in the gas-kinetic representation, all macroscopic flow variables are the moments of a single particle distribution function and the particle movement is basically the linear transport and collision. Moreover, in the past years, the gas-kinetic BGK scheme has been well developed for the compressible viscous flow computations [86], which is specifically accurate for the supersonic viscous and heat conducting flow [88, 71]. Since a gas distribution function describes both equilibrium and non-equilibrium flow properties, the inviscid and viscous fluxes are obtained simultaneously in the gas-kinetic scheme under the initial condition of a discontinuous flow distribution with slopes around a cell interface, i.e., the condition similar to the generalized Riemann problem.

This chapter focuses on the development of a gas-kinetic scheme on an adaptive grid for viscous flow computations. The scheme mainly composes two parts. The first part is about the mesh redistribution and adaptation and the second part is the flow update on a newly constructed mesh. The former one is based on the adaptive grid method proposed by Tang and Tang [74]. Since the velocity is an important quantity for the viscous flow, we are going to use velocity gradient to define the monitor function, then the mesh will be redistributed according to this function by solving elliptic equations. On other hand, the gas-kinetic BGK-NS scheme for the Navier-Stokes fluxes will be used to update the flow variables for each control volume [86, 88, 89], where a time-dependent gas distribution function is obtained from an initially discontinuous / continuous flow distribution. For the viscous flow computation, the use of adaptive mesh much improves the efficiency and accuracy of the method in comparison with the ones with static mesh points. Many numerical examples will be used to validate the current adaptive grid method, where the grid points can be easily moved and concentrated on the regions with large density and velocity gradients, such as the boundary layer and multi-material interface.

3.1 A Mesh-redistribution Method

This section contains two subsections, which are mesh generation based on the variational approach and conservative flow quantities redistribution to the newly generated mesh.

3.1.1 Mesh generation based on the variational approach

In this chapter we will focus on the two dimensional space (2D), since in the 1D case fine mesh can be adopted. In 2D, the widely used mesh generation techniques are based on the variational approaches. The pioneer work was proposed by Winslow [83]. Let $\vec{x} = (x, y)$ and $\vec{\xi} = (\xi, \eta)$ denote the physical and computational coordinates. A coordinate mapping from the computational domain Ω_c to the physical domain Ω_p is given by

$$x = x(\xi, \eta), \quad y = y(\xi, \eta), \tag{3.1}$$

and the inverse map is

$$\xi = \xi(x, y), \quad \eta = \eta(x, y). \tag{3.2}$$

The specific map is obtained by minimization of a mesh adaptation functional of the following form

$$E(\xi, \eta) = \frac{1}{2} \int_{\Omega_p} (\nabla \xi^T {G_1}^{-1} \nabla \xi + \nabla \eta^T {G_2}^{-1} \nabla \eta) dxdy, \tag{3.3}$$

where G_1 and G_2 are symmetric positive definite (SPD) matrices which are formally called monitor functions. The Euler-Lagrange equations for $E(\xi, \eta)$ become,

$$\nabla \cdot ({G_1}^{-1} \nabla \xi) = 0, \quad \nabla \cdot ({G_2}^{-1} \nabla \eta) = 0, \tag{3.4}$$

which gives the coordinate transformation in mesh generation and adaptation. Grid generation is basically to obtain the curvilinear coordinate system (3.1) from the above elliptic system (3.4). Usually, after solving the linear system (3.4) for $\vec{\xi}(\vec{x})$, we have to find the inverse map to obtain $\vec{x}(\vec{\xi})$, which will be expensive. On the contrary, we can directly solve the corresponding equations on the computational domain Ω_c by interchanging the dependent and independent variables in (3.4). However, the obtained equations are more complicated, and massive computations are required. An alternative approach, as suggested by Ceniceros and Hou [13], is to consider a functional defined in the computational domain directly,

$$\tilde{E}[\xi, \eta] = \frac{1}{2} \int_{\Omega_c} (\tilde{\nabla}^T x G_1 \tilde{\nabla} x + \tilde{\nabla}^T y G_2 \tilde{\nabla} y) d\xi d\eta, \tag{3.5}$$

to replace the conventional functional (3.3), where $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^T$. The corresponding Euler-Lagrange equations become

$$\tilde{\nabla} \cdot (G_1 \tilde{\nabla} x) = 0, \quad \tilde{\nabla} \cdot (G_2 \tilde{\nabla} y) = 0. \tag{3.6}$$

Therefore, the mesh distribution in the physical space can be directly obtained by solving (3.6), which is much simpler than the conventional variational approach (3.3). However, the system (3.6) can generate degenerating grids in some concave regions [21]. Even though it is more complicated, the original system (3.4) is more accurate and reliable than the above simple one (3.6). In the current thesis, all numerical examples have simple geometry, so Eq.(3.6) will be used for the mesh generation.

For any adaptive mesh method, choosing an appropriate monitor function is very important. Based on the simplicity considerations, we will use a directional splitting monitor function, i.e., $G_1 = G_2 = diag\{w_1, w_2\}$, where the functions w_1, w_2 are defined by

$$w_1 = (1 + \alpha_1|\psi|^2 + \beta_1|\frac{\partial\psi}{\partial\xi}|^2)^{\frac{\gamma_1}{2}}, \quad w_2 = (1 + \alpha_2|\psi|^2 + \beta_2|\frac{\partial\psi}{\partial\eta}|^2)^{\frac{\gamma_2}{2}}, \tag{3.7}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2$ are some nonnegative constants, and their optimum values depend on flow problems. The selection of these parameters is mainly based on the physical intuition and numerical testing. Here ψ can be chosen as density, velocity, entropy, or error estimate. But, for the viscous flow, the use of fluidb velocity seems to be a good choice.

The specific steps for constructing a adaptive mesh are the following. We firstly divide the computational domain $\Omega_c = \{(\xi,\eta)|1 \geq \xi, \eta \geq 0\}$ into the square mesh:

$$\{(\xi_j, \eta_k)|\xi_j = \frac{j}{J_x+1}, \eta_k = \frac{k}{J_y+1}; 0 \leq j \leq J_x + 1, 0 \leq k \leq J_y + 1\},$$

then, a second-order central difference scheme is used to discretize the mesh generation Eq.(3.6)

$$\begin{aligned} \frac{\Delta_-^j((w_1)_{j+\frac{1}{2},k}\Delta_+^j x_{j,k})}{(\Delta\xi)^2} + \frac{\Delta_-^k((w_2)_{j,k+\frac{1}{2}}\Delta_+^k x_{j,k})}{(\Delta\eta)^2} &= 0 \\ \frac{\Delta_-^j((w_1)_{j+\frac{1}{2},k}\Delta_+^j y_{j,k})}{(\Delta\xi)^2} + \frac{\Delta_-^k((w_2)_{j,k+\frac{1}{2}}\Delta_+^k y_{j,k})}{(\Delta\eta)^2} &= 0, \end{aligned} \tag{3.8}$$

where Δ_+^l and Δ_-^l, $l = j, k$ denote forward and backward difference operators. The mesh mapping (3.1) are denoted by $x_{j,k} = x(\xi_{j,k}, \eta_{j,k})$, $y_{j,k} = y(\xi_{j,k}, \eta_{j,k})$. The above equations with boundary condition are solved by iterative methods, i.e., Jacobi iteration or Gauss-Seidel iterative method. The numerical experiments show that the Gauss-Seidel iteration with a fixed number of iteration is very robust in the mesh construction.

3.1.2 Conservative solution interpolation

After generating new mesh at each time step according to the monitor function, we need to interpolate flow variables from the old to the newly obtained mesh. Even though many interpolating schemes have been suggested, such as the non-conservative one for the nonlinear Hamilton-Jacobi equation [75], for the compressible flow equations a conservative interpolation scheme has to be used. In the current research, the method proposed in [76] for a finite volume formulation is adopted. Let $(x_{j,k}, y_{j,k})$ and $(\tilde{x}_{j,k}, \tilde{y}_{j,k})$ be coordinates of the old and new grid points, respectively, and at the same time $\tilde{A}_{j+\frac{1}{2},k}$ and $A_{j+\frac{1}{2},k+\frac{1}{2}}$ denote the quadrangles with four vertices $(\tilde{x}_{j+p,k+q}, \tilde{y}_{j+p,k+q})$, and $(x_{j+p,k+q}, y_{j+p,k+q})$, p, q are integer numbers with $0 \leq p, q \leq 1$, see Fig.3.1. A perturbation method is applied in the derivation of the conservative interpolation scheme. Assuming $(\tilde{x}, \tilde{y}) = (x - c^x(x,y), y - c^y(x,y))$ with small displacement (c^x, c^y), we have

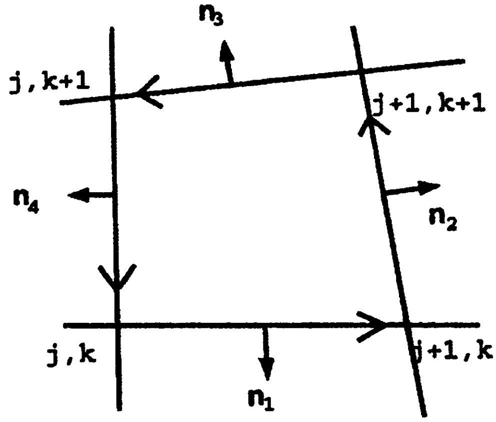


Figure 3.1: A schematic of two dimensional finite control volume $A_{j+1/2,k+1/2}$

$$\begin{aligned}
\int_{\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}} U(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y} &= \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(x-c^x, y-c^y)det(\frac{\partial(\tilde{x},\tilde{y})}{\partial(x,y)})dxdy \\
&\approx \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} (U(x,y) - c^x U_x - c^y U_y)(1 - c^x_x - c^y_y)dxdy \\
&\approx \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} [U(x,y) - (c^x U)_x - (c^y U)_y]dxdy \\
&= \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(x,y)dxdy - [(c^2_n U)_{j+1,k+\frac{1}{2}} + (c^4_n U)_{j,k+\frac{1}{2}}] \\
&\quad -[(c^3_n U)_{j+\frac{1}{2},k+1} + (c^1_n U)_{j+\frac{1}{2},k}], \qquad (3.9)
\end{aligned}$$

where both the coordinate transformation from $(\tilde{x},\tilde{y})$ to (x,y) and the Taylor expansion of $U(\tilde{x},\tilde{y})$ at (x,y) are used in the above derivation. Higher order terms have been neglected. Here $c_n^l := c^x n_x^l + c^y n_y^l$ for $l = 1,2,3,4$, where $n^l = (n_x^l, n_y^l)$ is the unit outward normal direction on the corresponding surface of the control volume $A_{j,k}$, namely,

$$\begin{aligned}
\mathbf{n}^1 &= (y_{j+1,k} - y_{j,k}, x_{j,k} - x_{j+1,k})/s_1 \\
\mathbf{n}^2 &= (y_{j+1,k+1} - y_{j+1,k}, x_{j+1,k} - x_{j+1,k+1})/s_2 \\
\mathbf{n}^3 &= (y_{j,k+1} - y_{j+1,k+1}, x_{j+1,k+1} - x_{j,k+1})/s_3 \\
\mathbf{n}^4 &= (y_{j,k} - y_{j,k+1}, x_{j,k+1} - x_{j,k})/s_4.
\end{aligned}$$

where s_1, s_2, s_3, s_4 are the corresponding surface length of the control volume $A_{j,k}$. The grid 'moving' speeds of internal grid points $\vec{x}$ are given by

$$(c^x, c^y)_{j,k} := (x_{j,k} - \tilde{x}_{j,k} \quad , \quad y_{j,k} - \tilde{y}_{j,k}), \qquad 1 \le j \le J_x, 1 \le\le J_y.$$

Then, we have

$$\begin{aligned}
c_n^l &= \frac{1}{2s_1}(c^x_{j,k} + c^x_{j+1,k})(y_{j+1,k} - y_{j,k}) - \frac{1}{2s_1}(c^y_{j,k} + c^y_{j+1,k})(x_{j+1,k} - x_{j,k}) \\
c_n^2 &= \frac{1}{2s_2}((c^x_{j+1,k} + c^x_{j+1,k+1})(y_{j+1,k+1} - y_{j+1,k}) - (c^y_{j+1,k} + c^y_{j+1,k+1})(x_{j+1,k+1} - x_{j+1,k})) \\
c_n^3 &= \frac{1}{2s_3}((c^x_{j+1,k+1} + c^x_{j,k+1})(y_{j,k+1} - y_{j+1,k+1}) - (c^y_{j+1,k+1} + c^y_{j,k+1})(x_{j,k+1} - x_{j+1,k+1})) \\
c_n^4 &= \frac{1}{2s_4}(c^x_{j,k+1} + c^x_{j,k})(y_{j,k} - y_{j,k+1}) - \frac{1}{2s_4}(c^y_{j,k+1} + c^y_{j,k})(x_{j,k} - x_{j,k+1}).
\end{aligned}$$

More detailed explanation can be found in [74]. From (3.9), the conservative quantities are interpolated into the newly constructed mesh with the values,

$$\begin{aligned}
|\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{U}_{j+\frac{1}{2},k+\frac{1}{2}} &= |A_{j+\frac{1}{2},k+\frac{1}{2}}|U_{j+\frac{1}{2},k+\frac{1}{2}} - [(c_n^2 U)_{j+1,k+\frac{1}{2}} + (c_n^4 U)_{j,k+\frac{1}{2}}] \\
&\quad -[(c_n^3 U)_{j+\frac{1}{2},k+1} + (c_n^1 U)_{j+\frac{1}{2},k}]. \qquad (3.10)
\end{aligned}$$

Taking the summation on both sides in the above equation for j,k, the conservative property of the above scheme can be proved, i.e.,

$$\sum_{j,k} |\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{U}_{j+\frac{1}{2},k+\frac{1}{2}} = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}|U_{j+\frac{1}{2},k+\frac{1}{2}}. \qquad (3.11)$$

3.2 Gas-kinetic BGK-NS flow solver

Let denote $\mathbf{x}_{j+1/2,k+1/2} = (x_{j+1/2,k+1/2}, y_{j+1/2,k+1/2})$ as the center coordinate of the finite control volume $A_{j+1/2,k+1/2}$ in Fig.3.1, which is different from the notation in Chapter 2, where $\mathbf{x}_{j,k} = (x_{j,k}, y_{j,k})$ is the center point of $A_{j,k}$. After interpolating the flow variables from the old to the new mesh, as any other finite volume method the fluxes will be evaluated at a cell interface to update the solution to the next time level. Here, the gas-kinetic BGK scheme introduced in Chapter 2 will be used to construct the numerical flux at the cell interface, where the inviscid and viscous parts are obtained simultaneously.

The gas-kinetic BGK scheme has been introduced in the last chapter. Here we only outline its application in the curvilinear mesh. Firstly, at the beginning of each time step we need to reconstruct the initial data from the cell averaged value at a cell interface such as the cell interface $\mathbf{P}_{j+1,k}\mathbf{P}_{j+1,k+1}$ with the unit outward normal direction $\mathbf{n}^2 = (\cos\theta, \sin\theta)$ in Fig.3.1 and the unit tangential direction $\mathbf{t}^2 = (-\sin\theta, \cos\theta)$. Then, the macroscopic flow velocities in the neighboring cells of the interface are projected into the normal direction $\mathbf{n}^2$ and the tangential direction $\mathbf{t}^2$ respectively, namely

$$\begin{cases} \tilde{U} = (U, V)\cdot\mathbf{n}^2 = U\cos\theta \ + V\sin\theta \\ \tilde{V} = (U, V)\cdot\mathbf{t}^2 = -U\sin\theta \ + V\cos\theta \end{cases} \tag{3.12}$$

and $W = (\rho, \rho U, \rho V, \rho E)$ become $\widetilde{W} = (\rho, \rho\tilde{U}, \rho\tilde{V}, \rho E)$. After the projection, the initial data reconstruction is the same as the one in Chapter 2.2.2. Similarly, in the local system $(\mathbf{n}^2, \mathbf{t}^2)$, the microscopic particle velocity (u, v) becomes

$$\begin{cases} \tilde{u} = (u, v)\cdot\mathbf{n}^2 = u\cos\theta \ + v\sin\theta \\ \tilde{v} = (u, v)\cdot\mathbf{t}^2 = -u\sin\theta \ + v\cos\theta \end{cases} \tag{3.13}$$

inversely,

$$\begin{cases} u = \tilde{u}\cos\theta \ - \tilde{v}\sin\theta \\ v = \tilde{u}\sin\theta \ + \tilde{v}\cos\theta \end{cases} \tag{3.14}$$

Therefore, based on Eq.(3.14) the time-dependent numerical fluxes in the normal-direction $\mathbf{n}^2$ across the cell interface $\mathbf{P}_{j+1,k}\mathbf{P}_{j+1,k+1}$ can be calculated by

$$
\begin{aligned}
\begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_m \\ \mathcal{F}_n \\ \mathcal{F}_E \end{pmatrix}
&= \int \tilde{u} \begin{pmatrix} 1 \\ u \\ v \\ \frac{1}{2}(u^2+v^2+\xi^2) \end{pmatrix} f(x_{j+1,k+1/2}, y_{j+1,k+1/2}, t, \tilde{u}, \tilde{v}, \xi) d\Xi \\
&= \int \tilde{u} \begin{pmatrix} 1 \\ \tilde{u}\cos\theta - \tilde{v}\sin\theta \\ \tilde{u}\sin\theta + \tilde{v}\cos\theta \\ \frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\xi^2) \end{pmatrix} f(x_{j+1,k+1/2}, y_{j+1,k+1/2}, t, \tilde{u}, \tilde{v}, \xi) d\Xi \\
&= \begin{pmatrix} \mathcal{F}_\rho \\ \cos\theta\mathcal{F}_{\tilde{m}} - \sin\theta\mathcal{F}_{\tilde{n}} \\ \sin\theta\mathcal{F}_{\tilde{m}} + \cos\theta\mathcal{F}_{\tilde{n}} \\ \mathcal{F}_E \end{pmatrix}
\end{aligned}
$$

where $f(x_{j+1/2,k}, y_{j+1/2,k}, t, \tilde{u}, \tilde{v}, \xi)$ is the standard BGK flux function defined by Eq.(2.51) in Chapter 2.3, Eq.(2.63) in Chapter 2.4, and Eq.(2.68) in Chapter 2.5. The corresponding numerical fluxes $(\mathcal{F}_\rho, \mathcal{F}_{\tilde{m}}, \mathcal{F}_{\tilde{n}}, \mathcal{F}_E)^T$ can be obtained similarly by,

$$
\begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_{\tilde{m}} \\ \mathcal{F}_{\tilde{n}} \\ \mathcal{F}_E \end{pmatrix}
= \int \tilde{u} \begin{pmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \frac{1}{2}(\tilde{u}^2+\tilde{v}^2+\xi^2) \end{pmatrix} f(x_{j+1,k+1/2}, y_{j+1,k+1/2}, t, \tilde{u}, \tilde{v}, \xi) d\Xi.
$$

3.3 Numerical experiments

The gas-kinetic scheme is a Navier-Stokes solver, the physical viscosity coefficient is proportional to the particle collision time. The inviscid flow is considered as a limiting solution when the viscosity coefficient is small. In this section, both inviscid and viscous flow test cases will be presented by the current adaptive grid method.

3.3.1 Inviscid flow

For the inviscid flow computation, the collision time τ in the gas-kinetic flux function is defined by

$$\tau = 0.05\Delta t + \frac{|P_l - P_r|}{P_l + P_r}\Delta t$$

where Δt is the CFL time step, and P_l, P_r are the corresponding pressure in the states g_l, g_r of the initial discontinuous gas distribution function f_0. Due to the large density gradient at the material interface, in the following example the density is used in the monitor function (3.7).

Case(1) A Mach=1.22 shock wave in air hitting a helium cylindrical bubble

A Mach 1.22 planar shock wave, moving through air, impinges on a cylindrical helium bubble. The flow is often modeled by two-dimensional compressible Euler equations of a two-component fluid (air-helium). The earlier result using the gas-kinetic BGK scheme has been reported in [45]. Here we mainly demonstrate that the adaptive grid method can give much high resolution solution with less grid points, especially in the regions with large density gradient, such as the shock waves and the air-helium interface. The initial condition for this case is

$$(\rho, U, V, P, \gamma) = \begin{cases} (1, 0, 0, 1, 1.4) & , \quad \text{pre-shock air} \\ (1.3764, -0.394, 0, 1.5698, 1.4) & , \quad \text{post-shock air} \\ (0.1358, 0, 0, 1, 1.67) & , \quad \text{helium.} \end{cases}$$

In this case, inflow and outflow boundary conditions are used on the right and left sides of the computational domain, and reflection conditions are imposed on the top and bottom.

The computational mesh used in the current adaptive grid method is 260 × 71 which is only 4% of the total number of uniform grids 1300 × 356 used in [45]. Fig.3.2

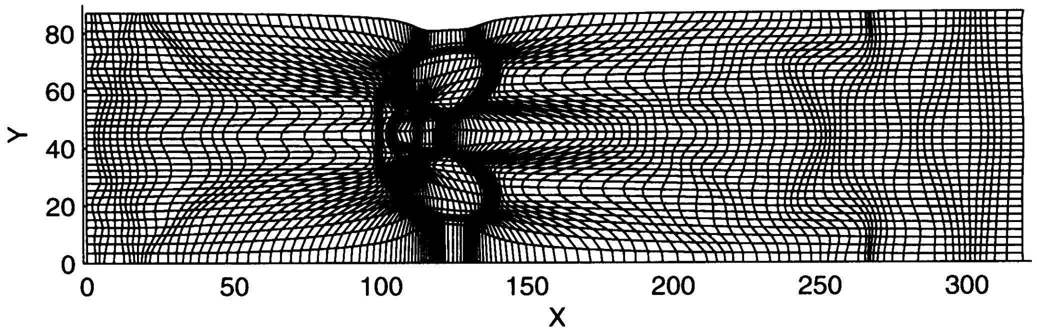


Figure 3.2: Adaptive mesh distribution at time $t = 150.00$.

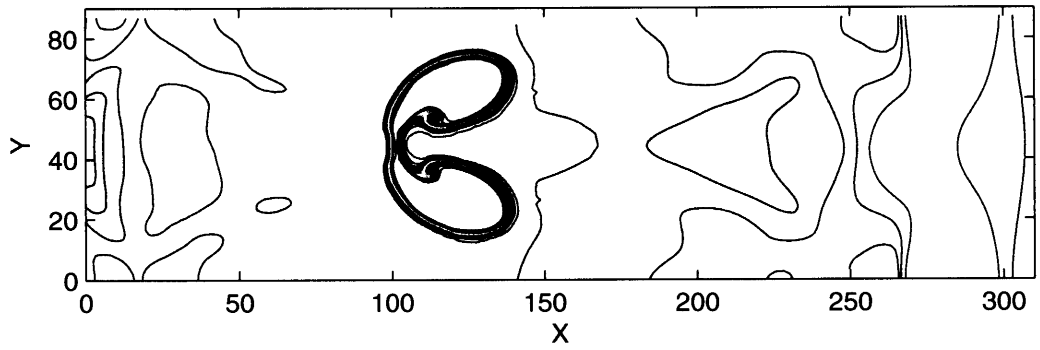


Figure 3.3: Density contours on the adaptive mesh at time $t = 150.0$.

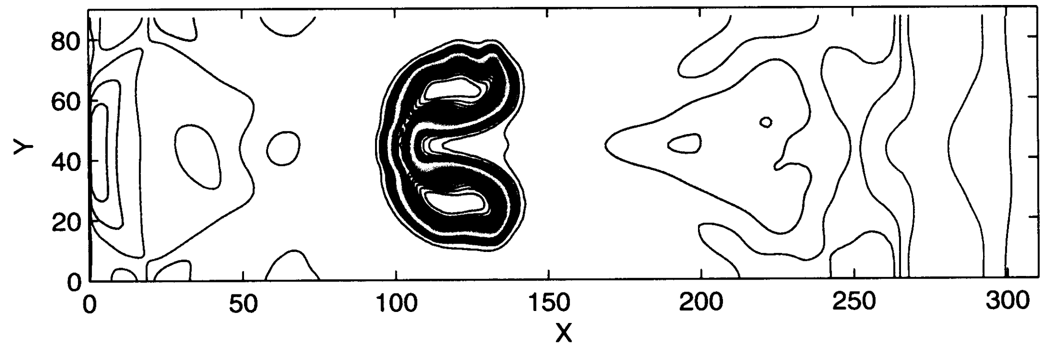


Figure 3.4: Density contours on a uniform mesh at time $t = 150.0$.

is the adaptive grids from the monitor function (3.7) with $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1.0$ at time $t = 150.0$. With the similar resolution at the material interface, the total computational time using the adaptive method is less than 30% of the one used in a uniform fine mesh 1300 × 356. The density distribution is shown in Fig.3.3, which clearly shows the coupling between the computational mesh and flow resolution. As expected, the adaptive grid method improves the interface resolution greatly in comparison with the uniform mesh result in Fig.3.4, where the same number of grid points are used as Fig.3.3.

Case(2) A 2D steady Riemann problem

The case is designed to test a numerical method for its crisp capturing of slip lines [31]. In this example, there are shock wave, slip-line and expansion fan, where an analytical solution for the Euler equations is available. It is a good benchmark problem for the verification of numerical dissipation in a numerical method. The initial condition for this case is

$$(p, \rho, M, \theta) = \begin{cases} (0.25, 0.5, 7, 0) & , \quad y > 0 \\ (1, 1, 2.4, 0) & , \quad y < 0 \end{cases}$$

Fig.(3.5) is the adaptive grid generated using a monitor function depending on the density gradients, where grid points are concentrated in three regions, namely shock wave, slip line and rarefaction wave. Fig.(3.7) shows that with the mesh adaptation the numerical solutions in shock wave, slip line and rarefaction have basically high resolutions, where less numerical dissipation involved in these regions in comparison with the results with uniform mesh in Fig.(3.6). Still, there exist a few numerical points in the slip region in Fig.(3.7), whose densities are smaller than the exact ones. This may be explained in the following. Even though we choose a small particle collision time in this case, the gas-kinetic BGK scheme is still solving the Navier-Stokes equations. Around the slip line, due to the strong shear velocity the viscous heating will emerge and heat up the gas to expand. Therefore, a density dip appears.

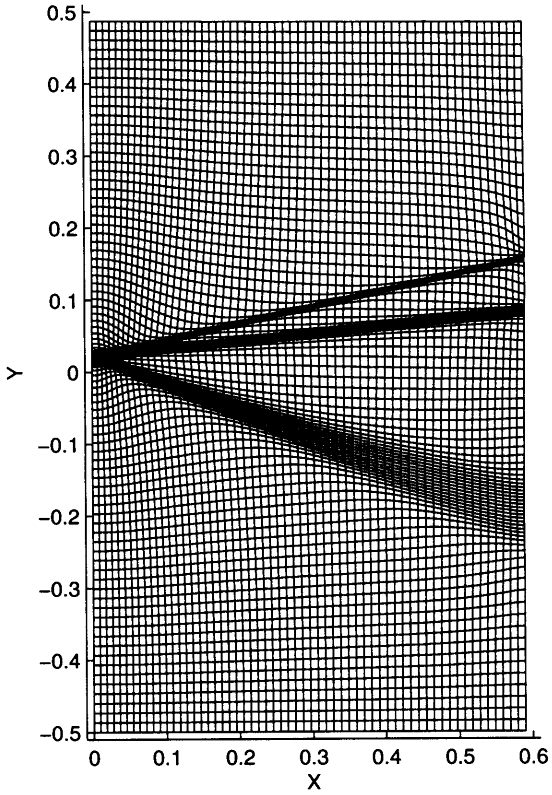


Figure 3.5: Adaptive meshes generated by the current method

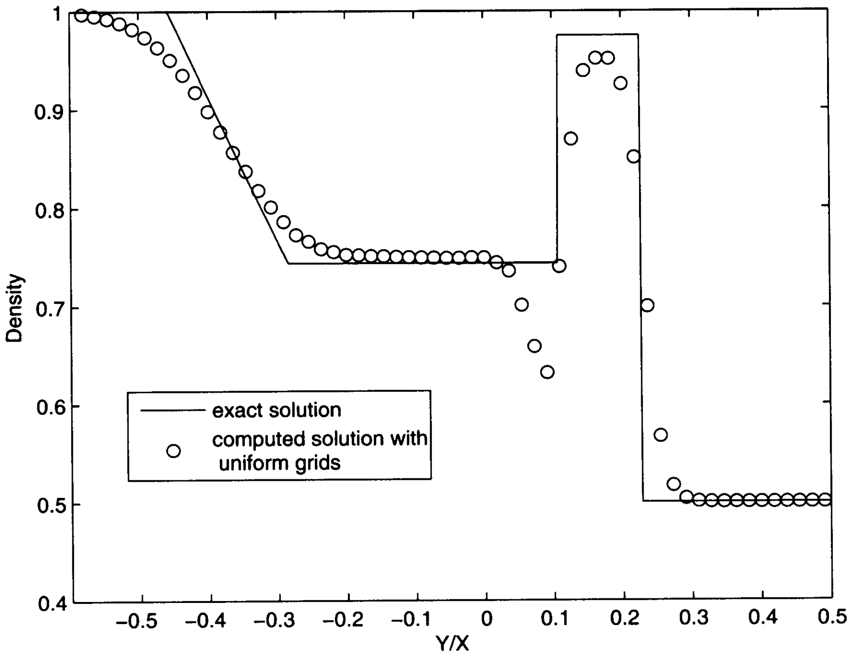


Figure 3.6: Density distribution in a steady Riemann problem computed by the gas-kinetic BGK scheme with uniform grids

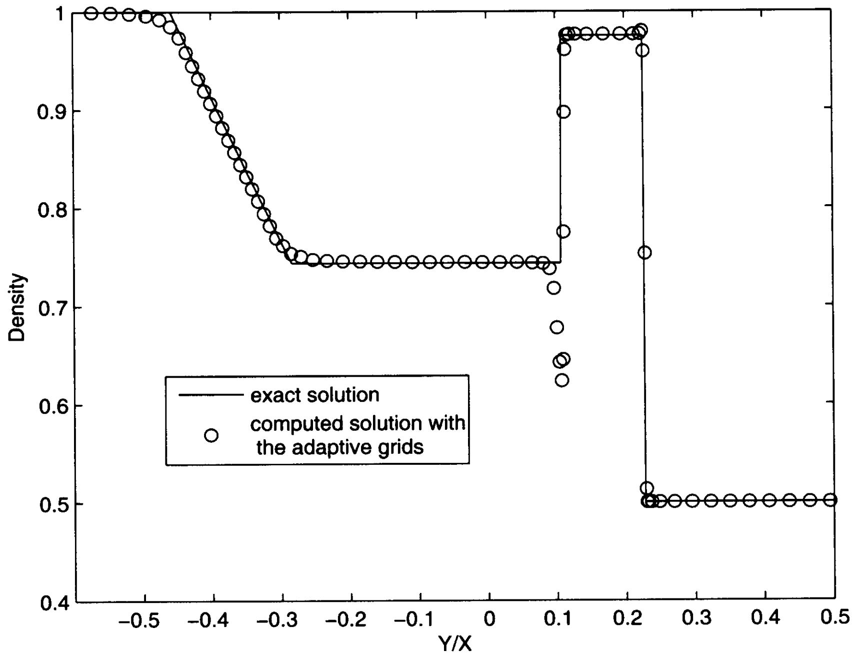


Figure 3.7: Density distribution in a steady Riemann problem computed by the gas-kinetic BGK scheme with adaptive grids

3.3.2 viscous flows

In the viscous flow calculations, the collision time τ is set to be $\tau = \mu/P$, or $\tau = 2\nu\lambda$ with $\lambda = \frac{\rho}{2P}$, where μ and ν are the dynamical and kinematic viscosity coefficients, ρ is the fluid density, and P is the local pressure. For the viscous flow simulation, the viscous effect is closely related to the velocity gradient. Therefore, ψ in the monitor function (3.7) will take the fluid velocity $q = \sqrt{U^2 + V^2}$.

Case(3) Viscous flow above an oscillating plate

This is called Stokes's second problem, which considers fluid motion above an infinite flat plate which executes sinusoidal oscillations parallel to itself. The fluid above the plate is initially stationary. The governing equation for the fluid velocity U in the

x-direction is

$$\frac{\partial U}{\partial t} = \nu \frac{\partial^2 U}{\partial y^2}, \tag{3.15}$$

with the boundary conditions

$$U(0,t) = U_0 \cos \omega t \quad , \quad U(\infty, t) = 0.$$

The exact solution for the above problem is,

$$U = U_0 e^{-y\sqrt{\omega/2\nu}} \cos(\omega t - y\sqrt{\frac{\omega}{2\nu}}). \tag{3.16}$$

At $y = 4\sqrt{\nu/\omega}$, the amplitude of U is equal to $U_0 \exp(-4/\sqrt{2}) = 0.05U_0$, which means that the influence from the wall is confined within a distance of order $\delta \sim 4\sqrt{\nu/\omega}$. Since the gas-kinetic scheme solves the compressible Navier-Stokes equations, in order to simulate the above incompressible limiting solution the Mach number for the compressible flow must take a small value, i.e., $M = 0.15$, and the gas-kinetic scheme becomes an artificial compressibility method for incompressible flow. In the current case, the kinematic viscosity coefficient takes a value $\nu = 0.00046395$, and a mesh size 10×20 is used in the computation. Obviously, we only need to move the mesh in the y direction, where $\alpha_2 = 0$, $\beta_2 = 1.0$ are used in the corresponding monitor function. Fig.3.8 shows the two adaptive grids at times $\omega t = \frac{\pi}{10}$ and $\frac{\pi}{2}$. The numerical results are compared with the exact solutions (3.16) in Fig.3.9. Certainly, the grid points concentrate in the regions with large velocity gradient. The results clearly show that the adaptive mesh method is very effective to capture the viscous solution and the artificial compressibility method is accurate for the incompressible flow solution.

Case(4) Shock reflection from a wedge

This is an unsteady Mach reflection problem, where a plane shock with Mach number $M = 1.3$ moves from left to right hitting a wedge with 25^0. The initial flow state is given by $(p, \rho, u, v) = (1/1.4, 1, 0, 0)$ inside the computational domain, and the left boundary condition is $(p, \rho, u, v) = (1.2893, 1.5157, 0.44231, 0)$, which satisfies

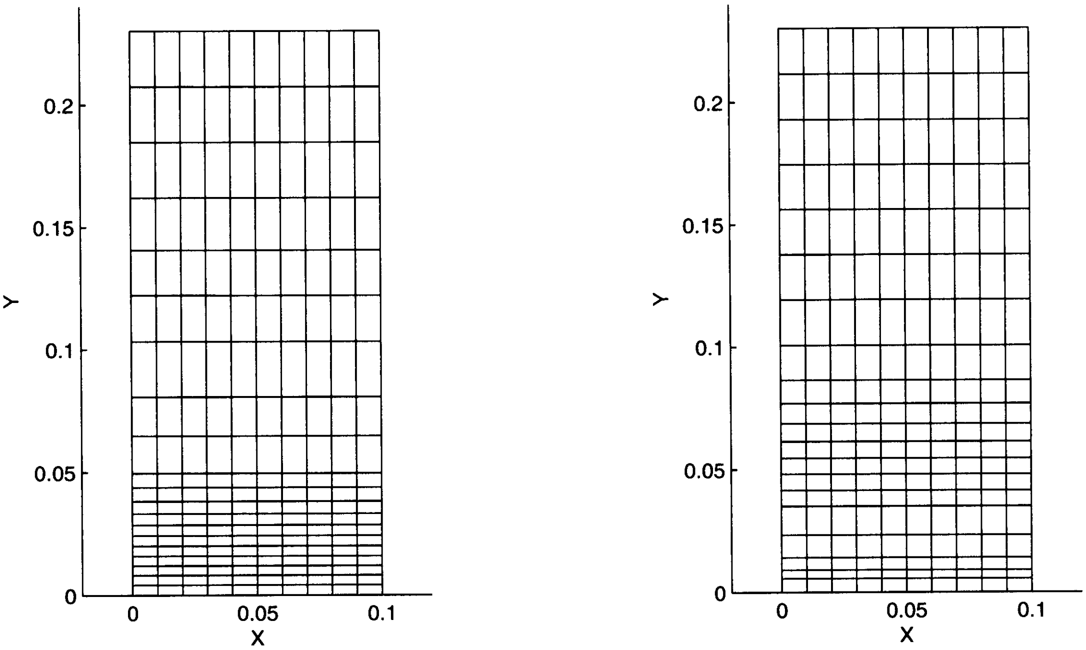


Figure 3.8: Adaptive mesh at $\omega t = \frac{\pi}{10}$ (left) and $\frac{\pi}{2}$ (right).

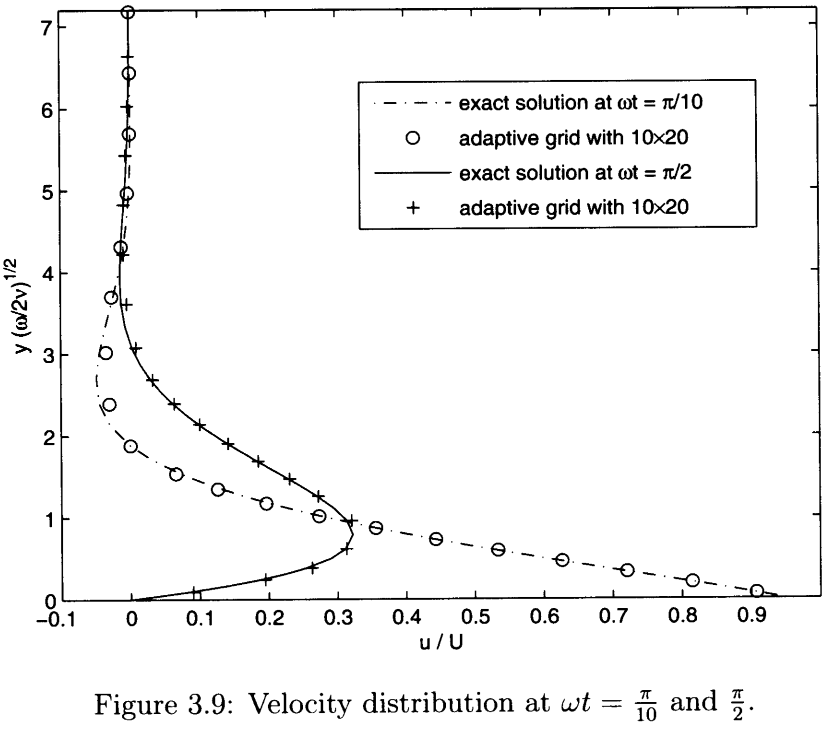


Figure 3.9: Velocity distribution at $\omega t = \frac{\pi}{10}$ and $\frac{\pi}{2}$.

the shock jump condition. Non- slip boundary condition is imposed on the surface of the wedge. Initially, the computational domain $\{0 \leq x \leq 2.0, 0 \leq y \leq 0.75\}$ is covered by a uniform mesh with 200×200 grid points and cell size $\Delta x = 0.01$ and $\Delta y = 0.00375$. The kinematic viscosity coefficient takes a value $\nu = 0.0000369$. The adaptive mesh is constructed with the parameters $\alpha_1 = \alpha_2 = 0$ and $\beta_1 = \beta_2 = 1.0$ in the monitor function. With the flow movement, the mesh starts to concentrate in the regions with high velocity gradients, such as the shock front and boundary layer. Fig. 3.10 shows the adaptive mesh in the physical domain. The Mach number contours at time $t = 1.25$ are shown in Fig.3.11, which is close to the experiment picture in [79].

In order to demonstrate the accuracy of the mesh moving method, we also used a much refined uniform mesh with 200×1000 grid points to do the same calculation. The boundary layer profile at the location $x = 1.175$ above the wedge surface is presented in Fig. 3.12. Obviously, the results from the adaptive mesh method is consistent with the mesh refinement results. However, the results with uniform coarse mesh (200×200 grid points) are inferior in comparison with others. In other words, the adaptive grid method with much less number of grid points can obtain accurate solution. The adaptive mesh method (200×200 grid points) uses less than 30% of computational time than that with a uniform refined mesh (200×1000 grid points).

Case(5) 2D cavity flow

The cavity flow inside a square is a well-defined example for validating numerical methods. There is highly accurate solutions for this case, which were obtained by Ghia et al. [24] and Botella and Peyret [7]. The square has three fixed walls and the upper moving boundary with unit velocity. Hence, the non-slip boundary condition relative to the walls are imposed. In the current simulation, we only calculate the flow at Reynolds number Re= 1000 with 65×65 grid points in the computational domain. The mesh adaptation is done with $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1.5$ in the monitor function. Fig.3.13 shows the adaptive grids, where the mesh points concentrate on

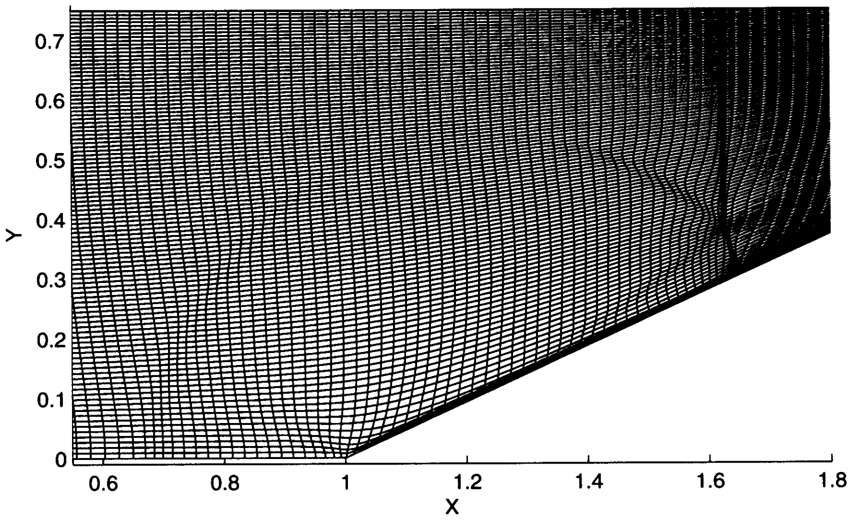


Figure 3.10: Adaptive mesh after the shock reflection from a wedge.

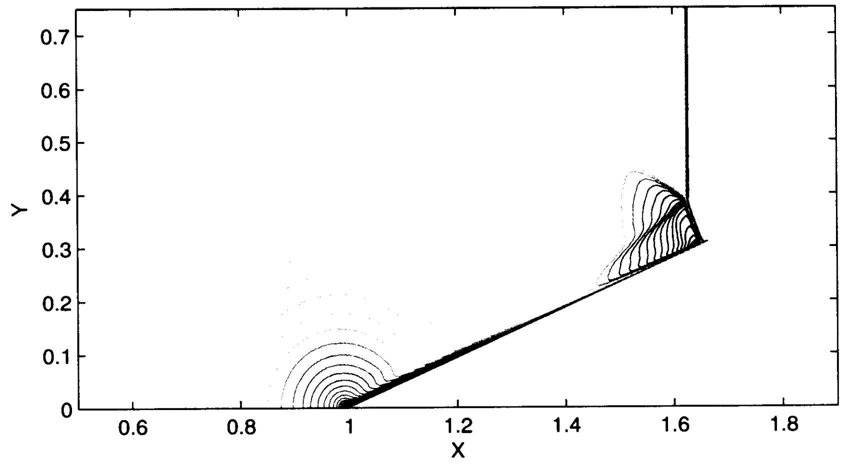


Figure 3.11: Mach contours.

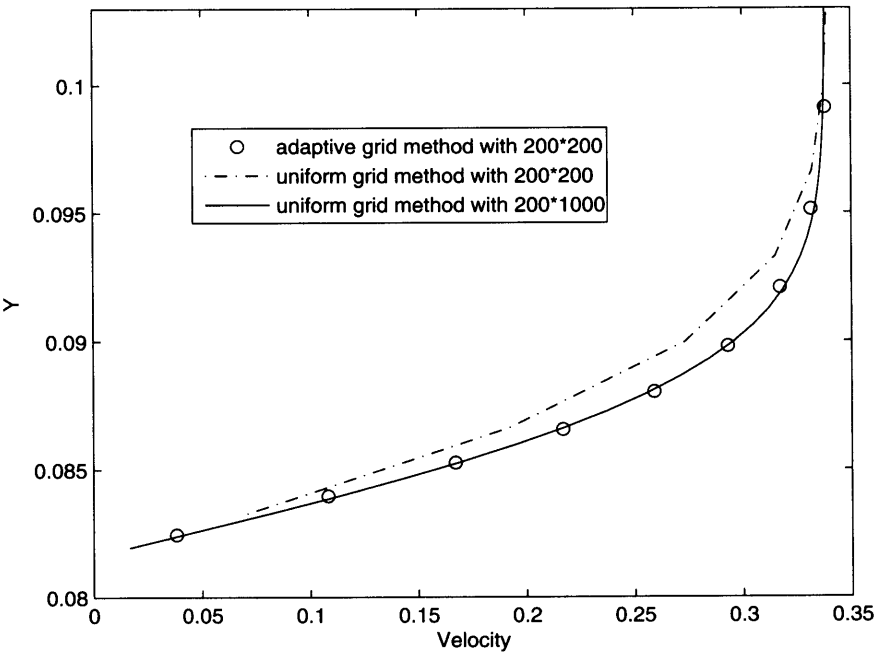


Figure 3.12: Velocity boundary layer at $x = 1.175$ above the wedge surface.

the high velocity gradient regions. The simulation results, as well as the benchmark solutions from Ghia et al. [24] and Botella and Peyret [7], are presented in Figs.3.13-3.17. The solutions from the adaptive mesh have better agreement with the benchmark results than that from uniform mesh. With similar accuracy, the computational time here only takes 23.5% of that used in the uniform mesh method with 255 × 255 grid points in [89].

In this chapter, an adaptive grid method has been successfully developed for the simulation of inviscid and viscous flows. For unsteady flow computation, the adaptive grid method has obvious advantage in terms of the accuracy and efficiency in comparison with the static mesh method. The good agreement between the numerical results and benchmark solutions validates the current approach. With the velocity-gradient dependent monitor function, the boundary layer and shock wave have been well captured for the viscous flow. In the next chapter, we are going to develop a unified moving mesh method. In order to get a robust numerical method with moving mesh, the

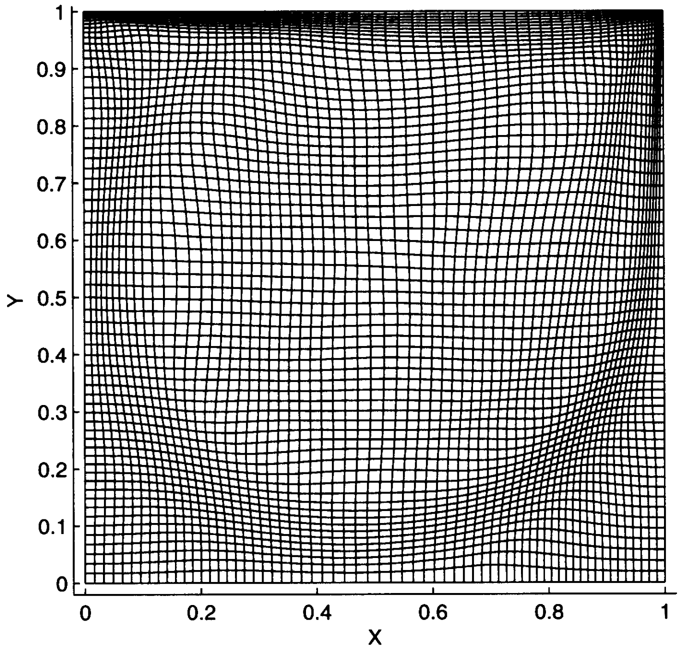


Figure 3.13: Adaptive mesh for the cavity flow.

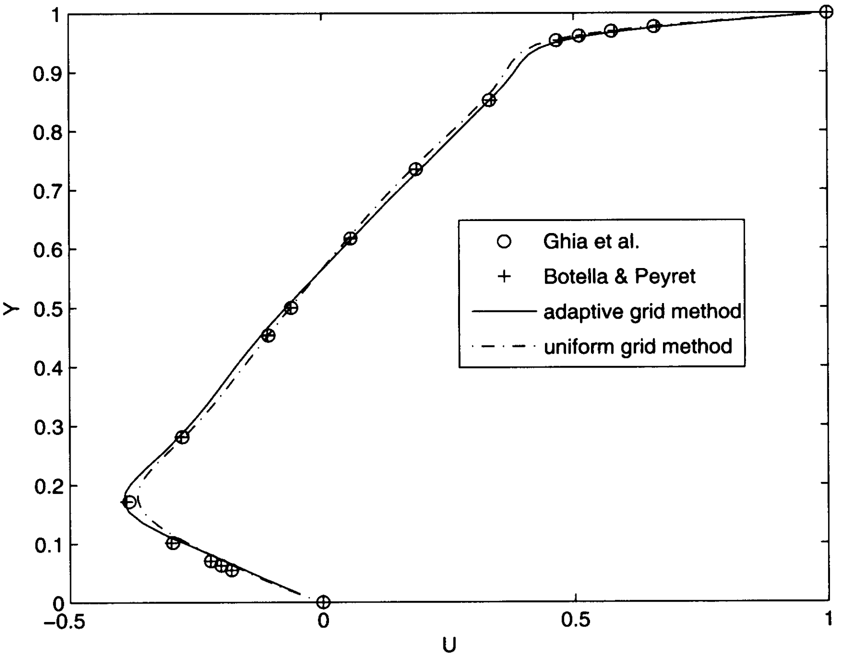


Figure 3.14: Velocity U profile along the central vertical line at $Re = 1000$.

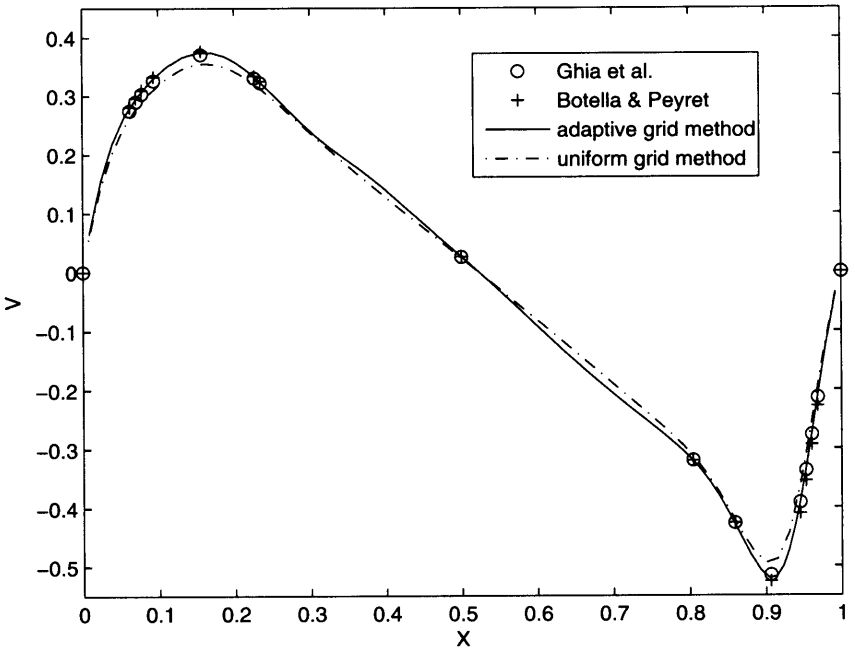


Figure 3.15: Velocity V profile along the central horizontal line at $Re = 1000$.

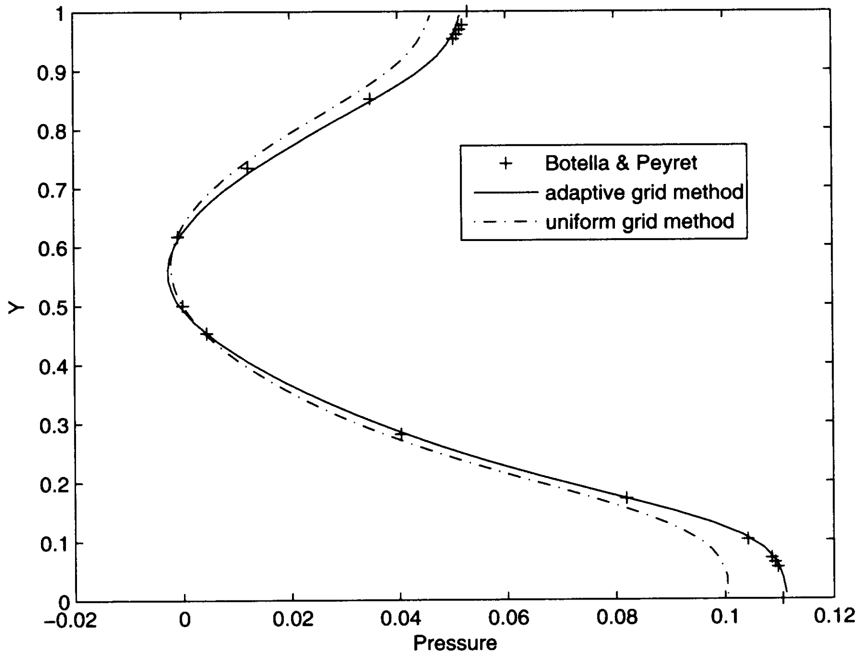


Figure 3.16: Pressure profile along the central vertical line at $Re = 1000$.

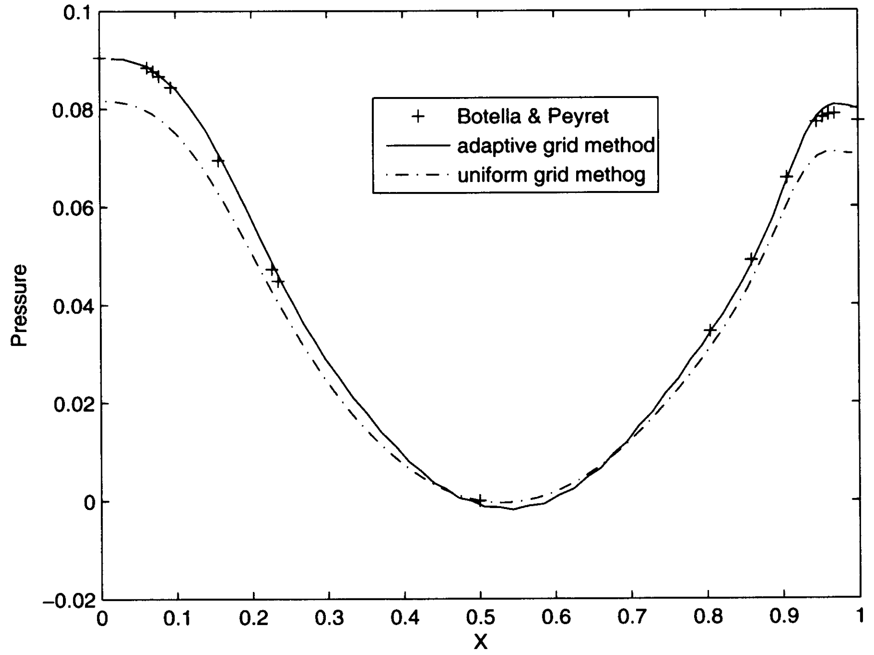


Figure 3.17: Pressure profile along the central horizontal line at $Re = 1000$.

adaptive grid method in the current chapter will be implemented there as well. On the adaptive mesh method itself, it can be predicted that it will become an important working horse for the unsteady viscous flow computation in the future, especially in the cases where fine grid resolution is needed, such as DNS and turbulent flow.

Chapter 4

Unified moving mesh gas-kinetic method for viscous flows

There are two different coordinate system for description of fluid motion: the Eulerian one describes fluid motion at fixed locations, whereas the Lagrangian one follows fluid particles. Considerable progress has been made over the past two decades on developing computational fluid dynamics (CFD) methods based on the above two coordinates system. As the unsteady flow calculations for problems with moving boundaries and interfaces become important, such as found in the flutter simulation of wings, turbomachinery blades, and multi-phase flow, the development of fast and reliable methods for dynamically deforming computational domain is required [67]. This research will help significantly the industry, such as the aerodynamic shape optimization studies and the detonative chemical reactive flow computation.

There are many moving mesh methods in the literature. One example is the static mesh movement method, where the new mesh is generated at each time step according to certain monitor function and the flow variables are interpolated into the newly generated mesh. Then, the flow update through the cell interface fluxes is done constructed on a static mesh. In order to increase the accuracy, the mesh can be properly adapted [29]. Another example is the dynamical one, where the mesh is moving according to certain velocity. At the same time, the fluid variables are updated inside each moving control volume within a time step. The second method is mostly used to track the interface location, to account for changes in the interface topology, and to resolve small-scale structure at singular point. The most famous one for this dynamical mesh moving method is the Lagrangian method. Through the research in the past decades, it has been well recognized that the Lagrangian method is always associated with the mesh tangling once the fluid velocity is used as the mesh moving velocity. In order to avoid severe mesh distortion in the Lagrangian method, many techniques have been developed. The widely used one at present time is the Arbitrary Lagrangian-Eulerian (ALE) technique, which uses continuous re-zoning and re-mapping to the Eulerian grid. Unfortunately, this process requires interpolations of geometry and flow variables [47]. In aerospace engineering, in order to re-distribute the boundary deformation dynamically into the whole computational domain a spring network approach has been usually used to determine the motion of the mesh point, such as those around a deforming airfoil [4, 64, 78]. Here, a smoothing global operator is applied to maintaining grid smoothness and grid angles. This process is always associated with iterative methods resembling an elliptic grid generator. With a general transformation between the physical (t, x, y) and the computational space (λ, ξ, η), the Navier-Stokes equations can be written in a conservative form in [28]. Many numerical schemes have been developed based on the above formulation for the Navier-Stokes equations directly, such as in the cases of fluid-structure interaction and fluid induced vibration. Instead of constructing an exact Riemann solver, an efficient approximate Riemann solution has been obtained [15], where the grid velocity is explicitly used in the wave decomposition. Even without conservative governing equations in a moving mesh system explicitly, many moving mesh methods for incompressible Navier-Stokes equations, hyperbolic system, or chemical reactive flow, have also been developed with detailed consideration of numerical cell deformation [70, 11, 2].

Recently, a successful moving mesh method for inviscid Euler equations has been developed by Hui et al on the target of crisp capturing of slip line [31]. In this unified coordinates method, with a prescribed grid velocity, the inviscid flow equations are written in a conservative form in the computational domain (λ, ξ, η), as well as the geometric conservation laws which control the mesh deformation. The most distinguishable merit in the unified coordinate method [31] is that the fluid equations and geometric evolution equations are written in a combined system, which is different from the fluid equations alone [28]. Furthermore, due to the coupling in the fluid and geometric system, for the first time the multidimensional Lagrangian gas dynamic equations have been written in a conservative form. As a consequence, theoretically it has been shown that the multidimensional Lagrangian system is only weakly hyperbolic. Numerically, in the unified coordinate system the fluid and geometric variables can be updated simultaneously. In order to overcome the disadvantage in the Lagrangian method, in the unified coordinate system the grid velocity is set to be $h\mathbf{q}$, where $\mathbf{q} = (U, V)$ is the fluid velocity and h is a parameter which is to be determined by conditions, such as the mesh alignment with the slip surface, or keeping grid angle during the mesh movement. Therefore, the grid velocity can be changed locally according to the value of h. In a recent paper [32], the grid velocity has been further generalized to (hU, kV), where h and k are two parameters to be determined. The great achievement of the unified coordinate method is that the numerical diffusion across the slip line is reduced to a minimum with the crisp capturing of contact discontinuity. However, in the complicated flow movement, in order to avoid the severe mesh distortion, the constraint, such as keeping mesh orthogonality and grid angles, has to be used in the unified coordinate system. As a result, in most cases, the constraint automatically enforces the mesh velocity being zero, such as in the case of gas implosion inside a square. Otherwise, for flow problems with circulations, any mesh movement method, once the grid speed is coupled with the fluid velocity, will distort the mesh eventually and stop the computation. Also, in order to capture the slip line, the unified method is mainly focusing on the solution of the Euler equations. For the viscous flow, the equations, see the Appendix, become much more complicated in a unified coordinate system.

Based on the gas-kinetic Boltzmann equation, the Navier-Stokes equations can be derived using the Chapman-Enskog expansion [14]. As presented early in this thesis, in the gas-kinetic representation all macroscopic flow variables are the moments of a single particle distribution function and the particle movement is basically the linear transport and collision. In the past years, the gas-kinetic scheme has been developed in the Eulerian space with stationary grid. The gas-kinetic BGK scheme has been well developed for the compressible Navier-Stokes solutions [86], and the scheme is especially accurate for the supersonic viscous and heat conducting flow [88, 72]. Since a gas distribution function includes both equilibrium and non-equilibrium flow properties, the inviscid and viscous fluxes are obtained simultaneously in the gas-kinetic scheme. Hence, in this chapter, we are going to represent a gas-kinetic method for the viscous solution in a moving frame of reference.

In the gas-kinetic approach, the gas distribution function is a scalar function, which can be easily transferred from one reference of frame to another one once the relative speed between them is given. The purpose of this chapter is to develop a unified gas-kinetic method based on the transformation in [31]. Firstly, the gas-kinetic BGK equation is transformed from the physical space (t, x, y) to the computational space (λ, ξ, η). Then, the corresponding conservative Euler and Navier-Stokes equations in the computational domain are obtained using the Chapmann-Enskog expansion. In the computational space (λ, ξ, η), the gas-kinetic BGK equation is solved for the solution of the gas distribution function while standing on a moving cell interface, which is subsequently used to evaluate the flux across a moving cell interface in the Eulerian space in order to update the flow variables inside each controlling volume with moving boundaries. Since the inviscid and viscous fluxes are included simultaneously in the gas-kinetic formulation, the Navier-Stokes fluxes are obtained automatically across the moving cell interface. The advantage of the current method, as for the unified coordinate system [31], is that an exact conservative mathematical formulation for both fluid and geometrical variables is constructed and used in the numerical scheme. If the unified system is not explicitly constructed, the satisfaction of geometrical conservation law, as most moving mesh methods in the literature, are likely based on the physical intuition in the multidimensional cases, such as the flux vector splitting method for the inviscid Euler equations in [60]. Furthermore, in order to avoid the severe mesh distortion due to the coupling of grid speed with the fluid velocity, an adaptive mesh method is also going to be used to regularize the mesh distribution in the domain. In other words, once the quality of the mesh is deteriorated, at a fixed time instant the grid points are redistributed according to a monitor function, and all corresponding conservative flow variables are interpolated from the old to a newly generated mesh [74, 33]. The detail procedure is presented in the last chapter. To include this step has two fold functions. Firstly, it is necessary for a unified scheme to have some mechanism to regularize the mesh from tangling in order to make the scheme robustness of the method. Secondly, for the current viscous flow computation, by using a proper monitor function in the adaptive mesh step the grid points can be easily concentrated on the high gradient regions, where the accuracy of the solution can be much improved, such as inside the boundary layer. With the unique coupling between the moving and adaption in the unified scheme, we have successfully simulated the free surface flow using the Lagrangian gas-kinetic scheme.

In this chapter, firstly under a generalized coordinate transformation with arbitrary grid velocity, the gas-kinetic BGK equation is reformulated in a moving frame of reference. Then, a unified conservative gas-kinetic scheme is developed for the viscous flow in the moving grid system in the Eulerian space. Due to the coupling between the grid velocity and the overall solution algorithm, the Eulerian and Lagrangian methods become two limiting cases in the current gas-kinetic method. A fully conservative formulation can be obtained even in the Lagrangian limit. The moving grid method extends the applicable regime of the gas-kinetic scheme to the flows with free surface and moving boundaries, such as dam break problem and airfoil oscillations. In order to further increase the robustness of the moving grid method, similar to the Arbitrary Lagrangian-Eulerian (ALE) method, a conservative adaptive grid technique is also implemented in the current method to redistribute the mesh concentration to the high gradient flow region and remedy the distorted moving mesh due to the coupling between grid velocity and fluid speed. Many numerical examples from incompressible flow to the supersonic shock interaction are presented. The test cases verify the accuracy and robustness of the unified moving grid gas-kinetic method.

4.1 Gas-kinetic BGK model under coordinate transformation

The BGK model of the approximate Boltzmann equation in two dimensional space (2D) can be written as [6],

$$f_t + uf_x + vf_y = \frac{g-f}{\tau}, \tag{4.1}$$

where f is the gas distribution function and g is the equilibrium state approached by f. Both f and g are functions of space (x, y), time t, particle velocity (u, v), and internal variable ζ. As presented in [31], a coordinate transformation can be constructed from the physical domain (t, x, y) to the computational domain (λ, ξ, η),

$$\begin{cases} dt = d\lambda, \\ dx = U_g d\lambda + Ad\xi + Ld\eta, \\ dy = V_g d\lambda + Bd\xi + Md\eta, \end{cases} \tag{4.2}$$

where (U_g, V_g) are the grid velocity, and (A, B, L, M) are determined by the compatibility conditions or the geometric conservative laws,

$$
\begin{cases}
\frac{\partial A}{\partial \lambda} = \frac{\partial U_g}{\partial \xi}, \\
\frac{\partial B}{\partial \lambda} = \frac{\partial V_g}{\partial \xi}, \\
\frac{\partial L}{\partial \lambda} = \frac{\partial U_g}{\partial \eta}, \\
\frac{\partial M}{\partial \lambda} = \frac{\partial V_g}{\partial \eta}.
\end{cases}
\tag{4.3}
$$

With the above transformation (4.2), the gas-kinetic BGK equation becomes

$$
\frac{\partial}{\partial \lambda}(\triangle f) + \frac{\partial}{\partial \xi}\{[(u-U_g)M-(v-V_g)L]f\} + \frac{\partial}{\partial \eta}\{[-(u-U_g)B+(v-V_g)A]f\} = \frac{g-f}{\tau}\triangle,
\tag{4.4}
$$

where $\triangle = AM - BL$ is the Jacobian of the transformation. The schematic relation between particle velocity, flow velocity, and mesh moving velocity is shown in fig.(4.1).

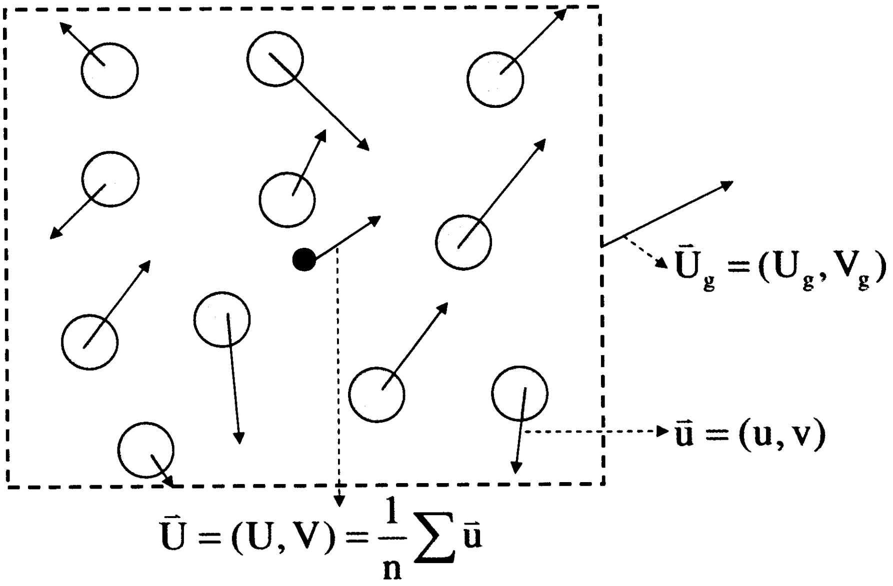


Figure 4.1: Schematic relation between particle velocity (u, v), flow velocity (U, V), and mesh moving velocity (U_g, V_g).

For an equilibrium flow with distribution $f = g$, by taking the conservative moments $\phi = (1, u, v, \frac{1}{2}(u^2 + v^2 + \zeta^2))^T$ to Eq.(4.4), the corresponding Euler equations under the moving mesh in the Eulerian space can be obtained,

$$\frac{\partial}{\partial\lambda}\begin{pmatrix}\rho\Delta\\ \rho\Delta U\\ \rho\Delta V\\ \rho\Delta E\end{pmatrix}+\frac{\partial}{\partial\xi}\begin{pmatrix}\rho(I-I_g)\\ \rho U(I-I_g)+PM\\ \rho V(I-I_g)-PL\\ \rho E(I-I_g)+PI\end{pmatrix}+\frac{\partial}{\partial\eta}\begin{pmatrix}\rho(J-J_g)\\ \rho U(J-J_g)-PB\\ \rho V(J-J_g)+PA\\ \rho E(J-J_g)+PJ\end{pmatrix}=0,(4.5)$$

where U and V are fluid velocity in the x- and y-directions, $I = UM - VL$, $I_g = U_gM - V_gL$, $J = AV - BU$, and $J_g = AV_g - BU_g$. Note again that the time evolution of the physical quantities is still in the Eulerian space. The difference between the Euler equations in Eulerian space and the above system is due to the mesh moving in space and time.

For the viscous and heat conducting flow, the Chapman-Enskog expansion of Eq. (4.4)up to the 1st order of τ gives

$$f = g-\frac{\tau}{\Delta}\left(\frac{\partial}{\partial\lambda}(\Delta g)+\frac{\partial}{\partial\xi}\{[(u-U_g)M-(v-V_g)L]g\}+\frac{\partial}{\partial\eta}\{[-(u-U_g)B+(v-V_g)A]g\}\right).$$

Taking moments of ϕ again to the Eq. (4.4) with the above NS distribution function, we can get the Navier-Stokes equations in moving space, which are presented in Appendix A. Numerically, instead of solving the viscous governing equations Eq.(A.8) in Appendix A, we are going to solve the gas-kinetic equation for the viscous solution.

4.1.1 Numerical BGK-NS scheme in a moving mesh system

In this section, we are going to present the gas-kinetic method to solve Eq.(4.4) by a directional splitting method. For example, the BGK model (4.4) in the ξ-direction is,

$$\frac{\partial}{\partial\lambda}(\Delta f)+\frac{\partial}{\partial\xi}\{[(u-U_g)M-(v-V_g)L]f\}=\frac{g-f}{\tau}\Delta. \qquad (4.6)$$

In order to evaluate the fluxes across a moving interface ξ = constant, let's first define its normal direction and tangent directions,

$$\vec{n}=\nabla\xi/|\nabla\xi|=(M,-L)/S \quad , \quad \vec{t}=(L,M)/S,$$

where $S = \sqrt{M^2 + L^2}$ is the physical length of the cell interface, see Fig. 4.2 for the schematic configuration.

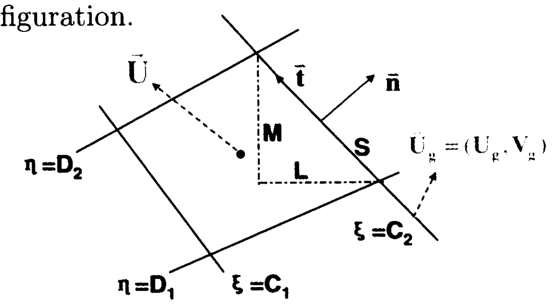


Figure 4.2: Schematic configuration around a moving cell interface.

Then, the particle velocity $(u - U_g, v - V_g)$ relative to a moving cell interface can be decomposed into the normal $\tilde{u}$ and tangential $\tilde{v}$ velocities as well, namely

$$\begin{cases} \tilde{u} = (u - U_g)M/S - (v - V_g)L/S, \\ \tilde{v} = (u - U_g)L/S + (v - V_g)M/S. \end{cases} \tag{4.7}$$

Hence, with the above transformation, Eq.(4.6) in the $\xi-$ direction becomes,

$$\frac{\partial}{\partial \lambda}(\triangle f) + \frac{\partial}{\partial \xi}(S\tilde{u}f) = \frac{g - f}{\tau}\triangle. \tag{4.8}$$

This is the basic equation to be solved to determine the gas distribution function f at the moving cell interface ξ = constant, then calculate the numerical fluxes. In the above equation, $\triangle$ is basically the cell area and S is the cell interface length. At the center of a moving cell interface the above equation can be re-written as

$$\frac{\partial}{\partial \lambda}f + \frac{\partial}{\partial \tilde{x}}(\tilde{u}f) = \frac{g - f}{\tau}, \tag{4.9}$$

where $\tilde{x}$ is the length scale in the normal direction of the moving cell interface in the physical space. Since $d\lambda = dt$, the integral solution of the above equation becomes

$$f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta) = \frac{1}{\tau}\int_0^t g(\tilde{x}', t', \tilde{u}, \tilde{v}, \zeta)e^{-(t-t')/\tau}dt' + e^{-t/\tau}f_0(\tilde{x}_{i+1/2} - \tilde{u}t, \tilde{v}), \tag{4.10}$$

where $\tilde{x}' = \tilde{x}_{i+1/2} - \tilde{u}(t - t')$ is the trajectory of a particle motion relative to the moving cell interface and f_0 is the initial gas distribution function f at the beginning of each time step ($t = 0$). The scheme based on the above solution will be identical to the BGK-NS method [86], even though the system here $(\lambda, \tilde{x})$ is moving relative to the stationary system (t, x). The difference only appears in the construction of the equilibriums state, which is presented in the following.

In the local moving frame of reference at interface $\xi = \xi_{i+1/2}$, the Maxwellian distribution should have the form

$$g = \rho(\frac{\lambda}{\pi})^{(K+2)/2} \exp\{-\lambda[(\tilde{u} - \tilde{U})^2 + (\tilde{v} - \tilde{V})^2 + \zeta^2]\},$$

where the averaged macroscopic fluid velocity $(\tilde{U}, \tilde{V})$ is related to the fluid velocity (U, V) in the inertia frame of reference,

$$\begin{cases} \tilde{U} = (U - U_g)M/S - (V - V_g)L/S, \\ \tilde{V} = (U - U_g)L/S + (V - V_g)M/S. \end{cases} \tag{4.11}$$

Numerically, Eq.(4.9) is basically the same equation as the one we have solved in the Eulerian space, where $\tilde{u}, \tilde{v}$ are the particle velocity, and $\tilde{U}, \tilde{V}$ are the macroscopic velocity in the $\vec{n}$ and $\vec{t}$ directions. Then, the standard BGK-NS method [86] can be used to solve Eq.(4.9) to evaluate the time-dependent gas distribution function $f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta)$ at the cell interface. The detailed formulation of the gas-kinetic BGK-NS scheme for the Navier-Stokes solutions is given in [86]. Therefore, standing on the moving cell interface the fluxes for unit length can be explicitly obtained,

$$\begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_{\rho\tilde{u}} \\ \mathcal{F}_{\rho\tilde{v}} \\ \mathcal{F}_{\tilde{E}} \end{pmatrix}_{i+1/2,j} = \int \tilde{u} \begin{pmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \zeta^2) \end{pmatrix} f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta) d\Xi. \tag{4.12}$$

Since different numerical cells can be deformed with different grid velocity, in order to update the flow variables inside each time-dependent computational cell we need to update the conservative variables inside each cell in the common inertia frame of reference, i.e., the so-called Eulerian space. Therefore, we need to transfer the fluxes in Eq.(4.12) standing on the moving cell interface into the fluxes for the mass, momentum and energy transport in the inertia frame of reference. In other words, the above obtained gas distribution function $f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta)$ and its mass flux across the moving cell interface $\tilde{u}f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta)$ will carry the mass, momentum and energy densities $(1, u, v, \frac{1}{2}(u^2 + v^2 + \zeta^2))$ of each particle defined in the inertia frame of reference. So, the time-dependent numerical flux in the Eulerian space in the $\vec{n}$ direction across the cell interface $\xi = \xi_{i+1/2}$ should be calculated as

$$\begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_m \\ \mathcal{F}_n \\ \mathcal{F}_E \end{pmatrix}_{i+1/2,j} = \int S\tilde{u} \begin{pmatrix} 1 \\ u \\ v \\ \frac{1}{2}(u^2 + v^2 + \zeta^2) \end{pmatrix} f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta) d\Xi. \qquad (4.13)$$

In the above equation, the distribution function f is a scalar function, which is invariant under coordinate transformation, but the particle velocities $(\tilde{u}, \tilde{v})$ and (u, v) are defined differently in the different frame of reference. In order to evaluate the above flux integration, the easiest way is to write the (u, v) velocities in terms of $(\tilde{u}, \tilde{v})$. Based on the transformation (4.7), we have

$$u = U_g + \frac{M\tilde{u} + L\tilde{v}}{S}, \qquad v = V_g + \frac{-L\tilde{u} + M\tilde{v}}{S}.$$

Therefore, Eq.(4.13) becomes

$$\mathbf{F}_{i+1/2,j} = \begin{pmatrix} S\mathcal{F}_\rho \\ M\mathcal{F}_{\rho\tilde{u}} + L\mathcal{F}_{\rho\tilde{v}} + SU_g\mathcal{F}_\rho \\ -L\mathcal{F}_{\rho\tilde{u}} + M\mathcal{F}_{\rho\tilde{v}} + SV_g\mathcal{F}_\rho \\ (MU_g - LV_g)\mathcal{F}_{\rho\tilde{u}} + (LU_g + MV_g)\mathcal{F}_{\rho\tilde{v}} + S\mathcal{F}_{\tilde{E}} + \frac{S}{2}(U_g^2 + V_g^2)\mathcal{F}_\rho \end{pmatrix} \qquad (4.14)$$

where $\mathbf{F}_{i+1/2,j} = (\mathcal{F}_\rho, \mathcal{F}_m, \mathcal{F}_n, \mathcal{F}_E)^T_{i+1/2,j}$, $(\mathcal{F}_\rho, \mathcal{F}_{\rho\tilde{u}}, \mathcal{F}_{\rho\tilde{v}}, \mathcal{F}_{\tilde{E}})^T$ given in Eq.(4.12). So, the fluxes relative to the moving cell interface in the Eulerian space are just the linear combinations of the fluxes in the moving frame of reference due to the linear transformation between the inertia and moving space with relative velocity (U_g, V_g). Similarly, the fluxes at the cell interface $\eta = \eta_{j+1/2}$, i.e., $\mathbf{G}_{i,j+1/2}$, can be constructed as well.

With the above fluxes, the flow variables can be updated in each moving computational cell such as the cell (i, j) by

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} + \frac{1}{\Delta\xi}\int_{t^n}^{t^{n+1}} (\mathbf{F}_{i-1/2,j} - \mathbf{F}_{i+1/2,j})dt + \frac{1}{\Delta\eta}\int_{t^n}^{t^{n+1}} (\mathbf{G}_{i,j-1/2} - \mathbf{G}_{i,j+1/2})dt, \quad (4.15)$$

where $Q = (\rho\triangle, \rho\triangle U, \rho\triangle V, \rho\triangle E)^T$ and $\mathbf{F} = (\mathcal{F}_\rho, \mathcal{F}_m, \mathcal{F}_n, \mathcal{F}_E)^T$ are given in Eq.(4.14).

4.1.2 Adaptive grid method

If the moving grid velocity (U_g, V_g) is chosen according to the fluid velocity, such as the Lagrangian method with $(U_g, V_g) = (U, V)$, it is very hard to keep the mesh regularity. In almost all cases, the mesh will get tangled to generate a negative cell area $\triangle$. In order to prevent this from happening, many mesh remedy techniques have been developed, such as ALE method. Here, we are going to use the adaptive mesh method [74] presented in the last chapter to overcome mesh-tangling problem. The function of this is two folds. Firstly, a smooth mesh will be generated before grid crashing. Secondly, the mesh can be automatically redistributed and concentrated on the region with high velocity or pressure gradients to improve the computational accuracy. However, when more grid points are concentrated in the shock region, the grid size becomes very small, as well as the time step. Therefore, the computational cost will be much increased. Also, since we do not resolve the shock structure, the order of accuracy cannot be improved by the mesh adaptation [90]. But, the absolute value of accuracy can be improved due to the small cell size. Certainly, it depends on the accuracy control in the interpolation between different meshes. The method in detail is presented in Chapter 3.

As the monitor functions (3.7) are assumed to have $w_1 = w_2 = 1$, the Euler-Lagrange equation (3.6) becomes the Laplace's equation, where a smooth mesh can be generated. The frequency to apply the adaptive mesh method depends on the test cases. For example, in the first test case in this chapter, after a few testing calculations, we know precisely when the mesh will get severely distorted. Before that, the smoothing mesh method can be applied. After that, the mesh is smoothed every 200 time steps. However, for the examples (5) and (6), due to the requirement of accurate capturing of the shock and boundary layer, the adaptive grid technique is applied at every time step to redistribute the mesh concentration. Again, the purpose of the adaptive grid generation is to regularize the mesh in the physical space which may be distorted due to the mesh movement, or concentrate more grids on the region with high density, velocity or pressure gradients to enhance the numerical resolution. The newly generated mesh has only a small modification from the original mesh with a limited number of iterations in solving (3.6).

4.1.3 Numerical procedure

The numerical procedure for the gas-kinetic scheme on a moving and adaptive mesh can be summarized in the following. In some numerical examples, the adaptive mesh method is not used, and the corresponding step can be ignored. Also, all fluid quantities $(\rho, \rho U, \rho V, \rho E)$ and the geometric variables (A, B, M, L) are defined at each cell center. Any cell interface values are obtained through the average of the values in the adjacent cells.

Step 1: Initialize the conservative variables $(\rho, \rho u, \rho v, \rho E)$ and (A, B, L, M) at $t = \lambda = 0$ in the $x - y$ plane. Usually, ξ and η are the initial arc-length of their corresponding x- and y-coordinate lines. For example, for the rectangular domain, we take $\triangle\xi = \triangle x$, and $\triangle\eta = \triangle y$ when $\triangle x$ and $\triangle y$ are constants on the physical domain initially. Or, we can simply choose two constants to define $\triangle\xi$ and $\triangle\eta$. Then, (A, B, L, M) are determined according to the definition,

$$\begin{cases} A = \frac{\partial x}{\partial \xi}, \\ B = \frac{\partial y}{\partial \xi}, \\ L = \frac{\partial x}{\partial \eta}, \\ M = \frac{\partial y}{\partial \eta}, \end{cases} \tag{4.16}$$

Then, we calculate $(\triangle\rho, \triangle\rho u, \triangle\rho v, \triangle\rho E)$ inside each cell with $\triangle = AM - BL$.

Step 2: Construct or modify the grid using the adaptive grid method if the adaptive grid is needed, and update the conservative variables $(\rho, \rho u, \rho v, \rho E)$ inside each cell in the physical domain. Sequently, calculate (A, B, L, M) by (4.16) again and update $(\triangle\rho, \triangle\rho u, \triangle\rho v, \triangle\rho E)$ with the new $\triangle = AM - BL$. This step of re-distributing the conservative variables is fully conservative.

Step 3: Given a grid velocity (U_g, V_g) at the center of each cell, such as the fluid velocity $\mathbf{q}$ in the gas-kinetic Lagrangian method, at the cell interface $\xi = \xi_{i+1/2}$ the grid velocity is calculated as

$$(U_g)_{i+1/2,j} = \frac{(U_g)_{i,j} + (U_g)_{i+1,j}}{2},$$
$$(V_g)_{i+1/2,j} = \frac{(V_g)_{i,j} + (V_g)_{i+1,j}}{2},$$

which are basically the average of the velocities from the neighboring cells. Then, based on (4.14) the numerical fluxes are calculated across the moving interface. At the same time, the center of the cell, such as $(x_{i,j}^n, y_{i,j}^n)$, moves to a new location through

$$\begin{cases} x_{i+1/2,j}^{n+1} = x_{i+1/2,j}^{n} + U_g \times (t^{n+1} - t^n), \\ y_{i+1/2,j}^{n+1} = y_{i+1/2,j}^{n} + V_g \times (t^{n+1} - t^n), \end{cases} \tag{4.17}$$

The location of cell vertex is updated by averaging the centers of 4 neighboring cells.

Step 4: With the new cell location x^{n+1}, y^{n+1}, calculate (A, B, L, M) and $\triangle$ using (4.16). Then, update the conservative variables by the finite volume scheme (4.15) in the newly moved cells. It can be shown that the use of (4.16) to calculate (A, B, L, M) is the same as the geometrical conservation law (4.3), such as the update of A,

$$
\begin{aligned}
A_{i,j}^{n+1} &= (\frac{\partial x}{\partial \xi})_{i,j}^{n+1} \\
&= \frac{x_{i+1/2,j}^{n+1} - x_{i-1/2,j}^{n+1}}{\Delta \xi} \\
&= \frac{x_{i+1/2,j}^{n} + (U_g)_{i+1/2,j}^{n}\Delta\lambda^n - x_{i-1/2,j}^{n} - (U_g)_{i-1/2,j}^{n}\Delta\lambda^n}{\Delta \xi} \\
&= \frac{x_{i+1/2,j}^{n} - x_{i-1/2,j}^{n}}{\Delta \xi} + \frac{\Delta\lambda^n}{\Delta\xi}((U_g)_{i+1/2,j}^{n} - (U_g)_{i-1/2,j}^{n}) \\
&= A_{i,j}^{n} + \frac{\Delta\lambda^n}{\Delta\xi}((U_g)_{i+1/2,j}^{n} - (U_g)_{i-1/2,j}^{n}).
\end{aligned}
$$

Similarly, it is true for B, L, and M.

Then, go to step (2) to repeat the above process until the output time.

4.2 Numerical experiments

As mentioned earlier, the main advantage of the current method is that the fluid system under unified coordinate is coupled with the geometrical conservation laws. Mathematically, the unified system provides a complete enlarged system, which avoids the difficulties in other moving mesh methods to implement the geometrical conservation separately based on the physical intuition. Also, different from the Eulerian and Lagrangian methods, the choice of grid velocity (U_g, V_g) becomes a new degree of freedom in the current method. The proper choice of the grid speed depends on the specific applications in order to get the optimum accuracy in the numerical solution. For example, the grid velocity can be the fluid velocity in the free surface flow, or follow the oscillating angular velocity for a pitching airfoil. In the approach by Hui et al [31], to preserve the grid angle has been used to get the local value h in the determination of grid velocity (hU, hV). In this section, the BGK-NS scheme on a moving mesh will be tested on many examples. Different choice of the grid velocity will be used. The numerical results are compared with the exact solutions, experimental data, and the available solutions obtained by others.

Case (1): free surface flow

The current approach with the choice of $(U_g, V_g) = (U, V)$ becomes a purely gas-kinetic Lagrangian method even though it is solving the viscous governing equations. The use of the fluid velocity as the grid speed can naturally capture the free surface. The case we are going to study is the dam break problem, where a column of water is released by removing a vertical diaphragm. This becomes a standard benchmark problem due to its simple geometry and the available experimental measurement [27]. The initial configuration is shown in the upper left picture in Fig. 4.3. In this example, a rectangular column of water in hydrostatic equilibrium is confined initially between two vertical walls. The water column is 7.0 units wide and 3.5 units high. The gravity is acting downward with 0.05 unit magnitude.

After the diaphragm eruption at time $t = 0$, the water is pushing out and moves freely along a dry horizontal floor. The measured quantities include the water wave front location L on the floor. In our calculation, 40×20 rectangular mesh points are initially employed in a domain $0 \leq x \leq 3.5, 0 \leq y \leq 7.5$. Fig. 4.3 presents the moving mesh at 4 subsequent times. Since the grid speed is equal to the fluid velocity, the mesh distribution is the same as the water distribution. In a purely Lagrangian simulation, the mesh is easily tangled at a later time. Since we have used a mesh smoothing technique through the mesh adaptive method with a constant monitor function to equally distribute the mesh, the moving mesh becomes generally smooth all over the domain even though the mesh adaptation steps are applied only a few times.

Fig. 4.4 shows the water tip location vs. time for both simulation and experimental measurement. The non-dimensional time in the horizontal coordinate is normalized by $t\sqrt{2g/W}$. The numerical results are in excellent agreement with the experimental measurement.

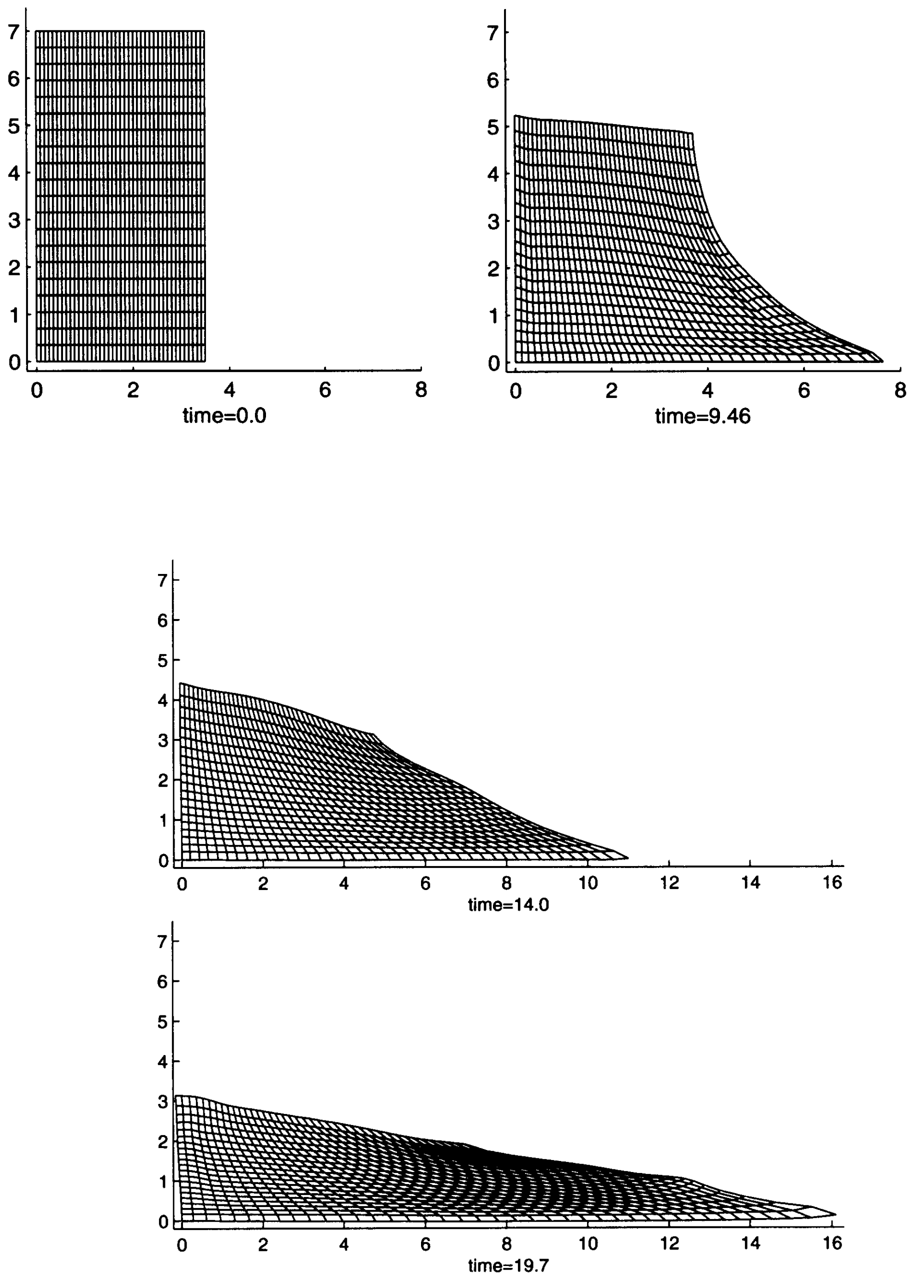


Figure 4.3: Moving mesh at time $t = 0, 9.46, 14.0, 19.7$.

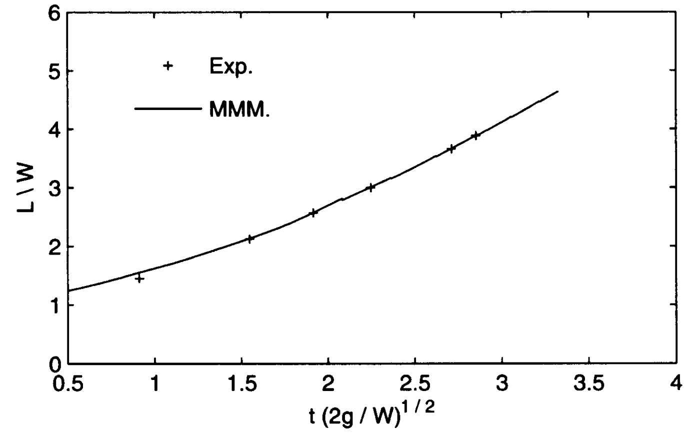


Figure 4.4: The time dependent location of water front. Experimental measurement vs. the calculation from moving mesh method (MMM).

Case (2): 2D model of a pitching airfoil

This is about the flow passing through a pitching airfoil. The airfoil undergoes pitching oscillations around a point on the chord with one quarter length from the leading edge. In the current case, a NACA 0012 airfoil was used and the chord was placed at the x axis initially. The free stream velocity U_∞ is parallel to the x axis with a Mach number $M = 0.755$. The oscillation cycle is defined by ,

$$\alpha = \alpha_m + \alpha_0 \sin \omega t, \quad \text{with} \quad \alpha_m = 0.016^\circ \quad \text{and} \quad \alpha_0 = 2.51^\circ,$$

where ω is the circular pitch frequency, c is chord length, and $\omega c / 2U_\infty = 0.0814$. In many computations, the mesh is fixed and the free stream U_∞ velocity is being rotated in the opposite direction as the airfoil pitching. However, in the present calculation, the whole mesh around the airfoil is moving with the airfoil and the grid velocity (U_g, V_g) is calculated according to the angular velocity of the pitching airfoil and the distance between the grid point and oscillating center. Since only mesh is rotating, all flow variables at ∞, such as U_∞, are unchanged in the current Eulerian space. In our calculation, a U-type mesh with 178 × 44 grid points on a span of 8 chord distance is used. In Fig. 4.5, the moving meshes near the airfoil are shown while the airfoil is at two different angles of attack. The oscillating center is at the original.

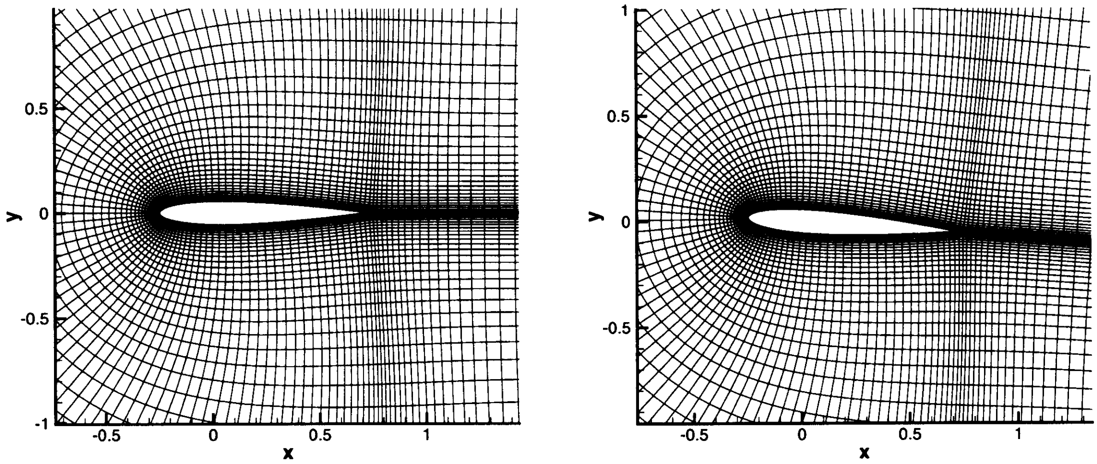


Figure 4.5: Moving meshes near the NACA 0012 airfoil with two different angles of attack: the left is $\alpha = \alpha_m$, the right is $\alpha = \alpha_m + \alpha_0$

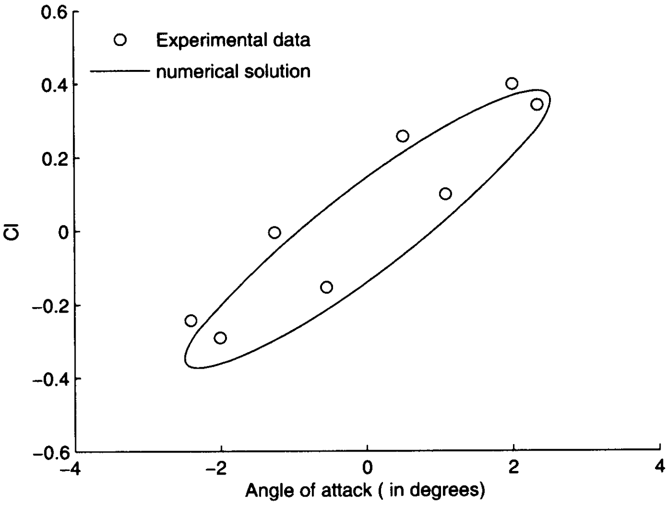
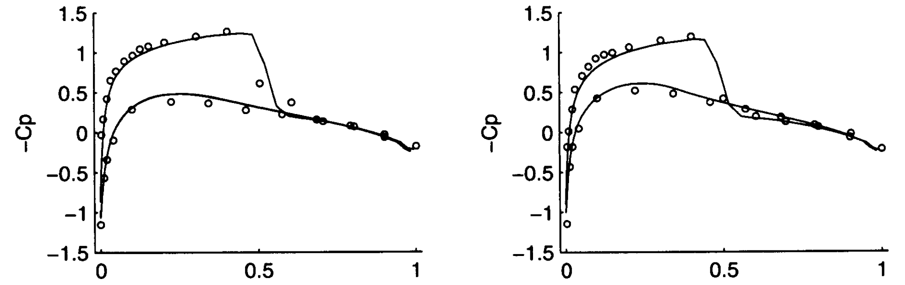
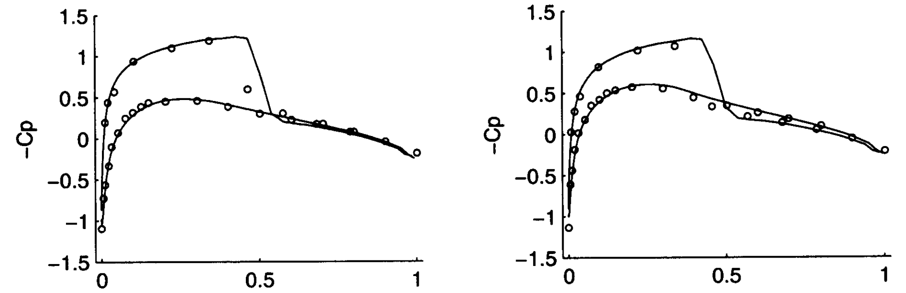


Figure 4.6: Lift coefficients versus angle of attack in the airfoil pitching problem.



Angle of attack = 2.01° (down)

Angle of attack = 0.52° (down)



Angle of attack = -2.0° (up)

Angle of attack = -0.54° (up)

Figure 4.7: Pressure distributions at different angle of attack.

Fig.4.6 presents the lift coefficient C_l versus instantaneous angle of attack α. The numerical solutions give good agreements with the experimental data. Fig. 4.7 presents the detailed pressure distribution C_p around the surface of the airfoil at different angle of attack α. Also, good agreement between the numerical and experimental results are obtained. Since the grid velocity in the current unified gas-kinetic scheme can be chosen freely, it may have advantage in applying this method in the biology insect flight computations where the mesh can follow the complicated wing movement.

Case (3): Shock reflection inside a channel with a ramp

In order to preserve the grid angles [31], the grid speed $(U_g, V_g) = (hU, hV)$ can be obtained by solving the following equation for h,

$$\frac{\partial}{\partial \lambda}\left[\frac{\nabla\xi}{|\nabla\xi|}\cdot\frac{\nabla\eta}{|\nabla\eta|}\right] = 0,$$

where $\nabla\xi = (M, -L)/\Delta, \nabla\eta = (-B, -A)/\Delta$, and θ is the flow angle: $U = q\cos\theta$,

$V = q\sin\theta$, $q = \sqrt{U^2+V^2}$. Let $\tilde{g} = \ln(hq)$, then the above equation becomes

$$\begin{aligned} S^2(A\sin\theta &- B\cos\theta)\frac{\partial\tilde{g}}{\partial\xi} + T^2(M\cos\theta - L\sin\theta)\frac{\partial\tilde{g}}{\partial\eta} \\ &= S^2\left(B\frac{\partial\cos\theta}{\partial\xi} - A\frac{\partial\sin\theta}{\partial\xi}\right) - T^2\left(M\frac{\partial\cos\theta}{\partial\eta} - L\frac{\partial\sin\theta}{\partial\eta}\right) \quad (4.18) \end{aligned}$$

Numerically, Eq.(4.18) can be solved by an iterative method to get h at different grid point. The detail description was given in [31].

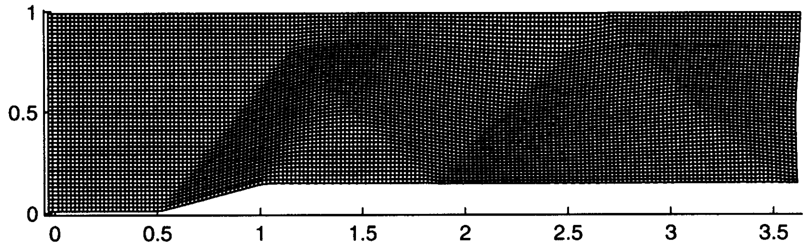


Figure 4.8: Computational mesh due to the implementation of grid angles preservation.

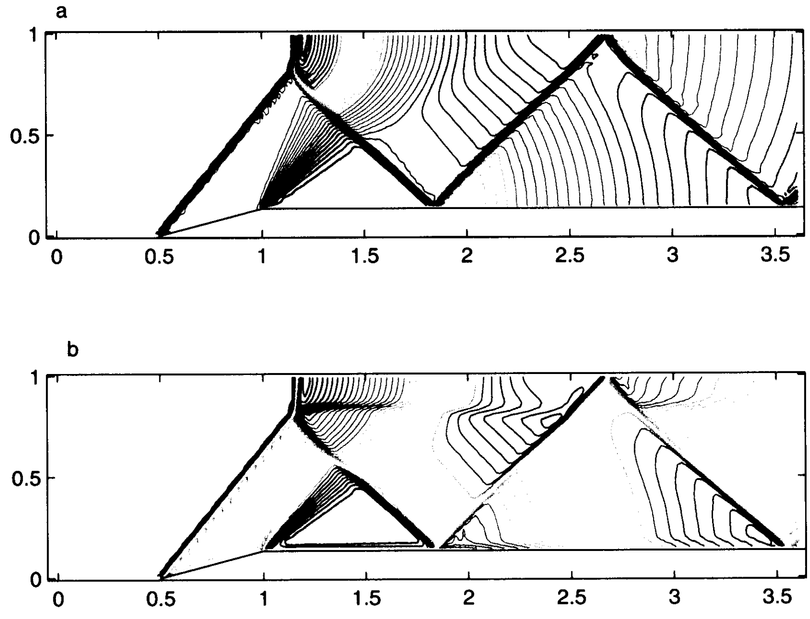


Figure 4.9: Pressure contours (up) and Mach (below) contours.

In the current supersonic flow of $M = 1.8$ passing through a ramp in a channel, Eq.(4.18) is used to calculate h, then determine the grid velocity. Here the ramp with 15° is placed at the lower wall between $x = 0.5$ and $x = 1.0$. A computational grid with 180 × 50 grid points in the physical domain $\{0 \leq x \leq 3.6, 0 \leq y \leq 1.0\}$ is initially generated. An initial flow data with $(p, \rho, M, \theta) = (1, 1, 1.8, 0)$ is imposed inside each cell, as well as at the left boundary. Reflection boundary conditions are used at the top and bottom walls. When h is chosen according to Eq. (4.18), the mesh will automatically preserve the grid angle, which is shown in Fig.4.8. Fig.4.9 presents pressure and Mach number distributions after the steady state solution is obtained.

Case (4) Viscous solution above an oscillating wall

This is called Stokes' second problem, which considers fluid motion above an infinite flat plate which executes sinusoidal oscillations parallel to itself. This problem has been simulated earlier in the last chapter for the purely adaptive mesh method. The fluid above the plate is initially stationary. The governing equation of velocity U in the x-direction is

$$\frac{\partial U}{\partial t} = \nu \frac{\partial^2 U}{\partial y^2}, \tag{4.19}$$

with the boundary conditions

$$U_{\text{wall}}(0,t) = U_0 \cos \omega t \quad , \quad U(\infty, t) = 0.$$

The exact solution for the above problem is,

$$U(y,t) = U_0 e^{-y\sqrt{\omega/2\nu}} \cos(\omega t - y\sqrt{\frac{\omega}{2\nu}}). \tag{4.20}$$

At $y = 4\sqrt{\nu/\omega}$, the amplitude of U is equal to $U_0 \exp(-4/\sqrt{2}) = 0.05 U_0$, which means that the influence from the wall is confined within a distance of order $\delta \sim 4\sqrt{\nu/\omega}$. Since the gas-kinetic scheme solves the compressible Navier-Stokes equations, in order to simulate the above incompressible limiting solution the Mach number for the compressible flow takes a small value, i.e., $M = 0.15$. The kinematic viscosity coefficient takes a value $\nu = 0.00046395$, and a mesh size 10 × 20 grid points is used.

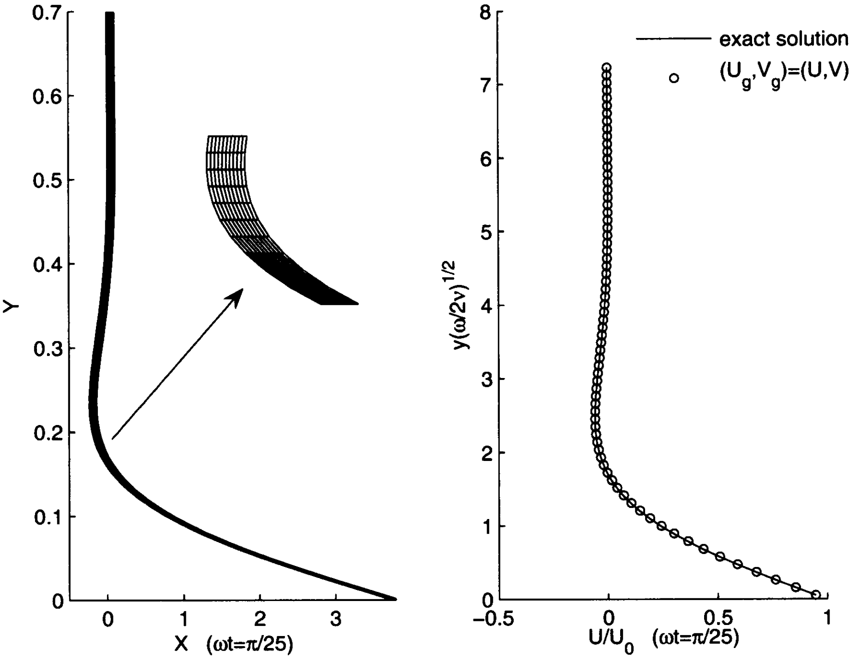


Figure 4.10: Lagrangian gas-kinetic scheme for viscous flow. Mesh (left) and velocity (right) distributions at time $\omega t = \pi/25$.

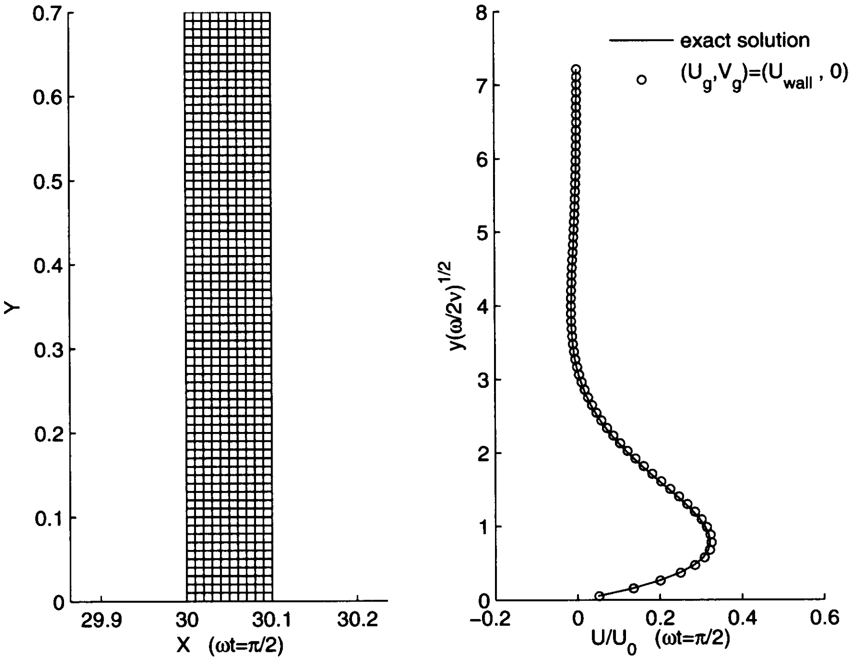


Figure 4.11: Gas-kinetic scheme for viscous flow with $(U_g, V_g) = (U_{wall}, 0)$. Mesh (left) and velocity (right) distributions at time $\omega t = \pi/2$.

In the current calculation, we have used two ways to determine the mesh velocity. In the first case, we used the purely Lagrangian method for the viscous solution, where the grid velocity follows the fluid one. Due to large velocity shear in the boundary layer, the Lagrangian method will stretch the mesh severely. This is the main reason why nobody really used Lagrangian method for the viscous flow computation. However, it is still theoretically interesting to see the solution using gas-kinetic Lagrangian method. Fig.4.10 shows the mesh (left) and velocity (right) at time $\omega t = \pi/25$ when the mesh follows the fluid velocity. Even though the mesh has been stretched greatly, it is surprising that the numerical solution is very close to the exact solution. This proves the robustness and accuracy of the current kinetic scheme for the viscous computation. With the time increasing, the mesh will be stretched further until 20 grid points are not enough to follow the time increasing velocity arc-length. Eventually, the computation will stop. If the mesh velocity at all grid points follow the wall velocity, such as $(U_g, V_g) = (U_{wall}, 0)$, the mesh will not get tangled. Fig.4.11 shows the mesh and simulation results at $\omega t = \pi/2$.

Case (5) Shock wave interaction with a rotating cylinder

An incident shock wave with Mach number $M_s = 1.19$ from the left impinges on an anti-clockwise rotating cylinder. The test gas is air at pressure $P_{-\infty} = 100kP_a$ and temperature $T_{-\infty} = 298K$ respectively. In the current computation, non-slip boundary condition is imposed around the surface of the rotating cylinder. The dynamical viscosity coefficient μ takes the power law,

$$\frac{\mu}{\mu_{-\infty}} = \left(\frac{T}{T_{-\infty}}\right)^n = \left(\frac{1/\lambda}{1/\lambda_{-\infty}}\right)^n, \qquad n = 0.666,$$

where $\lambda = \rho/2p$, $\lambda_{-\infty} = \rho_{-\infty}/2p_{-\infty}$, and $\mu_{-\infty} = 1.716 \times 10^{-5} Ns/m^2$. The cylinder has a radius $R = 0.07m$ and the surface rotating velocity has a value to be equivalent to Mach number $M = 0.3$. Besides the mesh velocity follows the rotating cylinder with a value $r\omega$, where r is the distance from the rotating center point to the mesh point,

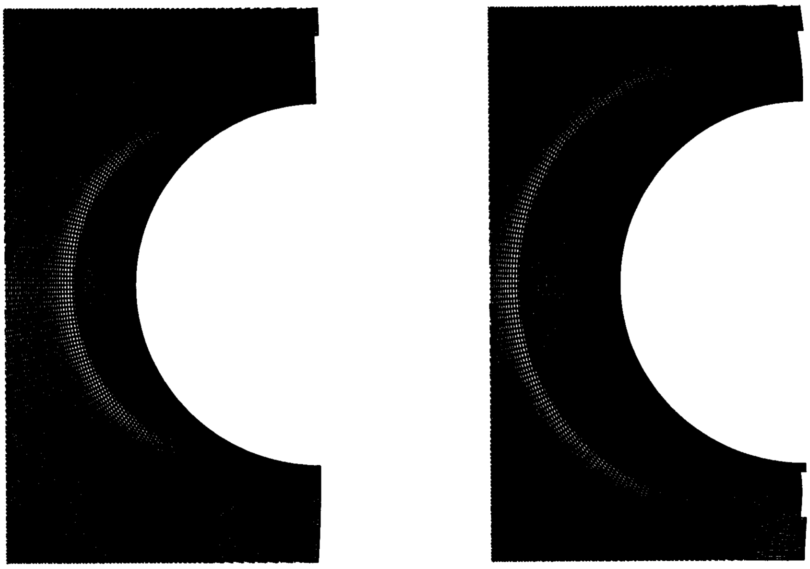


Figure 4.12: Moving and adaptive grid at time $t = 0.106, 0.179$.

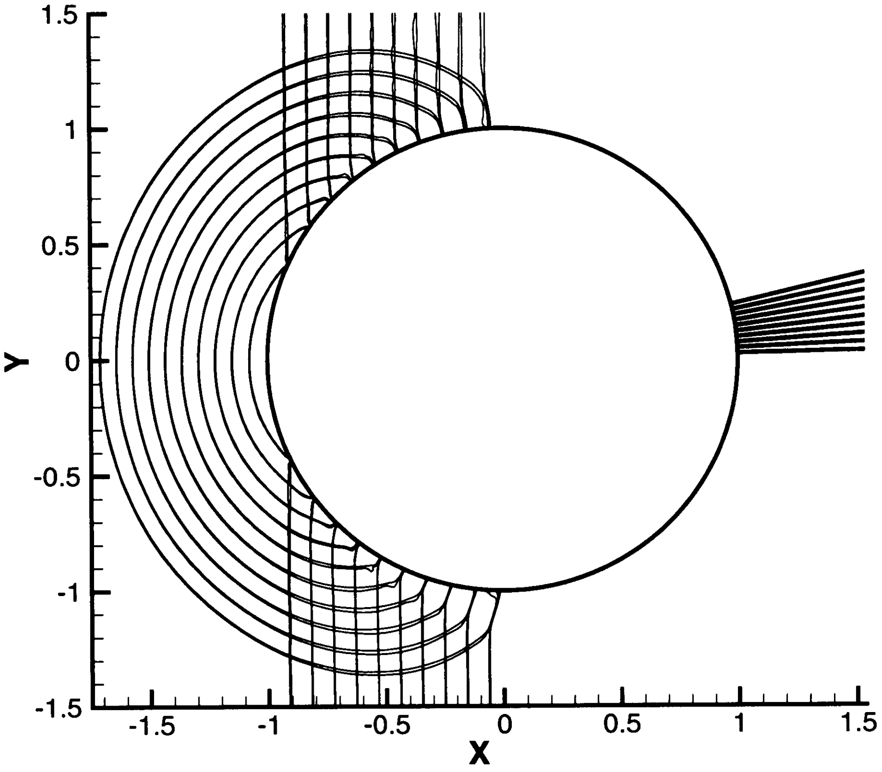


Figure 4.13: Sequential pressure contours after shock reflection from the rotating cylinder.

an adaptive mesh method is also used to concentrate the mesh in the high velocity gradient region. The solutions around the shock front and inside the boundary layer are captured accurately due to the mesh concentration. Numerical mesh at times $t = 0.106, 0.179$ are shown in Fig.4.12. Fig.4.13 shows the sequential pressure contours after shock reflection from the cylindrical surface. Because of the viscous effect, the locations of the triple points on the upper and lower surfaces are different, as well as the length of Mach stem. The Mach stem is longer on the lower surface where the shock wave propagates with the same direction as the rotating surface. On the contrary, on the upper surface the Mach stem is suppressed due to their opposite moving direction.

Although accurate measurement of shock wave over a moving body is rather difficult, the different shock wave distribution on both upper and lower surfaces has been observed experimentally. In [71], the experimental observation showed that the length of the Mach stem on the lower surface is indeed longer than that on the upper surface, which is consistent with the current computational result. For the inviscid flow computation, the solution is irrelevant with the rotation of the cylinder due to the implementation of slip boundary condition.

Case (6): Rotating cavity flow

The cavity flow inside a square cylinder with upper moving boundary has been calculated by many authors and becomes one of the benchmark test case for the incompressible Navier-Stokes equations [24]. Here, we design a new test case, where the whole 2D square cavity with boundary length 1 starts to rotate around its center at time $t = 0$ with a constant angular velocity $\omega = 1$, where the viscosity coefficient is set to a value with the Reynolds number $Re = 1000$ for $U = 1$ and $L = 1$. Since we are using a compressible code to calculate the incompressible solution, the initial Mach number according to the rotating velocity at the cavity surface is $M = 0.15$. A mesh with 65×65 grid points is used in the current computation. The whole computational mesh is rotating with the cavity, at the same time adaptated according to the velocity

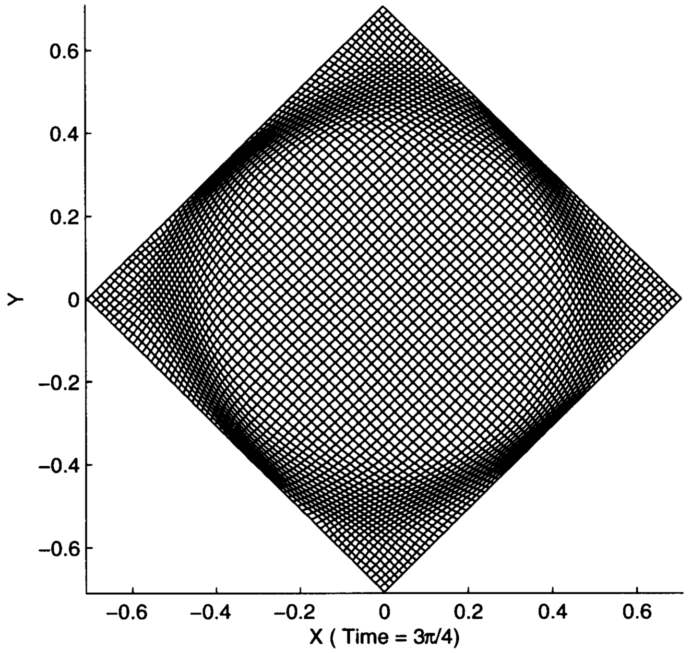


Figure 4.14: Moving and adaptive grid at time $t = \frac{3}{4}\pi$ for the rotating cavity case.

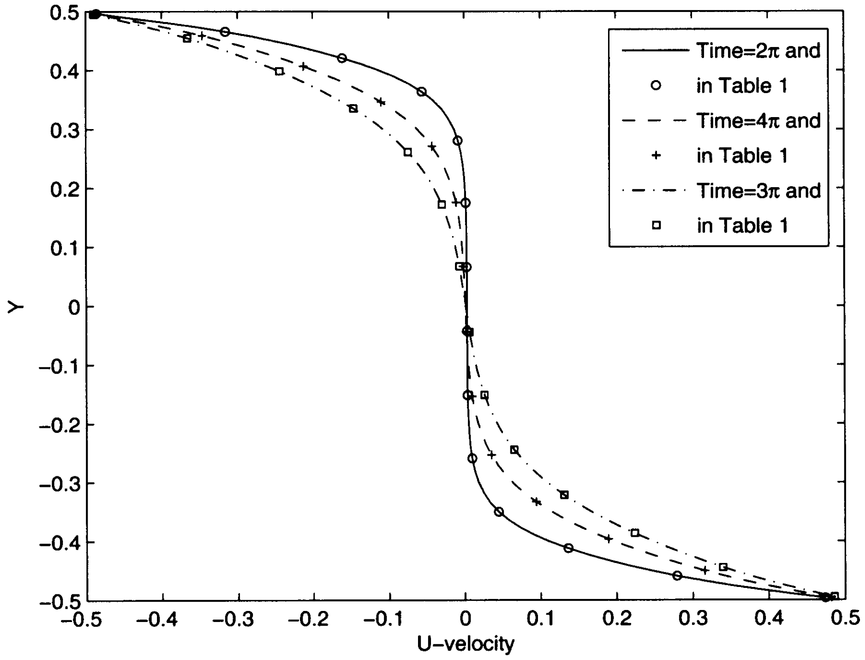


Figure 4.15: Velocity U profiles along the central vertical line at $Re = 1000$ after one, two and three full rotations.

Table 4.1: U-Velocity distributions along Vertical Central line at Re=1000

Time=2π		Time=4π		Time=6π	
y	U	y	U	y	U
−0.49674	0.47419	−0.49580	0.48181	−0.49513	0.48611
−0.45919	0.27910	−0.45029	0.31605	−0.44439	0.33977
−0.41211	0.13619	−0.39639	0.18935	−0.38667	0.22391
−0.35012	0.04468	−0.33326	0.09404	−0.32162	0.13113
−0.25932	0.00967	−0.25374	0.03519	−0.24469	0.06532
−0.15179	0.00382	−0.15394	0.00975	−0.15151	0.02615
−0.04298	0.00329	−0.04441	0.00235	−0.04476	0.00663
0.06586	0.00299	0.06659	−0.00114	0.06698	−0.00688
0.17473	0.00177	0.17518	−0.01107	0.17172	−0.02974
0.28056	−0.00831	0.27106	−0.04276	0.26130	−0.07437
0.36354	−0.05603	0.34621	−0.10971	0.33513	−0.14631
0.42069	−0.16095	0.40677	−0.21226	0.39825	−0.24436
0.46533	−0.31610	0.45867	−0.34656	0.45427	−0.36586
0.49594	−0.48614	0.49525	−0.48838	0.49474	−0.48976

gradients. Fig.4.14 shows the mesh in the physical domain at time=$\frac{3}{4}\pi$. Obviously, grid points are concentrated in regions close to the boundaries with large velocity gradients. Fig.4.15 displays the velocity U profiles along the central vertical line after one, two, and three rotations $\omega t = 2\pi, 4\pi$, and 6π. The specific velocity values are listed in Table 1 for future reference. Since this is an unsteady problem, the U-velocity along vertical central line depends on time in the early transition period.

In summary, in this Chapter a gas-kinetic BGK scheme is constructed under a unified coordinate system for the viscous solution is developed. This is a finite volume gas-kinetic scheme on a moving grid in the Eulerian space and the mesh velocity can be properly chosen to capture flow solution with moving boundaries. The Eulerian and Lagrangian methods are two limiting cases for the current scheme. In order to increase the robustness and accuracy of the current method, the mesh adaptation is also implemented in the current method, which not only remedies the distorted mesh, but also concentrates the mesh in the high gradient regions. For the first time, the Lagrangian gas-kinetic scheme has been used in the viscous boundary layer calculation. The great advantage of the current scheme with a variable mesh velocity is that the kinetic equation and geometrical conservation laws are combined as a unified system. The physical and geometrical variables can be updated simultaneously.

Chapter 5

Numerical study for unsteady aerodynamics of freely falling plates

It is a fact that not all falling objects travel straight downwards, such as leaves, tree seeds, and paper cards dropped from the table. Obviously, falling leaves and tumbling sheets of paper can get a lift momentarily to float upward against gravity as they flutter (oscillate from side to side) or tumble (rotate and drift sideways) through still air. To explain this complicated natural phenomena requires the knowledge of the instantaneous fluid forces. Previously, only few analytical results exist and they are in special limits, such as Stokes flow and inviscid irrotational flow around a steadily translating airfoil. However, most objects moving in a fluid encounter unsteady aerodynamic forces. Therefore, recently the problem of falling plate has been investigated numerically by solving two dimensional Navier-Stokes equations following a moving body [57], where the instantaneous fluid force for the fluttering and tumbling motion has been analyzed.

The fluttering, looping, and tumbling motions have been of interest to physicists since the 19th century, when famed Scottish physicist James Clerk Maxwell studied falling cards and offered a qualitative explanation of the correlation between the sense of rotation and the drift direction of a tumbling card [51], when the classical aerodynamic theory had not been established yet. In the past decades, most investigation only presented qualitative or average properties, where the instantaneous fluid forces were not obtained. Dupleich (1941) [20] measured the angle of descent and average tumbling frequency as functions of wing loading and ratio between plate length and width. Willmark, Hawk and Harvey (1964) [82] performed a beautiful set of falling disk experiments with steady descent, fluttering, and tumbling. And in the limit of small thickness to width ratio, they found that the final state depends only on the Reynolds number and the dimensionless moment of inertia [82]. Smith (1971) [68] also measured a phase diagram for falling plates qualitatively similar to the one for the falling disks. Field, Klaus, Moore and Nori (1997) [23] made further experiment on freely falling disks, and mapped out the phase diagram for falling disks showing the regions where one observes the different modes of motion. Belmonte, Eisenberg and Moses (1998) in [5] presented a simple laboratory experiment on flat strips dropped in a quasi two-dimensional geometry, and shown that the transition from flutter to tumble is determined by the Froude number, with a velocity scale set by the inertial drag. Subsequently, Mahadevan, Ryu and Samuel (1999) [49] observed a scaling law for the dependence of the rotational speed on the width of a tumbling card, which is valid in a regime far from the onset of tumbling and also from the region when the strip becomes extremely flexible and starts to bend. Mittal, Seshadri and Udaykumar (2004) in [54] considered a freely falling cylinder with rounded edges, and studied it by using the numerical solution of the two-dimensional Navier-Stokes equations. Recently, Jones and Shelley (2005) suggested a falling card model based on inviscid theory and the unsteady Kutta condition in [35]. Unfortunately, due to the limited capability of the proposed numerical methods, the detailed quantitative numerical computation of periodic fluttering and tumbling motion has been done only recently.

Among the most successful simulations is the study by Andersen, Pesavento, and Wang [J. Fluid Mech., vol. 541, pp. 65-90 (2005)], where the quantitative comparison between the experiment measurement and numerical computation was presented. The rich dynamical behavior, such as fluttering (oscillating from side to side) or tumbling (rotating and drifting sideways), was fully analyzed. However, obvious discrepancies between the experiment measurement and numerical simulations still exist, such as the forces and torque on the tumbling plates, as well as the plate trajectory. In this paper, we are going to present similar numerical computation, but with a newly developed moving mesh method for the viscous flow. In the unified moving mesh method in chapter 4, the computational grid is fixed rigidly with the plate and the grid velocity is determined according to the translational and rotational velocities of the falling plate. Even though the experiment was conducted by Andersen et al on a rectangular plate, the early numerical computation was solely based on the elliptic plate due to the requirement of the conformal mapping in their numerical method. However, our numerical method has no such a limitation and both elliptic and rectangular falling plates will be fully studied. Therefore, some observations presented in [1] will be examined in the current study. Our computation shows that the averaged translational velocity for both rectangular and elliptical plates are almost identical, which is different from early numerical observation. But, the plate rotation depends strongly on the shape of the plate. For the elliptic plate the pressure force is almost perpendicular to the longer axis of the plate during the tumbling movement. However, for the rectangular one the pressure force can be tilted greatly relative to the longer symmetric axis. Also, it is observed that the viscous force for the elliptic plate is always in the opposite direction of the plate movement, but it can be in the same direction when the rectangular plate moves upwards. In terms of fluid forces on the plates, excellent agreement between the numerical computation and experimental measurement has been obtained.

5.1 Falling plate motion in a viscous fluid

The motion of a plate falling freely in an incompressible fluid can be described by a coupled system of Navier-Stokes equations for the fluid and Newton's law for the solid plate. Based on the fluid solution, the forces (pressure force and viscous force) and the torque on the plate can be explicitly evaluated. Since the computational mesh is fixed on the falling plate and moves together with the plate, the mesh velocity can be automatically computed according to the movements of the plate, such as the translational and rotational motion. The viscous fluid solution around the moving plate is calculated using the moving mesh method in the last section. In this section, two plates movement, such as ellipse and rectangle, will be simulated.

5.1.1 Parametrization of the freely falling plates

A freely falling plate is characterized by six dimensional parameters: the width l, the thickness h, the density ρ_b of the plate, the density ρ_f of the fluid, the kinematic viscosity ν of the fluid, and the gravitational acceleration g. From the above six dimensional numbers three non-dimensional numbers can be defined: the thickness to width ratio, β, the dimensionless moment of inertial, I^*, and the Reynolds number, Re. Specifically, similar to the method used in [57], we define the ratios of the thickness and width on the plate cross-section,

$$\beta = \frac{h}{l}, \tag{5.1}$$

and the dimensional moment of inertia I and dimensionless moment of inertia I^* for the ellipse,

$$I = \frac{\pi l h (l^2 + h^2)\rho_b}{64}, \quad I^* = \frac{h(l^2 + h^2)\rho_b}{2l^3\rho_f}, \tag{5.2}$$

and for the rectangle,

$$I = \frac{l h (l^2 + h^2)\rho_b}{12}, \quad I^* = \frac{8h(l^2 + h^2)\rho_b}{3\pi l^3\rho_f}. \tag{5.3}$$

The Reynolds number is defined using the width of the plate l and the terminal velocity u_t,

$$Re = \frac{u_t l}{\nu}, \tag{5.4}$$

where u_t is estimated by balancing gravity against the fluid force on a plate with drag coefficient 1, which is

$$u_t = \sqrt{\frac{\pi h g}{2}(\frac{\rho_b}{\rho_f} - 1)} \tag{5.5}$$

for the ellipse, and

$$u_t = \sqrt{2hg(\frac{\rho_b}{\rho_f} - 1)} \tag{5.6}$$

for the rectangle.

5.1.2 Integration of fluid forces and torques on the falling plate

The fluid forces include pressure force and viscous force. These quantities can be calculated by integrating along the plate surface explicitly once the fluid solution is obtained around the falling plate. In the following, we will calculate the fluid forces and torque for both moving elliptic and rectangular plates through the integration of surface forces. The pressure force can be written as an integral of the pressure over the boundary Γ of the body,

$$\vec{F}_p = -\oint_\Gamma p\vec{n}ds = -\oint_\Gamma p(\cos\theta, \sin\theta)ds = -\oint_\Gamma (p_x, p_y)ds, \tag{5.7}$$

where p is the pressure, $\vec{n} = (\cos\theta, \sin\theta)$ is the normal direction to ds, and s is the arc length along the boundary of the ellipse or rectangle. Numerically, the above Eq.(5.7) becomes,

$$\vec{F}_p = -(\sum_i p_i \cos\theta \triangle s_i, \sum_i p_i \sin\theta \triangle s_i) = (P_x, P_y), \tag{5.8}$$

where i stands for the ith grid on the moving boundary, and $\triangle s_i$ is the length of the ith cell interface on the surface of the plate. The total pressure force is decomposed into P_x along x-direction and P_y along y-direction in the inertia frame of reference.

The viscous force is obtained by integrating the stress $\mu\partial U_t/\partial n$,

$$\vec{F}_\nu = -\oint_\Gamma \mu\frac{\partial U_t}{\partial n}\vec{t}ds = -\oint_\Gamma \mu\frac{\partial U_t}{\partial n}(-\sin\theta,\cos\theta)ds = -\oint_\Gamma(\nu_x,\nu_y)ds, \tag{5.9}$$

where $\mathbf{t} = (-\sin\theta,\cos\theta)$ is the tangent direction of ds, $U_t = \vec{U}\cdot\vec{t}$ is the velocity in the tangential direction, and μ is the dynamic viscosity coefficient. Numerically, the above Eq.(5.9) can be written as,

$$\vec{F}_\nu = -(\sum_i \nu_i\cos\theta\triangle s_i, \sum_i \nu_i\sin\theta\triangle s_i) = (\nu_x,\nu_y). \tag{5.10}$$

The viscous force is decomposed into ν_x along the x-direction and ν_y along the y-direction.

The pressure torque is also obtained by integrating along the surface of a falling plate,

$$\tau_p = \oint_\Gamma \vec{r}\times(-p\vec{n})ds = \sum_i \vec{r_i}\times(-p_i\vec{n})\triangle s_i = \sum_i[x_i(p_y)_i - y_i(p_x)_i], \tag{5.11}$$

where $\vec{r_i} = (x_i - x_c, y_i - y_c)$ is the vector from the geometrical center (x_c, y_c) of the plate to the center of the ith grid (x_i, y_i) on the plate surface. Similarly, the viscous torque can be calculated by

$$\tau_\nu = \oint_\Gamma \vec{r}\times\mu\frac{\partial U_t}{\partial n}\vec{t}ds = \sum_i \vec{r_i}\times\mu\frac{\partial U_t}{\partial n}\vec{t}\triangle s_i = \sum_i[x_i(\nu_y)_i - y_i(\nu_x)_i]. \tag{5.12}$$

Up to this point, we have obtained the pressure force, viscous force, the pressure torque, and viscous torque on the moving plate. The buoyancy corrected gravitational force $\vec{F_g}$ for the plate in negative y-direction becomes,

$$F_g = -\frac{\pi lh(\rho_b - \rho_f)g}{4}, \tag{5.13}$$

for the elliptic plate, and

$$F_g = -lh(\rho_b - \rho_f)g, \tag{5.14}$$

for the rectangular one. The total force $\vec{F}$ is equal to the summation of the pressure force $\vec{F_p}$, viscous force $\vec{F_\nu}$, and the buoyancy corrected gravity $\vec{F_g}$, namely,

$$\vec{F} = \vec{F_p} + \vec{F_\nu} + \vec{F_g}. \tag{5.15}$$

Further, the total force $\vec{F}$ can be decomposed into F_x on x-direction and F_y on y-direction,

$$\vec{F} = (F_x, F_y) = (P_x + \nu_x, P_y + \nu_y + F_g). \tag{5.16}$$

Similarly, the total torque can be decomposed into pressure and viscous ones $\tau = \tau_p + \tau_\nu$.

Let $\vec{a} = (a_x, a_y)$ and $\vec{\Omega}$ be the translational acceleration and the angular velocity of the plate respectively, which are determined by the forces through Newton's law,

$$F_x = ma_x, \quad F_y = ma_y, \quad \tau = Id\Omega/dt. \tag{5.17}$$

Due to the plate movement, the grid velocity $\vec{U_i}$ between two subsequent time steps at the center of the ith cell is given by

$$\vec{U}_i^{n+1} = \vec{U}_i^n + \triangle t^n \vec{a} + \triangle t^n \vec{r_i} \times \vec{\Omega}, \tag{5.18}$$

where $\vec{r_i}$ is the vector from the geometric center of the plate to the center of numerical cell. The above grid velocity is fed into the moving mesh method as the grid velocity (U_g, V_g).

5.2 Numerical study for unsteady falling plates movements

In the two-dimensional experiment [1], a small rectangular aluminum plate was designed to fall freely in a water tank. For the falling plate, many physical quantities were measured, such as the plate trajectories with falling velocities, and the fluid force and torque on the plate. Here, following the experiment, we apply the above moving mesh method in chapter 4 to solve the two dimensional Navier-Stokes equations to study the rich dynamical behavior of falling plates. In the following, we concentrate the plate falling simulation on the following conditions: $\beta = 1/8$, $h = 0.081$cm, $\rho_f = 1.0gcm^{-3}$,

$\rho_b = 2.7 gcm^{-3}$, and $\rho_b/\rho_f = 2.7$. In the current simulations, the Reynolds numbers for ellipse and rectangle are $Re = lu_t/\nu = 1100$ and 837 respectively.

In the early investigations [57, 1], due to the limitation of the numerical method, in order to avoid the singularities at the corners of a rectangular plate, an elliptic cross-section was used in the numerical computation. Since our numerical method has no constraint on the falling plate geometry, hence we will conduct the computations for both elliptic and rectangular cross-sections plates. The radius of the computational domain is about five times the longer axis of the plate, i.e., $r = 5L$ and L is the chord length.

Firstly, we present the simulation result for the elliptic plate. Figure 5.1 shows the computational mesh around an elliptic plate with an initial angle of 0.2 radian with respective to the horizontal axis. Figure 5.2 presents two representative trajectories of the falling plate, where the black one is the experimental measurement in [1] and the purple one is the trajectory from our computation. Overall, the two trajectories are very close to each other and have almost the same slope, but with obvious phase differences. In other words, the geometrical differences between the elliptic (numerics) and rectangular (experiment) plates will not effect the translational velocities of falling plates, but they will severely effect the rotation or angular velocity of the plates. Obviously, due to the sharp corner of the rectangular plate, the angular velocity of the rectangular one is much slower than that of elliptic one. The quantitative falling velocity in x- and y-directions, as well as angular velocity are listed in table 5.1. Except for the angular velocity ($\omega = 14.5 rad/s$) in experiment, the averaged translational velocities between the experiment ($V_x = 15.9 cm/s, V_y = -11.5 cm/s$) and the numerical computation ($V_x = 15.3 cm/s, V_y = -11.8 cm/s$) match reasonably. This observation is different from the numerical solutions ($V_x = 15.6 cm/s, V_y = -7.4 cm/s$) in [57, 1], where y-direction velocity as well as angular velocity have large discrepancies in comparison with the experiment measurement. Andersen, Persavento, and Wang attributed the discrepancy to the geometrical differences between the experimental rectangle and numerical ellipse. Based on our computation, it seems that the difference in the y-velocity is most likely due to the numerical algorithms instead of the geometry of the flying objects.

The vorticity field at four instants during a full rotation of the elliptical plate is presented in figure 5.3. The vorticity field is similar to the numerical one reported in [57]. In figure 5.4, the instantaneous fluid fores are also plotted, where the red one is the total fluid force on the plate, the green one is the pressure force, and the blue one is the viscous force. The constant buoyancy corrected gravitational force $m'g$, i.e., $m' = (\rho_b - \rho_f)\pi lh/4$, is not included there. Figure 5.4 clearly shows that the viscous force is almost always in the opposite direction of the elliptic plate movement except at the singular turning points. As shown later, this observation is violated by the rectangular plate.

The rectangle plate is released at an initial angle of 0.25π radian with respective to the horizontal axis, with an initial velocity $U = (-8.92cm/s, -8.92cm/s)$. The computational grid around the rectangle is shown in figure 5.5. For the rectangular plate, the plate motion is much more complicated than the ellipse, see figure 5.6 for the trajectory. For example, with the above initial condition, the rectangular plate first has a fluttering motion from side to side, then after a while the motion becomes tumbling. Since the experiment results presented in [1] is for the tumbling section, the numerical data in the corresponding section will be extracted and compared with the experiment. Figure 5.7 shows the rectangular plate trajectory of both numerical simulation and experimental measurement [1]. From this figure, the falling plate slope and the rotational velocity can be clearly identified. Quantitatively, the averaged x- and y-direction velocity components ($V_x = 15.1cm/s, V_y = -11.8cm/s$), and angular velocity ($\omega = 15.0rad/s$) have an excellent match with the experiment, especially the angular velocity is much improved in comparison with the elliptical one ($\omega = 16.9rad/s$). The vorticity field at four instants during a full rotation is presented in figure 5.8, which behaves more violent and unsteady than the corresponding one in the elliptic case. For example, the flow separation point around the rectangle is much close to the leading edge of the plate. Figure 5.9 shows the instantaneous fluid fores on the plate, where the red one is the total fluid force, the green one is the pressure force, and the blue one is viscous force.

Besides the magnitude deviation in the viscous force between elliptical and rectangular plates in figure 5.5 and figure 5.9, the astonishing differences between them are the behavior of the viscous force, especially during the process of the plate moving upward. For the elliptical plate, the viscous force is always in the opposite direction of the plate movement. However, for the rectangular one, due to the unsteadiness in the vorticity field, in certain flight period even with a small magnitude the viscous force can become a driving force for the moving plate due to the flow separation. In other words, during the up-rising period, the separated fluid around the plate moves faster than the plate itself. Also, from the numerical observation we can clearly realize that the total pressure force in the elliptic case is almost perpendicular to the long axis of the plate. However, for the rectangular one the pressure force can be tilted relative to the long symmetric axis of the plate due to the two small sections at the two end of the rectangle plate. In both elliptic and rectangular cases, the components of the fluid force and the torque as functions of the tumbling plate orientation (phase) and evolution time respectively are presented in figures 5.10-5.12, where the horizontal axis of the left plots are the angles of the tumbling plates, and of the right ones are the real time. Since the experimental measurements only presented the forces relative to the rotational angles, the phase error between ellipse and rectangle can be only observed through the right plots. From these figures, the good agreements between the experiment data [1] and our numerical solutions are obtained. Also, from the right plots it is obvious that the angular speed of elliptic plate is much faster than that of both experimental and rectangular one.

In order to avoid the boundary effect on the numerical solution, a large computational domain with radius equal to $10L$ is also used to repeat the above computations, where

L is the chord length. Then the numerical solutions are compared with the previous one obtained in the computational domain with radius equal to $5L$. By comparing figure 5.13 with figure 5.14, we find that the boundary has a little effect on the phase space. Figures 5.15-5.16 show the body forces and torques from both calculations. Basically, both computational results are close to each other, and the forces in large computational domain have been slightly improved in comparison with the experiment. Further for the rectangular plate we notice that the vorticity field in figure 5.18 becomes more steady than the one in figure 5.8. For the elliptic and rectangular trajectories in figures 5.19 and 5.20 respectively, almost the same conclusions as the previous numerical results in the small computational domain are obtained. Conclusively, in the computational domain with radius equal to or bigger than $5L$, the out boundary has only a little effect on numerical results.

With excellent agreements between our computation and experimental measurements, the discrepancies between them may be due to the following reasons. Firstly, even though the experiment is designed for the quasi two-dimensional flow [1], the three-dimensional effects can be brought into the measurement as well. Secondly, there are some uncertainties about the defined Reynolds number and the viscosity effect of the experimental fluid. Additionally, there are certain artificial dissipation in the numerical scheme.

Table 5.1: Experimental and numerical averaged translational and angular velocities on the falling plates

	U [cm/s]	V [cm/s]	ω [rad/s]
Experiment (Andersen et al. [1])	15.9±0.3	-11.5±0.5	14.5±0.3
Computations: Ellipse (Andersen et al. [1])	15.6±0.2	-7.4±0.3	18.0±0.3
Ellipse (current)	15.3±0.05	-11.2±0.16	16.9±0.01
Rectangle (current)	15.1±0.30	-11.8±0.37	15.0±0.36

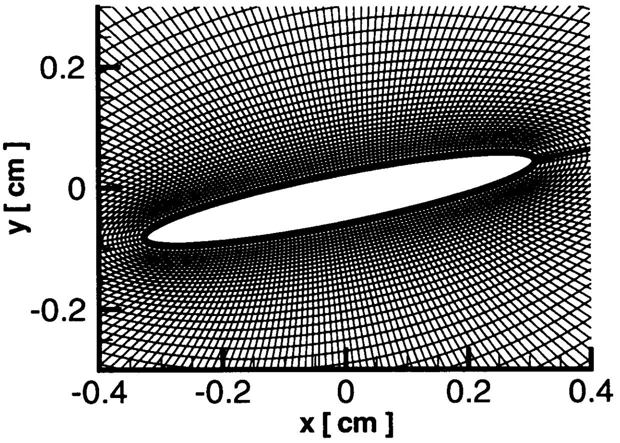


Figure 5.1: Computational grid around the ellipse.

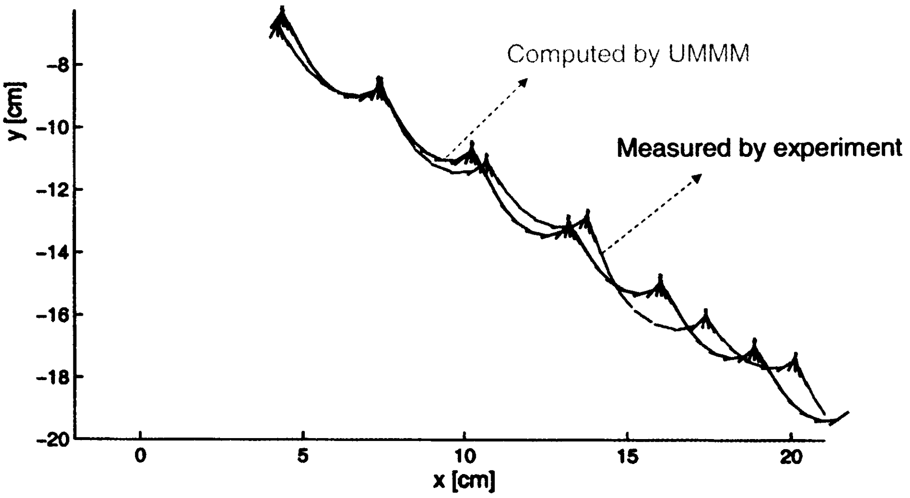


Figure 5.2: Trajectories of falling ellipse: experiment (black) and computation (purple).

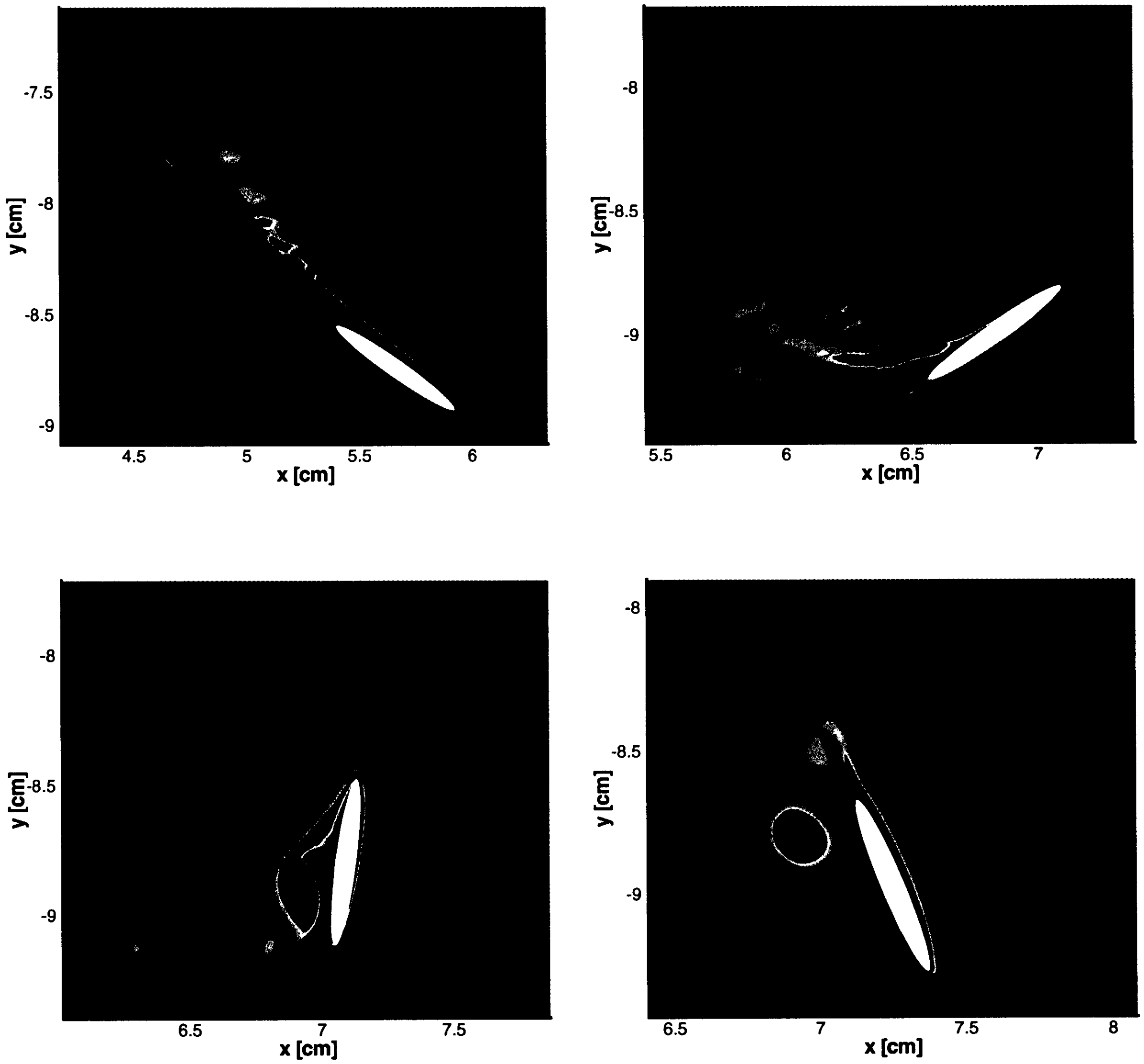


Figure 5.3: Vorticity field around a falling ellipse at four instants during a full rotation.

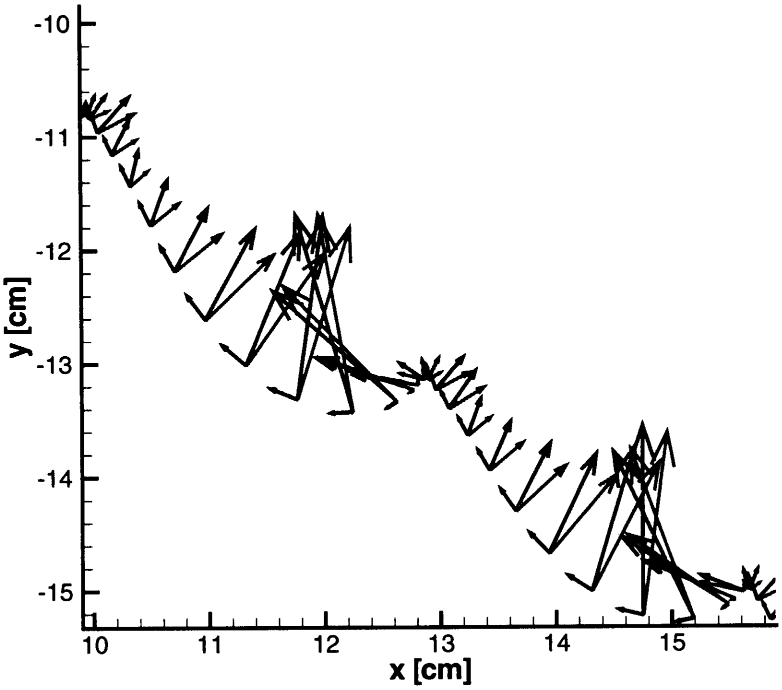


Figure 5.4: Fluid forces on the ellipse in the tumbling process: total fluid force (red), pressure force (green), and viscous force (blue).

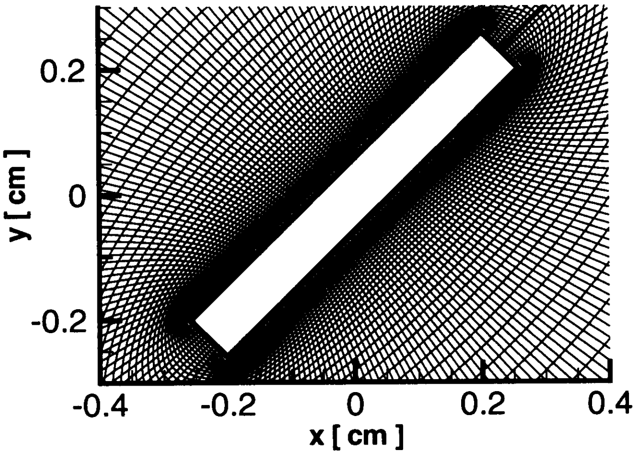


Figure 5.5: Computational grid around the rectangle.

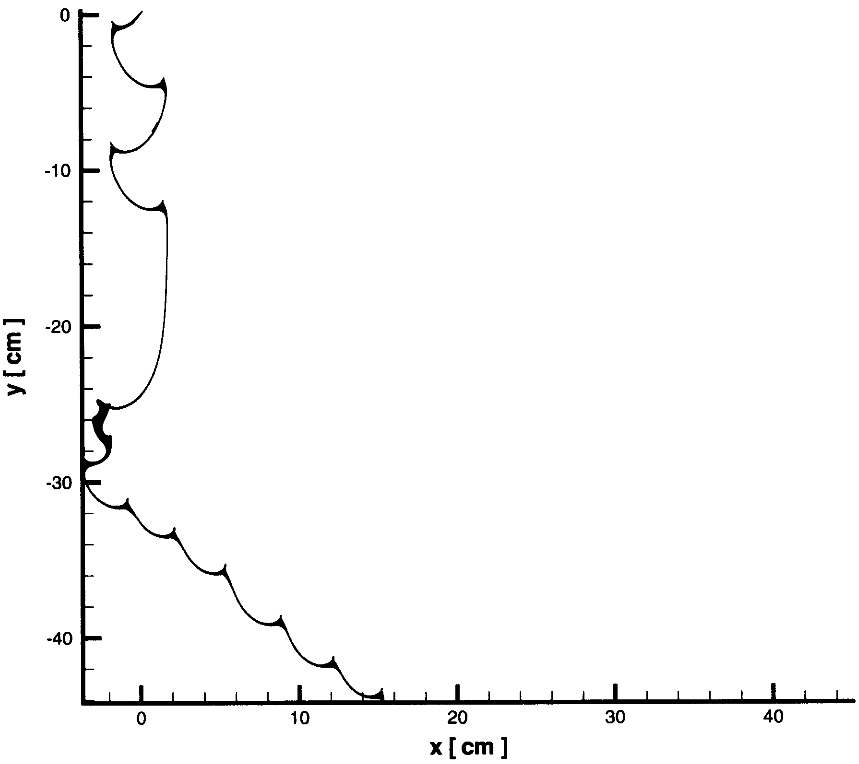


Figure 5.6: Falling trajectory of a rectangle from fluttering to tumbling motion.

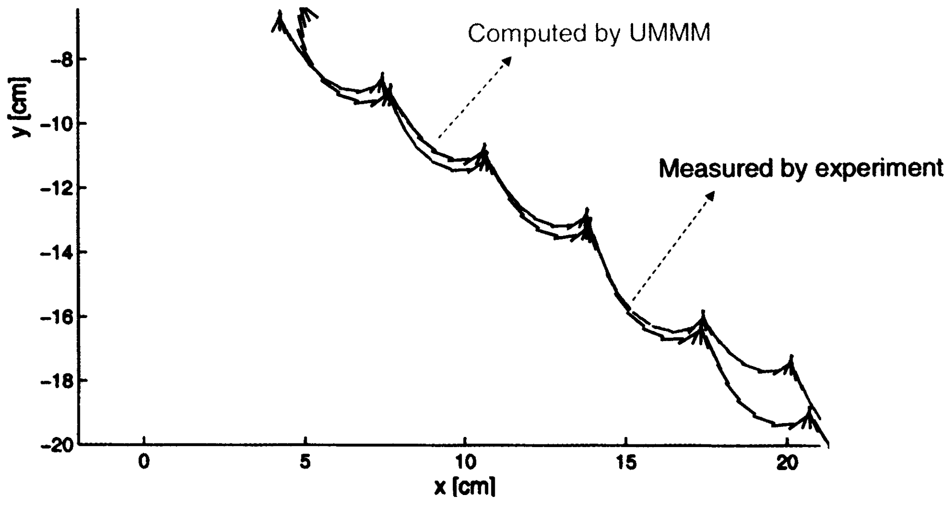


Figure 5.7: Trajectories of the falling rectangle in the tumbling section: experiment (black) and computation (red).

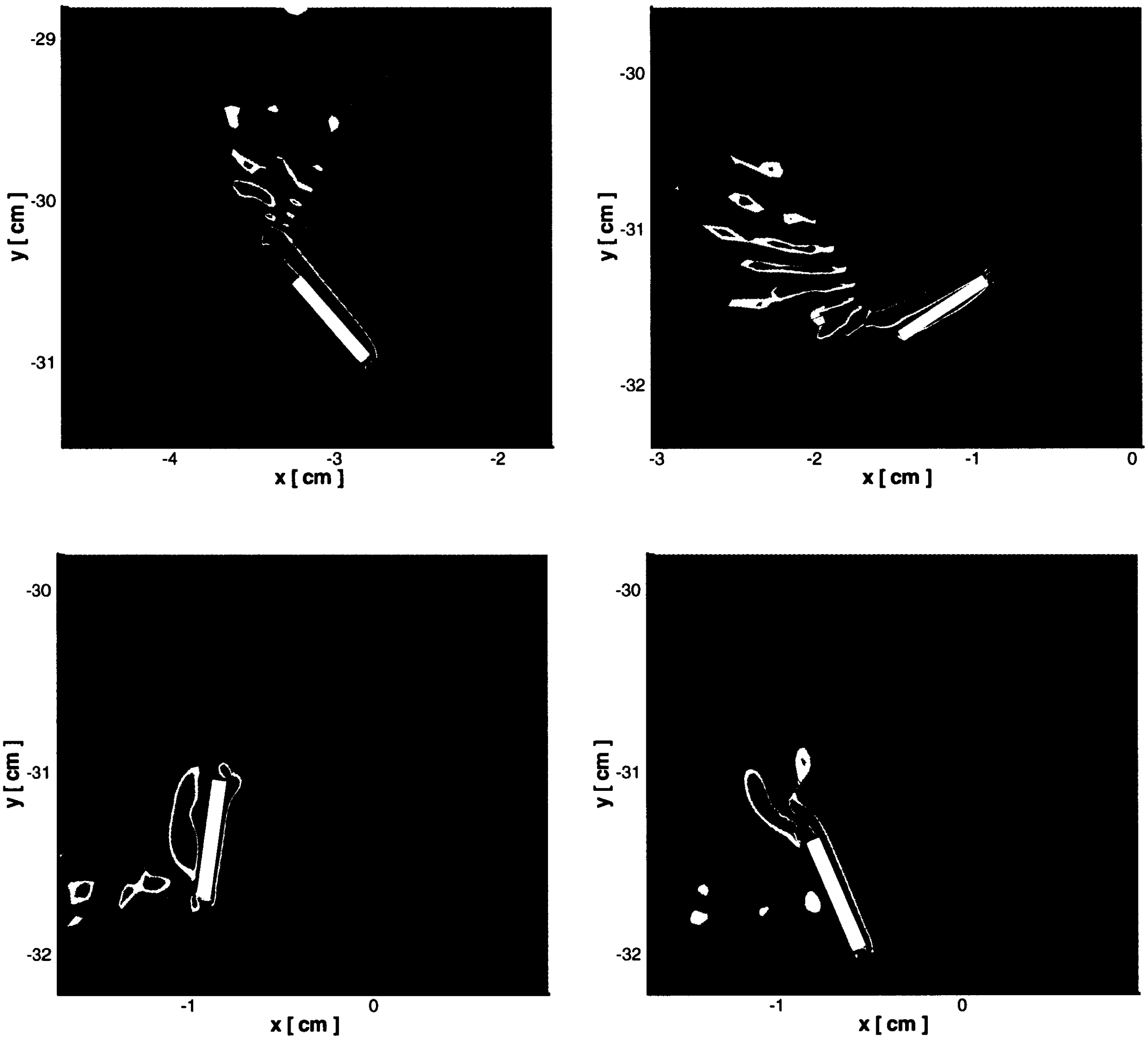


Figure 5.8: Vorticity field of a falling rectangle at four instants during a full rotation.

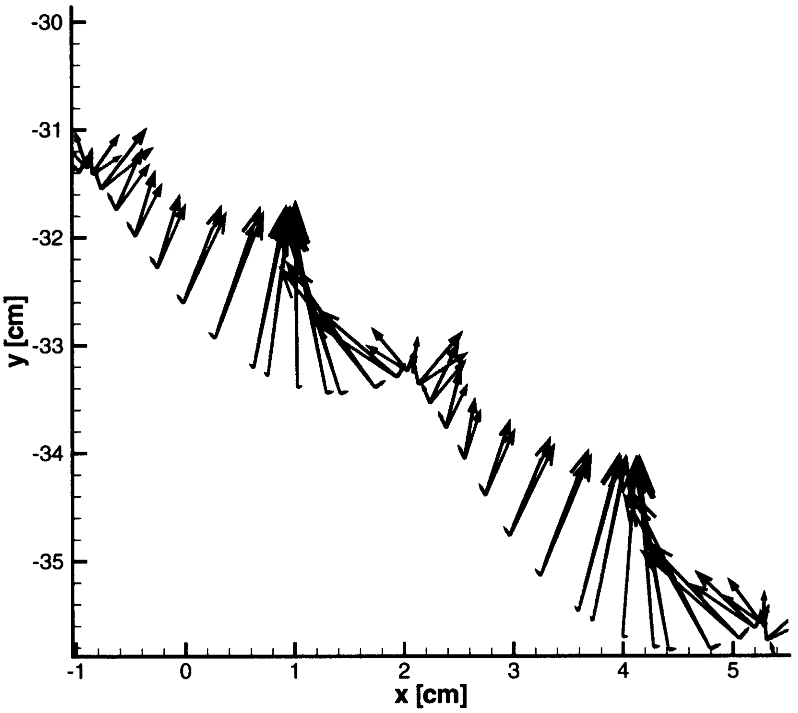


Figure 5.9: Fluid forces on the rectangle: total fluid force (red), pressure force (green), and viscous force (blue).

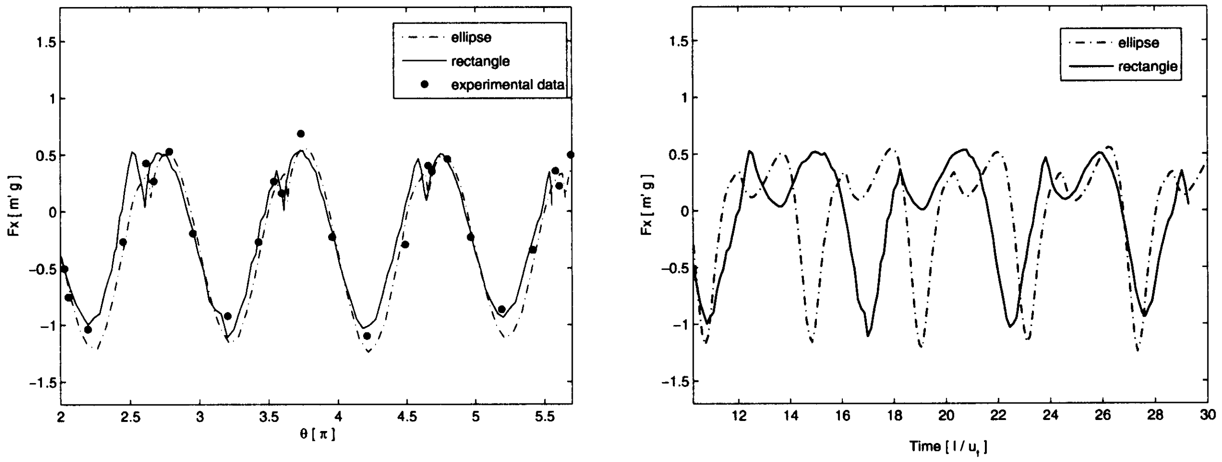


Figure 5.10: x-direction fluid force F_x on the plate during the tumbling process for both ellipse and rectangle. The horizontal axis are rotational angle (left) and dimensionless time (right).

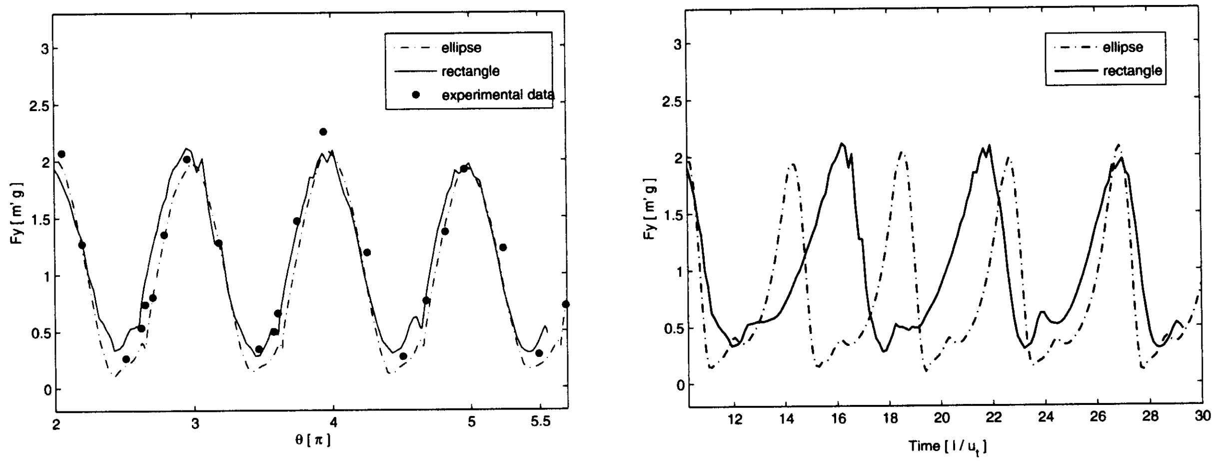


Figure 5.11: y-direction fluid force F_y on the plate during the tumbling process for both ellipse and rectangle. The horizontal axis are rotational angle (left) and dimensionless time (right).

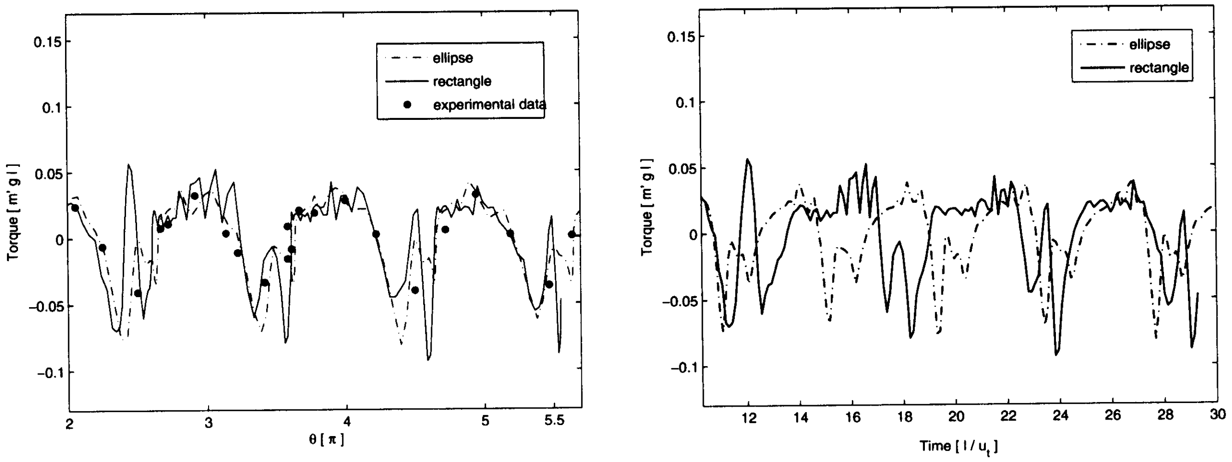


Figure 5.12: Torques on the plate during the tumbling process for both ellipse and rectangle. The horizontal axis are rotational angle (left) and dimensionless time (right).

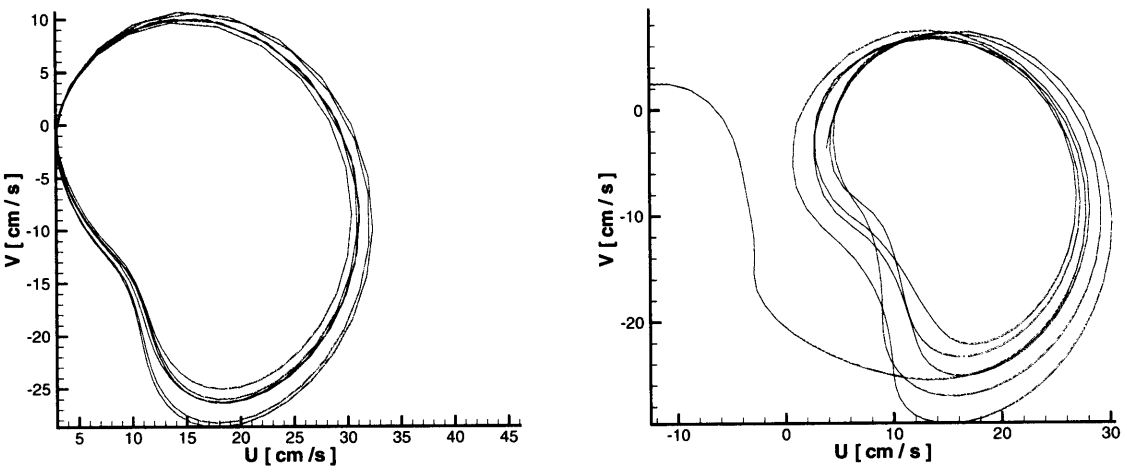


Figure 5.13: U-component versus V-component of the falling plate. The left one is for the elliptic cross-section and the right is for the rectangular one.

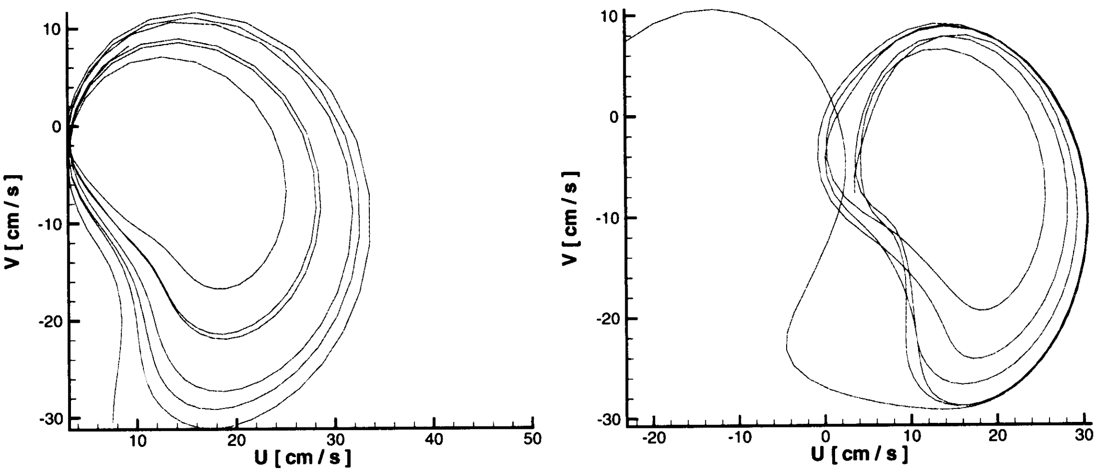


Figure 5.14: U-component versus V-component of the falling plate in the computational domain with radius equal to $10L$, where L is the chord length. The left one is for the elliptic cross-section and the right is for the rectangular one.

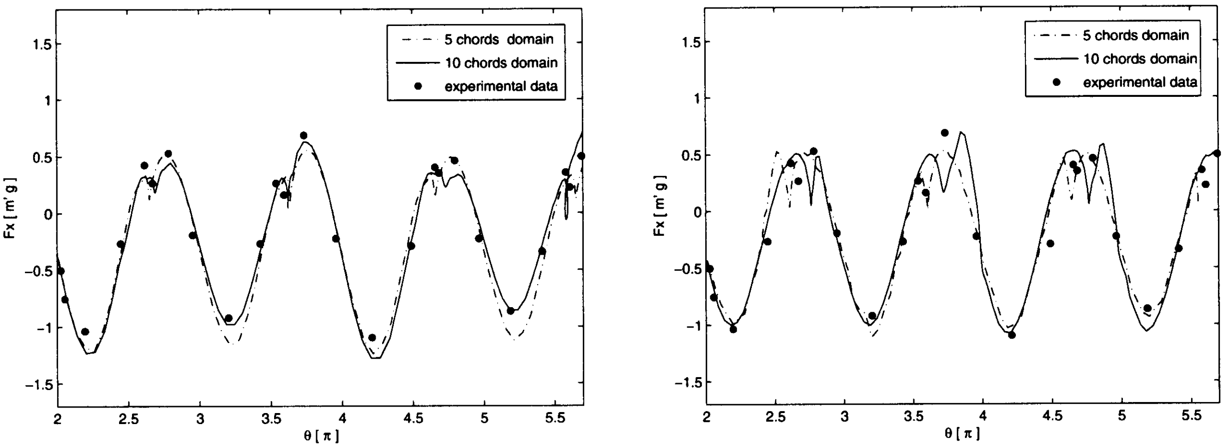


Figure 5.15: Comparison of the F_x force from two computations in different computational domain, i.e., radius $= 5L$ and radius$= 10L$, where L is the chord length, for falling ellipse (left) and rectangle (right).

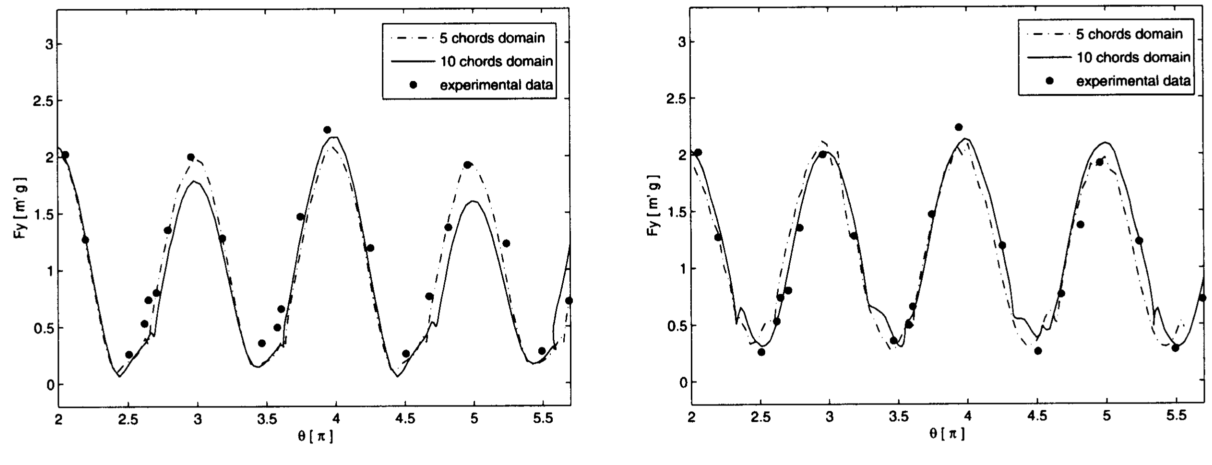


Figure 5.16: Comparison of the F_y force from two computations in different computational domain, i.e., radius$= 5L$ and radius$= 10L$, where L is the chord length, for falling ellipse (left) and rectangle (right).

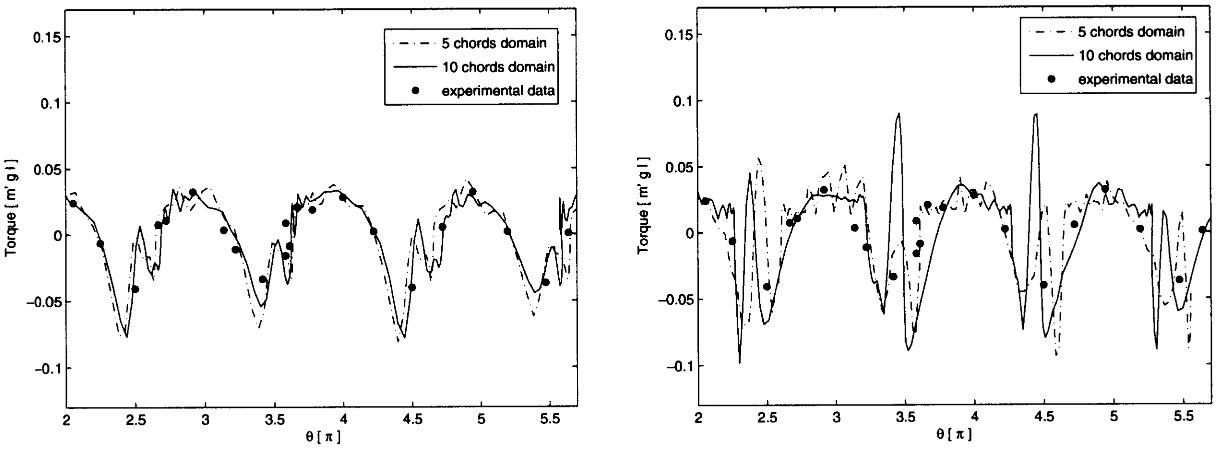


Figure 5.17: Comparison of torques from two computations in different computational domain, i.e., radius= $5L$ and radius= $10L$, where L is the chord length, for falling ellipse (left) and rectangle (right).

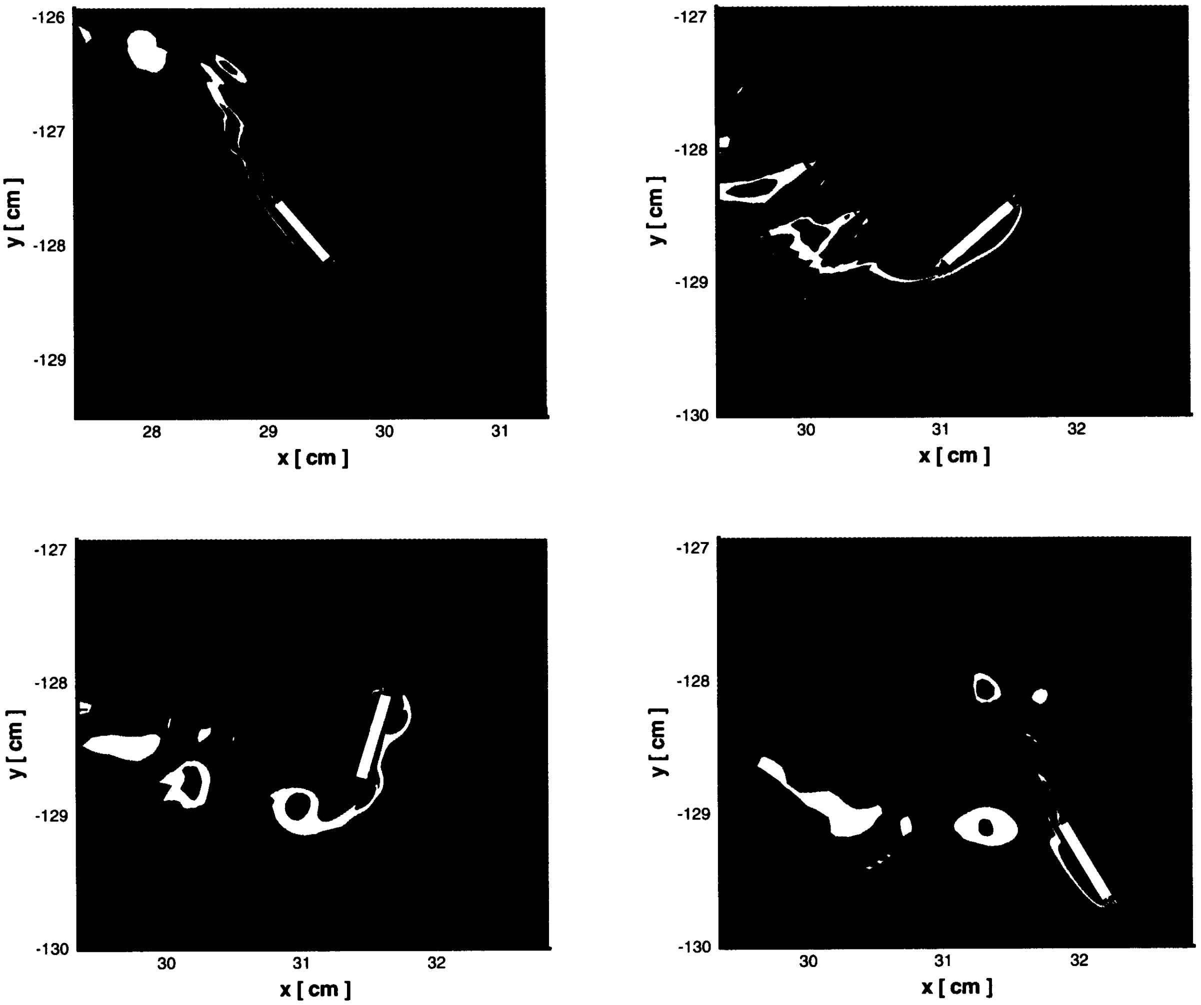


Figure 5.18: Vorticity field of a falling rectangular plate at four instants during a full rotation with the computational domain with radius equal to $10L$, where L is the chord length.

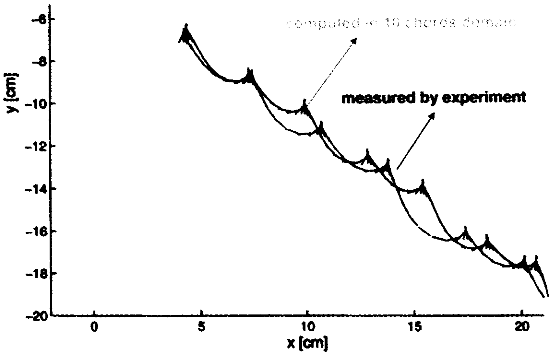


Figure 5.19: In the computational domain with radius equal to $10L$, where L is the chord length, falling elliptical plate trajectories based on the experiment (black) and the computation (green).

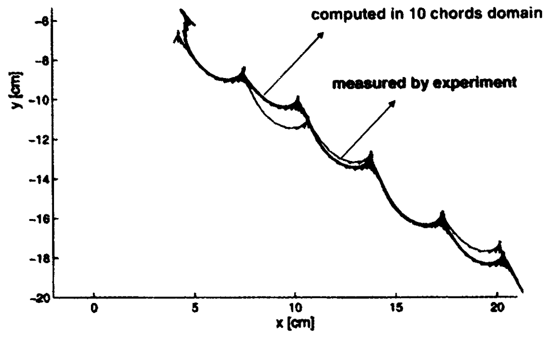


Figure 5.20: In the computational domain radius equal to $10L$, where L is the chord length, falling rectangle trajectories: the experimental one (black) and the computational one (red).

Chapter 6

Conclusion

In the first part of this thesis, an adaptive grid method based on the gas-kinetic BGK scheme has been successfully developed for the simulation of inviscid and viscous flows. For unsteady flow computation, the adaptive method has obvious advantage in terms of the accuracy and efficiency in comparison with the stationary mesh method. With the velocity-gradient dependent monitor function, the boundary layer and shock wave have been well captured for the viscous flow. It can be predicted that in the future the adaptive mesh method will become an important working horse for the unsteady viscous flow computation, especially in the cases where fine grid resolution is needed, such as DNS and turbulent flows. The good agreement between the numerical results and benchmark solutions validates the current adaptive grid approach.

In the second part, under the unified coordinate transformation the gas-kinetic BGK equation is transformed into a moving reference of frame and a unified moving grid gas-kinetic method in Eulerian space for viscous flow computation has been developed. The great advantage of the current scheme with a variable mesh velocity is that the gas-kinetic equation and geometrical conservation laws are combined as a unified system. The physical and geometrical variables can be updated simultaneously. The current method extends the gas-kinetic BGK-NS scheme to treat free surface and moving boundary problems. More specifically, this is the first time that the gas-kinetic BGK scheme for viscous solutions is constructed in a moving frame of reference in this thesis. Under a generalized coordinate transformation with arbitrary grid velocity (U_g, V_g), the fluxes across a moving cell interface for flow variables in the inertia reference of frame are explicitly calculated. The Eulerian and Lagrangian methods become two limiting cases from the current approach. Since the dissipative terms in the original Navier-Stokes equations are replaced by the relaxation term in the kinetic model, the merit of the current approach is that a viscous solution is obtained directly through the gas-kinetic scheme. Otherwise, the viscous macroscopic equations under a unified coordinate transformation become extremely complicated. In the current approach, a conservative adaptive grid is also implemented to increase the robustness and accuracy of the unified moving mesh method, which not only remedies the distorted mesh, but also concentrates the mesh in the high gradient regions. The mesh velocity can be properly chosen to capture flow solution. The current unified gas-kinetic method has been applied to many flow problems, such as the free surface flow, Mach reflection inside a channel, rotating cavity flow, and falling plates etc., where both inviscid and viscous solutions have been accurately obtained. Furthermore, for the first time, the Lagrangian gas-kinetic scheme has been constructed and used in the viscous boundary layer calculation.

The current unified moving mesh scheme is applied to the study of unsteady viscous aerodynamics of freely falling plates. In terms of the plate trajectories, fluid force, and translational as well as angular velocities, the agreements between our computation and experimental measurements are excellent. These solutions improve the early computations by Andersen, Persavento, and Wang [1], where a different numerical method was used. For example, the relation between the plate falling velocity and the plate geometry has been clarified. Basically, the whole flight process of the falling plate and its rich unsteady aerodynamics can be well captured by our computation. The moving mesh method presented in this thesis is an accurate and robust Navier-Stokes solver with moving meshes. It provides a powerful tool for the study of viscous flow with moving boundaries. The extension of the current method to three dimensional flow computation is straight forward. Clearly, the current method can be used in the study of unsteady aerodynamics, such as the insects flight.

Appendix A

The Navier-Stokes Equations under unified coordinate transformation

The flow equations in 2D Eulerian space can be expressed as,

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=\frac{\partial E_v}{\partial x}+\frac{\partial F_v}{\partial y}, \tag{A.1}$$

where

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho e\end{pmatrix},\quad E=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (\rho e+p)u\end{pmatrix},\quad E_v=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ u\tau_{xx}+v\tau_{xy}-q_x\end{pmatrix}, \tag{A.2}$$

$$F=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (\rho e+p)v\end{pmatrix},\quad F_v=\begin{pmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ u\tau_{yx}+v\tau_{yy}-q_y\end{pmatrix}. \tag{A.3}$$

For a Newtonian fluid,

$$\tau_{xx}=2\mu\frac{\partial u}{\partial x}+\kappa\nabla\cdot\vec{V},\quad \tau_{yy}=2\mu\frac{\partial v}{\partial y}+\kappa\nabla\cdot\vec{V},\quad \tau_{xy}=\tau_{yx}=\mu(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}), \tag{A.4}$$

and

$$q_x = -\frac{\mu c_p}{P_r}\frac{\partial T}{\partial x}, \qquad q_y = -\frac{\mu c_p}{P_r}\frac{\partial T}{\partial y}. \tag{A.5}$$

With the following transformation between the physical (t, x, y) and computational space (λ, ξ, η),

$$\begin{cases} dt = d\lambda, & , \\ dx = x_\lambda d\lambda + x_\xi d\xi + x_\eta d\eta, & , \\ dy = y_\lambda d\lambda + y_\xi d\xi + y_\eta d\eta, & , \end{cases} \tag{A.6}$$

we can get the inverse transformation,

$$\begin{pmatrix} d\lambda \\ d\xi \\ d\eta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \xi_t & \xi_x & \xi_y \\ \eta_t & \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} dt \\ dx \\ dy \end{pmatrix}. \tag{A.7}$$

The N-S Equations (A.1) in the computational space (λ, ξ, η) become

$$\frac{\partial \bar{Q}}{\partial \lambda} + \frac{\partial \bar{E}}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta} = \frac{\partial \bar{E}_v}{\partial \xi} + \frac{\partial \bar{F}_v}{\partial \eta}, \tag{A.8}$$

where

$$\begin{aligned} \bar{Q} &= \triangle Q, \\ \bar{E} &= \triangle(\xi_t Q + \xi_x E + \xi_y F), \\ \bar{F} &= \triangle(\eta_t Q + \eta_x E + \eta_y F), \\ \bar{E}_v &= \triangle(\xi_x E_v + \xi_y F_v), \\ \bar{F}_v &= \triangle(\eta_x E_v + \eta_y F_v), \end{aligned} \tag{A.9}$$

and $\triangle = \partial(x, y)/\partial(\xi, \eta) = x_\xi y_\eta - x_\eta y_\xi$.

For a system with mesh moving velocity (U_g, V_g), we have

$$\begin{pmatrix} 1 & 0 & 0 \\ x_\lambda & x_\xi & x_\eta \\ y_\lambda & y_\xi & y_\eta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ U_g & A & L \\ V_g & B & M \end{pmatrix}. \tag{A.10}$$

Therefore, we get

$$\begin{pmatrix} 1 & 0 & 0 \\ \xi_t & \xi_x & \xi_y \\ \eta_t & \eta_x & \eta_y \end{pmatrix} = \frac{1}{\triangle} \begin{pmatrix} \triangle & 0 & 0 \\ -I_g & M & -L \\ -J_g & -B & A \end{pmatrix}, \tag{A.11}$$

where $\triangle = AM - BL$, $I_g = MU_g - LV_g$, and $J_g = AV_g - BU_g$. Hence, the variables in Eq.(A.8) are

$$\begin{aligned} \bar{Q} &= \triangle Q, \\ \bar{E} &= (-I_g Q + ME - LF), \\ \bar{F} &= (-J_g Q - BE + AF), \\ \bar{E_v} &= (ME_v - LF_v), \\ \bar{F_v} &= (-BE_v + AF_v). \end{aligned} \tag{A.12}$$

The viscous shear stresses given by Eqs.(A.4) with Stokes' hypothesis $\kappa = -\frac{2}{3}\mu$ become the following in the transformed computational space,

$$\begin{aligned} \tau_{xx} &= \tfrac{\mu}{\triangle}[\tfrac{4}{3}(Mu_\xi - Bu_\eta) - \tfrac{2}{3}(-Lv_\xi + Av_\eta)], \\ \tau_{yy} &= \tfrac{\mu}{\triangle}[\tfrac{4}{3}(-Lv_\xi + Av_\eta) - \tfrac{2}{3}(Mu_\xi - Bu_\eta)], \\ \tau_{xy} &= \tfrac{\mu}{\triangle}(-Lu_\xi + Au_\eta + Mv_\xi - Bv_\eta), \end{aligned} \tag{A.13}$$

and the heat conduction terms in Eq.(A.5) are

$$\begin{aligned} q_x &= -\tfrac{\mu c_p}{p_r \triangle}(MT_\xi - BT_\eta), \\ q_y &= -\tfrac{\mu c_p}{p_r \triangle}(-LT_\xi + AT_\eta). \end{aligned} \tag{A.14}$$

In the computational space, the viscous flux vector $\bar{E}_v$ and $\bar{F}_v$ are

$$\bar{E}_v = \frac{\mu}{\triangle} \left(\begin{array}{l} 0 \\ (\frac{4}{3}M^2 + L^2)u_\xi - \frac{1}{3}MLv_\xi + (-\frac{4}{3}MB - AL)u_\eta + (-\frac{2}{3}AM + BL)v_\eta \\ \\ -\frac{1}{3}MLu_\xi + (M^2 + \frac{4}{3}L^2)v_\xi + (MA - \frac{2}{3}BL)u_\eta + (-MB - \frac{4}{3}AL)v_\eta \\ \\ \frac{1}{2}(\frac{4}{3}M^2 + L^2)(u^2)_\xi + \frac{1}{2}(M^2 + \frac{4}{3}L^2)(v^2)_\xi - \frac{1}{3}ML(uv)_\xi + \\ \frac{c_p(M^2+L^2)}{P_r}T_\xi + \frac{1}{2}(-\frac{4}{3}MB - AL)(u^2)_\eta + \frac{1}{2}(-MB - \frac{4}{3}AL)(v^2)_\eta + \\ (MA - \frac{2}{3}BL)vu_\eta + (BL - \frac{2}{3}MA)uv_\eta + \frac{c_p(-MB-AL)}{P_r}T_\eta \end{array} \right), \tag{A.15}$$

and

$$\bar{F}_v = \frac{\mu}{\triangle} \left(\begin{array}{l} 0 \\ (-\frac{4}{3}MB - AL)u_\xi + (MA - \frac{2}{3}BL)v_\xi + (\frac{4}{3}B^2 + A^2)u_\eta - \frac{1}{3}BAv_\eta \\ \\ (BL - \frac{2}{3}MA)u_\xi + (-MB - \frac{4}{3}AL)v_\xi - \frac{1}{3}ABu_\eta + (B^2 + \frac{4}{3}A^2)v_\eta \\ \\ \frac{1}{2}(-\frac{4}{3}MB - AL)(u^2)_\xi + \frac{1}{2}(-MB - \frac{4}{3}AL)(v^2)_\xi + (AM - \frac{2}{3}BL)uv_\xi + \\ (BL - \frac{2}{3}AM)vu_\xi + \frac{c_p(-MB-AL)}{P_r}T_\xi + \frac{1}{2}(\frac{4}{3}B^2 + A^2)(u^2)_\eta + \\ \frac{1}{2}(B^2 + \frac{4}{3}A^2)(v^2)_\eta - \frac{1}{3}AB(uv)_\eta + + \frac{c_p(B^2+A^2)}{P_r}T_\eta \end{array} \right). \tag{A.16}$$

Bibliography

[1] A. Andersen, U. Persavento and Z. Jane Wang, *Unsteady aerodynamics of fluttering and tumbling plates*, Accepted for publication in J. Fluid Mech. (2005).

[2] B.N. Azarenok and T. Tang, *Second-order Gounov-type scheme for reactive flow calculations on moving meshes*, J. Comput. Phys. **206** (2005), pp. 48-80.

[3] M.J. Baines, *Moving finite elements*, Oxford University Press, New York, (1994).

[4] J.T. Batina, *Unsteady Euler airfoil solutions using unstructured dynamic meshes*, AIAA J. **28** (1990), pp. 1381-1388.

[5] A. Belmonte, H. Eisenberg, and E. Moreses, *From flutter to tumble: inertial drag and froude similarity in falling paper*, Phys. Rev. Lett. **81** (1998), pp. 345-348.

[6] P.L Bhatnagar, E.P. Gross, and M. Krook, *A model for collision processes in gases I: small amplitude process in charged and neutral one-component systems*, Phys. Rev., **94**, (1954), pp. 511-525.

[7] O. Botella and R. Peyret, *Benchmark spectral results on the lid-driven cavity flow*, Computers Fluids **27** (1998), pp. 421-433.

[8] J.U. Brackbill, *An adaptive grid with directional control*, J. Comput. Phys. **108** (1993), pp. 38-50.

[9] J.U. Brackbill and J.S. Saltzman, *Adaptive zoning for singular problems in two dimensions*, J. Comput. Phys. **46** (1982), pp. 342-368.

[10] W. Cao, W. Huang, and R.D. Russel, *A moving mesh method based on the geometric conservation law*, Siam J. Sci. Comput. **24** (2002), pp. 118-142.

[11] W. Cao, W.Z. Huang, and R.D. Russell, *An r-adaptive finite element method based upon moving mesh PDEs*, J. Comput. Phys. **149**, pp. 221-244 (1999).

[12] W. Cao, W. Huang, and R.D. Russel, *Approaches for generating moving adaptive meshes: Local versus velocity*, Appl. Numer. Math., **47** (2003), pp. 121-138.

[13] H.D. Ceniceros and T.Y. Hou, *An efficient dynamically adaptive mesh for potentially singular solutions*, J. Comput. Phys., **172** (2001), pp. 609-639.

[14] S. Chapman and T.G. Cowling, *The Mathematical Theory of Non-uniform Gases*, Cambridge University Press (1990).

[15] X.Y. Chen and G.C. Zha, *Fully coupled fluid-structural interactions using an efficient high resolution upwind scheme*, J. Fluids and Structures **20** (2005), pp. 1105-1125.

[16] C.K. Chu, Phys. Fluids, (1965), **8**,12.

[17] S.F. Davis and J.E Flaherty, *An adaptive finite element method for initial-boundary value problems for partial differential equations*, SIAM J. sci. Stat. Comput. **3** (1982), pp. 6-27.

[18] H. Deshpande, *A second order accurate, kinetic-theory based, method for inviscid compressible flows*, NASA Langley Tech. paper No. **2613** (1986).

[19] Y. Di, R. Li, T. Tang and P. Zhang, *Moving mesh finite element methods for incompressible Navier-Stokes equations*, J. Comput. Phys. **190** (2003), pp. 100-117.

[20] P. DUPLEICH, *Rotation in free fall of rectangular wings of elongated shape*, NACA Technical Memo **1201** (1941), pp. 1-99.

[21] A.S. DVINSKY, *Adaptive grid generation from harmonic maps on Riemannian manifolds*, J. Comput. Phys. **95** (1991), pp. 221-244.

[22] T.G. ELIZAROVA AND B.N. CHETVERUSHKIN, *Kinetic algorithms for calculating gas dynamic flows*, J. Comput. Phys. **25** (1985), pp. 1526-1533.

[23] S.B. FIELD, M. KLAUS, M.G. MOORE AND F. NORI, *Chaotic dynamics of falling disks*, Nature **388** (1997), pp. 252-254.

[24] U. GHIA, K.N. GHIA AND C.T. SHIN, *High-Resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys. **48** (1982) pp. 387-411.

[25] A. HARTEN, *The artificial compression method for computation of shocks and contact discontinuities: III. Self-adjusting bybrid schemes*, Math. Comp. **32** (1978), pp. 363.

[26] A. HARTEN, *ENO schemes with subcell resolution*, J. Comput. Phys. **83** (1989), pp. 148.

[27] 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), pp. 227.

[28] K.A. HOFFMANN AND S.T. CHIANG, *Computational fluid dynamics*, third edition, **2**, pp. 21-46, published by Engineering Education System, Wichita, Kansas.

[29] W. HUANG, *Mathematical principles of anisotropic mesh adaptation*, Commnu. Comput. Phys. **1** (2006), pp. 276-310.

[30] W. HUANG AND W. SUN, *Variational mesh adaptation II: Error estimates and monitor function*, J. Comput. Phys. **184** (2003), pp. 619-648.

[31] W.H. Hui, P.Y. Li and Z.W. Li, *A unified coordinated system for solving the two-dimensional Euler equations*, J. Comput. Phys. **153** (1999), pp. 596-637.

[32] W.H. Hui and G.P.Zhao, *Capturing contact discontinuities using the unified coordinates*, Proceedings of Seconding MIT Conference on Computational Fluid and Solid Mechanics (2003), pp. 2000-2003.

[33] C. Jin and K. Xu, *An adaptive grid method of two dimensional viscous flows*, J. Comput. Phys. 218 (2006), pp. 68-81.

[34] C. Jin and K. Xu, *A unified moving grid gas-kinetic method in Eulerian space for viscous flow computation*, accepted by J. Comput. Phys. (2006).

[35] M.A. Jones and M.J. Shelley, *Falling cards*, J. Fluid Mech. **540**(2005), pp. 393-425.

[36] P. Knupp, L.G Margolin, and M. Shashkov, *The error-minimization-based strategy for moving mesh methods*, Commun. Comput. Phys. **1** (2006), pp. 53-81.

[37] P. Knupp, L.G Margolin, and M. Shashkov, *Reference Jacobian Optimization—based rezone strategies for arbitrary Lagrangian Eulerian methods*, J. Comput. Phys. **176** (2002), pp. 93.

[38] M.N. Kogan, *Rarefied gas dynamics*, Plenum Press, New York, 1969.

[39] M. Kucharik, M. Shashkov, and B. Wendroff, *An efficient linearity-and-bound-preserving remapping method*, LA-UR-03-1766.

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

[41] D. Levermore, *Moments closure hierarchies for kinetic theory*, submitted to J. of statistical Physics 1995.

[42] Q.B. Li, *On the multidimensional gas-kinetic BGK scheme*, submitted to Elsevier Science.

[43] R. LI, T. TANG AND P.W. ZHANG, *Moving mesh methods in multiple dimensions based on harmonic maps*, J. Comput. Phys., **170** (2001), pp. 562-588.

[44] S. LI AND L. PETZOLD, *Moving mesh methods with upwinding schemes for time-dependent PDEs*, J. Comput. Phys. **131** (1997), pp. 368-377.

[45] Y.S. LIAN AND K. XU, *A gas-kinetic scheme for multimaterial flows and its application in chemical reactions*, J. Comput. Phys. **163** (2000), pp. 349-375.

[46] M.S. LIOU, *An extended Lagrangian mehod* , J. Comput. Phys. **118** (1995), pp. 294-309.

[47] K. LIPNIKOV AND M. SHASHKOV, *The error minimization based strategy for moving mesh methods*, Commun. Comput. Phys. **1** (2006), pp. 53-80.

[48] V. LISEIKIN, *Grid Generation mehods*, Springer, Berlin, Heidelberg, New York, (1999).

[49] L. MAHADEVAN, W.S. RYU AND A.D.T. SAMUEL, *Tumbling cards*, Phys. Fluids **11** (1981), pp. 1-3.

[50] L.G. MARGOLIN, *Introduction to An arbitrary Lagrangian—Eulerian computing method for all flow speeds*, J. Comput. Phys. **135** (1997), pp. 198.

[51] J.C. MAXWELL, *The scientific papers of james Clerk Maxwell*, Dover, New York, 1890, pp. 115-118.

[52] G. MAY AND A. JAMESON, *Improved gas-kinetic BGK finite-volumne multigrid method for three-dimensional computation of viscous flow*, AIAA-2005-5106, 17^th AIAA Computational Fluid Dynamics conference, Toronto, Ontario, June 6-9, 2005.

[53] K. MILLER AND R.N. MILLER, *Moving finite element*, I, SIAM J. Numer Anal. **18** (1981), pp. 1019-1032.

[54] R. MITTAL, V.SESHADRI AND H.S. UDAYKUMAR, *Flutter, tumble and vortex induced autorotation*, Theoret. Comput. Fluid Dynamics **17** (2004), pp. 165-170.

[55] T. OHWADA AND K. XU, *The kinetic scheme for the full Burnett equations*, J. Comput. Phys. **201** (2004), pp. 315-332.

[56] B. PEROT AND R. NALLAPATI, *A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows*, J. Comput. Phys. **190** (2003), pp. 100-117.

[57] U. PESAVENTO AND Z. JANE WANG, *Falling paper: Navier-Stokes solutions, model of fluid forces, and center of mass elevation*, Physical Review Letters **93** (2004), Num. 14.

[58] W. PRACHT, *Calculating three−dimensional fluid flows at all speeds with an Eulerian−Lagrangian computing mesh*, J. Comput. Phys. **17** (1975), pp. 132.

[59] D.I. PULLIN, *Direct simulation methods for compressible inviscid ideal gas flow*, J. Comput. Phys. **34** (1980), pp. 231-244.

[60] V. RAMESH, S.M. DESHPANDE, *Least squares kinetic upwind method on moving grids for unsteady Euler computations*, Computers & Fluids. **30** (2001), pp. 621-641.

[61] R.D. REITZ, *One-dimensional compressible gas dynamics calculations using the Boltzmann equations*, J. Comput. Phys. **42** (1981), pp. 108-123.

[62] J.-F. REMACLE, J.E. FLAHERTY, AND M.S. SHEPHARD, *An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems*, SIAM Rev. **45** (2003), pp. 53-72.

[63] W.Q. REN AND X.P. WANG, *An iterative grid redistribution method for singular problems in multiple dimensions*, J. Comput. Phys. **159** (2000), pp. 246-273.

[64] B.A. ROBINSON, J.T. BATINA, AND H.T.Y. YANG, *Aeroelastic analysis of wings using the Euler equations with a deforming mesh*, J. Aircraft **28** (1991), pp. 781-788.

[65] R.H. SANDERS AND K.H. PRENDERGAST, *The possible relation of the three-kiloparsec arm to explosions in the galactic nucleus*, Astrophysical Journal, **188** (1974), pp. 32.

[66] C.W. SHU AND S. OSHER, *Efficient implementation of essentially non– oscillatory shock – capturing schemes II*, J. Comput. Phys. **83** (1989), pp. 32.

[67] W. SHYY, H.S. UDAYKUMAR, M.M. RAO, AND R.W. SMITH, *Computational Fluid Dynamics with Moving Boundaries*, Taylor and Francis, Washington DC (1996).

[68] E.H. SMITH, *Autorotating wings: an experimental investigation*, J. Fluid Mech. **50** (1971), pp. 513-534.

[69] J.L. STEGER AND R.F. WARMING, *Flux vector splitting of the inviscid gas-dynamic equations with applications to finite difference methods*, J. Comput. Phys., **40** (1981), pp. 263-293.

[70] J.M. STOCKIE, J.A. MACKENZIE, AND R.D. RUSSELL, *A moving mesh method for one-dimensional hyperbolic conservation laws*, SIAM J. Sci. Comput., **22** (2001), pp1791-1813.

[71] M. SUN, K. YADA, G. JAGADEESH, O. ONODERA, T. OGAWA AND K. TAKAYAMA, *A study of shock wave interaction with a rotating cylinder*, Shock Waves **12** (2003), pp. 479-485.

[72] M. SUN, T. SAITO, P.A. JACOBS, E.V. TIMOFEEV, K. OHTANI, AND K. TAKAYAMA, *Axisymmetric shock wave interaction with a cone: a benchmark test*, Shock Waves, (2003), Volume **14**, Numbers 5-6 pp. 313-331.

[73] M. SUSSMAN, P. SMEREKA AND S. OSHER, *A level set method for computing solutions to incompressible two phase flow*, J. Comput. Phys. **119** (1994), pp. 146.

[74] H.Z. TANG AND T. TANG, *Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., **41** (2003), pp. 487-515.

[75] H.Z. TANG, T. TANG, AND P.W. ZHANG, *Adaptive mesh redistribution method for nonlinear Hamilton—Jacobin equations in two- and three-dimensions*, J. Comput. Phys., **188** (2003), pp. 543-572.

[76] H.Z. TANG, *A moving mesh method for the Euler flow calculations using a directional monitor function*, Communications in Computational Physics, **1** (2006).

[77] T. TANG, *Moving mesh methods for computational fluid dynamics*, Contemporary Mathematics, vol. 383, American Mathematical Society, 2005.

[78] H.M. TSAI, A.S.F. WONG, J. CAI, AND F. LIU, *Unsteady flow calculations with a parallel multiblock moving mesh algorithm*, AIAA J. **39** (2001), pp. 1021-1029.

[79] M. VAN DYKE, *An album of fluid motion*, The Parabolic Press, Stanford, CA, (1982), p. 143.

[80] B. VAN LEER, *Toward the Ultimate Conservative Difference Scheme IV, A New Approach to Numerical Convection*, J. Comput. Phys., **23** (1977), pp. 276-299.

[81] J. VON NEUMANN AND R.D. RICHTMYER, *A method for the numerical calculations of hydrodynamics shocks*, J. Appl. Phys. **21** (1950), pp. 232.

[82] W.W. WILLMARTH, N.E. HAWK AND R.L. HARVEY, *Steady and unsteady motions and wakes of freely falling disks*, Phys. Fluids **7** (1964), pp. 197-208.

[83] A. WINSLOW, *Numerical solution of the quasi-linear Poisson equation*, J. Comput. Phys. **1** (1967), pp. 149-172.

[84] A. WINSLOW, *Adaptive mesh zoning by equipotential method*, Report UCID-19062, Lawrence Livermore Laboratory (1981)

[85] P. WOODWARD AND P. COLELLA, *The numerical simulation of two dimensional fluid flow with strong shocks*, J. Comput. Phys. **54** (1984), pp. 115-173.

[86] 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), pp. 289-335.

[87] K. XU, *Gas-kinetic schemes for unsteady compressible flow simulation*, von Karman institute for fluid dynamics lecture series 1998.

[88] K. XU AND M. L. MAO, *A multidimensional gas-kinetic BGK scheme for hypersonic viscous flow*, J. Comput. Phys. **203** (2005), pp. 405-421.

[89] K. XU AND X. HE, *Lattice Boltzmann method and gas-kinetic BGK schemes in the low-Mach number viscous flow simulations*, von Karman institute for fluid dynamics lecture series 1998.

[90] N.K. YAMALEEV AND M.H. CARPENTER, *On accuracy of adaptive grid methods for captured shocks*, J. Comput. Phys. **181** (2002), pp. 280-316.