INVESTIGATION ON THE INTERACTION OF A SHOCK AND A LIQUID DROPLET WITH AND WITHOUT A VAPOR CAVITY INSIDE

LIU YAO

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Mechanical Engineering

Investigation on the Interaction of a Shock and a Liquid Droplet with and without a Vapor Cavity inside

LIU Yao

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

September 2020

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

LIU Yao (Name of student)

Abstract

The stripping breakup process of a single liquid droplet under the impact of a planar shock wave is the phenomenon that has been widely observed over the last 70 years. This droplet aerobreakup phenomenon is a fundamental and challenging two-phase flow problem that occurs in multi-application, such as secondary atomization and raindrop damage during an atmosphere reentry. A challenging problem which arises in this domain is how the internal structure and flow field inside the droplet affects the droplet's initial deformation.

In this thesis, the droplet breakup experiment is conducted in a horizontal shock tube and the evolution of the droplet interface is recorded by direct high-speed photography. Compressible Euler equations are solved using an in-house inviscid upwind characteristic space-time conservation element and solution element (CE/SE) method coupled with the HLLC approximate Riemann solver. A reduced five-equation model is employed to demonstrate the air/liquid interface.

First, we present the numerical and theoretical investigation targeting the establishment of an internal flow field inside a pure water droplet exposed to shock-wave impact. The main focus is on the description of the droplet internal flow pattern, which is

believed to be one of the dominant factors in initial droplet deformation. The droplet internal flow pattern holds steady for quite a long time after the incident shock passage, and a saddle point is observed for the first time. Accordingly, the saddle point inside the droplet flow is used as a characteristic point to describe the internal flow. Cases of different incident shock strengths and liquid are tested, and a theoretical prediction is proposed to delineate the correlation between the saddle point steady position and the strength of the incident shock wave. The numerical cases are found to be in good agreement with the prediction. The present study helps to complete the understanding of the overall droplet aerobreakup phenomenon.

Moreover, to bridging the knowledge gap that few studies have yielded the deformation and breakup regimes of a water droplet embedded with a vapor cavity in the high-speed airstream, the interaction of a shock wave and a water droplet embedded with a vapor cavity is experimentally investigated in a shock tube for the first time. The vapor cavity inside the droplet is generated by decreasing the surrounding pressure to the saturation pressure, and an equilibrium between the liquid phase and the gas phase is obtained inside the droplet. The formation of a transverse jet inside the droplet during the cavity-collapse stage is clearly observed. Soon afterwards, at the downstream pole of the droplet, a water jet penetrating into the surrounding air is observed during the cavity-

expansion stage. The evolution of the droplet is strongly influenced by the evolution of the vapor cavity. The phase change process plays an important role in vapor cavity evolution. The effects of the relative size and eccentricity of the cavity on the movement and deformation of the droplet are presented quantitatively. A modified Rayleigh-Plesset equation is derived that reasonably predicts the bubble collapse process.

Publications Arising from the Thesis

Journals

GUAN, B., LIU, Y., WEN, C. Y. & SHEN, H. 2018 Numerical study on liquid droplet internal flow under shock impact. *AIAA J.* **56** (9), 3382–3387.

LIANG, Y., JIANG, Y., WEN, C. Y., & LIU, Y. 2020 Interaction of a planar shock wave and a water droplet embedded with a vapour cavity. *J. of Fluid Mech*. ***885***.R6.

Conference

LIU, Y., WEN, C., SHEN, H., & GUAN, B. 2017 Investigation on shock induced stripping breakup process of a liquid droplet. *21st AIAA International Space Planes and Hypersonics Technologies Conference*. 2369.

Acknowledgments

This dissertation would not have been completed without the guidance, support, and encouragement of the following people. It is to them that I own my deepest gratitude.

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Chih-Yung Wen, for his great support of my Ph.D. studies. Thank you for your great guidance and scholarly inputs on my research. Despite your busy schedules, you have always made yourself available to discuss with me about research problems and provided me with generous financial support for my Ph.D. studies. Thank you also for the consistent spur that pushes me to my best.

I would like to thank all those who served on my thesis committee and my external examiners: Prof. WANG Bing and Prof. CHERN Ming-Jyh, for spending time giving me comments and suggestions on my thesis. I also greatly appreciate all the support I received from the faculty members and staffs of the ME Department, including Dr. LIU Yang, Dr. NG Chun, and Mr. YUEN Ka-on.

I would like to express my gratitude to my peers, including but not limited to Dr. Lu Shen, Dr. Lisong Shi, for the friendship and emotional support. They made my Ph.D. studies enjoyable!

Finally, and most importantly, I would like to thank my parents for their unconditional love and support. No matter what decisions I make, my loving parents always trust me and stand behind my back. Special thanks are given to my wife, Fangyuan Chen, and my 5-year-old son, Qingjing Liu. Ph.D. study is a long and challenging journal, thanks my wife for walking besides me during this journey and thanks my son for making our life full of fun and hope. Without my family's support, it would have been impossible for me to finish my PhD study successfully.

Table of Contents

List of Figures

List of Tables

Nomenclature

γ	ratio of specific heats
W_e	Weber number
R_e	Reynolds Number
Oh	Ohnesorge number
M_s	Mach number of shock wave
σ	surface tension $[N/m]$
μ	dynamic viscosity $[Pa \cdot s]$
ρ	fluid density $[kg/m^3]$
α_i	volume fraction of fluid i
E	total energy $[J]$
t^*	dimensionless time
L	distance between first atomization point and second atomization point $[m]$
l_{sp}	displacement of the SP $[m]$
a_0	sound speed in quiescent air $[m/s]$
t_s	duration in which the shock influences the LS area $[s]$
u_{sp}	velocity at which the SP moves $[m/s]$
t_{sp}^0	the instant when the SP appears $[s]$
D_d	diameter of the droplet $[m]$
D_c	cavity diameter $[m]$
S	eccentric distance between the cavity centre and the droplet centre$[m]$
τ_R	Rayleigh time $[s]$
L_c	length of the vapor cavity $[m]$
δ	diameter ratio
ϵ	eccentricity
Γ	Gamma function

Abbreviations

KHI	Kelvin-Helmholtz Instability
RTI	Ralrigh-Taylor Instability
RMI	Richtmyer-Meshkov Instability
CE/SE	Conservation Element and Solution Element
IS	incident shock
RS	reflected shock
MS	Mach stem
WS	windward stagnation point,
LS	leeward stagnation point
SP	saddle point
TS	transmitted shock
DS	Diffracted shock
RZ	recirculation zone
SV	shedding vortex
EOS	equation of state
UW	upstream wall of the droplet
DW	downstream wall of the droplet
UI	upstream interface of the vapor bubble
DI	downstream interface of the vapor bubble
UWB	upstream wall boundary
DWB	downstream wall boundary
UIB	upstream interface boundary
DIB	downstream interface boundary
RRR	regular refraction with reflected shock
FPR	free precursor reflection
FNR	free precursor von Neumann reflection
RTP	Rayleigh-Taylor piercing
SIE	shear-induced entrainment

REx	reflected expansion wave

1. Introduction

1.1 Backgrounds

After being impacted by the coming shock wave, the droplets will then deform and break up in the following subsonic or supersonic free stream gas flow. This phenomenon is a hot-spot flow problem focusing on the evolutions of the high-speed impact interface shape and the droplet deformation and fragmentation mechanism. Researchers have put attention on this problem for several decades, due to its wide applications in the industrial and natural environment.

As shown in Fig. 1.1(a), such industrial applications include the ablation management of space vehicles caused by rain droplet impingement during an atmosphere reentry (Reinecke & Waldman 1970). By optimizing the shape of the aircraft and improving the aerodynamic characteristics of the surface, the droplets will be broken before hitting the surface, thus reducing the energy of the droplets and effectively reducing the damage of the aircraft surface. Other applications involve the supersonic combustion in multiphase mixtures for scramjet engines (Fig. 1.1(b)), and the national security measures for the terrorist attack of an airborne bio/chemical weapon explosion, etc.

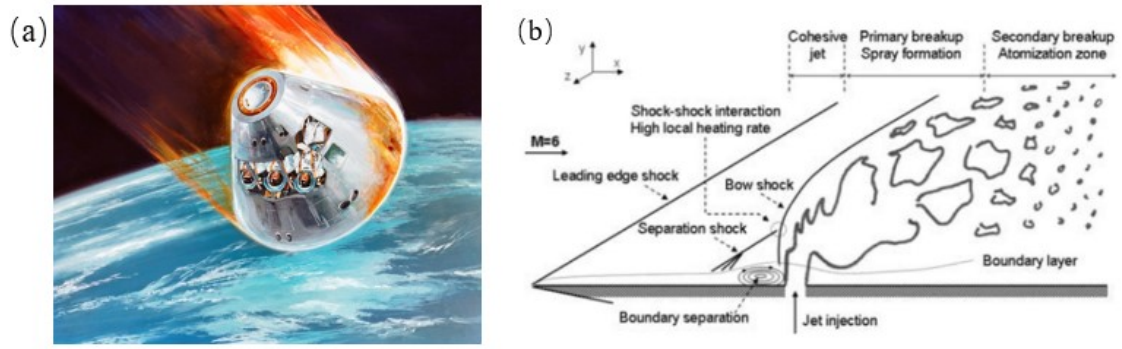


Figure 1.1 Examples of droplet aerodynamic breakup in engineering applications. (a) Apollo capsule during atmospheric reentry; (b) schematic diagram of a liquid jet injected into a high-speed cross flow (Perurena et al., 2009).

In the combustion chamber of the engine, as shown in Figure 1.1(b), the liquid jet that ejected from the fuel nozzle into the high-speed air stream first break into multiple droplets, these large droplets then secondary breakup into smaller droplet sizes (Perurena et al. 2009), accurately predicting droplet fragmentation times and the mist particle size distribution would be useful in optimizing engine design and raise the combustion efficiency. As the energy that leading to the liquid droplets breakdown comes from the kinetic energy of the high velocity airflow after the shock wave, the installation of a water curtain in the closed compartment can reduce the damage to the equipment and casualties caused by the blast wave(Van Wingerden 2000).

The fragmentation of isolated spherical droplets in airflow is a fundamental topic in the study of high-speed two-phase flow. A large number of scholars have studied and discussed the droplet fragmentation problem under different incoming flow conditions.

The early studies were mainly based on experimental observation and theoretical analysis, including the description of the morphology of droplets in the airflow under various incoming flow conditions, the measurement of droplet fragmentation time and the discrimination of several fragmentation mechanisms under different incoming flow conditions. As the experimental means and measurement accuracy are gradually enriched, the study of the droplet deformation and crushing mechanism is refined, and the droplet breakup experimental results are further enriched. At this stage, the transformation of the crushing mechanism and its influence by the incoming flow parameters are the hotspots of related research. Although the most fruitful achievements were obtained in experimental methods, the computational fluid dynamics played a significant role in understanding the droplet breakup mechanism. Because of the water's surface tension, the diameter of liquid droplets in nature is less than 3mm, and the wave system cannot be seen clearly using only experimental observation methods such as schlieren method. It is also difficult to measure the interfacial pressure distribution and temperature change of the deforming liquid droplets due to the experimental capability. With the help of CFD, a lot of studies have been carried out to simulate the droplet deformation and crushing process by numerical methods and to reveal its underlying mechanism. However, most of the existing studies are focused either on the first stage of droplet deformation or on the pure numerical validation.

The outer air flow is assumed to be the only drive of the droplet deformation, which leads to incomplete understanding of the droplet breakup mechanism. As a result, a comprehensive experimental and numerical investigation is still needed. The experimental images should illustrate the droplet interface evolution features from its early to relatively late stage. At the meantime, it is incomplete to only interpret the deformation from the perspective of outer flow. The inner liquid flow is also need be considered. Herein, we build upon the computational efforts of Shen and Wen (2016), and utilize a flow solver to numerically investigate this fundamental fluid dynamics problem.

The problem of aerodynamic deformation of bubbling droplets is also prevalent in engineering applications. In an experimental study of excitation wave impingement on a liquid column, Sembian et al. (2016) found that cavitation bubbles were generated downstream of the column, which they attributed to the convergence of expansion waves reflected from the interface downstream of the column at specific locations within the droplet, generating a low pressure sufficient to initiate cavitation; meanwhile, numerical simulations confirmed this, i.e., the location of the simulated low-pressure zone and the experimental cavitation. The location of the air bubbles is coincident. In ultrasound therapy, Shpak et al. (2016) used droplets to coat certain low-boiling drugs (e.g. volatile perfluoropentane) and they found that once a phase change was triggered, a large cavity

would form inside the droplet. In addition, a large number of vesicular-containing droplets are generated during vapor-containing column jet fragmentation (Lin et al., 2004). In contrast, the aerodynamic deformation and fragmentation kinetics of vesicular droplets is relatively complex, and there exists both aerodynamic deformation and fragmentation of liquid and bubble collapse, which belongs to the combination of the two. At present, there is a lack of research on the pneumatic deformation and fragmentation of bubbling droplets, and the underlying mechanism needs to be further explored.

Even with advancements in numerical methods, it is difficult to quantify the behavior of the interface shape of the droplet with cavitation bubble in aerobreakup experiments. This thesis represents the start of an attempt to bridge this gap in the current state of aerobreakup knowledge. The purpose of the present study is to elucidate the stripping breakup mode of a single liquid droplet embedded with a vapor cavity induced by a planar shock wave. Experimental results present the detailed deformation and breakup mechanism of this kind bubbling droplet.

1.2 Motivations

A detailed understanding of the physical mechanisms of the breskup process, and how they translate into final fragment sizes, has been a challenge to theorists, experimentalists, and computer modelers alike.

With the considerations all above, the motivations of the present subject are as follows:

1. Experimentally study the deformation and breakup patterns of millimeter-scale liquid droplets with or without a vapor cavity inside in the high-speed airstream behind a shock wave, with the focus on the influence of the parameters, such as We and Re on the development of the instability (R-T instability and K-H instability).

2. Numerical study the inner flow effect on droplet deformation in the shock/droplet interaction problem.

3. Through the complementarily numerical and experimental efforts, investigating the detailed stripping breakup governing mechanisms and the underlying physics.

1.3 Historical Perspective

1.3.1 Droplet Deformation Resulting from the Shock-Droplet Interaction

The interaction between shock wave and an individual liquid droplet, which yields the consequent aerodynamic breakup of the droplet, poses a fundamental and challenging two-phase flow problem. The rise of supersonic flight has created a requirement for better understanding of shock waves and of the properties of gases/liquids processed by shock waves. The comprehensive reviews on this subject can be found in Wierzba & Takayama (1988), Joseph et al. (1999), Guildenbecher et al. (2009) and Theofanous (2011).

The breakup of a liquid droplet is a complex multiple physics problem, characterized by many dimensionless parameters in a competitive manner (Theofanous, 2011). The effects of four dimensionless parameters were mainly discussed: Weber number We, Ohnesorge number Oh, Reynolds number Re and Bond number Bo. The definitions of these parameters are:

$$We = \frac{\rho_g u_g^2 D}{\sigma} \tag{1.1}$$

$$Oh = \mu_l / \sqrt{\rho_l \sigma D} = \frac{\sqrt{We}}{R_e} \tag{1.2}$$

$$Re = \rho_g u_g D / \mu_g \tag{1.3}$$

$$B_o = a\rho_l D^2 / 4\sigma \tag{1.4}$$

where ρ is density, u the velocity, D the diameter of the liquid droplet, σ the surface tension of the droplet, a the droplet acceleration, and μ the viscosity. Subscripts l and g indicate the liquid and gas behind the shock. The parameters W_e , Oh , R_e , and Bo present the ratios of the fluid's inertia force to its surface tension force, viscous force to its inertial and surface tension force, inertia force to viscous force and body force to surface tension force, respectively.

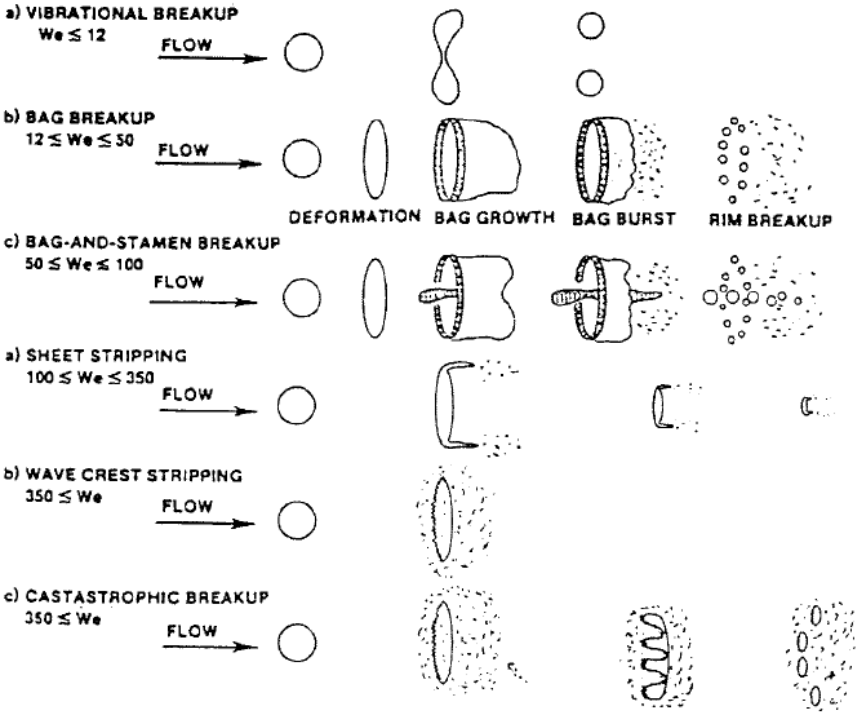


Figure 1.2 Six distinct mechanisms of droplet breakup as determined by the initial Weber number (Pilch and Erdman 1987).

The fascinating feature of liquid droplet breakup is the various distinct breakup regimes exhibited under different conditions. As shown in Fig 1.2, the types of liquid droplets breakup can be categorized into six modes according to different values of W_e

(Pilch & Erdman, 1987): vibrational mode, bag mode, bag-and-stamen mode, chaotic mode, stripping (shear) mode and catastrophic mode. These mechanisms are generally considered applicable to the breakup of droplets with very low viscosity (Oh<<1), in a qualitative way.

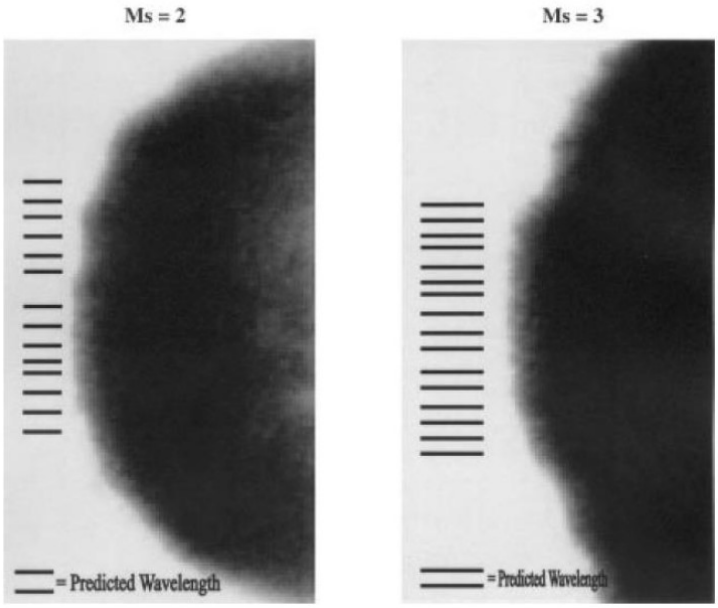


Figure 1.3 The castastrophic breakup regime (Joseph et al. 1999).

Wierzba & Takayama (1988) experimentally studied the mechanism of shear breakup of liquid droplets. A four-stage mechanism of the stripping breakup of liquid droplets was proposed by using holographic interferometry. Yoshida & Takayama (1990) showed that the patterns of the droplet breakup differed significantly with the methods of visualization. Images of the droplet breakup observed by double-exposure holographic interferometry were found to be different from those observed by shadowgraphs. To eliminate three-dimensional (3D) effects, Igra & Takayama (1999) visualized the two-

dimensional (2D) shock/water-column interactions in a shock tube by sequential double-exposure holographic interferograms. They mentioned that a 2D shock/water column interaction provides the only method for quantitative visualization of wave motions in water.

As shown in Fig. 1.3, Joseph et al. (1999, 2002) presented a series of shadowgraphs, covering a wide range of W_e (=11700−169000), Oh (=0.002−82.3) and R_e (40000−127600), and claimed that the droplet breakup is mainly caused by Rayleigh Taylor instability under high W_e (catastrophic mode). However, Theofanous et al. (2007, 2008 & 2011) argued that both temporal and spatial resolutions of shadowgraphs are not high enough to capture the fine structures during the breakup process and the "catastrophic mode" may not exist. As shown in Fig. 1.4, they investigated the aerodynamic breakup of Newtonian and viscoelastic droplets by laser-induced fluorescence visualizations at spatial resolutions of up to 200 pixels for millimeter and exposure times as low as 5 ns. They concluded: (1) as shown in Fig. 1.5, the droplet breakup can be mainly contributed by two major mechanisms: Rayleigh-Taylor piercing (RTP) (shown in Fig. 1.4(a),(b)) and shear-induced entrainment (SIE) (shown in Fig. 1.4(c),(d)), corresponding to low and high Weber number. (2) with increasing We, SIE is favored over RTP, and (3) there is no practical limitation for Oh to droplet breakup. Their observations are apparently different from previous studies.

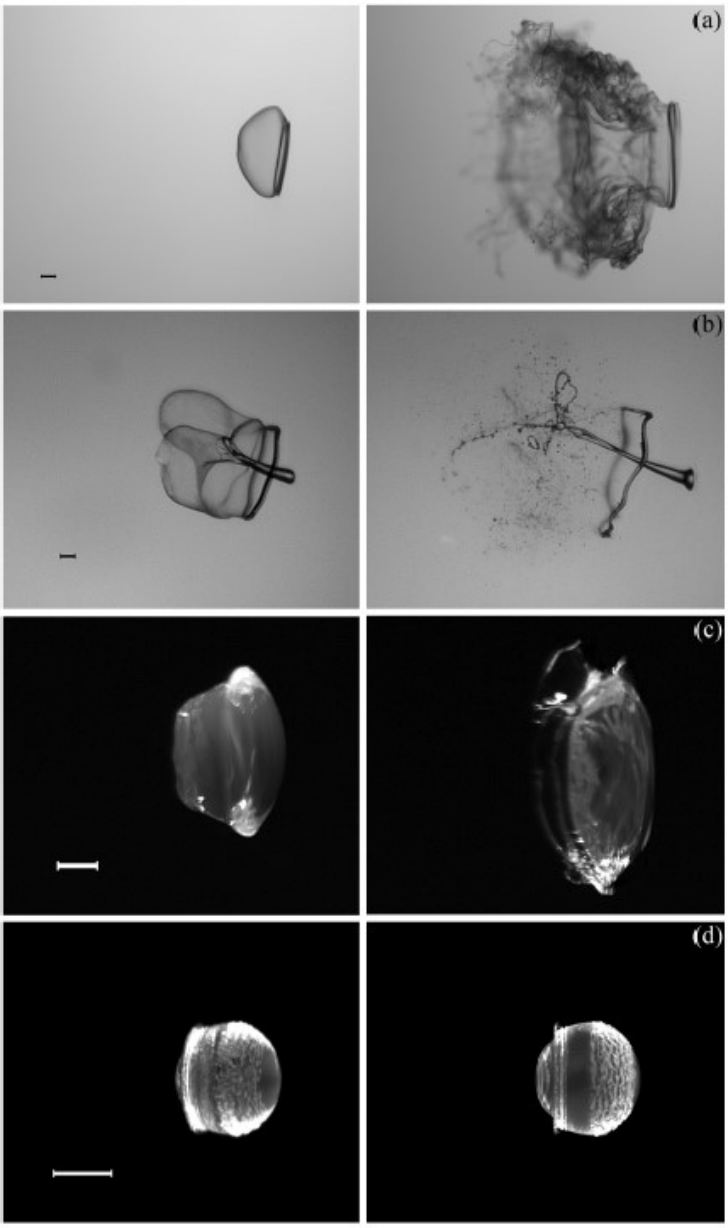


Figure 1.4 Illustration of the principal criticalities responsible for aerobreakup of liquid droplet. (a) the first mode of RTP; Run[SO1000 (silicon oils, viscosity equals 1000 cSt); W_e=1500; Oh=5.3]; (b) the second mode of RTP; Run[SO10 (silicon oils, viscosity equals 10 cSt); W_e=24.2; Oh=0.05]; (c) the onset of SIE; Run[Water (viscosity equals 1 cSt); W_e=210; Oh=0.002], and (d) the early stage of development of K-H SIE; Run[TBP4 (tri-butyl phosphate, viscosity equals 4 cSt); W_e=7000; Oh=0.018] (Mitikin & Theofanous 2017).

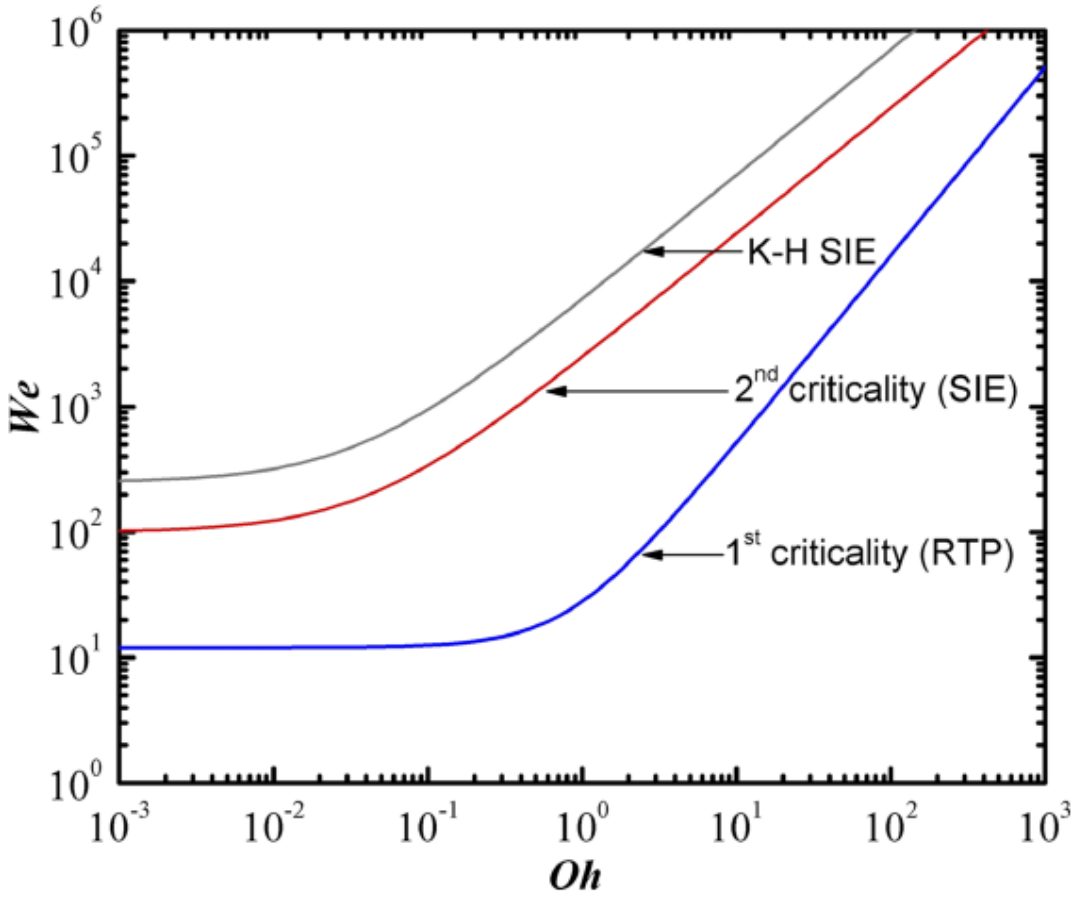


Figure 1.5 The three criticalities of aerobreakup with Newtonian liquids (Mitikin & Theofanous 2017).

Another important aspect of liquid droplet breakup is the time required for complete breakup. Liang et al. (1988) summarized the early experimental measurements of breakup time, including the finding of Ranger & Nicholls (1969) and Simpkins & Bales (1972) for shear breakup and those of Reinecke & McKay (1969) and Reinecke & Waldman (1970) for catastrophic breakup (all for shock wave disturbances at large ρ_l/ρ_g and low Oh). Under these conditions, the normalized breakup time t_b/t^* for both shear and catastrophic breakup modes was shown nearly constant over a large range of We. Here, the characteristic

breakup time $t^*=d_0(\rho_l/\rho_g)1/2/u_0$. Hsiang & Faeth (1992) found that $t_b/t^*=5$ provides a reasonably good correlation for all the measurements of the shear breakup mode with Oh<0.1 in Fig. 6. When considered the effect of Oh, an empirical correlation of $t_b/t^*=5/(1-Oh/7)$ were proposed for We<1000 and Oh<3.5. Joseph et al. (2002) applied the RT instability theory to analyze the catastrophic mode with the unstable disturbances of amplitude $A_{(t)}= A_0e^{nt}$, where A_0 is the initial amplitude and n the growth rate. A 'break-up' time $\hat{t_b}$ is arbitrarily defined as the time taken for the initial A_0 to grow to M times its value: $\hat{t_b} = (1/n)\, ln\, M$. Theofanous et al. (2007) presented data on deformation and breakup regimes, aided by numerical simulations. Different time scales were used to correlate the governing mechanisms in different breakup regimes. Their results agree with the breakup time correlation of Hsiang & Faeth (1992).

Focusing on the stripping mode, researchers conducted a number of classical experiments in various ways to study the breakup mechanisms. Wierzba and Takayama (1988) performed their shock tube experiments combined with five different photographic techniques and established the famous four-stage mechanism of the stripping type breakup. Hsiang and Faeth (1992) conducted a series of experiments concentrating on the atomization and spay problems and improved the famous droplet deformation and breakup regime map. Theofanous & Li (2008); Theofanous (2011); Theofanous et al. (2012) did a

number of studies in the aerodynamic breakup of liquid droplet in a rarefied supersonic flow using their LIF technique. They recorded clearly the droplet evolution and redefined the breakup mechanism. In order to minimize the uncertainty in visualizing the three-dimensional shock/droplet interaction, two dimensional liquid columns were adopted. Igra and Takayama (2001; 2003) performed experiments on shock/water column interactions in a shock tube by sequential double-exposure holographic interferograms. Sembian et al. (2016) successfully observed the propagation of waves inside the water column for incident Ma 1.75 and 2.4 by creating a water column of 22 mm in diameter using super-hydrophobic coating techniques.

The above-mentioned experiments provided a comprehensive database of the liquid droplet aerodynamic breakup phenomena. However, thorough understanding of the breakup mechanisms can only be addressed by direct numerical simulations (Joseph et al. 1999; Theofanous 2011). Various numerical works have been performed in recent years. Based on their experimental results, Igra and Takayama (2001) studied the shock/water column interaction using an interpolated pseudo-particle (CIP) scheme. Chen (2008) reported 2D simulations of stripping breakup of a water column where the five-equation model was solved by a Godunov-type scheme coupled with the HLLC-Riemann solver. Theofanous et al. (2012) combined the numerical code MuSiC+ and AROS to simulate the

droplet breakup process observed in their experiments. In their simulation, the MuSiC+ code provided the shear stress distributions and viscous boundary layer details around a drop as input to AROS, and the AROS predicted viscous KH dispersion on the drop surface. Meng and Colonius (2013; 2015) studied the effects of shock strength on the droplet breakup. The five-equation model was solved by a 3rd order WENO scheme coupled with the HLLC Riemann solver.

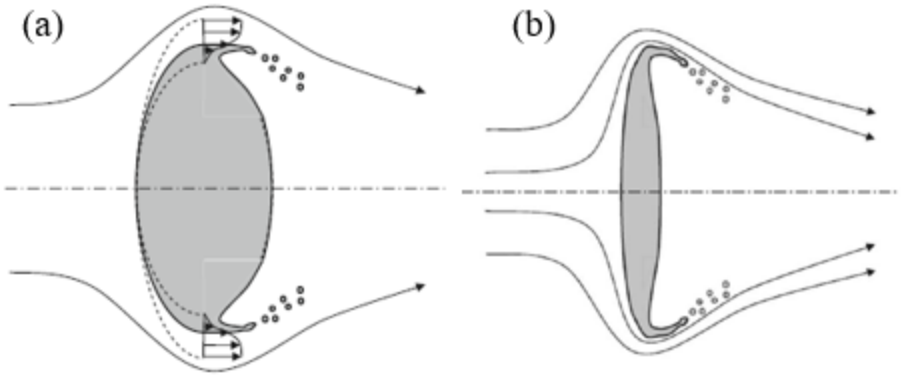


Figure 1.6 Two droplet breakup mechanism (Guildenbecher te al. 2009).

As shown in Fig. 1.6, Two theoretical models exist for the generation of liquid mist at the periphery of the droplets during the breakup process. Fig. 1.6 gives a diagram of the two theoretical models respectively. Fig. 1.6(a) shows the shear stripping breakup mechanism proposed by Ranger & Nochollos (1969). They assume that the droplets are spherical prior to fragmentation and ignore the pressure gradient and liquid swirl inside the droplets. A two-phase boundary layer forms gradually between the gas and liquid phases

and develops a K-H instability. The instability perturbation grows gradually until it breaks free of the surface tension bound and leaves the main droplet to form a liquid mist by entrainment of the airflow.

Another mechanism for the generation of liquid mist is the crushing mechanism for thin layer refinement proposed by LIU & Reita (1997). Under the combined effect of standing high pressure and low pressure near the equator, the droplet gradually flattens, and due to the shear of the airflow, a thin layer of liquid with a thickness comparable to the liquid inner boundary layer is formed in the outer layer of the droplet. This thin layer extends downstream, the thickness continues to thin, and finally due to capillary instability and broken into a large number of small droplets.

So far, a large number of experimental and numerical results have supported the sheet-thinning theory. However, the existence of shear stripping fragmentation cannot be ruled out at present. Therefore, along with the change of experimental parameters, the two liquid mist generation mechanisms can be switched between the above two.

1.3.2 Bubble Collapse

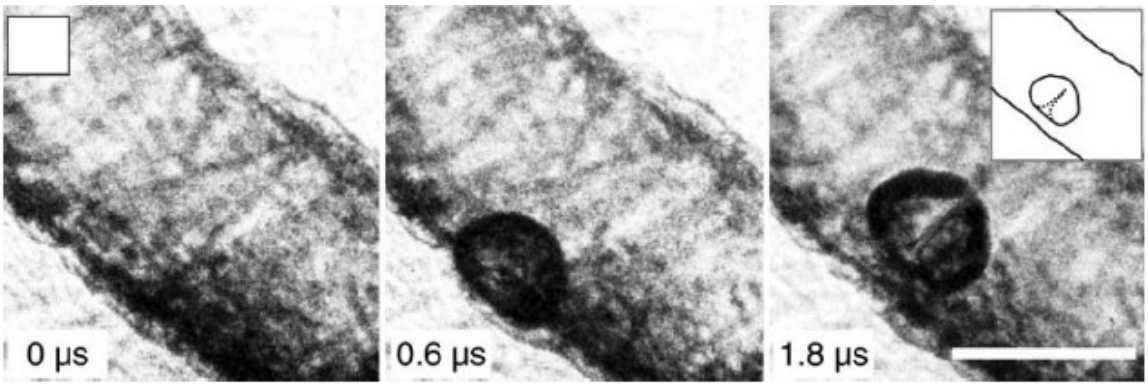


Figure 1.7 Examples of bubble collapse in medicine aera: the interaction of bubbles and blood vessels (Chen et al. 2011).

Bubble collapse refers to the process by which a bubble rapidly shrinks in volume, and rises in pressure and temperature in response to an external high-pressure fluid. Generally speaking, when the pressure inside the bubble rises to a level that is sufficient to resist the external high-pressure fluid, the bubble will begin to expand (along with a decrease in pressure) until it collapses again. Thus, due to inertia, this collapse-expansion process can continue several times, gradually decreasing in intensity, until the kinetic energy is dissipated.

In Extracorporeal Excitation Lithotripsy (EELT), which is commonly used in the medical field, ultrasound waves are converged and propagated to the patient's stone, inducing the collapse of microbubbles near the stone to generate a high-speed jet, which hits the stone or body fluids, generating an extremely high pressure, thus crushing the stone and achieving a therapeutic effect (Johnsen & Colonius, 2008, 2009). In the use of targeted

drug therapy, as shown in Fig. 1.7, drug-coated microbubbles are delivered to the diseased areas in the human body, and then ultrasound is used to induce microbubble collapse, which causes the drug to be released in the diseased area to achieve the targeted therapy (Chen et al., 2011; Wang et al., 2015). In addition, bubble collapse is also present in applications such as hot spot formation in explosives (Bowden & Yoffe, 1958; Bourne & Field, 1991) and ultrasonic cleaning (Hodnett & Zeqiri, 1997).

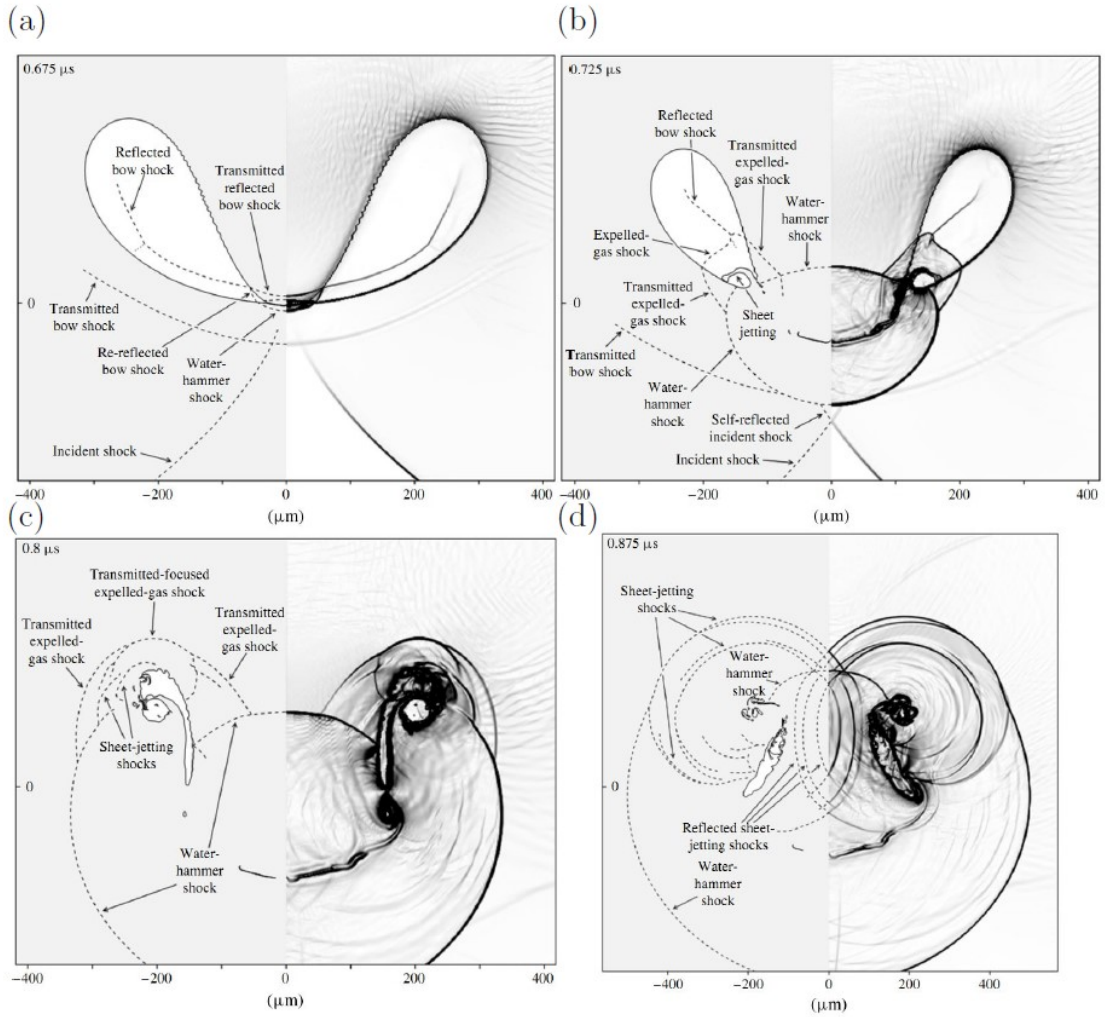


Figure 1.8 A series of pictures depicting the collapse of a 1 mm diameter cylindrical bubble driven by a 1 GPa incident shock.(Hawker & Ventikos 2012)

For an ideal spherical bubble collapse problem, the dynamics can be analyzed by solving the Rayleigh-Plesset equation (Brennen, 1995), which makes the assumption of no viscous incompressibility on the Navier-Stokes equation. In both natural and engineering applications, bubble collapse is often non-spherical due to a variety of perturbations: free interfaces, walls, volumetric forces (e.g., gravity), other bubbles, and shock waves or ultrasonic impacts.

A intense bubble collapse process generates an excitation wave (Supponen et al., 2016), which in turn acts on other nearby bubbles (Lauer et al., 2012a). Ohl & Ikink (2003) experimentally investigated bubble collapse in water induced by a weak excitation wave (peak pressure of about 22 MPa) with a range of bubble diameters of 7 to 55 μm. The jet tip length and the mean jet velocity were found to increase linearly with the bubble diameter. Ball et al. (2000) used the Free-Lagrange method to numerically simulate the problem of 6 mm cylindrical bubbles impacted by a strong surge of 1.9 GPa, and the simulation results obtained are in good agreement with experimental observations (Bourne & Field, 1992), including linear compression of bubble volume with time, collapse of the bubble tip, and the effect of the collapse of the bubble tip on the bubble diameter. time, location of hot gases and cold glow. They also give the evolution of the interface and wave system structure during bubble collapse, such as the generation of lateral jets, jets hitting

downstream interfaces, and the generation of explosive waves (water hammer excursions), and predict that the final gas temperature will rise to 12,000 K due to the strong excitation heating effect. Hawker & Ventikos (2012) use a highly accurate Front Frontier software for the Tracking method, simulates the problem of a single bubble in water impacted by a strong excitation wave (after-wave pressure of 100 MPa to 100 GPa) and obtains very detailed results on the interface and wave system evolution, as shown in Figure 1.8. Based on the difference in physical mechanisms, they divided the bubble collapse dynamics into three stages: incident excitation to generate lateral jet, jet impact to generate water hammer excitation, and vortex ring evolution. It was found that lateral jet impact is not the only cause of pressure peaks in the liquid phase, but that secondary jet structures and expelled-gas excitations in bubbles also induce pressure peaks in the liquid phase. By contrast, they also found that spherical bubbles produce 40% higher water hammer excitation pressures than columnar bubbles, due to the convergence of transverse jets being more violent in the three-dimensional case. Another counter-intuitive phenomenon is that the density peak in the bubble decreases with the enhancement of the incident excitation, which is due to the reduced reflection time of the bowed excitation in the bubble.

1.3.3 The Aerobreakup of Single Droplet Embedded with Cavitation Bubble

Droplet aerodynamic deformation and bubble collapse has been extensively studied, in which the flow mechanism has been largely clearer. In contrast, the aerodynamic deformation and fragmentation kinetics of bubbling liquid droplets is relatively complex, with both aerodynamic deformation and fragmentation of liquids and bubble collapse, belonging to the combination of the two. (Payri et al., 2005) found that cavitation in engine nozzles has a strong influence on the formation and atomization of fuel spray, and that a well-characterized spray can improve combustion efficiency, reduce pollutant emissions, and thus improve overall engine performance. At present, with the increase of fuel injection pressure, the cavitation phenomenon has become very prominent in the fuel injection process, and the fuel jet becomes a bubble-liquid two-phase flow, which has an important impact on the atomization of the fuel spray. However, at present, the research on pneumatic deformation and fragmentation of bubble-containing liquid droplets started late, and the research results are still lacking, and the flow mechanism still needs to be further explored.

Lin et al. (2004) combined experimental and numerical simulations to study the problem of gas-containing liquid column injection, and found that the aerated liquid jet produced by the nozzle can effectively enhance liquid atomization and gas-liquid mixing under subsonic conditions by prefilling a small amount of gas into the liquid. In the far

field typical of gas-containing liquid column jets, they observed good spray characteristics, including a larger spray cross-sectional area, smaller droplet size, higher droplet velocities, and a fairly uniform liquid-liquid mist distribution in the spray plume. Lüet al. (2017) simulated the bubbling diesel droplet collapse process using the Volume of Fluid numerical method, separately with and without considering the phase change. Without considering the phase transition, the bubbles undergo multiple collapse rebound processes, which resemble damped spring oscillations, and the bubble volume fluctuation interval becomes progressively narrower. In the case of phase change, the bubbles only collapse but not rebound, and the collapse velocity of the bubbles accelerates with increasing ambient pressure and decreasing saturated vapor pressure; in contrast, the influence of ambient pressure on the collapse velocity is greater than that of saturated vapor pressure on the collapse.

Last several years, the phenomenon that cavitation bubbles appear in a droplet was experimentally observed in shock–water-column interaction (Sembian et al. 2016) and droplet–solid-wall impingement (Field, Dear & Ogren 1989; Field et al. 2012), where rarefaction waves and tension waves, respectively, play important roles in the formation of the cavitation bubbles.As shown in Fig. 1.9, Sembian et al. (2016) observed the cavitation

bubbles causing by the focus of the reflected expansion wave in the 22mm diameter 2D water column.

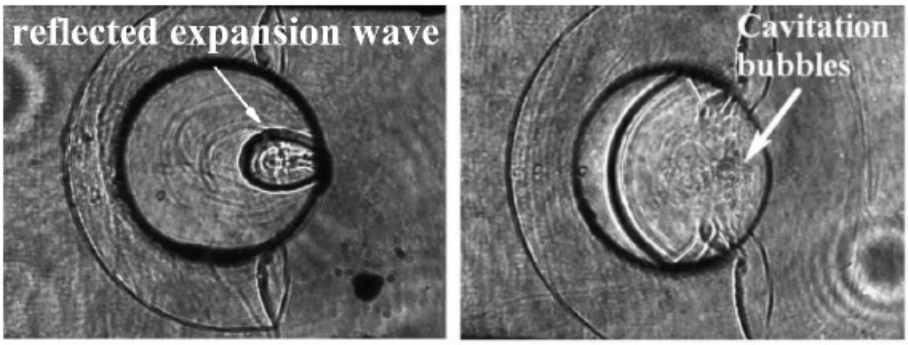


Figure 1.9 Experimental observation of reflected expansion waves and the resulting cavitation bubbles in a two-dimensional liquid column (Sembian et al. 2016).

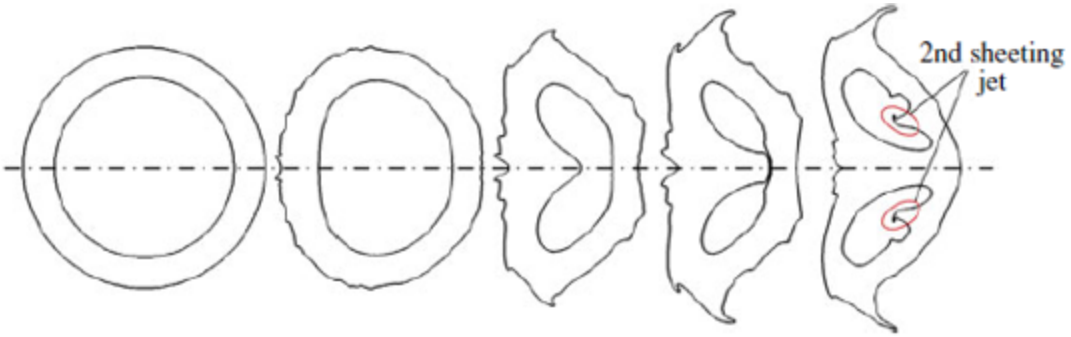


Figure 1.10 The interface evolutions of the shock/water-hollow at Mach number 2.4(Xiang & Wang 2017).

Additionally, Xiang & Wang (2017) numerically simulated the problem of excitation-impacted columnar pure droplets and cavity-containing droplets. In the study of pure droplets, the numerically simulated wave system structure and experimental observations (Sembian et al., 2016) are in good agreement. In particular, the expansion wave generated by the reflection from the wall downstream of the droplet converges inside

the droplet to produce a continuous low pressure that is sufficient to produce cavitation, and the numerically simulated low pressure region is in agreement with the experimentally observed location of cavitation. In the study of cavity-containing droplets, Xiang & Wang (2017), referring to the analytical approach of Hawker & Ventikos (2012), divided the evolution of the wave system structure and interface morphology into three stages: the establishment of the external flow field of the liquid ring, the generation of the jet and its evolution, and the impact of the jet on the downstream interface and its subsequent evolution. It is noteworthy that no water hammer excitation is generated during the third stage of jet impact, as the jet and downstream liquid ring quickly exchange momentum and fuse together to move downstream, thus reducing the pressure generated by the collision (peak impact in water is about 30 times atmospheric pressure). As shown in Fig. 1.10, the numerical simulation in Xiang & Wang (2017) indicated that cavitation bubbles can forms a transverse jet in the Mach 2.4 gas flow and a super high pressure region will be formed in near the impaction point. Then the liquid will be driven into the cavity to form the socalled second sheeting jet.

Wu et al. (2018) studied the problem of high-speed columnar droplet impact on the solid wall by numerical simulation with an initial impact velocity of 50∼200 m/s. It was found that when a columnar droplet hits the solid wall, a restricted bending excitation wave

is generated at the impacting surface, and the bending excitation wave propagates within the droplet towards the unimpacted surface; subsequently, due to the reflection of the surface of the unimpacted surface, a series of expanding waves converge at a distance of ~100 m/s from the droplet vertex. At one-third of the diameter, the local fluid sustained low pressure satisfies the cavitation conditions and produces a cavity. Within the parameters of the study, the location of the cavity was independent of the initial impact velocity; however, the higher the impact velocity, the larger the cavitation zone and the more dramatic the subsequent collapse.

Later, Wu, Wang & Xiang (2019) studied the problem of bubble-containing liquid column impingement on the solid wall, as shown in Figure 1.11, where the bubbles are air and saturated water vapor cases respectively. For the air bubble case, transmission waves are generated within the bubble and further compress the cavity; the main jet, lateral jet and sheet jet cause intermittent interfacial impingement fusion, and the impingement process generates a series of small bubbles, which collapse intermittently and produce intermittent collapsing wavelets. However, for the saturated water vapor case, no transmission waves are initially generated and the vapor cavity rapidly contracts with local condensation; as the cavity contracts, strong collapsing excitations are generated in the flow field. They improve on the spherical collapse theory and predict the non-spherical

collapse time, and the numerical simulations of the collapse time at different initial bubble radii and initial impact velocities are in good agreement with the improved theory.

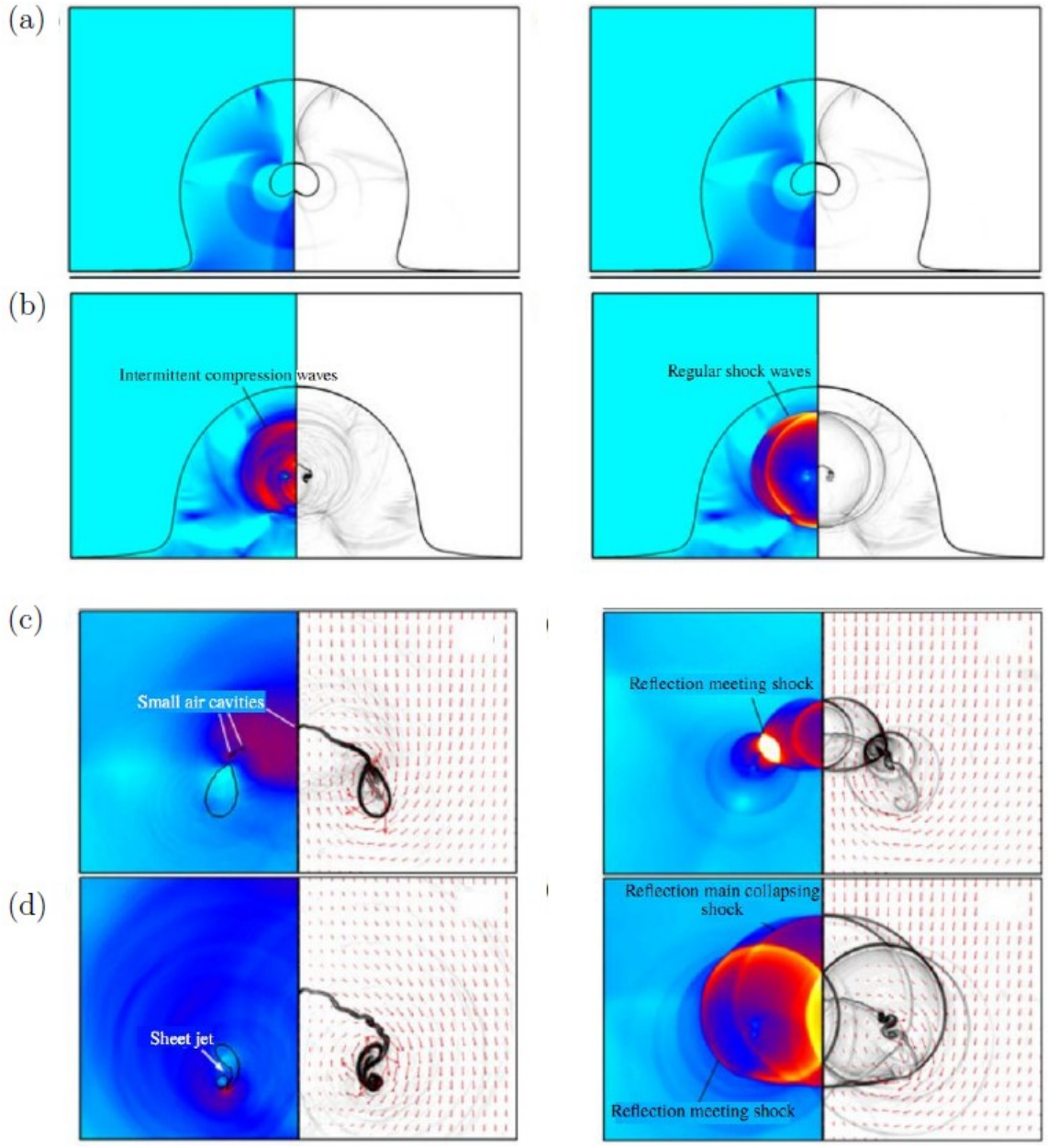


Figure 1.11 Numerical simulation of the evolution of interfaces and wave systems in the problem of solid-wall impingement of fluid columns embedded with bubble. The left-side figures are gas cavity, the right-side figures are vapor cavity (Wu et al. 2019).

1.3.4 The Instabilities in Aerobreakup

The potentially active modes for shock/droplet interection problem include Rayleigh-Taylor (RT) instability, Kelvin-Helvin-Helmholtz instability ((KHI)), and Richtmyer-Meshkov instability (RMI).

Harper et al. (1972) performed a systematic analysis of the RT instability induced by acceleration at the droplet surface, but their theoretical predictions are consistent with experiment only to a limited extent (windward side of the droplet in the later stages of fragmentation). Theofanous et al. (2011) and Chang et al. (2013) analyzed and simulated the droplet surface that KH instability mainly generate, and that RT puncture behavior is not present in the high Weber number condition. Wang et al. (2014) argued that there is RM instability involved in the droplet fragmentation process.

Theofanous et al. (2008; 2011; 2012) argue that Richtmeyer-Meshkov Instabilities can be neglected due to severe mismatch between the acoustic impedance of gases and droplets.

In the case of secondary atomization/droplet breaking in the high speed flow region after a shock wave, the competition which takes place on droplet surfaces is between K-H instability and R-T instability. The R-T instability occurs when a heavy fluid is being accelerated by a light fluid (shown in Fig.1.12). If Bond number $Bo < 10^5$, the R-T

instability will not the main reason for droplet breakup (Chen 2008). The K-H instability can occur where there is a velocity difference across the interface between two fluids (shown in Fig. 1.13). The K-H instability is caused by the sheer flow at the interface, and it can occur at very small scales in the horizontal system.

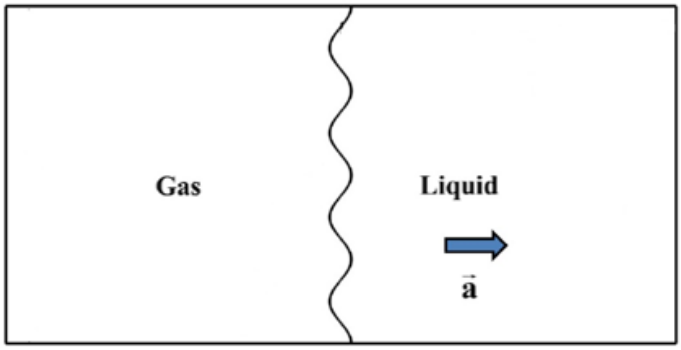


Figure 1.12 Schematic of Rayleigh–Taylor instability

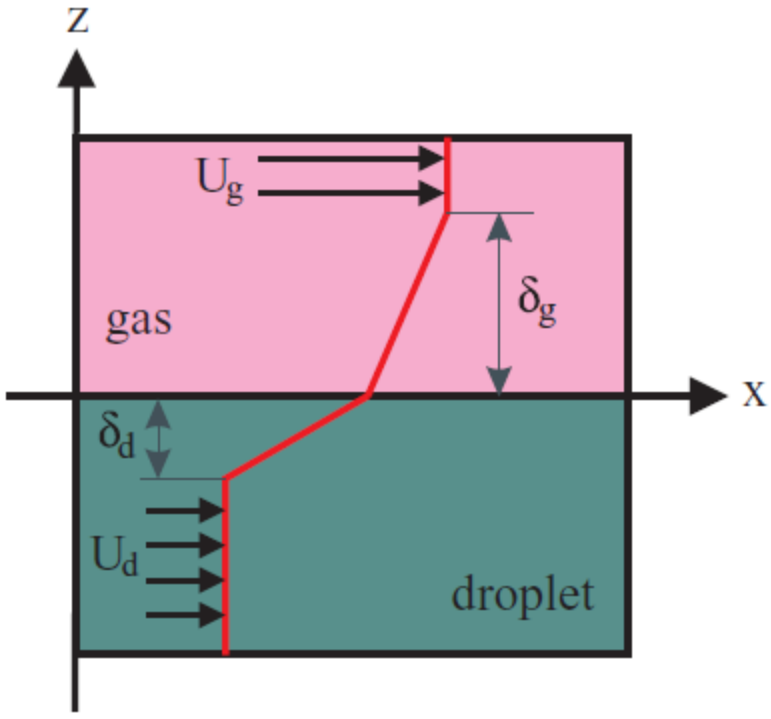


Figure 1.13 Schematic of Kelvin-Helmholtz instability (Jalaal & Mehravaran 2014).

The growth rate of KH and RT instabilities can be predicted by the linear-stability theory as (Kull, 1991)

$$n = \sqrt{k^2 \frac{\rho_1 \rho_2 [U]^2}{\rho_1 + \rho_2} + k \frac{[\rho\alpha]}{\rho_1 + \rho_2}} \tag{1.5}$$

where [U] and [ρα] are the jump of tangential velocity and normal acceleration across the interface. The 1st and 2nd terms in Eq. (1.5) are induced by KH and RT instabilities, respectively. It is difficult to evaluate the contribution of these two mechanisms experimentally or theoretically.

1.4 Scope of Thesis

There exists a gap in the current state of droplet aerobreakup knowledge associated with the underlying fundamental flow physics that dictate the experimentally observed shock/cavity-containing droplet interaction phenomena. Much work has yet to be done. Even with advancements in experimental methods, it is challenging to visualize and quantify the behavior of the gas phase in aerobreakup experiments.

The scope of this thesis is organized as follows.

The experimental and numerical methods are first laid out in Chapters 2. In Chapter 2, we describe the setup of the problem as it relates to previous experimental investigations.

All the experiments are conducted using the horizontal shock tube of the Department of Mechanical Engineering at the Hong Kong Polytechnic University. The governing equations and physical model for the in-house CE/SE numerical simulation method are then presented.

Investigation of aerodynamic breakup of a sphere liquid droplet behind a planer shock wave are described in Chapter 3. The inner flow effect on the droplet deformation is highlighted in this chapter.

Additionally, with the advanced experimental facilities established, we proceed to present the shock/vapor cavity containing droplet interaction experimental results in Chapters 4. Due to the the ignorance of the phase change in the present code, the code fails to simulate the evolution of both the droplet and cavity precisely. Therefore, the CE/SE simulations are not listed in this chapter. It is expected that subsequent numerical studies will fill this gap.

Lastly, conclusions and suggestions for future work are made in Chapter 5.

2. Method

This chapter is about the shock tube experimental setup and the numerical simulation method that this thesis uses and provides verification examples to assess the accuracy of the CE/SE numerical simulations.

2.1 Experimental Apparatus and Measurement Methodology

We conduct the experiments in a newly built self-designed shock tube facility with M_a number range 1.2~2.5, W_e range 10^3~10^4, and water droplets diameter D range 1~3mm.

A representative case conducted in this shock tube produces an incident shock with M_a number 1.52 with the post-shock air density $\rho_g = 1.14\ kg \cdot m^3$ and the free stream speed after the shock $u_g = 248.6\ m \cdot s^{-1}$. The water droplets in this case have a diameter D of 2.5 mm. Under this condition, the corresponding W_e number and R_e number are as follows:

$$W_e = \frac{\rho_g u_g^2 D}{\sigma} = 3.0 \times 10^3 \quad and \quad R_e = \frac{\rho_g u_g D}{\mu_g} = 3.3 \times 10^4 \tag{2.1}$$

where σ is the surface tension and μ_g is the viscosity.

2.1.1 Shock tube

We designed and built a special shock tube for studying the shock/droplet interaction. Fig. 2.1 shows a schematic of this specially designed shock tube facility,

consisting mainly of a shock tunnel, diaphragm, droplet generator, pressure supply, and observation windows.

The shock tube is made of a stainless-steel containing chromium and manganese and has high-strength resistance (2.5MPa). The overall inside length of the shock tube is 5450mm. The cross section is 80mm×80mm square. The lengths of the driver section and the driven section can be adjusted by switching the modular round tubes. Therefor this shock tube can provide at least 1 ms test time for all the preset experimental cases. A liquid dropper sits on top of the test section. Ball-bearing carriages that ride on two stainless steel rails support the entire tube. In addition to the tunnels, a boost control pump (SMC VBA11A, Japan, max pressure 20atm) and a vacuum pump (LEYBOLD DIVAC 1.2L, German) in the facility can run experimental cases with various Mach numbers (1.2~2.5) and Weber numbers (10^3~10^4). As shown in table 2.1, 16 preset running conditions should be able to be conducted in this shock tube. P_4 represents the pressure in the driver section, and P_1 represents the pressure in the driven section and test section. The variable v is the free stream velocity after the impact shock, ρ is the free stream density.

Generating a shock requires a quick rupture of the diaphragm. Cases which the pressure in the driver section is below 0.8MPa use a two-diaphragm system, due to its

repeat accuracy of higher than 90% under 1Mpa. For higher driven pressure, an electric bursting device can burst the aluminum foil diaphragm between the driver/driven sections.

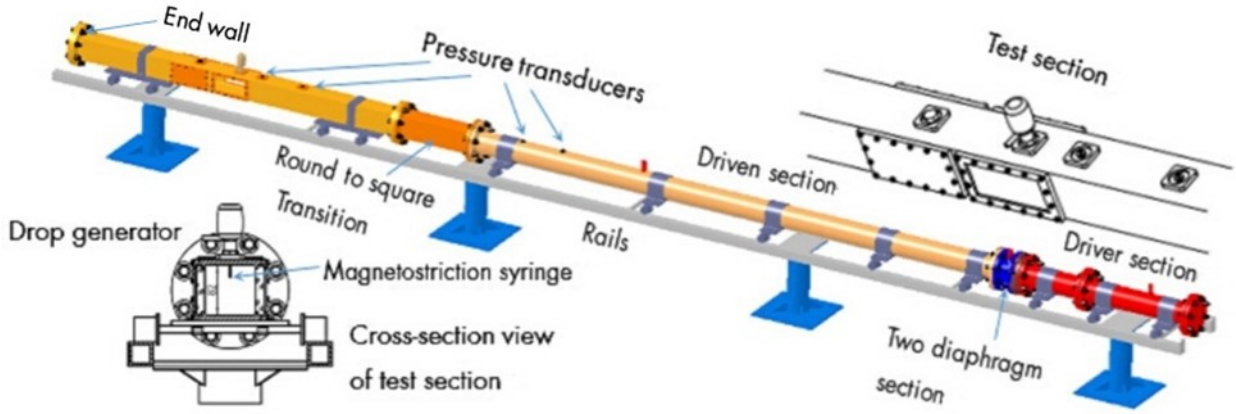


Figure 2.1 Schematic of the shock tube assembly.

We \ Ma	1.2 (P_4/P_1=2.35)				1.5 (P_4/P_1=7)				1.8 (P_4/P_1=18.4)				2.1 (P_4/P_1=45.3)				2.4 (P_4/P_1=108.7)			
	P_4(atm)	P_1(atm)	v(m/s)	ρ(kg/m^3)	P_4	P_1	v	ρ	P_4	P_1	v	ρ	P_4	P_1	v	ρ	P_4	P_1	v	ρ
3000	11	4.67	106	7.39	4.6	0.65	241	1.44	4.3	0.23	360	0.65	5.3	0.11	470	0.39	7.5	0.07	574	0.26
5000					7.6	1.09	241	2.38	7.2	0.39	360	1.09	8.7	0.19	470	0.64	12.3	0.11	574	0.43
7000					10.8	1.54	241	3.38	10	0.54	360	1.51	12.2	0.27	470	0.89	17.2	0.15	574	0.6
10000					15.5	2.21	241	4.86	14.3	0.78	360	2.16	17.5	0.38	470	1.28				

Table 2.1. The overall running conditions of the shock tube.

2.1.2 Electrical and Visualization Setup

Fig. 2.2 shows the schematics of the electrical outline and direct high-speed photography. First, a signal from the signal generator goes to the droplet generator, and then the signal generator sends a delayed signal to the diaphragm bursting device. The delay time is approximated by $\sqrt{D/g} - l/u_s - t_d$, so that the droplet falls to the center of

the shock tube when the shock impacts on it. Here, D is the diameter of the shock tube, g the acceleration of gravity, l the distance between the droplet and the diaphragm, u_s the shock speed, and t_d the diaphragm breakup time. Four piezoresistive pressure transducers that sit flush on the shock tube wall connect to a multi-channel signal conditioner and then a data acquisition system to record u_s and pressure behind the shock wave. The shock Mach number is then calculated basing on the record u_s and the gap between the piezoelectric transducers. The piezoresistive pressure transducers' signal triggers another delay generator, which then triggers the high-speed camera at a preset time before the impact of the shock on the droplet.

A Z-type schlieren system is used to verify the shock Mach number's accuracy before the running. Due to the rarely small size of the droplet, it will be too difficult to observe the wave pattern around the droplet. Refer to the visualization system in Theofanous 2011, a direct shadowgraph system containing two 150W LED light sources, a light diffuser, and a high-speed camera (Photron SA-Z, Japan) allows for flow visualization of the interfacial flow. With the effect of the scattered light from the light diffuser, this direct high-speed photography technology shows improved three-dimensional quality like the Planar laser induced fluorescence (PLIF) images in the early time of the droplet deformation. The sequential images have a resolution of 33.5 to 40

pixels/mm (110 pixels along the droplet diameter 2.5mm) and a frame interval of 20μs. Exposure time reaches as fast as 250ns..

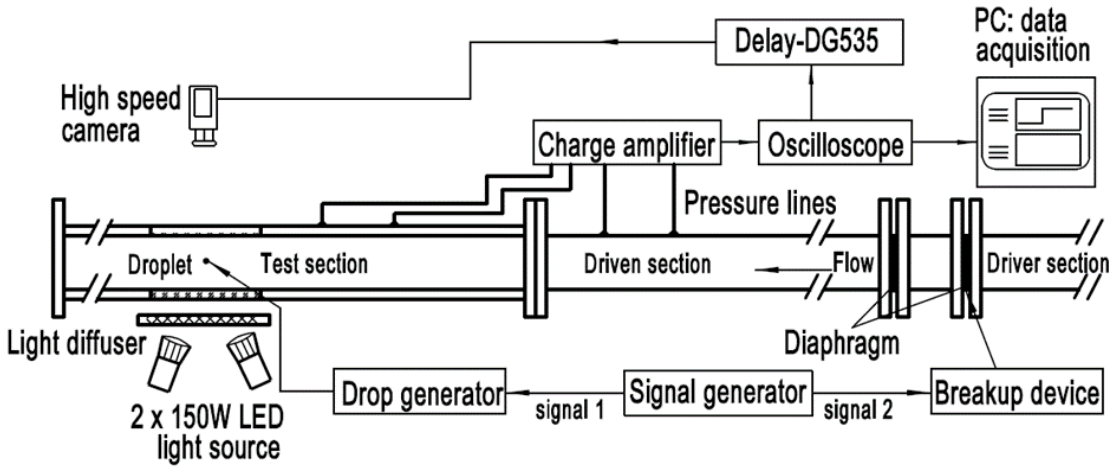


Figure 2.2 Schematics of the electrical outline and direct high-speed photography.

2.1.3 Creation of the Water Droplet

A specially designed injector device can inject a single drop of test liquid into the test section by controlling the injection volume. Fig. 3-3 shows a schematic of the droplet generator system. Details of the generation and control of water droplets are as follows. A syringe needle, connected to the solenoid valve, generates droplets, controlling the volume per drop and the dripping time. The water tank maintains the pressure of the water pipes, so that the repeat accuracy can achieve about 80%.

The typical droplet diameter is about 2.5mm. With further enlargement of the diameter, oscillations will occur on the droplet surface from the start. With the assistance of the vacuum pump in the text section of the shock tube, better spherical droplets with larger diameter can result.

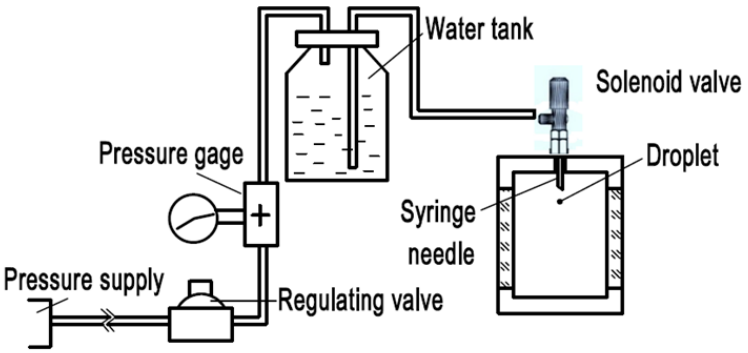


Figure 2.3 Schematic of the droplet generator system.

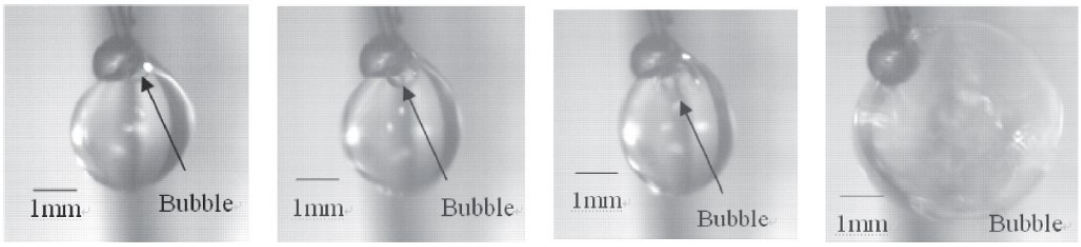


Figure 2.4 Bubble growth process (Liu et al. 2018).

In the present study, to experimental study how is the water-bubble affect the droplet's aerobreakup, depressurization of the air surrounding a water droplet (Liu et al. 2018) generates a droplet embedded with a vapor cavity (Fig. 2.4). Due to the existing experimental accuracy problem it is temporarily impossible to precisely control the liquid

flow rate and drop time, therefore the size and position of the vapor cavity inside the droplet is uncontrollable. Nevertheless, the process achieves satisfactory sphericity for the droplet and cavity. It is worth mentioning that the use of a diameter decreasing nozzle will facilitate the creation of droplets with vapor bubbles.

2.2. Numerical Methods and Validations

The experiment adopts an upwind space-time CE/SE (conservation element and solution element) method (Shen et al. 2015; Shen & Wen 2016; Shen et al. 2017) to simulate the process of a plane shock impacting with a spherical water droplet. This method has been well validated in capturing shocks and details of complex flow structures. Therefore, this chapter ignores the details of the method and provides only a brief description. For the computer programmer that interested in the CE/SE code, a sample code of using CE/SE method to simulation the deformation of the droplet after the shock wave (Ma=2.4, axisymmetric droplet embedded with a vapor cavity) is listed in the appendices. In this method, an upwind procedure determines the numerical flux through the interface of two different conservation elements. The upwind procedure breaks the space-time inversion invariance so that it can be directly applied to capture discontinuities without spurious oscillations. It has a second order precision in both temporal and spatial scales

and employs two-dimensional rectangular meshes.

2.2.1 Modeling Geometry and Problem Description

The problem geometry, as shown in Fig. 2.5, is a two-dimensional rectangular domain. The model simulates only half the water droplet as the experimental visualizations show no significant asymmetries along the test chamber's centerline. Hence, an axial symmetry plane boundary condition is set at the domain's bottom, and non-reflection wall boundary conditions are set at the remaining three boundaries. The width of the computational Region is 10R, where the R is the radius of the droplet. L_1 and L_2 are 15R and 20R, respectively. The mesh for the domain has a uniform grid of 7000×2000 points, with a grid resolution of 400 points along the water droplet diameter. In one case, while initializing, the entire domain is at rest with the following initial conditions:

$$\gamma_g = 1.4, \pi_g = 0, \gamma_l = 1.932, \pi_l = 1.1645 \times 10^9, M_s = 1.52$$

$$\rho_g = 0.6kg/m^3, P_g = 5000Pa, \rho_l = 1000kg/m^3, P_l = 5000Pa \quad (2.2)$$

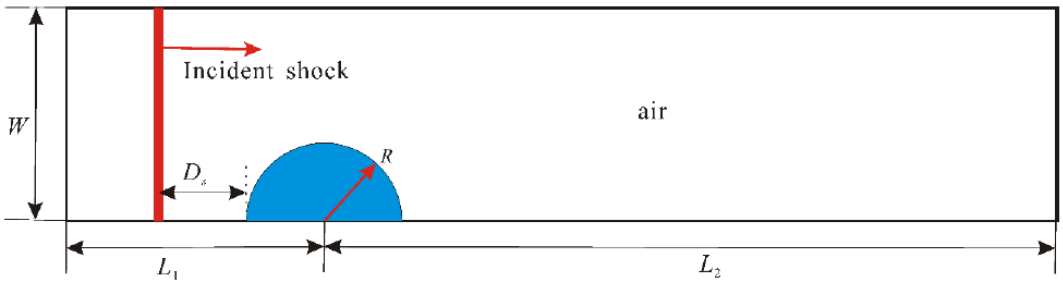


Figure 2.5 The schematic of the initial setup for the shock-water droplet interaction.

2.2.2 Equation of state

A solution to express the volume fraction-based five-equation model illustrates the gas/liquid interface evolution as follows:

$$\frac{\partial \alpha_i}{\partial t} + V \cdot \nabla \alpha_i = 0, \qquad i = 1,2$$

$$\frac{\partial \rho_s \alpha_s}{\partial t} + \nabla \cdot (\rho_s \alpha_s \mathrm{V}) = 0, \quad s = 1,2$$

$$\frac{\partial \rho V}{\partial t} + \nabla \cdot (\rho V \otimes V + p) = 0$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \big(V(E + p)\big) = 0 \tag{2.3}$$

where the α_i denotes the volume fraction of gas i, ρ_s the density of component fluid s, ρ the density of the mixture, V the velocity vector, p the pressure and E the total energy.

The stiffened gas equation of state (EOS) is adopted to close the system:

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho V \cdot V\right) - \gamma\pi \tag{2.4}$$

where,

$$\frac{1}{\gamma - 1} = \sum \frac{\alpha_i}{\gamma_i - 1} \quad and \quad \frac{\gamma\pi}{\gamma - 1} = \sum \frac{\alpha_i \gamma_i \pi_i}{\gamma_i - 1} \tag{2.5}$$

The total density and the sound speed of the mixture can be respectively calculated as follows:

$$\rho = \sum \alpha_i \rho_i \quad and \quad c = \sqrt{\gamma(p+\pi)/\rho} \tag{2.6}$$

Solving the five-equation model uses a maximum-principle satisfying upwind CE/SE scheme that proves capable of capturing the contact interface without obvious diffusion. The HLLC approximate Riemann solver is employed to get the numerical fluxes between the conservation element. This numerical method proves to be perfect in the numerical conservative properties in both space and time and accurate in capturing shock and contact discontinuities.

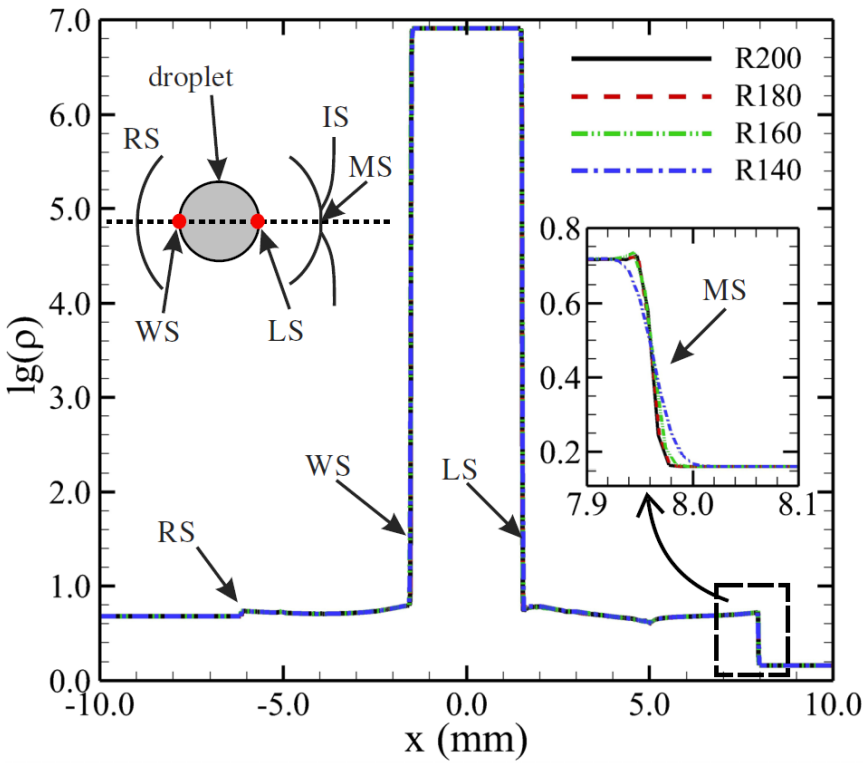


Figure 2.6. Grid convergence test showing density distributions at 20 μs after the incident shock touches the droplet with four different grid sizes (140, 160, 180, and 200 grids per droplet radius).

2.2.3 The Independence of the Computing Grid

First, we tested the grid convergence according to the density distributions $lg(\rho)$ along the axis of symmetry, as shown in Fig. 2.6, with four different grids arranged within the length of the droplet radius. The terms R140 ~ R200 denote that there were 140 ~ 200 grids arranged per droplet radius. The left inset presents a sketch of the droplet/shock-wave system after the shock-wave impact, within which RS denotes the reflected shock, IS is the incident shock, MS is the Mach stem, WS is the windward stagnation point, and LS is the leeward stagnation point. The dotted line indicates the axis of symmetry. The right inset presents an enlarged MS area (dashed square area). Clearly, the larger the grid number is, the sharper the pressure change across the MS shock front is. The shock fronts of cases R160, R180, and R200 nearly collapse with each other, which indicates good grid convergence. Throughout, this study adopts a mesh size of 200 grids per droplet radius. In a standard example of using the CE/SE method to simulate the droplet fragmentation and deformation process from the start of the incoming shock wave impact the droplet to the 160 μs instant that after the passage of the shock wave, the simulation takes about 70 to 80 hours to simulate the process using a mesh size of 200 grids per droplet radius on 64 intel Xeon E7-4850 v4 computing cores.

2.2.4 Validation Cases

2.2.4.1 Qualitative Analysis of Shock/Water-Droplet Breakup

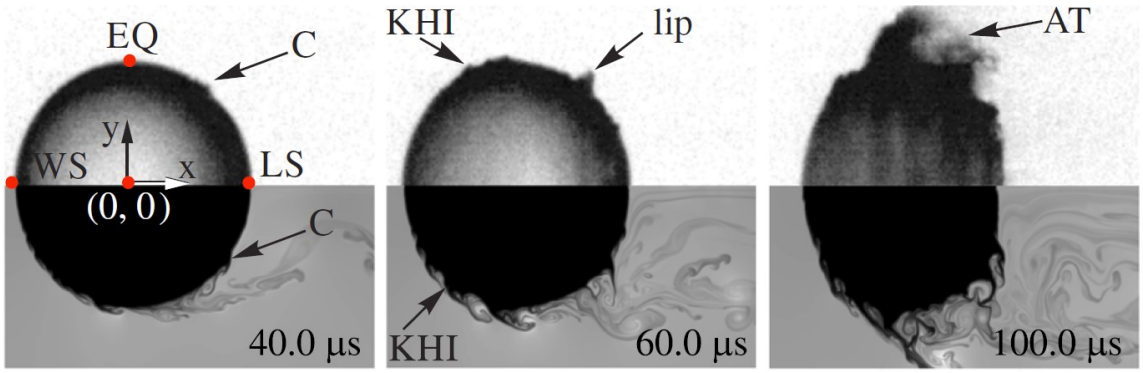


Figure 2.7 Comparison of the numerical and experimental results at different instants (upper part: experimental images, lower part: numerical results).

The numerical CE/SE method first simulates the experiment conducted by Yi et al. (2017) using a droplet diameter d_0 of 3.03 mm and an incident shock Mach number M_s of 1.39. Fig. 2.7 shows a comparison between the numerical results and the experimental images, selecting three distinct instants after the shock impact ($t = 0$). At $t = 40\mu s$, the experimental image shows tiny corrugation (C) at the leeward surface of the droplet between the equator (EQ) and the LS, whereas the rest of the surface remains intact. At this moment, a Kelvin–Helmholtz instability (KHI) develops at the windward surface in the numerical result but not in the experimental result. The reason for this difference derives from ignoring the surface tension term in the present numerical method. In the

experimental images, the first tiny corrugation grows at $t = 60\mu s$ and turns into the "lip." The KHI amplitude is another element. The KHI amplitude is a direct response to the shear velocity distribution, which increases from the WS to the EQ. At $t = 100\mu s$, although the main body of the droplet remains spherical to a certain extent, the protrusions are distinct, and atomization (AT) occurs obviously at the tips of these protrusions. In the in-house CE/SE numerical simulation method, because the continuous model with the volume fraction-based five-equation is used, it is possible to use different volume fractions (e.g., α_{water}=0.5) to distinguish between AT and droplet bodies. The capability of the present numerical method, according to this comparison, proves to be fairly good.

2.2.4.2 Qualitative Analysis of Shock/Water-Column Breakup

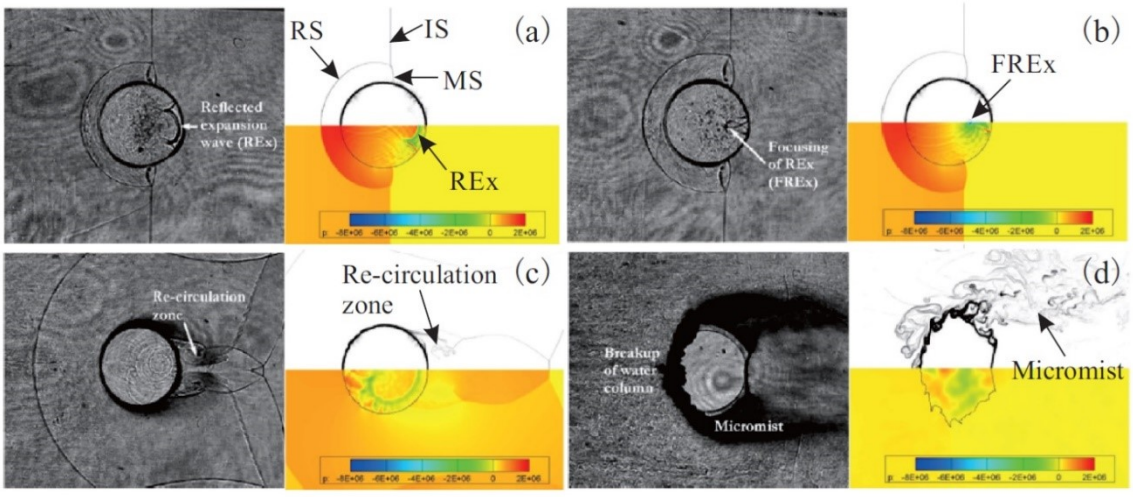


Figure 2.8 Sequential images of water column deformation and breakup for Mach 2.4 at t*=0.39 (a), 0.44 (b), 1.87 (c), 9.51 (d). The left shadow graphs of each sub-figures are experimental images of Sembian et al.(2016) the right upper half and lower half are numerical density gradient and pressure contour respectively.

Fig. 2.8 shows that simulating the shock/water-column interaction (Sembian et al. 2016) validates the numerical method. The zero time is the instant of shock impact, and the dimensionless time t* is defined as follows:

$$t^* = tu_g/D \tag{2.7}$$

The figure presents numerical density gradient (right-upper) and pressure distribution (right-lower) in comparison with the experimental image (left) at four distinct instants.

The incident shock propagates from left to right and impinges on the water column. At t*=0.39, the incident shock (IS), reflected shock (RS), and Mach stem (MS) intersect in a triple point, and Fig. 2.8(a) clearly shows the reflected expansion wave (REx) within the water column. Due to the acoustic impedance mismatch between the air and water, waves propagate much faster in the water than that in air. The REx focuses at a single point (FREx) immediately at t*=0.44 while the incident shock moves a very short distance in Fig. 2.8(b). After the passage of the incident shock, the flow field builds and a recirculation zone forms at the leeward side of the water column at t*=1.87 in Fig. 2.8(c). And finally, Fig. 2.8(d) presents the water column breaks up in a striping mode at t*=9.51. Images in Fig. 2.8 validate the present numerical method in capturing both wave structures and contact interfaces. Thus, the following study can proceed based on the present method.

3. Investigation of Aerodynamic Breakup of a Sphere Liquid Droplet behind a Shock Wave

Normal shockwaves have little effect on the droplet. However, these shock could generate a subsonic or supersonic flow around the droplet. The droplets will then deform and break up in the shock wave-induced gas flow. At present, a large amount of attention in the literature focuses on several typical fragmentation modes and their transformation patterns with respect to each other. In this chapter, based on experimental and numerical simulations, the effects of the external airflow and the internal flow of the droplet on the deformation of the droplet will be described in detail.

3.1. Outer airflow effect on droplet deformation

3.1.1. Detailed Shock/Droplet Interaction around the Interface

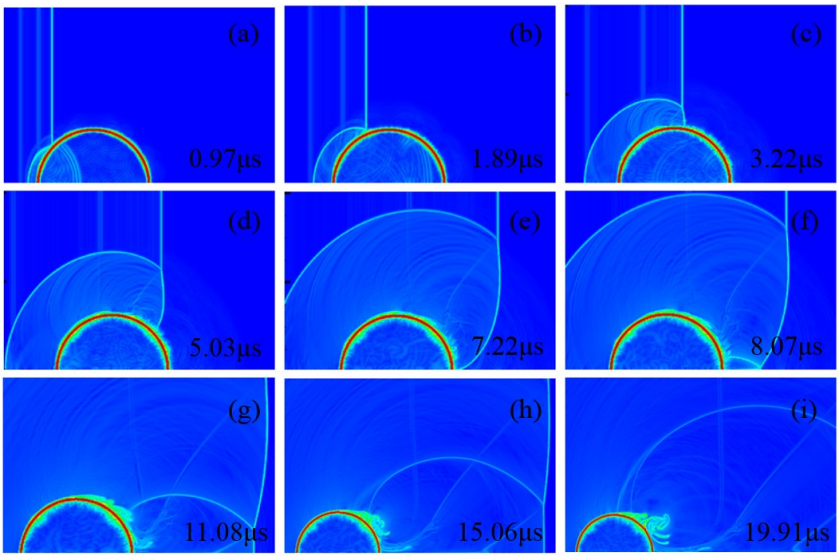


Figure 3.1 Numerical density gradient for a 2.5mm water droplet hit by a Mach 1.52 incident shock wave (W_e=3000).

Fig. 3.1 shows the process by which the shock wave is bypassed near the droplet, where t=0 is defined as the moment when incident shock and the droplet are about to impact. The duration of this phase is relatively short, usually less than 20 microseconds. Under the conditions of this thesis, no visible deformation of the droplet surface occurs during the bypassing phase of the shock wave. The shock wave impacts the windward side of the droplet and reflection occurs. As the wave propagates downstream, the angle of incidence increases, and near the middle of the windward surface, the reflection type changes from regular to Mach reflection (Fig. 3.1(c)). The development of the bow-shaped reflection wave is relatively simple: the intensity decreases as it propagates outward. In the preferred computational domain, the reflected excitation wave acts on the droplet attachment region after it is re-reflected at the wall, but its intensity is so low that its influence can be neglected. The expansion wave generated by the post-wave flow through the bent interface act on the Mach rod and bend it. The bent Mach steam, after passing through the region near the equator of the sphere, converges on the leeward side towards the backward stationary point. The Mach reflection waves converge and intersect at the backward standing point (Fig. 3.1(e)), creating extremely high pressures and temperatures in this region. The bypassing wave then travels upstream over the surface of the sphere, decreasing in intensity until it is finally dissipated. During the intersection of the bypassing waves, a Mach steam is formed at the intersection of the waves downstream of the rear stationary point and propagates downstream.

The reflected wave of the Mach-Mach collision sweeps across the entire surface of the droplet (Fig. 3.1(i)). Although there is a large pressure gradient at the moment and an extremely high acceleration peak is created, this peak acceleration does not stop at a fixed area of the droplet, but sweeps across the entire droplet, so there is no significant deformation of the droplet. In fact, droplet deformation during excitation is generally negligible and when the excitation ends and a stable ambient airflow field is formed, it is the constant pressure exerted by the ambient airflow on the surface of the droplet that is the primary cause of the droplet's degeneration.

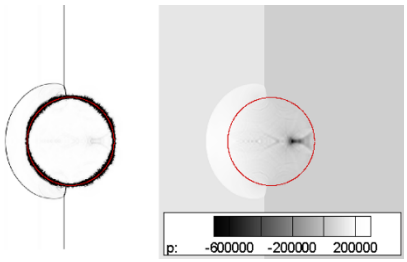


Figure 3.2 Numerical density gradient (left) and pressure contour (right) at t=2.5μs (t*=0.25).

The transmitted shock wave will be produced in the droplet when the incident shock impact the droplet, when this transmitted shock reaching the downstream water-air interface, the transmitted shock gets reflected as an expansion wave, since acoustic impedance $Z_{water} > Z_{air}$. Due to the droplet's downstream concave boundary the reflected expansion wave focuses at a point, (Fig. 3.2) creating "negative pressures." By negative pressure, we mean that the absolute pressure is below 0.

3.1.2. Pressure Distribution Effect on Droplet Deformation

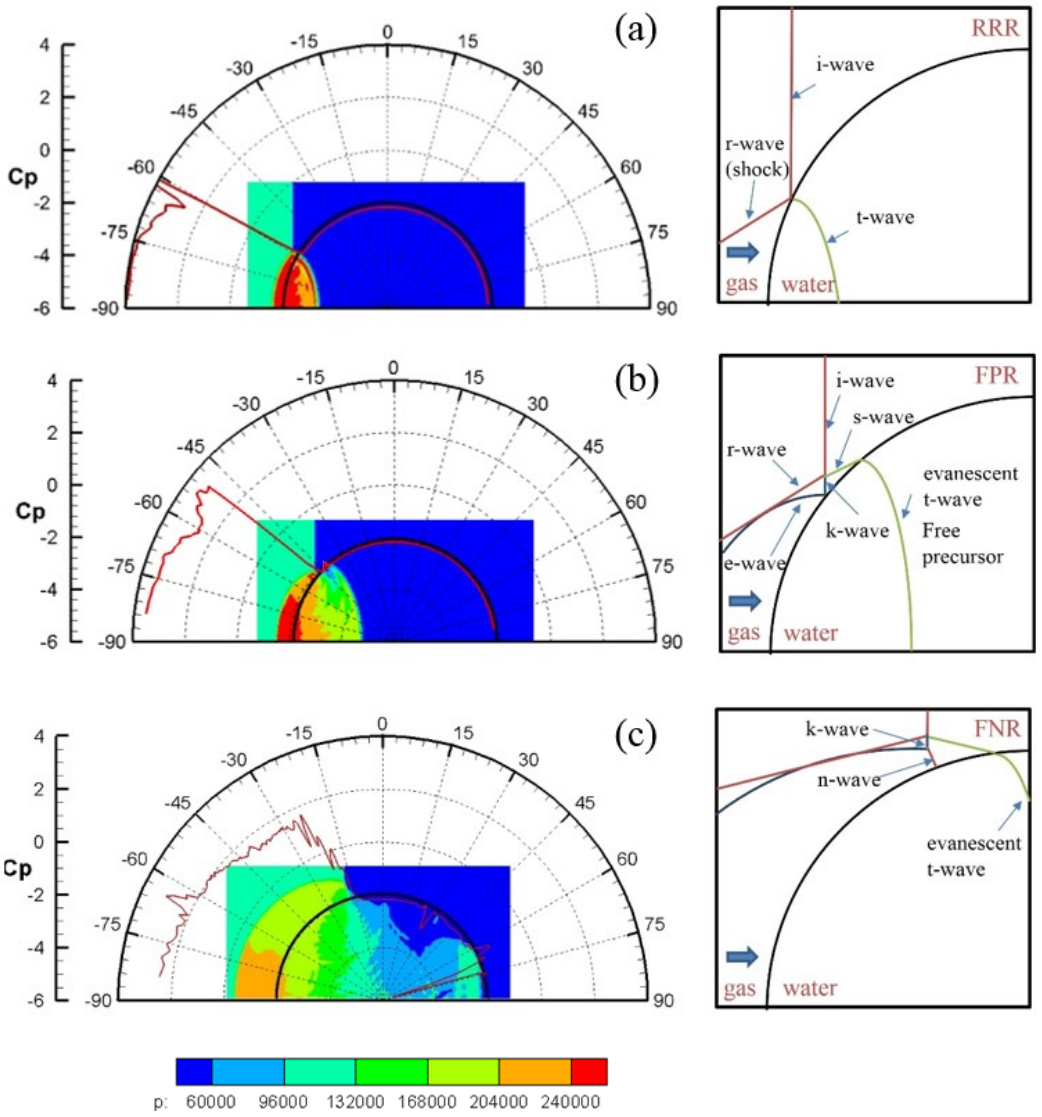


Figure 3.3 Pressure distributions on the droplet's surface at 0.69μs, 0.97μs and 2.03μs after the shock impingement (a) and the corresponding shock wave refraction patterns (b) (α_{water}=0.5). Pressure distribution effect on droplet deformation.

Two mechanisms can be responsible for the formation of the lips during the droplet deformation behind the shock wave. One is the squeezing and sucking effect of the nonuniform pressure field, which is referred to as the pressure mechanism in this study. And the other is the

accumulation effect of the shear induced surface flow, which is referred to as the shear mechanism. The action of pressure mechanism can be reflected by the pressure distribution. To analysis the pressure distribution, dimensionless pressure factor $C_p = \frac{p-p_\infty}{p_d}$ is introduced, where P_∞ is the free stream gas pressure, P_d is the flow dynamic pressure. If C_p is above 0, it means the area is under a squeezing effect, and vice versa.

Fig. 3.3 shows the pressure distributions on the droplet's surface at 0.69μs, 0.97μs and 2.03μs after the shock impingement and the corresponding shock wave refraction patterns: RRR, FPR, FNR. As for local sound speed, the shock/droplet interaction problem is a slow-fast, air-water interface case. Fig. 3.3(a) presents the RRR (regular refraction with reflected shock) shock refraction pattern. In this pattern, the transmitted t-wave is always ahead of the incident i-wave, and it moves along the interface at the same velocity. Figure 3.3(b) presents the FPR (free precursor reflection) shock refraction pattern. The t-wave breaks loose from the i-shock and reflected r-waves, running significantly ahead along the interface. The t-wave is a evanescent wave, which is refracted back into a side, s-wave. The s-wave interacts with the i-wave, transform into a k-wave, which is reflected as a centered expansion e-wave. The e-wave interacts with the r-wave, causing weaken each other. The local C_p has a small peak between the i-wave and t-wave on the interface. Figure 3.3(c) presents the FNR (free precursor von Neumann reflection) shock refraction pattern. The different between FNR and FPR is the formation of a weak Mach reflection n-wave.

Fig. 3.4 presents the pressure distributions on the droplet's surface at different instants in the early stage after the shock impingement at Ms=1.52 and We=3000. Compared with wave movement in Fig. 3.1, show the wave effect on the droplet deformation.

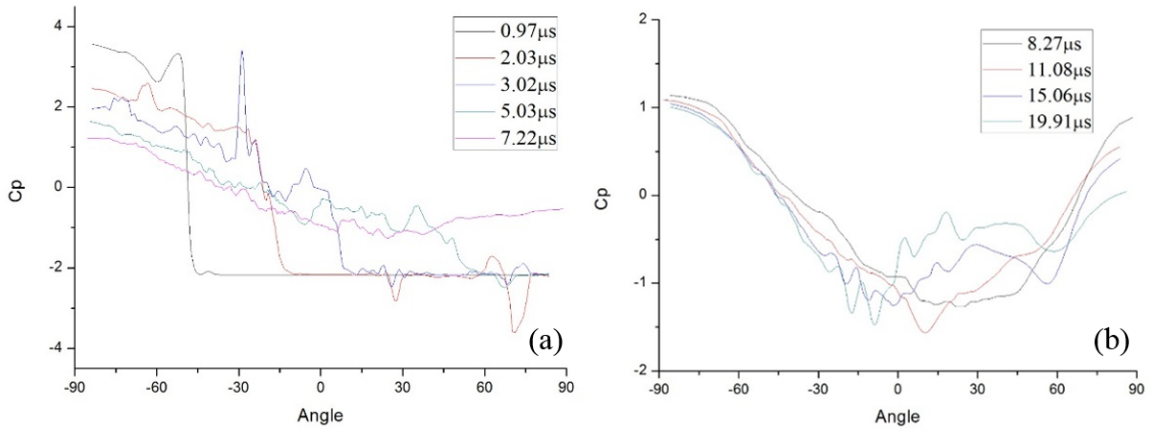


Figure 3.4 Pressure distributions on the droplet's surface at different times in the early period after the shock impingement at Ms=1.52 and We=3000 (α_{water}=0.5).

Fig. 3.5(a) presents the pressure distributions on the droplet's surface at different time after the shock impingement along with the vortex separation development at Ms=1.52 (α_{water}=0.95), with -90˚ indicating the windward stagnation point and 0˚ the equator. Low pressure trough on the C_p graph means that special areas are under sucking effect, which may lead to formation of lips on the droplet's surface. It can be observed that at 40μs, four low pressure trough angles on the droplet's surface at -18°, -1°, 23°, 45°. However, till 100μs only the -12° and 25° low pressure trough angles become the main force that control the deformation of the droplet. It coincides well with the experimental results.

Fig. 3.5(b) shows the numerical Schlieren of spherical droplet's deformation a t*=9.8. It shows clearly that the liquid bulges/rings angles on the leeside water surface at t*=9.8 (-1.2°, 25.2°, 45.2°) are almost the same as low pressure peaks angle on the droplet's lee surface at t*=4 (-1°, 23°, 45°). After the votex separation fully developed, the four low pressure areas on the lee surface of droplet will generate liquid rings later. And different from the windward surface, the liquid bulges/rings on the leeside droplet surface will maintain the same angles for a period.

As shown in Fig. 3.5(b), In the early stages of breakup, the liquid mist is produced in two main regions: near the equator and at the top of the leeward surface liquid ring. In the former, the direction of spraying is the same as that of the ambient airflow, while in the latter the direction of spraying is opposite to that of the ambient airflow. Of the two, generally speaking, the amount of liquid fog generated near the equator is much higher than that of the leeward liquid ring, which is the main component of the early liquid fog. In the middle and late stages of droplet fragmentation, due to the flattening of the droplet body, the two areas of liquid mist generation will gradually approach each other.

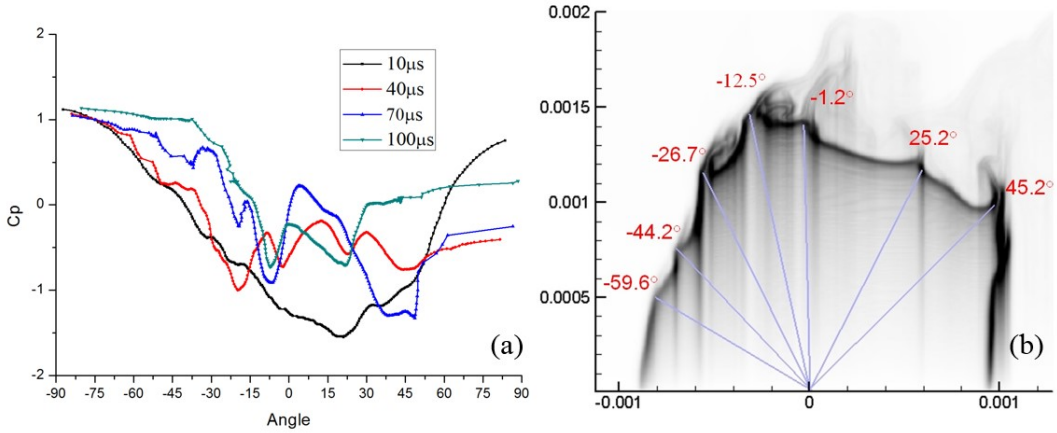


Figure 3.5 Pressure distributions on the droplet's surface at 10μs, 40μs, 70μs and100μs after the shock impingement (a) and numerical Schlieren of spherical droplet's deformation at t=100μs (b) (t*=10) at Ms=1.52 and We=3000 (α_{water}=0.5).

3.1.3. Qualitative Analysis of Water-Droplet Breakup

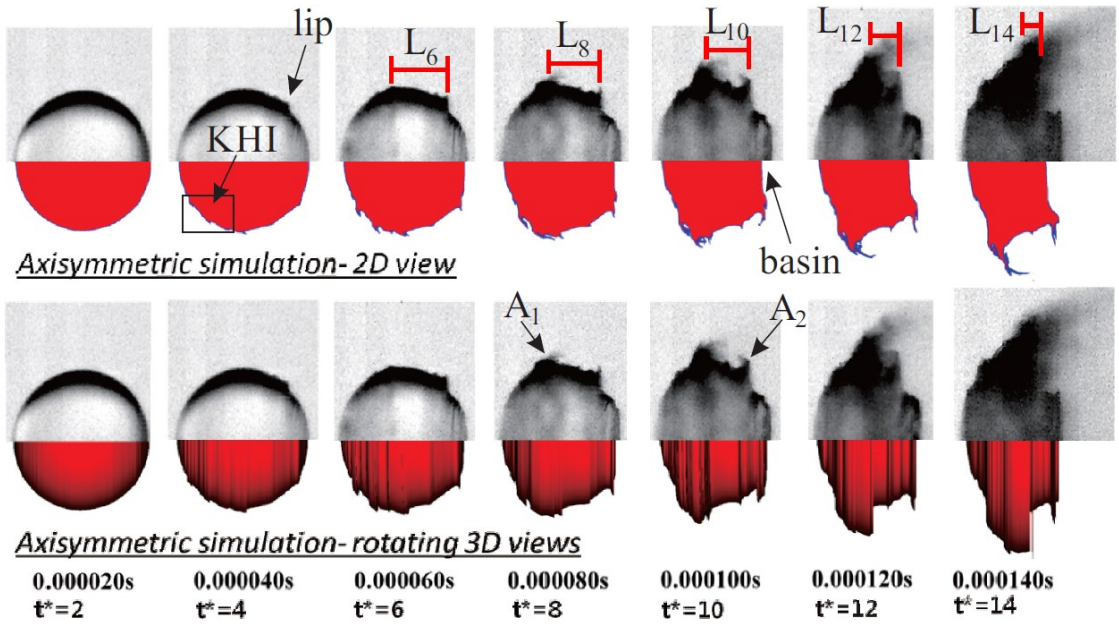


Figure 3.6 Sequential images of spherical drop deformation and breakup at Mach 1.52 and W*e*=3000 and comparison with the corresponding axisymmetric CE/SE simulation (α_{water} ≥0.05).

Fig. 3.6 shows comparisons of sequential experimental images of a spherical drop deformation and breakup for a typical case with shock M_a=1.52 and W_e=3000 and the corresponding numerical results. The comparisons are made in two ways: each lower half image in the upper row presents a 2D view where the simulation results represent a middle slice of the water droplet, and each lower half image in the lower row presents a 3D view where an iso-surface with water volume fraction 0.05 represents the air/droplet interface. As shown in Fig. 3.6, the entire deformation and breakup process can be well simulated. In this case, the incident shock wave propagates from left to right, and the starting time is the instant when the incident shock wave impinges on the liquid surface at the windward stagnation point. At t* = 2, the incident shock wave passes the liquid droplet, and the flow field is initially established. At this early moment, the droplet can be deemed a solid sphere where a perfect spherical form is maintained. At t* = 4, the lip structure forms at the leeward surface of the droplet, and shear induced ripples form at the windward surface (marked as KHI which is caused by the Kelvin−Helmholtz instability). However, both the windward and leeward stagnation areas maintain smooth surfaces. The size of lip and the shear induced ripples grow with time, and the width of the droplet in the transverse direction compresses (t* = 6 ∼ 14, see also Fig. 3.7(a)). The mark of breakup occurs at t* = 8 when atomization first happens at the protrusion near the droplet equator (denoted as

A_1). The second atomization point occurs at the lip tip (t* = 10, denoted as A_2), although the lip appears first. It is interesting to note that the atomized lip tip (A_2) is left pointing. As time elapses, the distance between the first atomization point (A_1) and the second atomization point (A_2), labeled as L, becomes shorter. The figure illustrates this change as $L_6 < L_8 < L_{10} < L_{12} < L_{14,}$ where the subscripts denote the time. This indicates that the flow direction at the left and right sides of the droplet are opposite. The opposite flow direction enhances the formation of the vortices at the lateral side, which benefits the stripping breakup process. From the instant t* = 8, a basin-shaped leeward surface emerges and its depth increases. Fig. 3.7(a) presents the interface evolution history. The development of the basin depth leads to further droplet lateral growth, making the striping phenomenon even more severe.

Figure 3.7(b) presents the vorticity distribution of the flow field at the early stage (t = 20μs). After the incident shock passage, the vorticity that results concentrates mainly in a region near the droplet surface. At the windward side of the droplet, the vorticity distribution remains confined on the droplet surface except in the area near the stagnation point where the angle between the fluid flow and the surface normal direction is small.

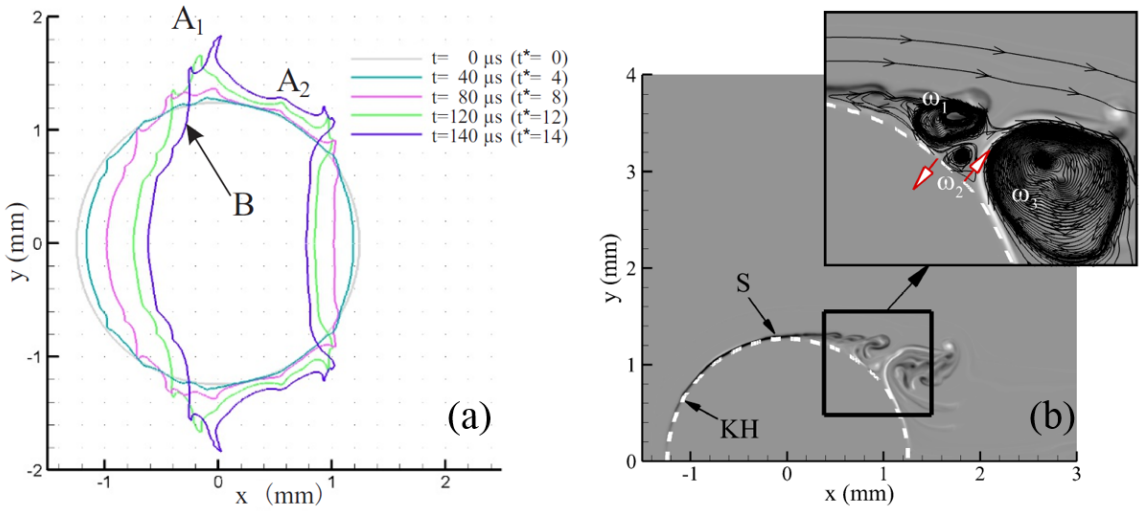


Figure 3.7 Deformation history of the water droplet surface by the CE/SE simulation (a) at Ms=1.52 and We=3000, and vortices distribution (b) at t=20 μs (t*=2) in Figure 4-2. S denotes the separation point, KH the area where the KH instability emerges. ω_1, ω_2 and ω_3 denotes three distinct vortices. The white dashed line represents the air/water interface (volume fraction of heavy fluid 95%). Light color indicates positive vorticity and dark color indicates negative vorticity.

The separation occurs near the droplet equator (S). In this case, the vorticity acts as an index of flow direction. It shows that the air flow separates tangentially near the droplet equator. At the droplet windward surface out of the stagnation area, an interface protrusion occurs (KH), which is induced by the K-H instability. The vorticity at the leeward stagnation area is as weak as the windward stagnation area. However, the vorticity distribution between the separation point and the leeward stagnation point is complex and fascinating. The inset of Fig. 3.7(b) enlarges the complex part of the vorticity and illustrates streamlines. The streamlines out of the separation region are regular while the streamlines

in the separation region are twisted. In the separation region, three distinct vortices form, labelled ω_1, ω_2, and ω_3 separately. This flow structure is in accordance with the research of Theofanous et al. (2012) and Meng et al. (2013). The three vortices rotate in different directions: ω_1 clockwise, ω_2 anti-clockwise, and ω_3 clockwise. These vortices locate closely at the droplet surface and force the surface to deform. As the arriving shock wave passes through the equator, negative vorticity ω_2 is generated by the baroclinic vorticity term, $\frac{1}{\rho^2}\nabla\rho \times \nabla p$, which relying on the gas flow after the shock wave to propagate downstream. The surface area between ω_1 and ω_2 is pressed to "sink" into the droplet, while the surface area between ω_2 and ω_3 is "pulled up" into the air (red arrows depict the deformation trend). The clockwise rotating vortices ω_1 and ω_3 are the reason why the protrusion A_2 atomization happens windward. And the distance decreasing between A_1 and A_2 has an explanation. Fig. 3.7 shows that the outer air flow field of the shock/droplet interaction builds up at a very early stage. There is an obvious hysteresis of the droplet deformation after the flow field has built up.

In the numerical results, the liquid protrusion in the middle of the leeward droplet surface is slightly weaker than the experimental images. The main reason is that the numerical code does not take the surface tension variation into account. As the shock wave

passes by, the surface tension becomes smaller with the increase of gas temperature. In that case, the small liquid protrusion easily extends into lips.

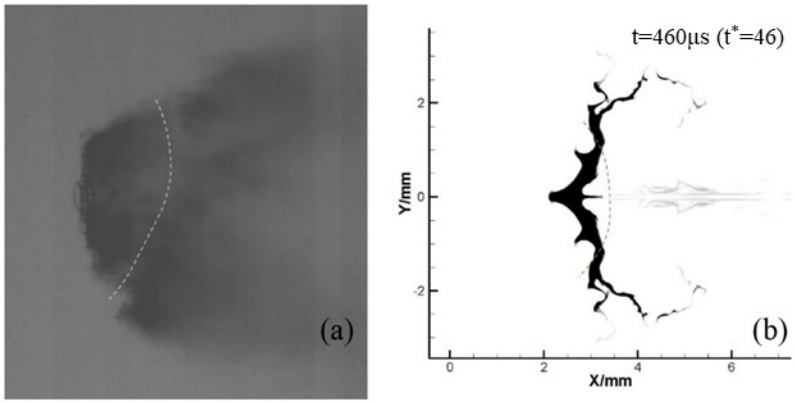


Figure 3.8 Breakup of the droplet hit by Ms=1.52 shock at t=460μs (t*=46) (left is the experimental photo, right is the numerical density gradient). ($\alpha_{water} \geq 0.05$).

Fig. 3.8(a) shows the deformation and breakup of the droplet hit by Ms=1.52 shock at t=460μs(t*=46), clearly revealing a discontinuous gap in the atomization cloud. Shown in Fig. 3.8(b), this discontinuous gap may be due to the development of instability on the windward surface. The B area in Fig. 3.7(a) continues to cave during the flattening of the droplet, and then the outer atomization cloud will detach from the main drop body at a certain time.

3.2. Inner flow effect on droplet deformation

3.2.1. Internal Flow Field Description

Previous work has seldom considered the inner flow of the water droplet in the shock/droplet interaction because of its low flow speed. However, there is no doubt that the droplet deformation is a combined action of the inner and outer flow of the droplet. It is incomplete to interpret the deformation only from the perspective of outer flow. Fig. 3.9 presents the inner flow field of the water droplet by illustrating the pressure distribution as well as the streamlines at a series of instants. When the incident shock impinges a spherical droplet, Mach stem emerges at the droplet surface, and then it diffracts (DS) at the leeward surface; see (a), 5 μs. At this moment, the much faster transmitted shock has already built up the inner flow field. Streamlines shown within the droplet coincide with the precursory transmitted shock in the droplet. At instant (b) 7 μs, the inner flow basically remains intact, except for the minor change effected by the diffracted shock (DS). At (c) 9 μs, the DS near the leeward stagnation point forms a high pressure zone, and re-diffracts to the windward. This high pressure zone is vital to the inner flow field. It produces a negative speed in the x direction, and a half saddle point (HS) is formed at the axis of symmetry. The windward-moving re-diffracted shock imposes direct disturbance only at the leeward stagnation area.

The vortices formed near the leeward surface well "protect" the droplet surface from disturbance of any shock wave (d, 11 μs). After its formation, the half saddle point HS moves windward until the moment (e), 20 μs. From 20 μs on, the location of HS stays steady at approximately a fixed point (x =0.4∼0.5 mm), and the droplet can be considered as two parts divided by the streamlines emitted from the HS point. It also is worth noting that, up to this time (e, 20 μs), the droplet surface contour (the white solid line) remains a perfect circle. Also at this time, both the outer and inner flow fields have built up without the droplet deformation. A prediction of droplet deformation rests on the inner flow: two protrusions will emerge at the droplet surface, one at the intersection of droplet surface and the streamlines emitted from the HS point (i.e. A_2) and the other at the leeward surface of the droplet surface near the largest vortex (i.e. A_3). Apparently, these two protrusions form via different mechanisms: the inner fluid flow driven by the high pressure zone at the leeward stagnation point forms A_2, while the outer fluid flow driven by the vortex forms A_3. The deformation at (f, 50 μs) proves this prediction. In addition, the third protrusion (A_1) is visible near the droplet equator at this moment; it is formed mainly by the outer low pressure zone. As time elapses, the droplet deforms obviously (g, 100 μs) in both the lateral and transverse directions, but the HS point remains nearly still (h, 140 μs).

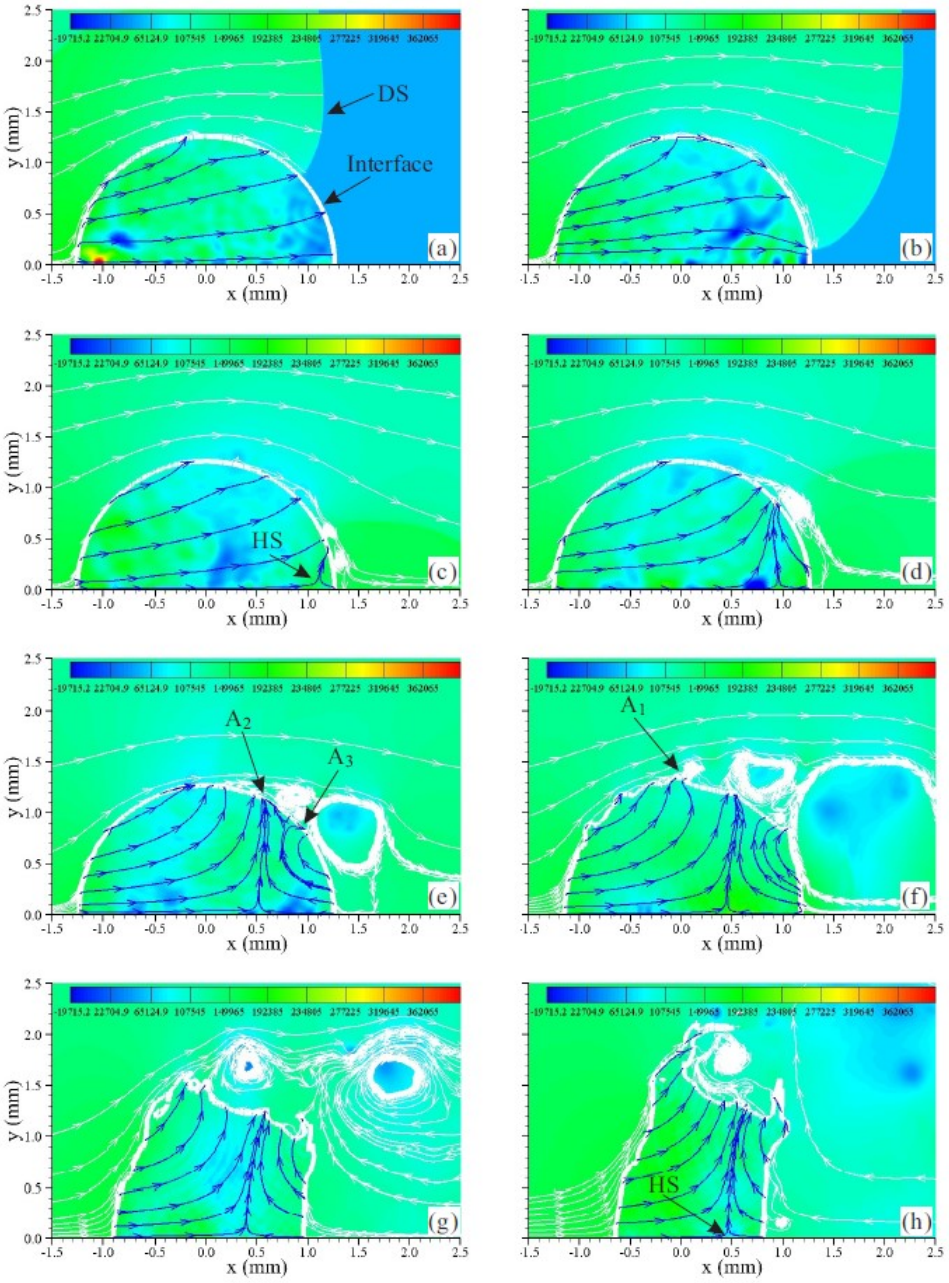


Figure 3.9 Numerical pressure distribution and streamlines at (a) 5 μs, (b) 7 μs, (c) 9 μs, (d) 11 μs, (e) 20 μs, (f) 50 μs, (g) 100 μs and (h) 140 μs. The white solid line denotes the air/water interface, DS the diffracted shock, and Bif the bifurcated point.

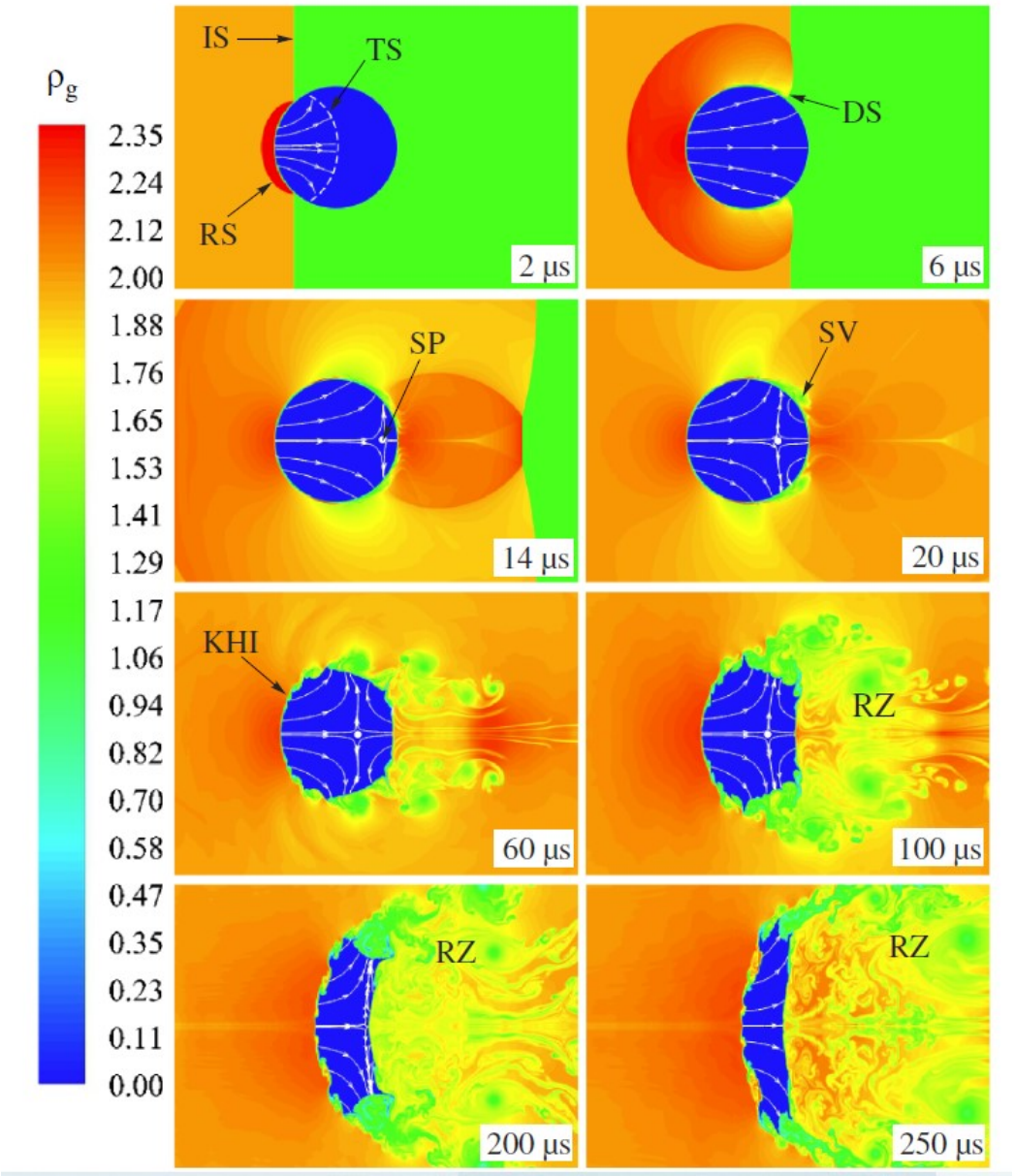


Figure 3.10 Numerical airstream density distribution and the internal flow streamlines of Yi et al.'s case.

This study then employs the numerical CE/SE method to simulate the experiment conducted by Yi et al. (2017), using a droplet diameter d_0 of 3.03 mm and an incident shock Mach number M_s of 1.39. Fig. 3.10 depicts the flow field build-up process of this

shock–droplet interaction by the air density contours. The white solid lines with arrows illustrate the temporal streamlines. These streamlines are truncated to focus only on the liquid flow inside the droplet rather than on the outer airflow and the interfacial boundary flow. After the incident shock (IS) impacts the windward surface of the droplet at $t = 2\mu s$, the reflected shock (RS) forms and propagates upstream, whereas a transmitted shock (TS, shown by the white dash line) propagates inside the droplet, which is much faster than the incident shock in air. Before the diffracted shock (DS) collides at the droplet LS point, the droplet internal flow keeps developing, induced by the internal diverging TS ($t = 6\mu s$) when all of the streamlines within the droplet are pointing in the downstream direction. At $t = 14\mu s$, high pressure forms in the LS area because of the DS collision. Accordingly, this high pressure induces the LS and the part of the internal liquid close to the LS to flow upstream. As a result, there must be a point inside the droplet on the axis of symmetry where the velocity is zero to balance the downstream and upstream liquid flow momentums. Observed for the first time, a saddle point (SP, labeled as the white dot) forms. In the following instants, although the shedding vortex (SV), KHI, and chaotic recirculation zone (RZ) form in sequence ($t = 20, 60, and\ 100\mu s$), the SP keeps its position nearly unchanged. This shock–droplet interaction process shows that the forming of the internal flow field is in quite a short time after the incident shock sweeps over the droplet. The

existence of an SP suggests that the droplet is suspended in air relative to the SP position, before the free stream tears apart and blows downstream the droplet.

Late-stage images at $t = 200\ and\ 250\mu s$ show where the droplet is severely deformed into a crescent shape. At $200\mu s$, although the droplet experiences severe deformation, the internal flow field maintains the same pattern as before. The distance between the windward stagnation point (WS) and leeward stagnation point (LS) decreases, and the SP nearly touches the LS. At $250\mu s$, the SP disappears, and all of the streamlines point in the downstream direction. No longer suspended, the whole droplet drifts downstream.

To delineate the relative positions of the WS, LS, and SP, Fig. 3.11 records the trajectories of these three points. The distance between the WS and LS illustrates that the droplet becomes narrow in the streamwise direction. Interestingly, the SP trajectory remains steady after the formed internal flow field (at $\mathrm{t} = 25\mu s$) and moves toward the LS when the droplet deforms severely (at t= $180\mu s$). After the trajectories of the LS and the SP intersect, the LS trajectory stops moving upstream and turns downstream instead.

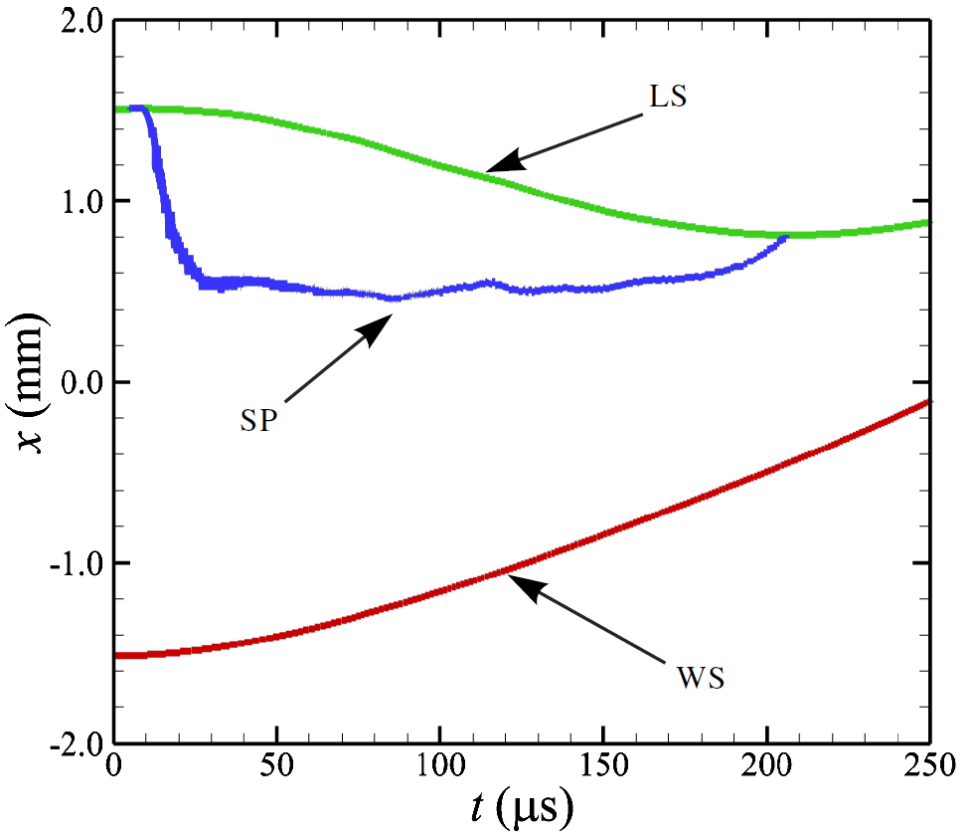


Figure 3.11 Trajectories of windward stagnation point (WS), leeward stagnation point (LS), and saddle point (SP) of Yi et al.'s case.

More numerical cases examined if this internal flow pattern is universal in a larger parameter space. The focus was on the droplet diameter and density differences. Regarding the droplet diameter, additional simulations examined $d_0 = 2.5$ and 3.5 mm. Regarding the density difference, while maintaining the incident shock strength at $M_s = 2.4$, simulations included three different kinds of droplets: gelatin (heavier than water), fat (lighter that water), and dodecane (much lighter than water). Table 3.1 lists the corresponding densities and stiffened gas EOS parameters.

	ρ	γ	π
Gelatin	1061.0	4.04	6.1×10^8
Fat	920.0	4.18	4.74×10^8
Dodecane	749.5	2.35	4.0×10^8

Table 3.1 The parameters used for the different liquids.

Dimensionless trajectories of the WS, LS, and SP are extracted from the numerical results mentioned previously (see Fig. 3.12). The quantities are nondimensionalized as

$$t^* = \frac{t}{\left[\left(d_0/u_g\right)\sqrt{\rho_l/\rho_g}\right]},$$

$$x^* = x/d_0,$$

$$and\ t_{sp}^* = (t - t_{sp}^0)/\left[\left(d_0/u_g\right)\sqrt{\rho_l/\rho_g}\right] \tag{3.1}$$

where t^* is the dimensionless time of the droplet evolution, u_g is the post-shock air velocity, ρ_l is the liquid density, ρ_g is the post-shock air density, x^* is the dimensionless time, t^* is the dimensionless time of SP development, and t_{sp}^0 is the instant when the SP appears. Fig. 3.12(a) reveals that the dimensionless trajectories of the WS and LS collapse perfectly into each other. The SP trajectories, as shown in Fig. 3.12(b), hold steady at around $x^* = 0.3$ after $t_{sp}^* = 0.05$, which indicates that this internal flow pattern is universal in similar shock/droplet interaction phenomena.

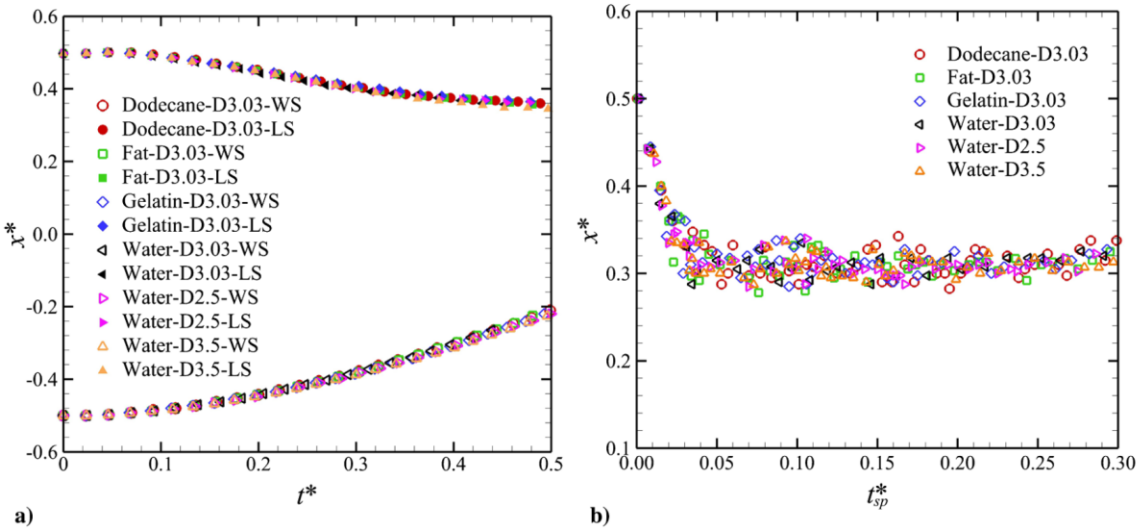


Figure 3.12 Trajectories of a) WS/LS, and b) SP, for different droplets at $M_s = 2.4$ in their dimensionless form.

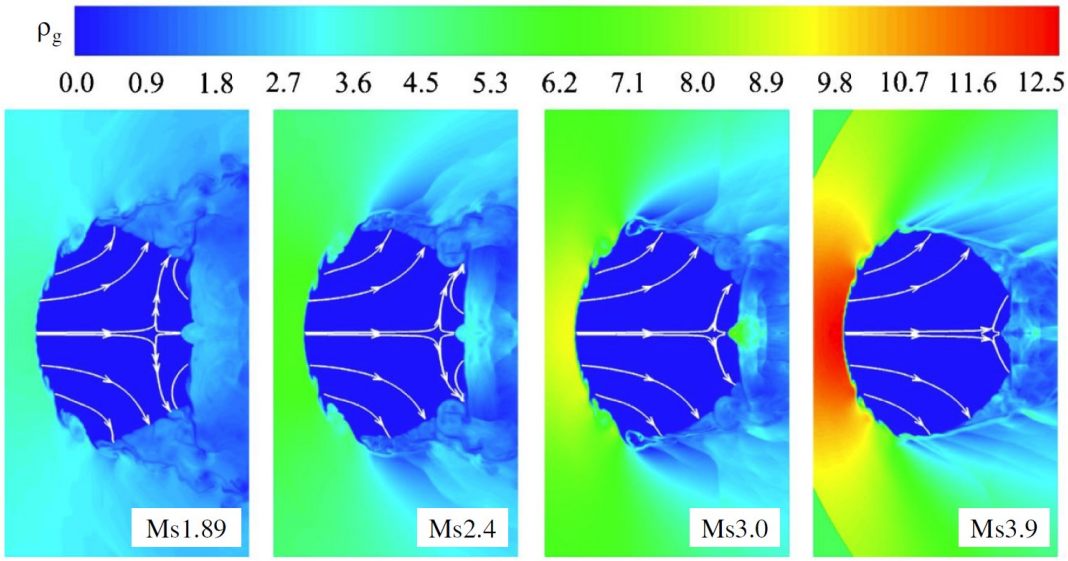


Figure 3.13 Water droplet morphologies at $t^* = 0.3$ for cases with different incident shock strengths.

3.2.2. Internal Flow Field with Different Shock Strengths

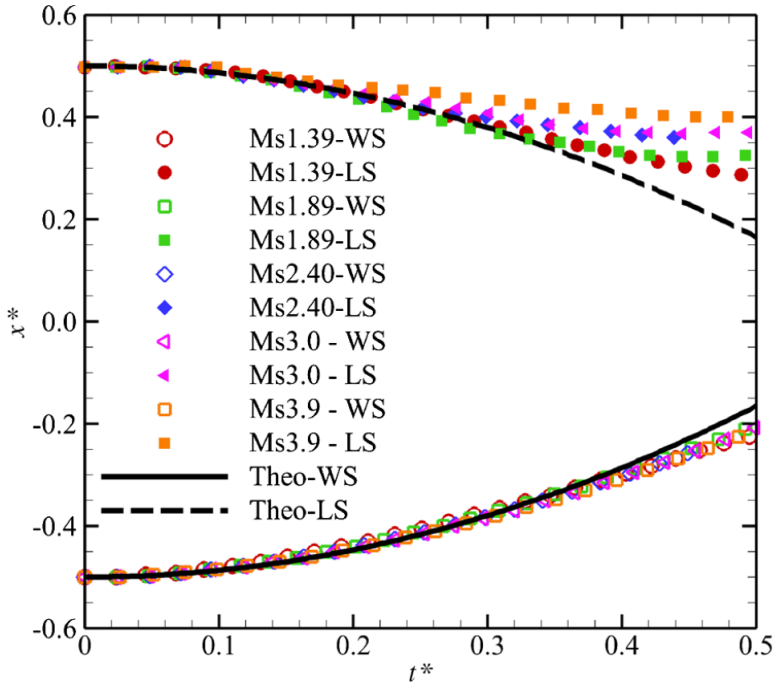


Figure 3.14 Trajectories of WS (open symbols) and LS (solid symbols) of cases with different incident shock Mach numbers.

To correlate M_s with the water droplet internal flow pattern, cases simulate different shock strengths, and make comparisons at the same dimensionless time. Fig. 3.13 presents the droplet internal flow pattern at the instant when $t^* = 0.3$ for four different M_s values. With the increase of shock strength, the SP locates itself closer and closer to the LS, and the droplet morphology varies accordingly.

The momentum transportation from the high-pressure zones at the WS and the LS obviously affects the position of the SP. This momentum transportation presents itself by the movement of the positions of the stagnation points. Fig. 3.14 shows the trajectories of

the WS and LS of the preceding five different M_s cases and compares them. It is interesting to see that both the WS and LS trajectories fits perfectly to the potential theory prediction (Engel 1958) at the early stage.

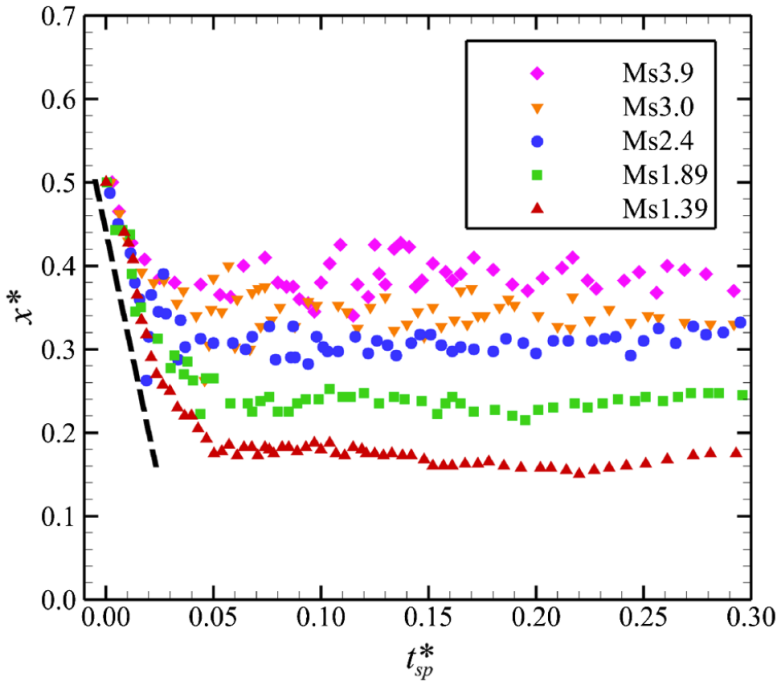


Figure 3.15 SP trajectories for cases with different incident shock strengths.

Because the internal flow field is simple at the early stage, an easy way to describe the internal flow pattern is to record the position of the SP, which remains stationary in space after the initial flow development process. Fig. 3.15 presents the trajectories of the SPs for different incident shock strengths. As shown, SPs form very close to the LS initially and move upstream. Although oscillations exist in all five cases because of the repeated internal wave reflection, the positions of the SPs hold relatively steady after $t_{sp}^* = 0.05$. Furthermore, Fig. 3.15 well reflects the trend shown in Fig. 3.13 in that the SP position

drifts farther downstream (larger x coordinate) for the relatively strong shock cases. The trajectories of the SPs before they reach their stationary position collapse to the same line (with the same slope as the black dash line denoted in Fig. 3.15). This indicates the ability to correlate the rate of change of the internal flow pattern using the preceding nondimensionalization method.

3.2.3. Theoretical Prediction

Following the preceding discussion, the position of the SP, l_{sp}, after the initial shock–droplet interaction, can be estimated based on the velocity at which the SP moves (u_{sp}) and the duration in which the shock influences the LS area (t_s), i.e., $l_{sp} \sim u_{sp} t_s$. u_{sp} is the outcome of the change of internal flow, and it connects to the liquid flow velocity u_l (i.e., $u_{sp} \sim u_l$). The term t_s closely relates to the shock propagation outside of the droplet, and it connects to the shock velocity u_s by $t_s \sim (d_0/u_s) = (d_0/a_0\, M_s)$, where a_0 is the sound speed in quiescent air. In this way, we present the dimensionless SP displacement $l_{sp}^* = l_{sp}/d_0$ by a simple linear approximation:

$$l_{sp}^* = A\frac{u_l\, t_s}{d_0} + B = A\frac{t_s}{d_0/u_l} + B = A\frac{t_s}{\tilde{t}} + B \tag{3.2}$$

where d_0 on the right-hand side of the first equals sign is used to nondimensionalize the term $u_l\, t_s$. The liquid flow velocity is obtained qualitatively by $u_l \sim u_g \sqrt{\rho_g/\rho_l}$ as applied

by Nicholls & Ranger (1969), which arises directly from an application of Newton's second law to droplet displacement. The important measure of the intensity of the interaction is the gas-flow dynamic pressure ($(1/2)\rho_g u_g^2$) behind the shock); the momentum flux change $\rho_l u_l^2$ inside the droplet is proportional to $(1/2)\rho_g u_g^2$ as a prompt consequence of the interfacial response to this gas dynamic impulse (i.e., $\rho_l u_l^2 \sim (1/2)\rho_g u_g^2$)). Therefore, the liquid velocity $u_l \sim u_g \sqrt{\rho_g/\rho_l}$ can be qualitatively obtained. The term $\tilde{t} = d_0/u_l$ can be considered a characteristic time. A and B are constants to be determined.

Manipulating equation 3.2 and correlating the dimensionless SP displacement to the incident shock strength, we have

$$l_{sp}^* = \frac{A}{d_0} u_g \sqrt{\frac{\rho_g}{\rho_l}} \frac{d_0}{u_s} + B$$

$$= A \sqrt{\frac{\rho_0}{\rho_l}} \frac{2(M_s^2+1)}{M_s \sqrt{M_s^2(\gamma^2-1)+2(\gamma+1)}} + B \tag{3.3}$$

where ρ_0 is the density in quiescent air, and γ = 1.4 is the ratio of the specific heats of air. Note that

$$\rho_g = \rho_0 \frac{(\gamma+1)M_s^2}{(\gamma-1)M_s^2+2} \tag{3.4}$$

$$u_g = (1 - \frac{\rho_0}{\rho_g}) u_s \tag{3.5}$$

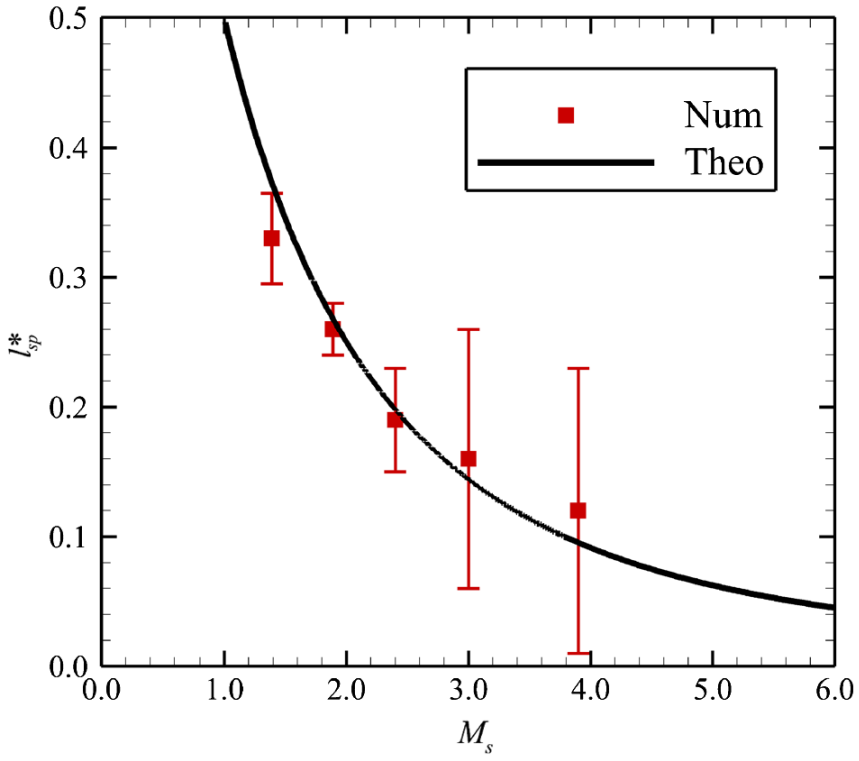


Figure 3.16 Comparison of numerical results and the theoretical prediction of Equation 3.3.

Asymptotic conditions confine the correlation between l^*_{sp} and M_s in Equation 3.3. When M_s approaches 1, the incident shock is infinitely weak, and the SP is located at the center of the droplet (i.e., l^*_{sp} goes to 0.5). However, when M_s approaches infinity, the SP reaches the LS (i.e., l^*_{sp} goes to zero). From this, constants A = −7.14 and B = 0.5 are derived. Equation 3.3 relates the stationary SP position solely to the incident shock strength M_s.

Following the preceding discussion, Fig. 3.16 depicts the relationship between l^*_{sp} and M_s in Equation 3.3, together with the numerical data in Fig. 3.15. The five cases' numerical error bars characterize the vibrations of the SPs due to the constantly repeating

reflection of the internal wave system. It is easy to understand that the larger M_s is, the larger the vibration amplitude of SP. It is seen that the theoretical prediction agrees well with the numerical simulations. We can quantitatively predict the characteristic internal flow field pattern with different incident shock strengths, including the stationary SP position and the trajectories of the WS and LS.

3.3 Summary

In summary, this study investigates, both numerically and theoretically, the internal flow pattern of a single water droplet under shock impact. Similar internal flow patterns occur in cases with different incident shock strengths, in which the SP forms and remains stationary soon after the passage of the incident shock. With the increase in the incident shock strength, the SP position varies, and the droplet presents different morphologies. A simple theory can predict the stationary position of SP in accordance with the incident shock Mach number. This correlation connects the shock strength with the droplet internal flow field. It infers that the shock inputs the droplet internal flow pattern into the droplet at the initial impact stage rather than following the post-shock airstream. More research should be done on this point in the near future.

4. Investigation of a Shock Interacting with a Sphere Water Droplet Embedded with a Large Vapor Bubble Inside

4.1 Experimental Specifications

Since Chapter 2 schematically explains the experiment setup used in this study, this chapter will not be describe it further. Some specifications for setup and operation are given below.

A key point to mention is, a vacuum pump (Leybold DIVAC) removes the air from the test and driven sections before each experiment, and a vacuum pressure gauge (SMCGZ46-K2K) measures the final pressure at $(5.3 \pm 0.5) \times 10^3$ Pa. Meanwhile, as a result of the heating effect of the LED light source, the temperature inside the test section is 34℃, which is higher than the laboratory temperature (23℃). Thereby, the ambient pressure in the test section reaches the saturation vapor pressure of water (5.3×10^3 Pa) at 34℃. The equilibrium between the liquid phase and gas phase is reached inside the droplet, and a large vapor bubble (cavity) is gradually generated within the droplet before the droplet falls from the needle. However, each experiment cannot control the sizes of the droplet and the bubble, as well as the position of the bubble relative to the droplet. The droplet's diameter (D_d) ranges from 1.14 mm to 3.04 mm, and the bubble's diameter (D_c)

ranges from 0.68 mm to 2.71 mm. Table 4.1 lists the dimensions of D_d and D_c for each case.

Case no.	1	2	3	4	5	6	7	8	9
D_d (mm)	1.232	1.375	1.135	1.538	3.038	2.218	1.554	1.295	1.143
D_c (mm)	0.679	0.857	0.750	1.231	2.712	2.055	1.143	0.911	0.679
$\delta=D_c/D_d$	0.55	0.62	0.66	0.80	0.89	0.92	0.74	0.70	0.59
S (mm)	0.063	-0.027	0.0	0.058	0.029	-0.027	-0.116	0.112	0.196
$\epsilon=S/(D_d/2)$	0.10	-0.04	0.00	0.08	0.02	-0.02	-0.15	0.17	0.34
U_{flow} (m/s)	457.8	577.0	480.6	480.6	481.6	456.0	511.7	512.0	510.0
M	2.08	2.43	2.15	2.15	2.15	2.08	2.24	2.24	2.23
Re	3372	5002	3261	4419	8747	6048	4873	4063	3556
We	618	1278	649	880	1745	1103	1051	877	764
$We^*=p_gD_d/\sigma$	455	700	449	609	1202	820	669	558	488

Table 4.1 The initial physical parameters in all cases(Liang et al. 2020). Here D_d represents the droplet diameter; D_c represents the cavity diameter; S represents the eccentric distance between the cavity center and the droplet center, and a positive or negative value indicates whether the cavity center is closer to the upstream or downstream wall of the droplet, respectively; U_{flow} and p_g represents the flow speed after the incident shock and pressure in the surrounding gas; M represents the incident shock Mach number; R_e represents the Reynolds number; W_e represents the Weber number; and W_e^* represents the defined Weber number in present study.

As measured by two piezoelectric transducers, the incident shock Mach number is 2.25 ± 0.15. Reynolds numbers and Weber numbers of flows under different working conditions are also calculated by the following definitions:

$$R_e = \frac{\rho_g U_{flow} D_d}{\mu_g}, W_e = \frac{\rho_g U_{flow}^2 D_d}{\sigma}, \tag{4.1}$$

where ρ_g, μ_g, and U_{flow} are the density, viscosity, and flow velocity of the post-shock gas. D_d represents the diameter of the droplet, and σ is the surface tension coefficient of water at the pre-shock temperature 34 ºC.

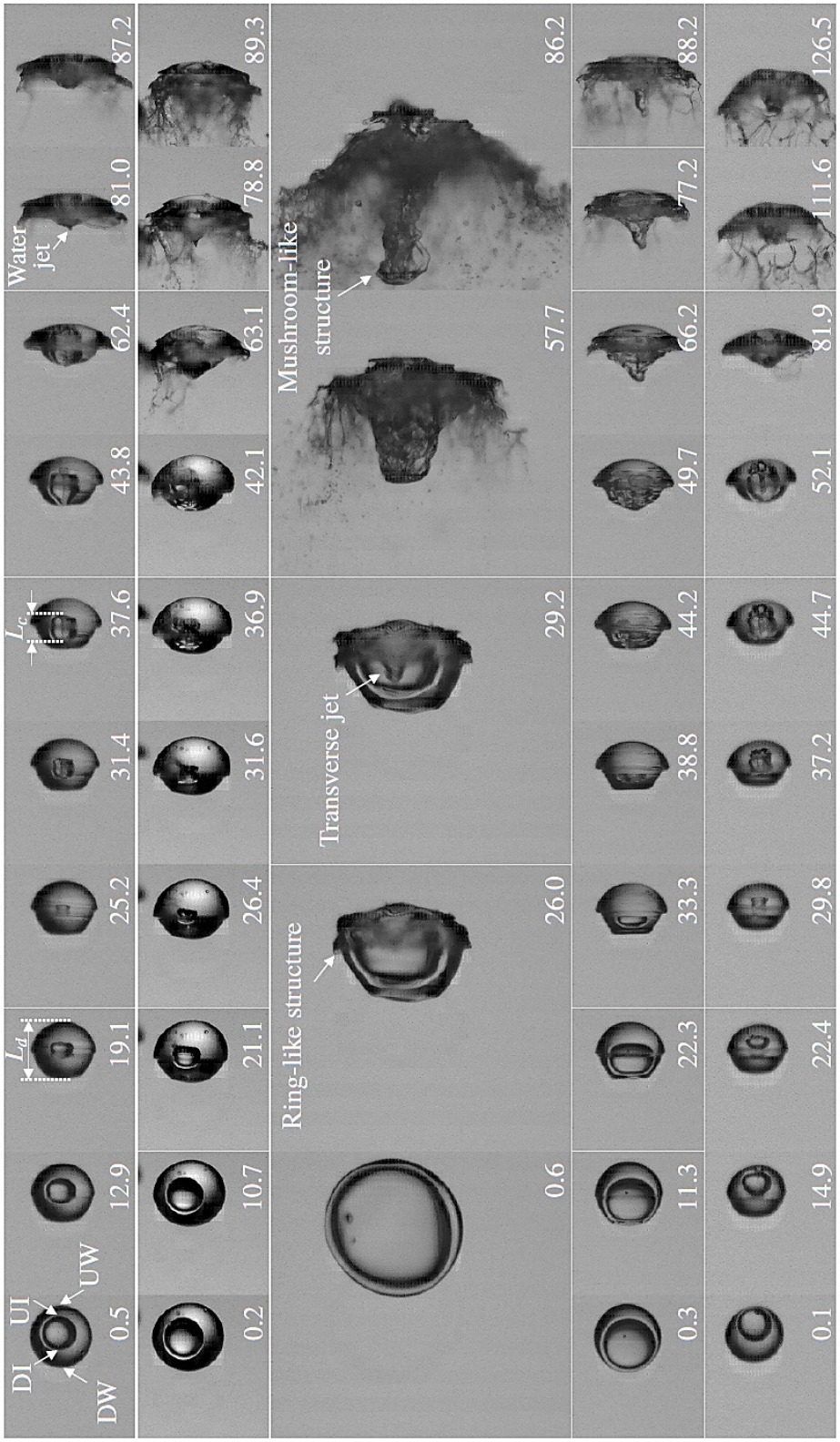
Here, another dimensionless parameter $W_e^*(=p_g D_d/\sigma)$ is also defined to compare the post-shock gas pressure p_g and the Laplace pressure. Table 4.1 summarizes geometrical and flow parameters in different cases. By using the relations of normal shock, W_e^* can be readily expressed in terms of the shock Mach number M_s and the traditional Weber number W_e:

$$\frac{W_e}{W_e^*} = \frac{4\gamma(M_s^2-1)^2}{[2\gamma M_s^2-(\gamma-1)][2+(\gamma-1)M_s^2]}, \tag{4.2}$$

where γ is the specific heat ratio of air. Since the shock Mach numbers of different cases are all approximately 2, the ratios of W_e to W_e^* are all approximately 1.5 in the present study.

4.2 Qualitative Analysis

Fig. 4.1 shows experimental images of the hollow droplets in five representative cases. Time zero is defined as the instant at which the shock wave impacts the upstream wall of the droplet, and the time is normalized by D_d/U_{flow}. Fig. 4.1(a) indicates the upstream wall of the droplet (UW), the downstream wall of the droplet (DW), the upstream interface of the cavity (UI), and the downstream interface of the cavity (DI). Note that the incident shock wave propagates from right to left in each image.



dded with a vapor cavity under the imp
e) correspond to cases 1, 2, 5, 7 and 9 i
d DW denote the upstream wall and th
ace and the downstream interface of the
the cavity, respectively. The dimensio
image (Liang et al. 2020).

Case 5 listed in the table 4.1 has the largest average droplet diameter of 3.04 mm; therefore, it is the best image resolution to use as an example of the entire interaction process, as shown in Fig. 4.1(c). During the interaction of the incident shock wave and the droplet embedded with a vapor cavity (dimensionless time from 0 to 0.6), the external shape of the droplet remains almost undisturbed. However, the UI of the vapor cavity moves downstream slightly because the transmitted shock wave and the compression waves reflect from the UW impact on the UI repeatedly.

From dimensionless time 0.6–26.0, the DW becomes flattened and moves upstream because the pressure difference between the high pressure generated by the shock–shock interaction behind the droplet and the low pressure generated by rarefaction waves inside the liquid drives the DW. This observation is similar to the results of Xiang & Wang (2017), Meng & Colonius (2018) and Guan et al. (2018). In addition, the cavity shrinks substantially. In this process, local condensation of vapor can take place (Wu et al. 2019) because the local pressure inside the cavity can be higher than the saturation pressure after repeated impacts of shock/compression waves. In addition, a ring-like structure emerges on the outside surface of the droplet.

The third frame of Fig. 4.1(c) for a dimensionless time of 29.2 clearly shows a transverse jet along the symmetric axis of the droplet. Two explanations arise for the

formations of the transverse jet. The first explanation is derives from the Richtmyer–Meshkov instability (Richtmyer 1960; Meshkov 1969). As the Richtmyer–Meshkov instability (RMI) develops on the liquid–gas interface (UI) with an extreme Atwood number (Apazidis 2016), vorticity deposited by the misalignment between the pressure gradient and the density gradient drives the UI to penetrate into the vapor cavity (Xiang & Wang 2017). The second explanation emphasizes the mechanism that, after the transmitted shock inside the droplet impacts the vapor cavity, rarefaction waves reflected from the vapor cavity relax the pressure near the UI. The resultant temporal pressure gradient accelerates the flow in the streamwise direction. Due to the curved shape of the UI, this acceleration focuses the flow to one point, eventually causing the flow to evolve and form the transverse jet (Hawker & Ventikos 2012). Comparing the positions of the jet tip between two sequential images results in a measure of the mean velocity of the transverse jet in case 5, and this calculation is $V_{jet} = 21.9 \pm 0.1.9\ m/s$. Here, V_{jet} decomposes into three parts. The first part is the post-shock velocity of the interface UI, denoted by V_0. Based on the experimental images at the early stage of evolution, V_0 is evaluated to be 4:73 m/s. The second part is the RMI growth rate V_{RMI} of the interface UI, which equals 6.69 m/s. The impulsive theory (Richtmyer 1960) calculates this rate as follows:

$$V_{RMI} = kaAV_0\ , \tag{4-3}$$

where $k = \sqrt{2}/a$ is the wavenumber (Haas & Sturtevant 1987), a is the cavity radius, and $\mathrm{A} = 1$ is the Atwood number. The rest (approximately 10:48 m/s) is the velocity induced by other mechanisms, such as the flow penetration due to the reflected rarefaction waves and the effect of cavity collapse. When the transverse jet impacts the DI with a large momentum, a great temperature rise develops at the impact point (Bourne & Field 1992; Hawker & Ventikos 2012). Therefore, the vapor cavity expands, accompanied by local evaporation. As the volume of the vapor cavity increases, the DI catches up with the DW. Later, a water jet appears at the downstream pole of the droplet (dimensionless time 57.7). To the best of the authors' knowledge, this water jet has not been observed in previous experimental studies on the shock–droplet interaction. Eventually, the fine mist generated by the droplet breakup at dimensionless time 86.2 covers the water jet.

Other interesting findings include the effects of the cavity size and position on droplet deformation. When the ratio of the cavity diameter to the droplet diameter σ is sufficiently large (Fig. 4.1c), a mushroom-like vortex ring appears on the head of the water jet because of the Kelvin–Helmholtz instability induced by the velocity shear, and the surface of the UW is prominently rippled, which is different from the smooth liquid surface in the classical shock–droplet interaction experiments (Theofanous & Li 2008). The eccentricity of the vapor cavity ϵ also influences the droplet deformation. When the cavity

is closer to the downstream wall, the water jet is long and thin, as shown in Fig. 4.1(d). When the cavity is closer to the upstream wall, the water jet is short and thick, as shown in Fig. 4.1(e).

In summary, cavity evolution significantly influences droplet deformation, and cavity evolution has two stages: a cavity-collapse stage and a cavity-expansion stage. Multiple physical mechanisms, including shock/rarefaction wave dynamics, interface instabilities, and gas–liquid phase changes, govern the entire flow process.

4.3 Droplet Evolution

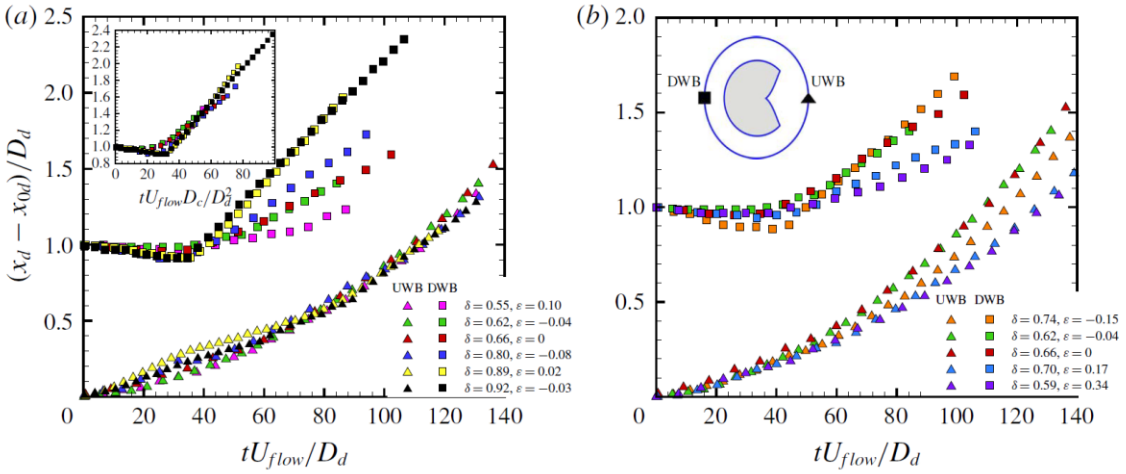


Figure 4.2 Comparison of the dimensionless displacements on the characteristic points of the droplets for different cases: (a) different δ and small ϵ (| ϵ |≤ 0.1); (b) different ϵ and similar δ (0.59 ≤ δ ≤ 0.74). Triangle symbols and square symbols represent the UWB displacements and the DWB displacements obtained from the experimental images, respectively.

Fig. 4.2 shows the time-varying displacements of the upstream wall boundaries (UWBs) and the downstream wall boundaries (DWBs) in different cases. The dimensionless displacement is defined as $(x_d - x_{0d})/D_d$, where x_d is the displacement of the UWB or DWB, and x_{0d} is the initial position of the UWB. For all cases, after the incident shock wave impacts on the hollow droplet, the UWB displacement (triangle symbol) increases gradually with a positive acceleration because of the accelerating effect of the rarefaction wave reflected from the DW. Moreover, the displacement of the DWB (square symbol) decreases slightly at first, and then increases because both the transverse jet impingement and the local evaporation of water push the DI downstream.

In Fig. 4.2(a), when the eccentricity is sufficiently small ($|\epsilon| \leq 0.1$), the time-varying displacements of the UWBs for different δ values almost coincide, which indicates that δ has little influence on the motion of the UW. In addition, the movement of the DWB increases when δ is larger. In the inset of Fig. 4.2(a), the dimensionless time is redefined as $\mathrm{t}U_{flow}D_c/D_d^2 = (\mathrm{t}U_{flow}/D_d)\delta$. The time-varying displacements of the DWBs almost coincide with this new dimensionless method, which means that the water jet velocity is nearly proportional to δ.

In Fig. 4.2(b), when δ varies over a small range ($0.59 \leq \delta \leq 0.74$), the time-varying displacements of the UWBs in the cases with the negative ϵ (≤ -0.1) are smaller

than those in the cases with $\epsilon \sim 0$, while those in the cases with positive ϵ (≥ 0.1) reduce even more. Therefore, when the vapor cavity center misalignes with the droplet center, the vapor cavity impedes the motion of the UW, especially when the vapor cavity is closer to the UW. In addition, the time-varying displacement of the DWB with negative ϵ (≤ -0.1) is similar to those in the cases with $\epsilon \sim 0$, but those in the cases with positive ϵ (≥ 0.1) are smaller. In general, as ϵ increases, the time-varying displacement of the DWB decreases.

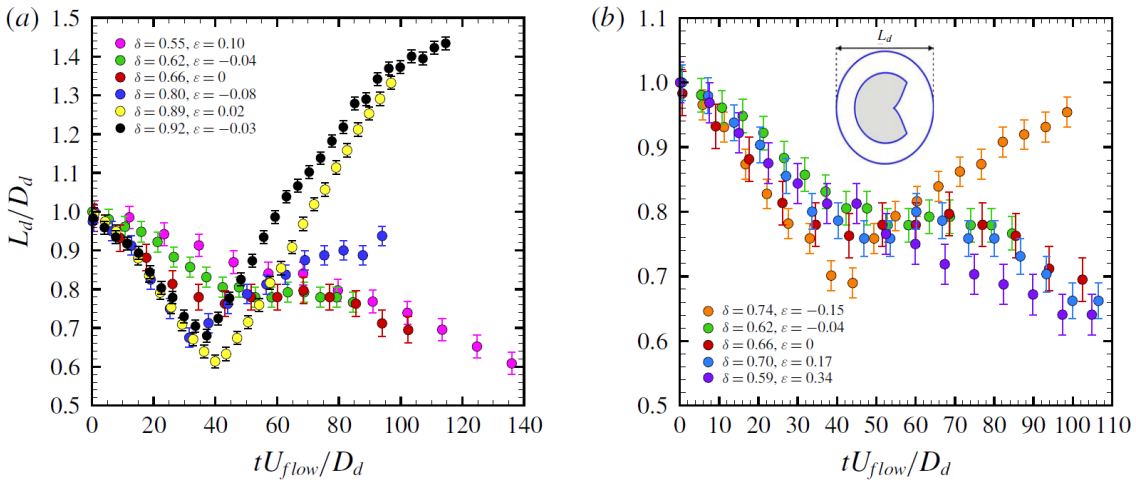


Figure 4.3 Comparison of the dimensionless streamwise lengths of the droplets for different cases: (a) different δ and small ϵ (| ϵ |$\leqslant$ 0.1); (b) different ϵ and similar δ (0.59 $\leqslant$ δ $\leqslant$ 0.74).

Fig. 4.3 shows the time-varying lengths of the droplets (L_d) in different cases. The time is normalized as tU_{flow}/D_d, and the length is normalized as L_d/D_d. Fig. 4.3(a) demonstrates the comparison between cases with sufficiently small eccentricity ($|\epsilon| \leq 0.1$) and different values of δ. When $\delta \geq 0.8$, L_d first decreases, and then increases because the water jet velocity is larger than the velocity of the UWB. However, when $\delta \leq$

0.66, due to the limited momentum of the transverse jet, the induced velocity of the water jet is larger than the velocity of the UWB only during a short period. Consequently, L_d first decreases, then increases slightly, and finally decreases again during the observation time. In Fig. 4.3(b), δ varies within a small range $(0.59 \leq \delta \leq 0.74)$, but ϵ varies significantly. When $\epsilon = -0.15$, L_d first decreases and then increases. However, for the cases with $\epsilon \geq -0.04$, L_d first decreases, then increases for a short period, and finally decreases again. In summary, when the size of the cavity is relatively large or the position of the cavity is closer to the downstream wall, the water jet is faster, and thus the length of the droplet is longer.

4.4 Cavity Evolution

Fig. 4.4(a) shows the time-varying displacements of the upstream interface boundaries (UIBs) and the downstream interface boundaries (DIBs) in different cases. The normalized displacement is defined as $(x_c - x_{0c})/D_c$, where x_c is the displacement of the UIB or DIB, and x_{0c} is the initial position of the UIB. After the transmitted shock wave passes through the vapor cavity, the UI moves downstream as a result of the shock impact and the cavity collapse. Meanwhile, the DI moves upstream because of the high pressure

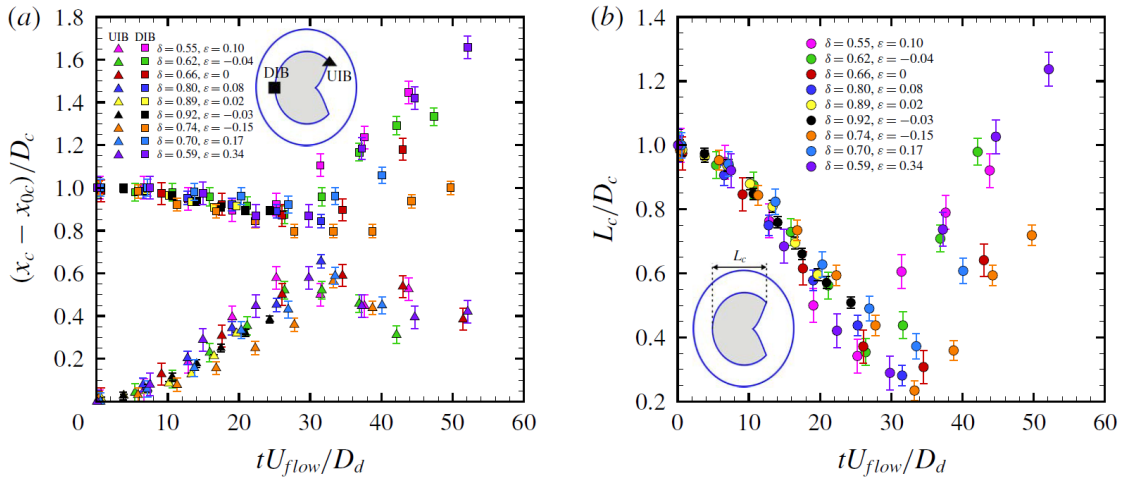


Figure 4.4 (a) Comparison of the dimensionless displacements of the UIBs and DIBs for different cases. Triangle symbols and square symbols represent the UIB displacements and the DIB displacements obtained from the experimental images, respectively. (b) Comparison of the dimensionless lengths of cavities for all cases.

generated by the shock–shock interaction near the downstream pole of the cavity as well as the cavity collapse. Before the vapor cavity shrinks to a tiny core, the transverse jet inside the cavity impacts the DI, and the volume of the cavity decreases at this time. Thereafter, the momentum exchange between the transverse jet and the DI and the local evaporation of water, which expands the vapor cavity, pushes the DI downstream and drives the UI upstream. The time-varying displacements of the UIBs and DIBs in almost all the cases show agreement in their variation trends.

Fig. 4.4(b) indicates the time-varying lengths of the vapor cavity (L_c) in different cases. The normalized length is defined as L_c/D_c. In general, both δ and ϵ have limited influences on the evolution in the vapor cavity size. The decrease in L_c corresponds to the

cavity-collapse stage, which is caused by shock compression and possibly vapor condensation. The subsequent increase in L_c corresponds to the cavity-expansion stage, which comes from the transverse jet impingement and the water evaporation.

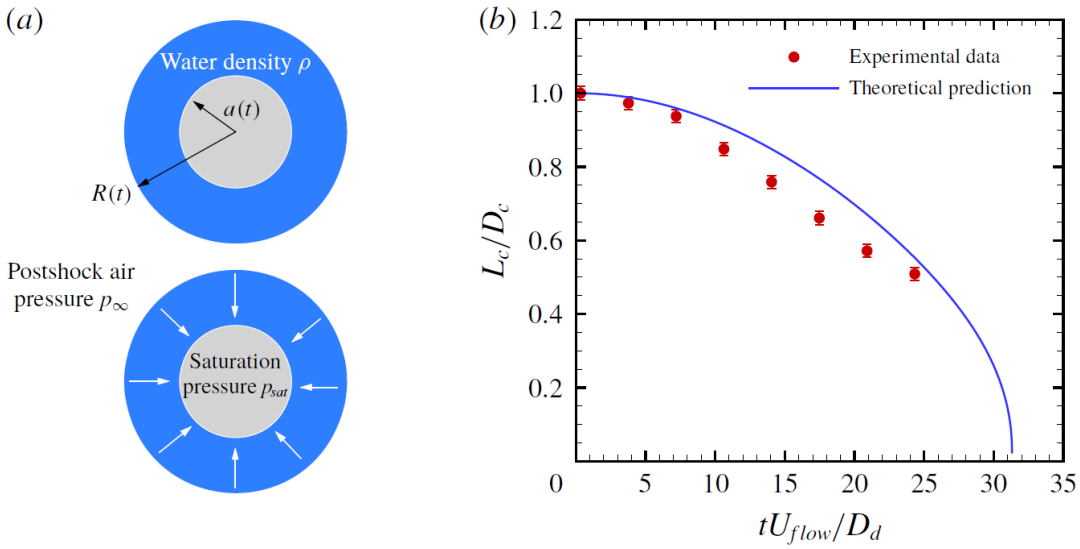


Figure 4.5 (a) Schematics of a spherical collapsing model for a water droplet embedded with a vapor cavity. The droplet is immersed in uniform air with a pressure equal to the postshock air pressure in the present experiment. The pressure inside the cavity is assumed to be the saturation pressure of water. (b) Theoretical prediction and experimental data for the time-varying cavity size in the collapsing stage of case 6.

A theoretical estimation of the time-varying cavity length can emerge from solving a simplified cavity collapsing problem, which is assumed to be of spherical symmetry, as sketched in Fig. 4.5(a). Extending the Rayleigh–Plesset equation (Brennen 1995) to the scenario, as shown in Fig. 4.5(a), can describe the evolution of the cavity radius $a(t)$. First of all, due to the liquid density ρ only changing minorly after the incident shock wave, then

by ignoring the change in liquid density ($\nabla \cdot \mathrm{u} = 0$, here u represents the velocity vector), the continuity equation for the liquid phase can be expressed in spherical coordinates as

$$\frac{1}{r^2}\frac{\partial(r^2 u(t,r))}{\partial r} = 0\,, \tag{4.4}$$

where the flow velocity $u(t,r)$ is in the radial direction. From equation 4.4 and the boundary condition for velocity at the liquid–vapor interface ($\mathrm{r} = \mathrm{a(t)}$),

$$\mathrm{u\big(t, a(t)\big)} = \frac{da(t)}{dt}\,, \tag{4.5}$$

The velocity field can be derived and expressed in term of the cavity radius $\mathrm{(a(t))}$ as

$$u(t,r) = \frac{da(t)}{dt}\frac{a^2(t)}{r^2} \tag{4.6}$$

The velocity at $\mathrm{r = R(t)}$ is equal to the change rate of the droplet radius $(\mathrm{dR(t)}/dt)$, thus

$$\frac{dR(t)}{dt} = \mathrm{u\big(t, R(t)\big)} = \frac{da(t)}{dt}\frac{a^2(t)}{R^2(t)}\,. \tag{4.7}$$

Integrating equation 4.7 yields

$$R^3(t) = a^3(t) + R_0^3 - a_0^3\,, \tag{4.8}$$

where R_0 and a_0 are the radius of droplet and cavity at $t = 0$, respectively.

By ignoring the viscous force, the surface tension and the gravity, the momentum equation for the liquid phase can be written in spherical coordinates as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial t} = -\frac{1}{\rho}\frac{p}{r} \tag{4.9}$$

where ρ and p are the density and pressure in the liquid phase at the experimental temperature.

Assume that the droplet is immersed in a uniform gas with a pressure p_∞ and the cavity is filled with a saturated water vapor that has a pressure p_{sat}. Then, integrating equation 4.9 with respect to r, we obtain

$$\int_{a(t)}^{R(t)}\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial t}\right) dr = \frac{p_{sat}-p_\infty}{\rho} \tag{4.10}$$

After substituting equation 4.6 into equation 4.10 and performing the integration, the resulting equation is the modified Rayleigh–Plesset equation:

$$\left(a(t) - \frac{a^2(t)}{R(t)}\right)\frac{d^2a(t)}{dt^2} + \left[\frac{3}{2} - 2\left(\frac{a(t)}{R(t)}\right) + \frac{1}{2}\left(\frac{a(t)}{R(t)}\right)^4\right]\left(\frac{da(t)}{dt}\right)^2 = \frac{p_{sat}-p_\infty}{\rho}, \tag{4.11}$$

where the relation between $R(t)$ and $a(t)$ are provided by equation 4.8.

In the limit of $R(t) \to \infty$, equation 3.2 reduces to the classical Rayleigh–Plesset equation for a vapor bubble in an infinite volume of liquid. A limitation of the present modelling comes from the assumption of undisturbed vapor pressure p_{sat}. Accordingly, future work should incorporate into the collapse dynamics the effects of thermodynamics and heat transfer on the evaluation of p_{sat}.

With initial conditions and physical parameters in accordance with those in experimental case 6, the above equations are numerically solved, and Fig. 4.5(b) plots the collapsing history of the vapor cavity. The comparison between the theoretical results and corresponding experimental data indicates that the idealized theoretical model in bubble dynamics still provides a good estimation of the time-varying cavity length, even though the present problem involves complex flow physics. For a vapor cavity in an infinite volume of water with initial radius a_0, the total collapse time has the following theoretical expression, known as the Rayleigh time (Lord Rayleigh 1917; Brennen 1995):

$$\tau_R = \sqrt{\frac{3\pi}{2}} \frac{\Gamma(5/6)}{\Gamma(1/3)} a_0 \sqrt{\frac{\rho}{p_\infty - p_{sat}}}, \tag{4.12}$$

where Γ is the Gamma function. Based on the parameters in case 6, the normalized Rayleigh time $\tau_R U_{flow}/D_d$ is equal to 42.62, which overestimates the collapse time, in comparison with the results in Fig. 4.5(b). Therefore, it is necessary to consider the effect of finite droplet size in modelling the collapse dynamics for the present problem.

4.4 Summary

In this chapter, the evolution of the droplet with a vapor bubble inside under shock impact is experimentally investigated, and the development of multiphase systems under shock wave is investigated.

1). For the first time, shock-tube experiments on the interaction of shock wave and a water droplet embedded with a vapor cavity are performed to investigate both the evolutions of water droplet and vapor cavity.

2). An equilibrium between the liquid phase and the gas phase inside the droplet is obtained by the depressurization of the surrounding gas to the saturated vapor pressure.

3). The deformations of the vapor cavity can be separated into two stages: (1) the vapor cavity shrinks because of the shock compression and the possible local condensation; (2) the cavity expands because of the transverse jet impact and the consequent local evaporation.

4). When the vapor cavity size is larger and the vapor cavity center is closer to the downstream wall of the droplet, the water jet is longer, and its speed is higher.

5). A modified Rayleigh-Plesset equation is derived that reasonably predicts the bubble collapse process.

5. Conclusions and Suggestions for Future Research

5.1 Conclusions

In summary, the main conclusions of Chapter 3 include:

The internal flow pattern of a single water droplet under shock impact is investigated numerically and theoretically. Similar internal flow patterns are found in cases with different incident shock strengths, in which the SP forms and remains stationary soon after the passage of the incident shock. With the increase in the incident shock strength, the SP position varies, and the droplet presents different morphologies. A simple theory is proposed to predict the stationary position of SP in accordance with the incident shock Mach number. This correlation connects the shock strength with the droplet internal flowfield. It infers that the droplet internal flow pattern is input into the droplet at the initial impact stage by the shock rather than following the postshock airstream. Upon this point, more research should be done in the near future.

Moreover, the evolution of the droplet with a vapor bubble inside under shock impact is experimentally investigated, and the development of multiphase systems under shock wave is investigated. The main conclusions of Chapter 4 include：

The interaction of excitation waves with droplets containing vapor bubbles was studied using shock tube experiments to investigate the deformation of droplets and

bubbles in a multiphase system. Vapor bubbles containing different sizes and eccentricities were generated in the droplets by reducing the ambient pressure. The evolution of the vapor bubbles was found to significantly affect the deformation of the outer surface of the droplets. The deformation of bubbles and droplets can be divided into two phases based on the change in bubble volume: bubble-collapse phase and bubble-expansion phase. In the bubble-collapse phase, the transverse jet at the upstream interface of the bubble and its collision with the downstream interface of the bubble was observed. After the transverse jet impacts the downstream interface of the bubble, the bubble and interface deformation enters the bubble expansion phase, and a water jet appears at the downstream pole of the droplet. The effect of the size and eccentricity of the bubble relative to the droplet on the motion and deformation of the bubble and the droplet was quantitatively analyzed. When the size of the bubble was relatively large or when the size of the bubble was closer to the downstream wall of the droplet, the water jet was longer and thus the length of the droplet was greater. However, the effect of the size and position of the bubble relative to the droplet on the bubble deformation is limited. Finally, a modified Rayleigh-Plesset equation is given that reasonably predicts the bubble collapse process considering a finite droplet volume.

The interaction of a shock wave and a liquid droplet embedded with a vapor cavity poses a great challenge for the numerical simulation method, considering the complex wave system, unstable liquid–gas interfaces, and the possible phase change process. Therefore, the present experiments provide valuable benchmarks for numerical solvers aimed at compressible two-phase flows. The present experimental data and images can serve as reference results for numerical validation purposes. Future works will include developing numerical methods for multiphase flows with viable phase-change models and numerical studies of shock–droplet interactions considering vapour cavities inside the droplets.

5.2 Suggestions for Future Research

As the position of the SP is related to the momentum transportation from the high pressure regions (WS and LS), for industrial applications, it should be of great practical value to study the parameters that affect the steady time of the SP.

The Mach number studied in the experiments with shock waves impacting droplets containing vapor bubbles was around 2.2. To future study the evolution of the sphere droplet containing large vapor bubble under the impact of the planar shock wave, more experiments need to be conducted at a larger range of Mach numbers. In addition, although

the CE/SE numerical simulations agree well with the previous shock-droplet experiments. The numerical code still needs major improvement to fit the shock-droplet embedded with vapor bubble experiment results, because the phase change can not be neglected in these experiments. It is believed that the future development of numerical simulation programs that consider the solution of the compressible N-S equations for reasonable phase transition equations is needed to study in detail the evolution of such multiphase systems.

Reference

APAZIDIS, N. 2016 Numerical investigation of shock induced bubble collapse in water. *Phys. Fluids* **28** (4), 046101.

BHATTACHARYA, S. 2016 Interfacial wave dynamics of a drop with an embedded bubble. *Phys. Rev. E* **93**, 023119.

BOIKO, V. M., & POPLAVSKI, S. V. 2009 On the dynamics of drop acceleration at the early stage of velocity relaxation in a shock wave. *Combust., Explos. Shock Waves* **45**(2), 198-204.

BOURNE, N. K. & FIELD, J. E. 1992 Shock-induced collapse of single cavities in liquids. *J. Fluid Mech.* **244**, 225–240.

BRENNEN, C. E. 1995 *Cavitation and Bubble Dynamics*. pp. 34–37. Oxford University Press.

BRUJAN, E. A., KEEN, G. S., VOGEL, A. & BLAKE, J. R. 2002 The final stage of the collapse of a cavitation bubble close to a rigid boundary. *Phys. Fluids* **14** (1), 85–92.

CHEN, H., KREIDER, W., BRAYMAN, A. A., BAILEY, M. R. & MATULA, T. J. 2011 Blood vessel deformations on microsecond time scales by ultrasonic cavitation. *Physical review letters* **106** (3), 034301.

ENGEL, O. G. 1958 Fragmentation of Waterdrops in the Zone. *J. Res. Natl. Bur. Stand.* **60**(3), 245-280.

FIELD, J. E., DEAR, J. P. & OGREN, J. E. 1989 The effects of target compliance on liquid drop impact. *J. Appl. Phys.* **65** (2), 533–540.

FIELD, J. E., CAMUS, J. J., TINGUELY, M., OBRESCHKOW, D. & FARHAT, M. 2012 Cavitation in impacted drops and jets and the effect on erosion damage thresholds. *Wear* **290**, 154–160.

GUAN, B., LIU, Y., WEN, C. Y. & SHEN, H. 2018 Numerical study on liquid droplet internal flow under shock impact. *AIAA J.* **56** (9), 3382–3387.

GUILDENBECHER, D., LÓPEZ-RIVERA, C. & SOJKA, P. 2009 Secondary atomization. *Exp. Fluids* **46** (3), 371–402.

HAAS, J. F. & STURTEVANT, B. 1987 Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *J. Fluid Mech.* **181**, 41–76.

HAWKER, N. A. & VENTIKOS, Y. 2012 Interaction of a strong shockwave with a gas bubble in a liquid medium: a numerical study. *J. Fluid Mech.* **701**, 59–97.

IGRA, D., & TAKAYAMA, K. 2001 Investigation of aerodynamic breakup of a cylindrical water droplet. *Atomization Sprays* **11**(2), 20.

JALAAL, M., & MEHRAVARAN, K. 2014 Transient growth of droplet instabilities in a stream. *Phys. Fluids*, **26**(1), 012101.

JOSEPH, D. D., BELANGER, J. & BEAVERS, G. S. 1999 Breakup of a liquid drop suddenly exposed to a high-speed airstream. *Intl J. Multiphase Flow* **25** (6), 1263–1303.

KODAMA, T. & TOMITA, Y. 2000 Cavitation bubble behavior and bubble–shock wave interaction near a gelatin surface as a study of in vivo bubble dynamics. *Appl. Phys. B* **70** (1), 139–149.

KONDO, T. & ANDO, K. 2016 One-way-coupling simulation of cavitation accompanied by high-speed droplet impact. *Phys. Fluids* **28** (3), 033303.

LÜ, M., NING, Z. & SUN, C. 2017 Numerical simulation of cavitation bubble collapse within a droplet. *Computers and Fluids* **152**, 157–163.

LIANG, Y., JIANG, Y., WEN, C. Y., & LIU, Y. 2020 Interaction of a planar shock wave and a water droplet embedded with a vapour cavity. *J. of Fluid Mech.* **885**.R6.

LIN, K., KENNEDY, P. & JACKSON, T. 2004 Structures of water jets in a mach 1.94 supersonic crossflow. *42nd AIAA Aerospace Sciences Meeting and Exhibit* .

LIU, L., MA, W., LIU, Y. & CUI, J. 2018 Study on mechanism of bubble growth within a water droplet under rapid depressurization. *Intl J. Heat Mass Transfer* **119**, 709–719.

MENG, J. C. & COLONIUS, T. 2018 Numerical simulation of the aerobreakup of a water droplet. *J. Fluid Mech.* **835**, 1108–1135.

MESHKOV, E. E. 1969 Instability of the interface of two gases accelerated by a shock wave. *Fluid Dyn.* **4**, 101–104.

NICHOLLS, J. A., & RANGER, A. A. 1969 Aerodynamic shattering of liquid drops. *AIAA J.* **7**(2), 285-290.

PAYRI, R., GARCIA, J.M. & SALVADOR, F.J. 2005 Using spray momentum flux measurements to understand the influence of diesel nozzle geometry on spray characteristics. *Fuel* **84**, 551–561.

PERURENA, J. B., ASMA, C. O., THEUNISSEN, R. & CHAZOT, O. 2009 Experimental investigation of liquid jet injection into mach 6 hypersonic crossflow. *Experiments in* fl*uids* **46** (3), 403–417.

PHILIPP, A. & LAUTERBORN, W. 1998 Cavitation erosion by single laser-produced bubbles. *J. Fluid Mech.* **361**, 75–116.

PILCH, M. & ERDMAN, C. A. 1987 Use of breakup time data and velocity history data to predict the maximum size of stable fragments for acceleration-induced breakup of a liquid drop. *International journal of multiphase ow* **13** (6), 741–757.

QU, Q., MA, P., LIU, P., LI, S., & AGARWAL, R. K. 2016 Numerical study of transient deformation and drag characteristics of a decelerating droplet. *AIAA J.* **54**(2), 490-505.

RAYLEIGH, LORD 1917 On the pressure developed in a liquid during the collapse of a spherical cavity. *Phil. Mag.* **34** (200), 94–98.

REINECKE, W. G. & WALDMAN, G. D. 1970 The study of drop breaking behind strong shocks with applications to flight. *REP. SAMCO-TR-70-142* .

RICHTMYER, R. D. 1960 Taylor instability in shock acceleration of compressible fluids. *Commun. Pure Appl. Maths* **13**, 297–319.

SEMBIAN, S., LIVERTS, M., TILLMARK, N. & APAZIDIS, N. 2016 Plane shock wave interaction with a cylindrical water column. *Phys. Fluids* **28** (5), 056102.

SHAO, C., LUO, K., & FAN, J. 2017 Detailed numerical simulation of unsteady drag coefficient of deformable droplet. *Chem. Eng. J.* **308**, 619-631.

SHEN, H., WEN, C. Y., & ZHANG, D. L. 2015 A characteristic space–time conservation element and solution element method for conservation laws. *J. Comput. Phys.* **288**, 101-118.

SHEN, H., & WEN, C. Y. 2016 A characteristic space–time conservation element and solution element method for conservation laws II. Multidimensional extension. *J. Comput. Phys.* **305**, 775-792.

SHEN, H., WEN, C. Y., PARSANI, M., & SHU, C. W. 2017 Maximum-principle-satisfying space-time conservation element and solution element scheme applied to compressible multifluids. *J. Comput. Phys.* **330**, 668-692.

SHPAK, O., VERWEIJ, M., DE JONG, N. & VERSLUIS, M. 2016 *Droplets, Bubbles and Ultrasound Interactions*. pp. 157–174. Springer International.

THEOFANOUS, T. G. 2011 Aerobreakup of Newtonian and viscoelastic liquids. *Annu. Rev. Fluid Mech.* **43**, 661–690.

THEOFANOUS, T. G. & LI, G. J. 2008 On the physics of aerobreakup. *Phys. Fluids* **20** (5), 052103.

THEOFANOUS, T. G., MITKIN, V. V., & NG, C. L. 2013 The physics of aerobreakup. III. Viscoelastic liquids. *Phys. Fluids* **25**(3), 032101.

VAN WINGERDEN, K. 2000 Mitigation of gas explosions using water deluge. *Process Safety Progress* **19** (3), 173–178.

WADHWA, A. R., MAGI, V., & ABRAHAM, J. 2007 Transient deformation and drag of decelerating drops in axisymmetric flows. *Phys. Fluids* **19**(11), 113301.

WANG, Q., MANMI, K. & CALVISI, M. 2015 Numerical modeling of the 3d dynamics of ultrasound constrast agent microbubbles using the boundary integral method. *Physics of Fluids* **27** (2), 022104.

WANG, T., LIU, N., YI, X. & LU, X. 2017 Numerical study on shock/droplet interaction before a standing wall. *Communications in Computational Physics* **23** (4), 1052–1077.

WIERZBA, A. & TAKAYAMA, K. 1988 Experimental investigation of the aerodynamic breakup of liquid drops. *AIAA J.* **26** (11), 1329–1335.

WU, W., XIANG, G. & WANG, B. 2018 On high-speed impingement of cylindrical droplets upon solid wall considering cavitation effects. *Journal of Fluid Mechanics* **857**, 851 – 877.

WU, W., WANG, B. & XIANG, G. 2019 Impingement of high-speed cylindrical droplets embedded with an air/vapour cavity on a rigid wall: numerical analysis. *J. Fluid Mech.* **864**, 1058–1087.

XIANG, G. & WANG, B. 2017 Numerical study of a planar shock interacting with a cylindrical water column embedded with an air cavity. *J. Fluid Mech.* **825**, 825–852.

YI, X., ZHU, Y., & YANG, J. 2017 On the Early-Stage Deformation of Liquid Drop in Shock-Induced Flow. Int. Symp. Shock Waves, 30th **2**, 1269-1273.

Appendices

A. High Weber Number Shock/Droplet Interaction Experiment.

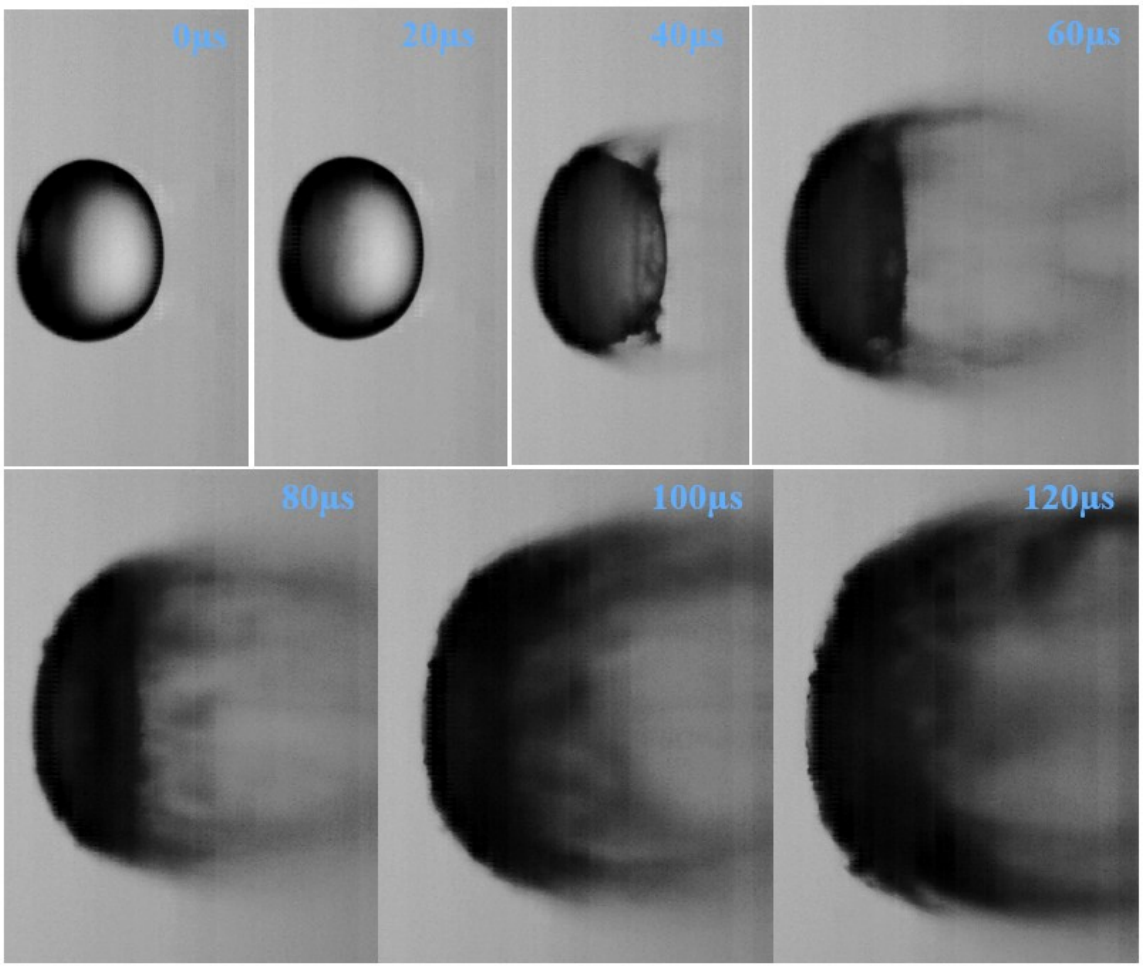


Figure A.1 Evolution of the droplet interface at Weber number equals to 9945.

As the experiment shown in Fig. A.1 has a high Weber number (9945), it may contributes to the discussion of the terminal breakup regime (SIE vs. catastrophic). As shown in Fig. A.1, in the early stages of fragmentation, the liquid mist is produced in two main regions: near the equator and at the top of the leeward surface liquid ring. In the

former, the direction of spraying is the same as that of the ambient airflow, while in the latter the direction of spraying is opposite to that of the ambient airflow. Of the two, generally speaking, the amount of liquid fog generated near the equator is much higher than that of the leeward liquid ring, which is the main component of the early liquid fog. In the middle and late stages of droplet fragmentation, due to the extreme flattening of the droplet body, the two droplet-generating regions will gradually get closer and closer together, which is difficult to distinguish from the image. Under the action of peripheral shear, the droplets in crushing show a similar "crescent" morphology, which is consistent with the shear stripping pattern. Because no RT punctures were observed in the middle region of the windward side of the droplet, the catastrophic phenomenon does not existence at least at the 10^4 Weber number. Moreover, this experiment run confirmed that K-H instability at the droplet surface dominates the generation of liquid fog near the equator in the presence of high dynamic pressure.

B. The CE/SE Simulation Results Comparing with the Existing Experiments.

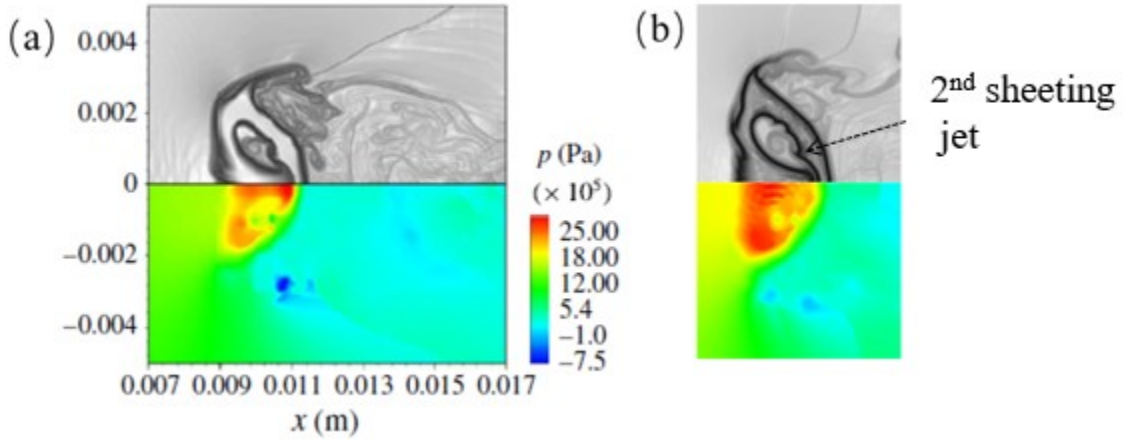


Figure B.2 The evolution of the shock/water-ring interaction at M_s=2.4, t^*=6.85. (a) the top part is the numerical schlieren and the bottom part is the pressure contours (Xiang & Wang 2017), (b) the CE/SE simulation results.

Since the experimental means used to generate bubble-containing droplets in this thesis are only applicable to the generation of vapor bubble-containing droplets, and our simulation procedure has so far not solved the phase transition problem, the authors did not present the results of our available numerical simulations in the main chapters, instead to illustrate them in the appendix. These information are provided for subsequent researchers to conduct their research.

First of all, A comparison between Xiang & Wang (2017) simulation work with our CE/SE simulation is provided in Fig. B.1, under the same test conditions, the second sheeting jet has also been observed in the current CE/SE code.

Table B.1 lists all the experimental cases' run conditions. Fig. B.2 and Fig. B.6 further present the sequential experiment photos for case 2 and 7. These two cases holding relatively large vapor bubble inside the droplet, the cavity-collapse stage and the cavity-expansion stage could also be clearly observed.

Fig. B.3 and Fig. B.7 presents the comparison between the sequential experiment results with the CE/SE numerical simulations. The 3D numerical results are generated by VisIt software basing on the results of axisymmetric numerical calculations. It can be observed intuitively that the numerical simulation results of the collapse process of the vapor bubble lag significantly behind the actual experimental results, mainly due to the lack of calculation of the phase transition of the vapor bubble in the shock-induced high pressure and high temperature conditions, but in general, the deformation of the outer surface of the droplet from the droplet is in basic agreement with the experiment.

Fig. B.4 present an interesting phenomenon that when the transverse jet is formed, there is a negative pressure region just in the liquid after the windward surface.

Fig. B.5 shows the comparison of the dimensionless displacements of the various interface for case 2 between experiment and numerical simulation. Fail to simulate the evolution of both the droplet and cavity, and which is ascribed to the ignorance of the phase change in the present code.

Case No.	1	2	3	4	5	6	7	8	9
diameter of droplet [mm]	1.232	1.375	1.135	1.538	3.038	2.218	1.554	1.295	1.143
diameter of cavity [mm]	0.679	0.857	0.750	1.231	2.712	2.055	1.143	0.911	0.679
eccentric distance [mm]	0.063	-0.027	0.000	0.058	0.029	-0.027	-0.116	0.112	0.196
post-shock flow speed [m/s]	457.8	577.0	480.6	480.6	481.6	456.0	511.7	512.0	510.0
Mach number	**2.08**	**2.43**	**2.15**	**2.15**	**2.15**	**2.08**	**2.24**	**2.24**	**2.23**
Reynolds number	**3372**	**5002**	**3261**	**4419**	**8747**	**6048**	**4873**	**4063**	**3556**
Weber number	**618**	**1278**	**649**	**880**	**1745**	**1103**	**1051**	**877**	**764**

Table B.1 The run conditions of all the experimental cases.

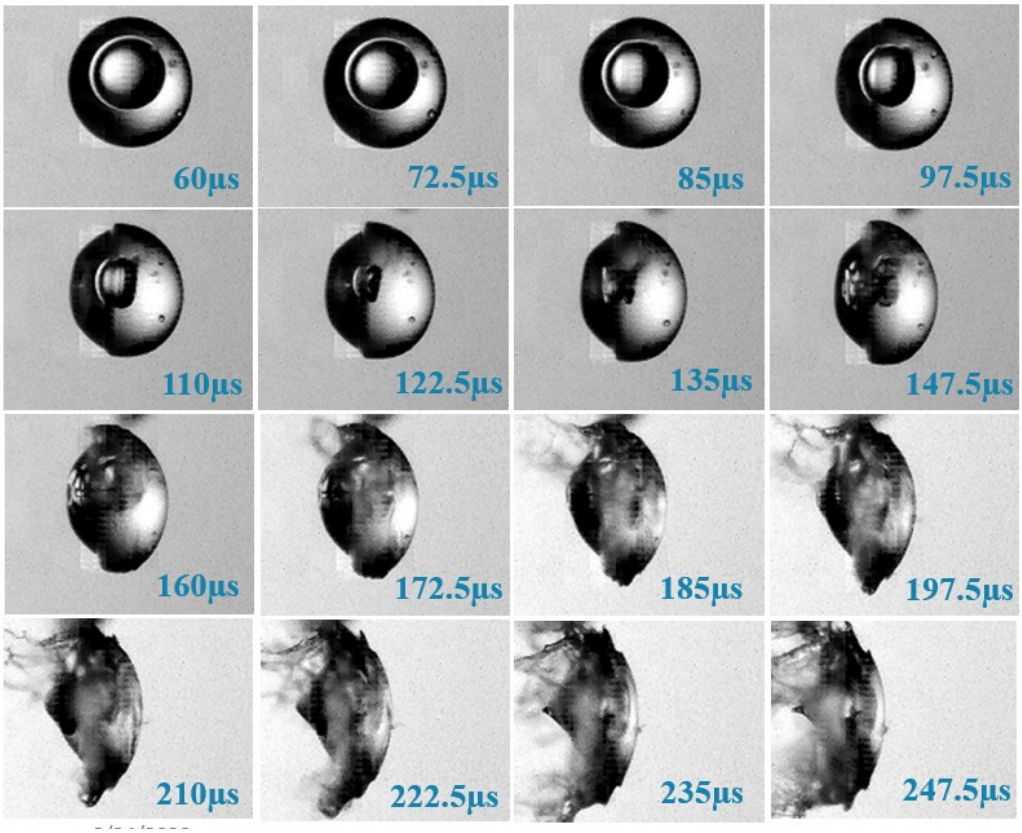


Figure B.3 The experimental photos of case 2 in the early stage of the breakup phenomena. The shock wave flows from right to left.

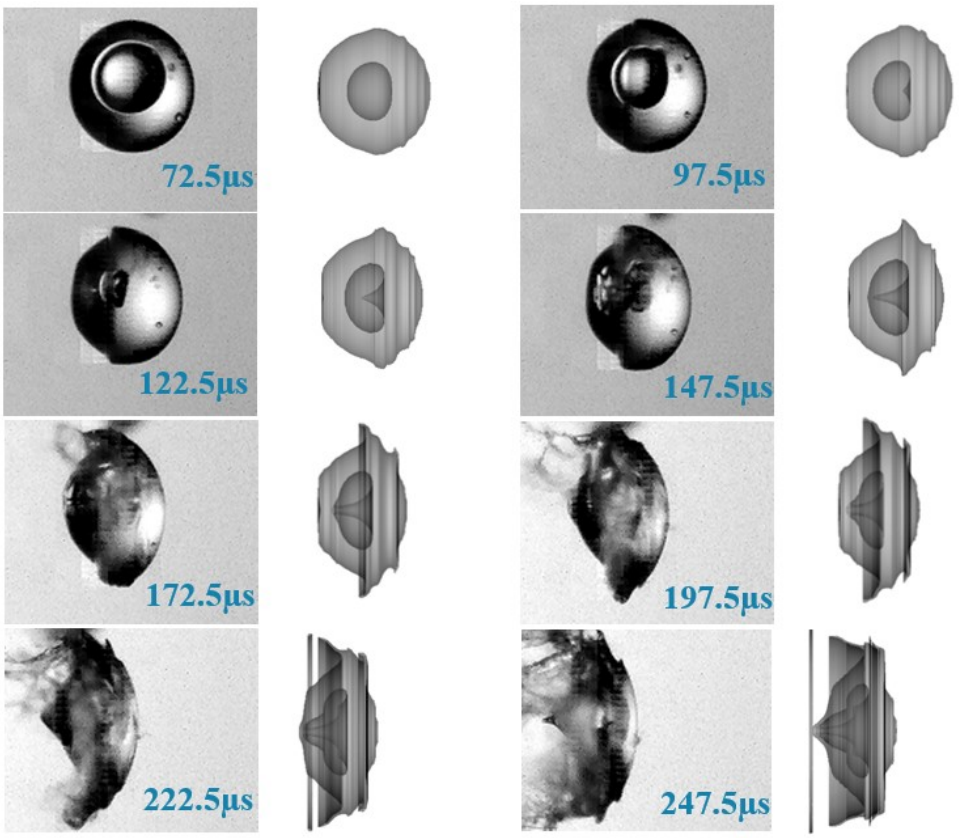


Figure B.4 Comparison of the experimental and numerical results for case 2.

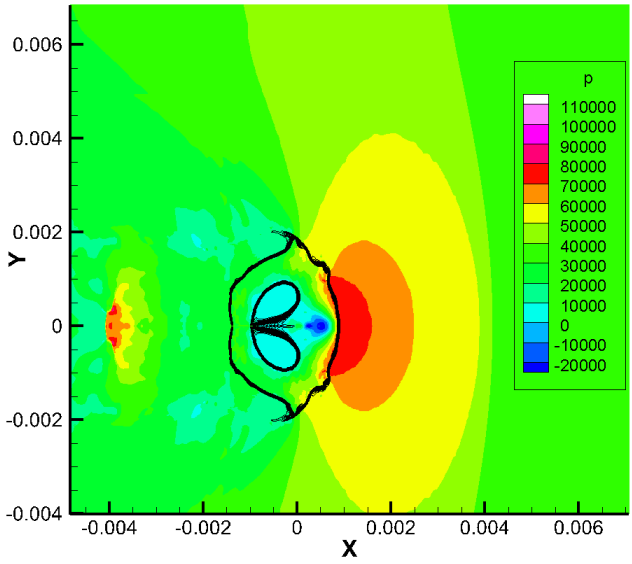


Figure B.5 The pressure counters for case 2 when the transverse jet is formed

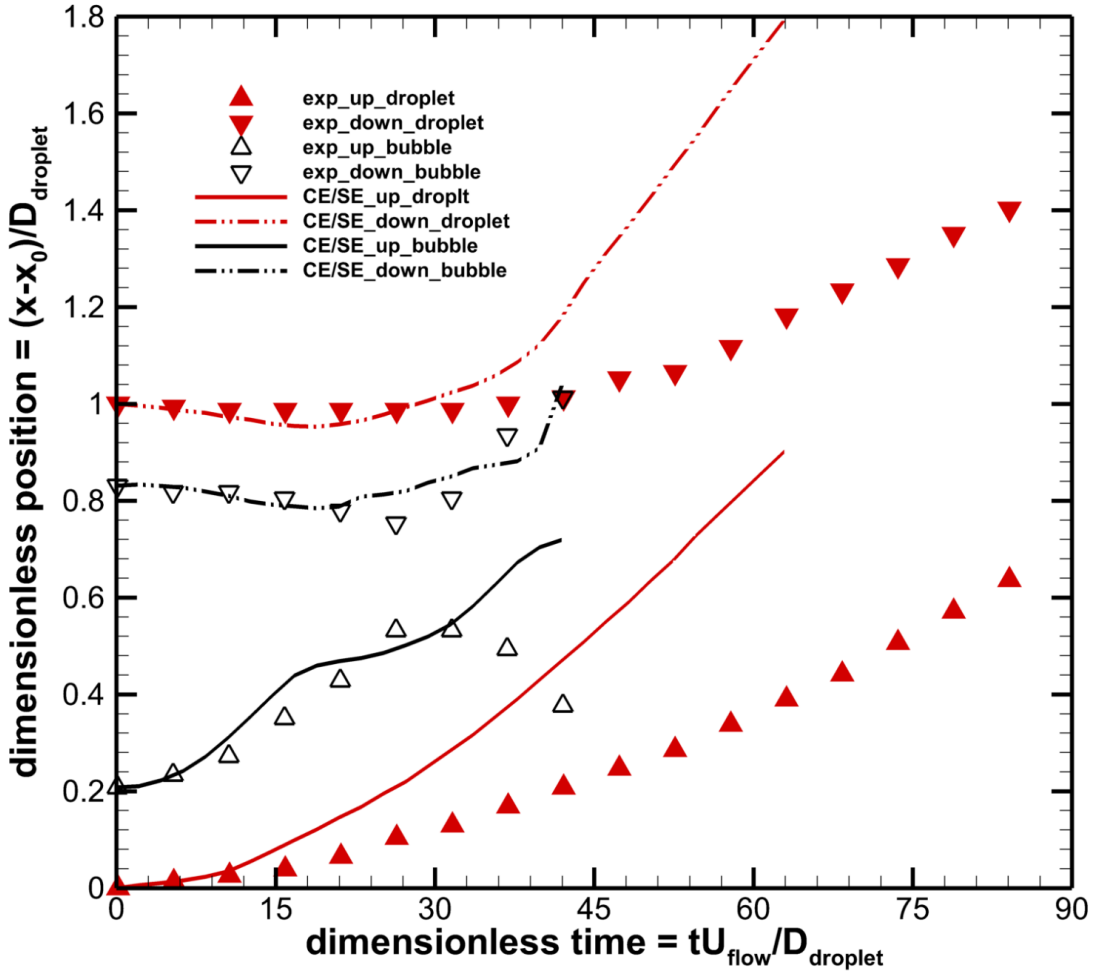


Figure B.6 Comparison of the dimensionless displacements of the various interface for case 2 between experiment and numerical simulation.

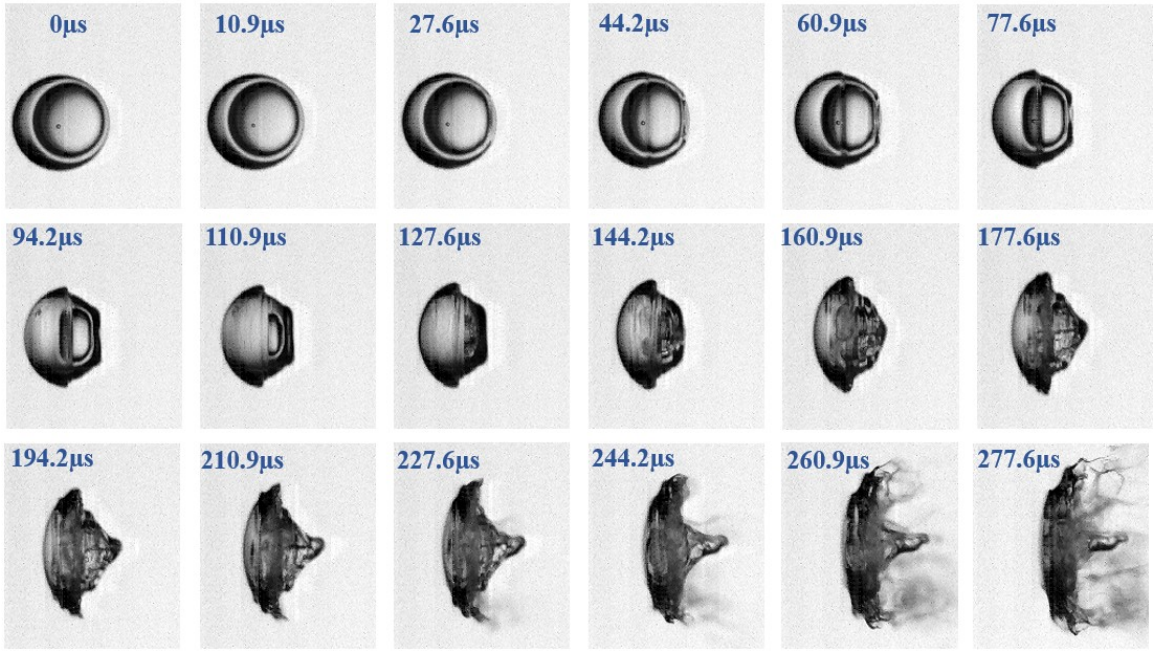


Figure B.7 The experimental photos of case 7 in the early stage of the breakup phenomena. The shock wave comes from left.

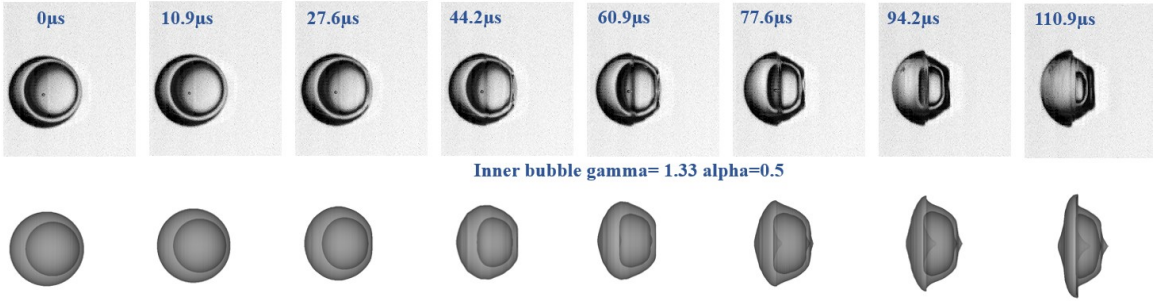


Figure B.8 Comparison of the experimental and CE/SE numerical results for case 7.

C. Illustration of the CE/SE Simulation Code for a Mach Number 2.4 Case with an Air Cavitation Bubble.

In order to record the latest developments in CE/SE simulation code to this day, the code for a Mach 2.4 case is appended as follows:

```
#include "stdafx.h"
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <omp.h>
#include <conio.h>
#include <ctype.h>
#include <string.h>
#include <time.h>

#define NumberOfThreads 39

#define time 7.0E-4
//computing time
#define Nx 1800
//grid number in x direction
#define Ny 480
//grid number in y direction
#define Neq 6
//Number of Equation
#define CFL_Number 0.4
//Courant number
#define WBAP_Parameter_n 5.0
#define SmallNumber 1e-15
#define pi 3.141592654
```

```
#define GAMA1 1.4
#define PAI1 0
#define GAMA2 7.15
#define PAI2 3.31e8
//EOS constant of materials
#define StoreFreq 500
//Output restore file & result file frequency
#define OutputFileType 1
//0--Ascii file 1--Binary type
#define is_symm 1
//algorithm selection: 1--symmetric flow(y=0 is axis of symmetry);0--plane flow
#define Xc 0.0
#define Yc 0.0
//coordinates of droplet's centroid (m)
#define DropletRadius 0.0015
#define innerDropletRadius 0.0008  // r/r0=0.75
//radius of droplet (m)
#define Ms 2.4
//shock Mach number
#define SurfaceTension 0.0
#define MAXIMUM 1.0
#define MINIMUM 0.0
//
#define RHOMIN 1E-15
#define PMIN 1E-15

#define rou1 0.06
#define u1 0
#define v1 0
#define p1 5000.00

#define rou2 1000.0
#define u2 0
#define v2 0
#define p2 5000.00
```

```
#define MAX(x,y) (((x)>(y))?(x):(y))
#define MIN(x,y) (((x)<(y))?(x):(y))

double ***U1, ***U2, ***Ux1, ***Uy1, ***Ux2, ***Uy2, ***Ut, ***F, ***Fy, ***Ft, ***G, ***Gx,
***Gt, ***S, ***Sx, ***Sy, ***St, **Curv, **Normx,**Normy;
double const Lx_S=-0.0135;  //-4.0*DropletRadius;
double const Lx_E=0.0135;  //8*DropletRadius;
double const Ly_S=0;
double const Ly_E=0.009;  //4*DropletRadius;
//size of domain
double const Xshock=-1.2*DropletRadius;
//shock position (m)
double const dx=double(Lx_E-Lx_S)/double(Nx),dy=double(Ly_E-Ly_S)/double(Ny);
double DeltaL=sqrt(dx*dx+dy*dy),sintheta=dy/DeltaL,costheta=dx/DeltaL;
//gird size
double InitialMass1,InitialMass2,TotalMass1,TotalMass2,InitialEnergy,TotalEnergy;

void ErrorHandler(int errcode);
void restoreall(char* sFName);
void restart(char* sFName);
void Initializer(void);
void BoundaryTreatment(double U[Nx+2][Ny+2][Neq],double Ux[Nx+2][Ny+2][Neq],double
Uy[Nx+2][Ny+2][Neq]);
void ComputeFluxesAndDerivatives(double ***U, double ***Ux, double ***Uy, int ishalf);
void Comput_Ut(double *U, double *Ux, double *Uy, double y, double *Qt);
bool Rotated_HLLC_RiemannSolver(double QL[Neq],double QR[Neq],double FHLLC[Neq],double
GridNormalx,double GridNormaly);
double WBAP_Limiter(double Theta1,double Theta2);
void CESE_2DTimeMarching(double ***U_old, double ***Ux_old, double ***Uy_old,
                         double ***U_new, double ***Ux_new, double ***Uy_new, double dt, int IsHalf);
void Compute_NormAndCurv(double ***U, int IsHalf);
void WriteAsciiData(char *filename,double U[Nx+2][Ny+2][Neq]);
void WriteBinaryData(char *filename,double U[Nx+2][Ny+2][Neq]);
void CESE_Solver(void);
void AllocateMemory(void);
void FreeMemory(void);
```

```
void AllocateMemory(void)
{
	int i,j;
	printf("Allocate memory...");
	U1=(double ***)malloc((Nx+2)*sizeof(double **));
	Ux1=(double ***)malloc((Nx+2)*sizeof(double **));
	Uy1=(double ***)malloc((Nx+2)*sizeof(double **));
	U2=(double ***)malloc((Nx+2)*sizeof(double **));
	Ux2=(double ***)malloc((Nx+2)*sizeof(double **));
	Uy2=(double ***)malloc((Nx+2)*sizeof(double **));
	Ut=(double ***)malloc((Nx+2)*sizeof(double **));
	F=(double ***)malloc((Nx+2)*sizeof(double **));
	Fy=(double ***)malloc((Nx+2)*sizeof(double **));
	Ft=(double ***)malloc((Nx+2)*sizeof(double **));
	G=(double ***)malloc((Nx+2)*sizeof(double **));
	Gx=(double ***)malloc((Nx+2)*sizeof(double **));
	Gt=(double ***)malloc((Nx+2)*sizeof(double **));
	S = (double ***)malloc((Nx + 2)*sizeof(double **));
	Sx = (double ***)malloc((Nx + 2)*sizeof(double **));
	Sy = (double ***)malloc((Nx + 2)*sizeof(double **));
	St = (double ***)malloc((Nx + 2)*sizeof(double **));
	Curv = (double **)malloc((Nx + 2)*sizeof(double *));
	Normx = (double **)malloc((Nx + 2)*sizeof(double *));
	Normy = (double **)malloc((Nx + 2)*sizeof(double *));
	for(i=0;i<Nx+2;i++)
	{
		U1[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Ux1[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Uy1[i]=(double **)malloc((Ny+2)*sizeof(double *));
		U2[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Ux2[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Uy2[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Ut[i]=(double **)malloc((Ny+2)*sizeof(double *));
		F[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Fy[i]=(double **)malloc((Ny+2)*sizeof(double *));
```

```
		Ft[i]=(double **)malloc((Ny+2)*sizeof(double *));
		G[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Gx[i]=(double **)malloc((Ny+2)*sizeof(double *));
		Gt[i]=(double **)malloc((Ny+2)*sizeof(double *));
		S[i] = (double **)malloc((Ny + 2)*sizeof(double *));
		Sx[i] = (double **)malloc((Ny + 2)*sizeof(double *));
		Sy[i] = (double **)malloc((Ny + 2)*sizeof(double *));
		St[i] = (double **)malloc((Ny + 2)*sizeof(double *));
		Curv[i] = (double *)malloc((Ny + 2)*sizeof(double));
		Normx[i] = (double *)malloc((Ny + 2)*sizeof(double));
		Normy[i] = (double *)malloc((Ny + 2)*sizeof(double));
		for(j=0;j<Ny+2;j++)
		{
			U1[i][j]=(double *)malloc((Neq)*sizeof(double));
			Ux1[i][j]=(double *)malloc((Neq)*sizeof(double));
			Uy1[i][j]=(double *)malloc((Neq)*sizeof(double));
			U2[i][j]=(double *)malloc((Neq)*sizeof(double));
			Ux2[i][j]=(double *)malloc((Neq)*sizeof(double));
			Uy2[i][j]=(double *)malloc((Neq)*sizeof(double));
			Ut[i][j]=(double *)malloc((Neq)*sizeof(double));
			F[i][j]=(double *)malloc((Neq)*sizeof(double));
			Fy[i][j]=(double *)malloc((Neq)*sizeof(double));
			Ft[i][j]=(double *)malloc((Neq)*sizeof(double));
			G[i][j]=(double *)malloc((Neq)*sizeof(double));
			Gx[i][j]=(double *)malloc((Neq)*sizeof(double));
			Gt[i][j]=(double *)malloc((Neq)*sizeof(double));
			S[i][j] = (double *)malloc((Neq)*sizeof(double));
			Sx[i][j] = (double *)malloc((Neq)*sizeof(double));
			Sy[i][j] = (double *)malloc((Neq)*sizeof(double));
			St[i][j] = (double *)malloc((Neq)*sizeof(double));
		}
	}
	printf("Done!\n");
}
void FreeMemory(void)
{
```

```
int i,j;
printf("Free memory...");
for(i=0;i<Nx+2;i++)
{
	for(j=0;j<Ny+2;j++)
	{
		free(U1[i][j]);
		free(Ux1[i][j]);
		free(Uy1[i][j]);
		free(U2[i][j]);
		free(Ux2[i][j]);
		free(Uy2[i][j]);
		free(Ut[i][j]);
		free(F[i][j]);
		free(Fy[i][j]);
		free(Ft[i][j]);
		free(G[i][j]);
		free(Gx[i][j]);
		free(Gt[i][j]);
		free(S[i][j]);
		free(Sx[i][j]);
		free(Sy[i][j]);
		free(St[i][j]);
	}
	free(U1[i]);
	free(Ux1[i]);
	free(Uy1[i]);
	free(U2[i]);
	free(Ux2[i]);
	free(Uy2[i]);
	free(Ut[i]);
	free(F[i]);
	free(Fy[i]);
	free(Ft[i]);
	free(G[i]);
	free(Gx[i]);
```

```
                free(Gt[i]);
                free(S[i]);
                free(Sx[i]);
                free(Sy[i]);
                free(St[i]);
                free(Curv[i]);
                free(Normx[i]);
                free(Normy[i]);
        }
        free(U1);
        free(Ux1);
        free(Uy1);
        free(U2);
        free(Ux2);
        free(Uy2);
        free(Ut);
        free(F);
        free(Fy);
        free(Ft);
        free(G);
        free(Gx);
        free(Gt);
        free(S);
        free(Sx);
        free(Sy);
        free(St);
        free(Curv);
        free(Normx);
        free(Normy);
        printf("Done!\n");
}

void ErrorHandler(int errcode)
{
        switch(errcode)
        {
```

```
		case 0:
			{
				printf("Restore file does not match!\n");
				printf("The present configuration is: Nx=%d,Ny=%d\n",Nx,Ny);
				exit(0);
			}
		case 1:
			{
				printf("Can not open the File\n");
				exit(0);
			}
		case 2:
			{
				printf("File not exist!\n");
				exit(0);
			}
		}
}
void restoreall(char* sFName,double& comput_time,int& comput_num)
{
		int i,j,k;
		FILE* fp;
		int AA;
   if((fp=fopen(sFName,"wb"))==NULL)ErrorHandler(1);
   printf("Restoring data...");
   fwrite(&comput_num,sizeof(int),1,fp);
   fwrite(&comput_time,sizeof(double),1,fp);
   AA=Nx;
   fwrite(&AA,sizeof(int),1,fp);
   AA=Ny;
   fwrite(&AA,sizeof(int),1,fp);
		for(i=0;i<=Nx;i++)
		{
			for(j=0;j<=Ny;j++)
			{
				for(k=0;k<Neq;k++)
```

```
                        {
                                fwrite(&U1[i][j][k],sizeof(double),1,fp);
                                fwrite(&Ux1[i][j][k],sizeof(double),1,fp);
                                fwrite(&Uy1[i][j][k],sizeof(double),1,fp);
                        }
                }
        }
        fclose(fp);
  printf("Done!\n");
}

void restart(char* sFName,double& comput_time,int& comput_num)
{
        FILE* fp;
        int i,j,k,A1,A2;
        if((fp=fopen(sFName,"rb"))==NULL)ErrorHandler(2);
  printf("Reading Data...");
        fread(&comput_num,sizeof(int),1,fp);
        fread(&comput_time,sizeof(double),1,fp);
        fread(&A1,sizeof(int),1,fp);
        fread(&A2,sizeof(int),1,fp);
        printf("The configuration of restart file is: Nx=%d,Ny=%d\n",A1,A2);
  if(A1!=Nx||A2!=Ny) ErrorHandler(0);
        for(i=0;i<=Nx;i++)
  {
    for(j=0;j<=Ny;j++)
    {
      for(k=0;k<Neq;k++)
      {
        fread(&U1[i][j][k],sizeof(double),1,fp);
        fread(&Ux1[i][j][k],sizeof(double),1,fp);
        fread(&Uy1[i][j][k],sizeof(double),1,fp);
      }
    }
  }
        fclose(fp);
```

```
	printf("Done!\n");
}
double ComputingTimeStep(double ***U)
{
	int i,j;
	double VOF1,gama,pai,density,x_velocity,y_velocity,pressure,SoundSpeed,MaximumWaveSpeed;
	MaximumWaveSpeed=0;
#pragma omp parallel for private(j,VOF1,gama,pai,density,x_velocity,y_velocity,pressure,SoundSpeed)
	for(i=0;i<Nx+1;i++)
	{
		for(j=0;j<Ny+1;j++)
			{
				VOF1=U[i][j][0];
				density=U[i][j][1]+U[i][j][2];
				x_velocity=U[i][j][3]/density;
				y_velocity=U[i][j][4]/density;
				gama=1.0/(VOF1/(GAMA1-1)+(1-VOF1)/(GAMA2-1))+1;
				pai=(VOF1*GAMA1*PAI1/(GAMA1-1)+(1-
VOF1)*GAMA2*PAI2/(GAMA2-1))*(gama-1)/gama;
				pressure=(gama-1)*(U[i][j][5]-
0.5*density*(x_velocity*x_velocity+y_velocity*y_velocity))-gama*pai;
				if(density<=0||(pressure+pai)<=0) continue;
#pragma omp critical
				{

				  SoundSpeed=sqrt(gama*(pressure+pai)/density);

MaximumWaveSpeed=MAX(MaximumWaveSpeed,SoundSpeed+sqrt(x_velocity*x_velocity+y_velocity*
y_velocity));
				}
			}
	}
	return CFL_Number*MIN(dx,dy)/MaximumWaveSpeed;
}
//Computing time step according to Courant number
```

```
void Initializer()
{
	int i,j,k;
	double x,y,alpha,rou,gama,pai;
	double xLD,yLD,xRD,yRD,xLU,yLU,xRU,yRU,d1,d2,d3,d4;
	double innerd1, innerd2, innerd3, innerd4;
	double rouf,uf,vf,pf,a;
	a=sqrt(GAMA1*(p1+PAI1)/rou1);
	rouf=rou1*(GAMA1+1)*Ms*Ms/(2+(GAMA1-1)*Ms*Ms);
	uf=(1-rou1/rouf)*Ms*a;
	vf=0;
	pf=p1*(2*GAMA1*Ms*Ms-(GAMA1-1))/(GAMA1+1);
#pragma omp parallel for
private(j,k,x,y,alpha,rou,gama,pai,xLD,yLD,xRD,yRD,xLU,yLU,xRU,yRU,d1,d2,d3,d4)
	for(i=0;i<Nx+1;i++)
	{
		x=Lx_S+i*dx;
		for(j=0;j<Ny+1;j++)
		{
			y=Ly_S+j*dy;
			if(x<Xshock)
			{
				U1[i][j][0]=1.0;
				U1[i][j][1]=rouf;
				U1[i][j][2]=0;
				U1[i][j][3]=rouf*uf;
				U1[i][j][4]=rouf*vf;
				U1[i][j][5]=(pf+GAMA1*PAI1)/(GAMA1-1)+0.5*rouf*(uf*uf+vf*vf);
			}
			else
			{
				xLD=xLU=x-0.5*dx;
				xRD=xRU=x+0.5*dx;
				yLD=yRD=y-0.5*dy;
				yLU=yRU=y+0.5*dy;
```

```
d1=sqrt(xLD*xLD+yLD*yLD)-DropletRadius;
d2=sqrt(xRD*xRD+yRD*yRD)-DropletRadius;
d3=sqrt(xRU*xRU+yRU*yRU)-DropletRadius;
d4=sqrt(xLU*xLU+yLU*yLU)-DropletRadius;

innerd1 = sqrt(xLD*xLD + yLD*yLD) - innerDropletRadius;
innerd2 = sqrt(xRD*xRD + yRD*yRD) - innerDropletRadius;
innerd3 = sqrt(xRU*xRU + yRU*yRU) - innerDropletRadius;
innerd4 = sqrt(xLU*xLU + yLU*yLU) - innerDropletRadius;

if(d1>=0&&d2>=0&&d3>=0&&d4>=0) alpha=1.0;
if(d1<0&&d2<0&&d3<0&&d4<0) alpha=0.0;
{
        if (innerd1<0 && innerd2<0 && innerd3<0 && innerd4<0)
alpha = 1.0;

        if (innerd1<0 && innerd2 >= 0 && innerd3 >= 0 &&
innerd4 >= 0) alpha = 0.5*innerd1*innerd1 / (innerd2 - innerd1) / (innerd4 - innerd1);
        if (innerd1 >= 0 && innerd2<0 && innerd3 >= 0 &&
innerd4 >= 0) alpha = 0.5*innerd2*innerd2 / (innerd3 - innerd2) / (innerd1 - innerd2);
        if (innerd1 >= 0 && innerd2 >= 0 && innerd3<0 &&
innerd4 >= 0) alpha = 0.5*innerd3*innerd3 / (innerd4 - innerd3) / (innerd2 - innerd3);
        if (innerd1 >= 0 && innerd2 >= 0 && innerd3 >= 0 &&
innerd4<0) alpha = 0.5*innerd4*innerd4 / (innerd1 - innerd4) / (innerd3 - innerd4);

        if (innerd1 >= 0 && innerd2<0 && innerd3<0 && innerd4<0)
alpha = 1.0 -0.5*innerd1*innerd1 / (innerd2 - innerd1) / (innerd4 - innerd1);
        if (innerd1<0 && innerd2 >= 0 && innerd3<0 && innerd4<0)
alpha = 1.0 -0.5*innerd2*innerd2 / (innerd3 - innerd2) / (innerd1 - innerd2);
        if (innerd1<0 && innerd2<0 && innerd3 >= 0 && innerd4<0)
alpha = 1.0 -0.5*innerd3*innerd3 / (innerd4 - innerd3) / (innerd2 - innerd3);
        if (innerd1<0 && innerd2<0 && innerd3<0 && innerd4 >= 0)
alpha = 1.0 -0.5*innerd4*innerd4 / (innerd1 - innerd4) / (innerd3 - innerd4);

        if (innerd1<0 && innerd2 >= 0 && innerd3 >= 0 &&
innerd4<0) alpha = 1.0 -0.5*(innerd2 / (innerd2 - innerd1) + innerd3 / (innerd3 - innerd4));
```

```
                    if (innerd1 >= 0 && innerd2<0 && innerd3<0 && innerd4 >=
0) alpha = 1.0 -0.5*(innerd1 / (innerd1 - innerd2) + innerd4 / (innerd4 - innerd3));
                    if (innerd1 >= 0 && innerd2 >= 0 && innerd3<0 &&
innerd4<0) alpha = 1.0 -0.5*(innerd1 / (innerd1 - innerd4) + innerd2 / (innerd2 - innerd3));
                    if (innerd1<0 && innerd2<0 && innerd3 >= 0 && innerd4 >=
0) alpha = 1.0-0.5*(innerd4 / (innerd4 - innerd1) + innerd3 / (innerd3 - innerd2));


                }

                if(d1<0&&d2>=0&&d3>=0&&d4>=0)        alpha=1-0.5*d1*d1/(d2-
d1)/(d4-d1);
                if(d1>=0&&d2<0&&d3>=0&&d4>=0)        alpha=1-0.5*d2*d2/(d3-
d2)/(d1-d2);
                if(d1>=0&&d2>=0&&d3<0&&d4>=0)        alpha=1-0.5*d3*d3/(d4-
d3)/(d2-d3);
                if(d1>=0&&d2>=0&&d3>=0&&d4<0)        alpha=1-0.5*d4*d4/(d1-
d4)/(d3-d4);

                if(d1>=0&&d2<0&&d3<0&&d4<0) alpha=0.5*d1*d1/(d2-d1)/(d4-d1);
                if(d1<0&&d2>=0&&d3<0&&d4<0) alpha=0.5*d2*d2/(d3-d2)/(d1-d2);
                if(d1<0&&d2<0&&d3>=0&&d4<0) alpha=0.5*d3*d3/(d4-d3)/(d2-d3);
                if(d1<0&&d2<0&&d3<0&&d4>=0) alpha=0.5*d4*d4/(d1-d4)/(d3-d4);

                if(d1<0&&d2>=0&&d3>=0&&d4<0)   alpha=0.5*(d2/(d2-d1)+d3/(d3-
d4));
                if(d1>=0&&d2<0&&d3<0&&d4>=0)   alpha=0.5*(d1/(d1-d2)+d4/(d4-
d3));
                if(d1>=0&&d2>=0&&d3<0&&d4<0)   alpha=0.5*(d1/(d1-d4)+d2/(d2-
d3));
                if(d1<0&&d2<0&&d3>=0&&d4>=0)   alpha=0.5*(d4/(d4-d1)+d3/(d3-
d2));

                U1[i][j][0]=alpha;
                U1[i][j][1]=rou1*alpha;
                U1[i][j][2]=rou2*(1-alpha);
```

```
                                        rou=U1[i][j][1]+U2[i][j][2];
                                        gama=1.0/(alpha/(GAMA1-1)+(1-alpha)/(GAMA2-1))+1;
                                        pai=(alpha*GAMA1*PAI1/(GAMA1-1)+(1-
alpha)*GAMA2*PAI2/(GAMA2-1))*(gama-1)/gama;
                                        U1[i][j][3]=rou*u1;
                                        U1[i][j][4]=rou*v1;
                                        U1[i][j][5]=(p1+gama*pai)/(gama-1)+0.5*rou*(u1*u1+v1*v1);
                                }
                                for(k=0;k<Neq;k++)
                                {
                                        Ux1[i][j][k]=0;
                                        Uy1[i][j][k]=0;
                                }
                        }
                }
}
/*void Initializer()
{
        int i,j,k;
        double x,y;
        double rouf,uf,vf,pf,a;
        a=sqrt(GAMA1*(p1+PAI1)/rou1);
        rouf=rou1*(GAMA1+1)*Ms*Ms/(2+(GAMA1-1)*Ms*Ms);
        uf=(1-rou1/rouf)*Ms*a;
        vf=0;
        pf=p1*(2*GAMA1*Ms*Ms-(GAMA1-1))/(GAMA1+1);
#pragma omp parallel for private(j,k,x,y)
        for(i=0;i<Nx+1;i++)
        {
                x=Lx_S+i*dx;
                for(j=0;j<Ny+1;j++)
                {
                        y=Ly_S+j*dy;
                        if(x<Xshock)
                        {
                                U1[i][j][0]=1.0;
```

```
                    U1[i][j][1]=rouf;
                    U1[i][j][2]=0;
                    U1[i][j][3]=rouf*uf;
                    U1[i][j][4]=rouf*vf;
                    U1[i][j][5]=(pf+GAMA1*PAI1)/(GAMA1-1)+0.5*rouf*(uf*uf+vf*vf);
                }
                else
                {
                    if(sqrt(x*x+y*y)>=DropletRadius)
                    {
                        U1[i][j][0]=1.0;
                        U1[i][j][1]=rou1;
                        U1[i][j][2]=0;
                        U1[i][j][3]=rou1*u1;
                        U1[i][j][4]=rou1*v1;
                        U1[i][j][5]=(p1+GAMA1*PAI1)/(GAMA1-
1)+0.5*rou1*(u1*u1+v1*v1);
                    }
                    else
                    {
                        U1[i][j][0]=0.0;
                        U1[i][j][1]=0.0;
                        U1[i][j][2]=rou2;
                        U1[i][j][3]=rou2*u2;
                        U1[i][j][4]=rou2*v2;
                        U1[i][j][5]=(p2+GAMA2*PAI2)/(GAMA2-
1)+0.5*rou2*(u2*u2+v2*v2);
                    }
                }
                for(k=0;k<Neq;k++)
                {
                    Ux1[i][j][k]=0;
                    Uy1[i][j][k]=0;
                }
            }
    }
```

```
}
//Initialization*/
void BoundaryTreatment()
{
	int i,j,k;
#pragma omp parallel private(k)
	{
#pragma omp for
		for(j=1;j<=Ny;j++)
		{
			for(k=0;k<Neq;k++)
			{
				U2[0][j][k]=U2[1][j][k];
				Ux2[0][j][k]=0;
				Uy2[0][j][k]=0;
			}
		}//Left boundary-inflow
#pragma omp for
		for(j=1;j<=Ny;j++)
		{
			for(k=0;k<Neq;k++)
			{
				U2[Nx+1][j][k]=U2[Nx][j][k];
				Ux2[Nx+1][j][k]=0;
				Uy2[Nx+1][j][k]=0;
			}
		}//Right boundary-nonreflection
#pragma omp for
		for(i=0;i<=Nx+1;i++)
		{
			for(k=0;k<Neq;k++)
			{
				if(k==-4)
				{
					U2[i][Ny+1][k]=-U2[i][Ny][k];
					Ux2[i][Ny+1][k]=-Ux2[i][Ny][k];
```

```
                                        Uy2[i][Ny+1][k]=Uy2[i][Ny][k];
                                }
                                else
                                {
                                        U2[i][Ny+1][k]=U2[i][Ny][k];
                                        Ux2[i][Ny+1][k]=0;
                                        Uy2[i][Ny+1][k]=0;
                                }
                        }
                }
                //Upper boundary-nonreflection
#pragma omp for
                for(i=0;i<=Nx+1;i++)
                {
                        for(k=0;k<Neq;k++)
                        {
                                if(k==4)
                                {
                                        U2[i][0][k]=-U2[i][1][k];
                                        Ux2[i][0][k]=-Ux2[i][1][k];
                                        Uy2[i][0][k]=Uy2[i][1][k];
                                }
                                else
                                {
                                        U2[i][0][k]=U2[i][1][k];
                                        Ux2[i][0][k]=Ux2[i][1][k];
                                        Uy2[i][0][k]=-Uy2[i][1][k];
                                }
                        }
                }
                //Lower boundary-symmetry
        }
}


void ComputeFluxesAndDerivatives(double ***U,double ***Ux,double ***Uy,int ishalf)
```

```
{
	int i,j,k;
	double
y,VOF1,rou,u,v,p,gama,pai,VOF1x,roux,ux,vx,px,gamax,paix,VOF1y,rouy,uy,vy,py,gamay,paiy,VOF1t,ro
ut,ut,vt,pt,gamat,pait;
#pragma omp parallel for
private(j,k,y,VOF1,rou,u,v,p,gama,pai,VOF1x,roux,ux,vx,px,gamax,paix,VOF1y,rouy,uy,vy,py,gamay,pai
y,VOF1t,rout,ut,vt,pt,gamat,pait)
	for(i=0;i<Nx+2;i++)
	{
		for(j=0;j<Ny+2;j++)
		{
			y = Ly_S + j*dy-ishalf*0.5*dy;
			VOF1=U[i][j][0];
			rou=U[i][j][1]+U[i][j][2];
			u=U[i][j][3]/rou;
			v=U[i][j][4]/rou;
			gama=1.0/(VOF1/(GAMA1-1)+(1-VOF1)/(GAMA2-1))+1;
			pai=(VOF1*GAMA1*PAI1/(GAMA1-1)+(1-VOF1)*GAMA2*PAI2/(GAMA2-
1))*(gama-1)/gama;
			p=(gama-1)*(U[i][j][5]-0.5*rou*(u*u+v*v))-gama*pai;
			VOF1x=Ux[i][j][0];
			roux=Ux[i][j][1]+Ux[i][j][2];
			ux=Ux[i][j][3]/rou-U[i][j][3]/rou/rou*roux;
			vx=Ux[i][j][4]/rou-U[i][j][4]/rou/rou*roux;
			gamax=-(1.0/(GAMA1-1)-1.0/(GAMA2-1))*(gama-1)*(gama-1)*VOF1x;
			paix=(GAMA1*PAI1/(GAMA1-1)-GAMA2*PAI2/(GAMA2-
1))*VOF1x*(gama-1)/gama+pai/(gama-1)/gama*gamax;
			px=gamax*(U[i][j][5]-0.5*rou*(u*u+v*v))+(gama-1)*(Ux[i][j][5]-
0.5*roux*(u*u+v*v)-rou*(u*ux+v*vx))-gamax*pai-gama*paix;

			VOF1y = Uy[i][j][0];
			rouy = Uy[i][j][1] + Uy[i][j][2];
			uy = Uy[i][j][3] / rou - U[i][j][3] / rou / rou*rouy;
			vy = Uy[i][j][4] / rou - U[i][j][4] / rou / rou*rouy;
```

```
                    gamay = -(1.0 / (GAMA1 - 1) - 1.0 / (GAMA2 - 1))*(gama - 1)*(gama -
1)*VOF1y;
                    paiy = (GAMA1*PAI1 / (GAMA1 - 1) - GAMA2*PAI2 / (GAMA2 -
1))*VOF1y*(gama - 1) / gama + pai / (gama - 1) / gama*gamay;
                    py = gamay*(U[i][j][5] - 0.5*rou*(u*u + v*v)) + (gama - 1)*(Uy[i][j][5] -
0.5*rouy*(u*u + v*v) - rou*(u*uy + v*vy)) - gamay*pai - gama*paiy;

                    F[i][j][1]=U[i][j][1]*u;
                    F[i][j][2]=U[i][j][2]*u;
                    F[i][j][3]=U[i][j][3]*u+p;
                    F[i][j][4]=U[i][j][4]*u;
                    F[i][j][5]=(U[i][j][5]+p)*u;

                    G[i][j][1]=U[i][j][1]*v;
                    G[i][j][2]=U[i][j][2]*v;
                    G[i][j][3]=U[i][j][3]*v;
                    G[i][j][4]=U[i][j][4]*v+p;
                    G[i][j][5]=(U[i][j][5]+p)*v;

                    if (is_symm&&y>0.1*dy)
                    {
                        S[i][j][1] = -U[i][j][1] * v/y;
                        S[i][j][2] = -U[i][j][2] * v/y;
                        S[i][j][3] = -U[i][j][3] * v / y - 2*SurfaceTension*Curv[i][j] *
Normx[i][j];
                        S[i][j][4] = -U[i][j][4] * v / y - 2*SurfaceTension*Curv[i][j] *
Normy[i][j];
                        S[i][j][5] = -(U[i][j][5] + p)*v/y;

                        for (k = 0; k < Neq-1; k++)
                        {
                            Sx[i][j][k] = -(Ux[i][j][k] * v + U[i][j][k] * vx) / y;
                            Sy[i][j][k] = -(Uy[i][j][k] * v + U[i][j][k] * vy) / y + U[i][j][k]
* v / y/y;
                        }
                        Sx[i][j][5] = -(Ux[i][j][5] + px)*v / y - (U[i][j][5] + p)*vx / y;
```

```
                Sy[i][j][5] = -(Uy[i][j][5] + py)*v / y - (U[i][j][5] + p)*vy / y + (U[i][j][5]
+ p)*v / y/y;
            }
            else
            {
                S[i][j][1] = 0;
                S[i][j][2] = 0;
                S[i][j][3] = -SurfaceTension*Curv[i][j]*Normx[i][j];
                S[i][j][4] = -SurfaceTension*Curv[i][j]*Normy[i][j];
                S[i][j][5] = 0;
                for (k = 0; k < Neq; k++) Sx[i][j][k] = Sy[i][j][k] = 0;
            }

            Gx[i][j][1]=Ux[i][j][1]*v+U[i][j][1]*vx;
            Gx[i][j][2]=Ux[i][j][2]*v+U[i][j][2]*vx;
            Gx[i][j][3]=Ux[i][j][3]*v+U[i][j][3]*vx;
            Gx[i][j][4]=Ux[i][j][4]*v+U[i][j][4]*vx+px;
            Gx[i][j][5]=(Ux[i][j][5]+px)*v+(U[i][j][5]+p)*vx;

            Fy[i][j][1]=Uy[i][j][1]*u+U[i][j][1]*uy;
            Fy[i][j][2]=Uy[i][j][2]*u+U[i][j][2]*uy;
            Fy[i][j][3]=Uy[i][j][3]*u+U[i][j][3]*uy+py;
            Fy[i][j][4]=Uy[i][j][4]*u+U[i][j][4]*uy;
            Fy[i][j][5]=(Uy[i][j][5]+py)*u+(U[i][j][5]+p)*uy;


            Ut[i][j][0]=-u*Ux[i][j][0]-v*Uy[i][j][0];
            Ut[i][j][1] = -(Ux[i][j][1] * u + U[i][j][1] * ux) - (Uy[i][j][1] * v + U[i][j][1] * vy)
+ S[i][j][1];
            Ut[i][j][2] = -(Ux[i][j][2] * u + U[i][j][2] * ux) - (Uy[i][j][2] * v + U[i][j][2] * vy)
+ S[i][j][2];
            Ut[i][j][3] = -(Ux[i][j][3] * u + U[i][j][3] * ux + px) - (Uy[i][j][3] * v + U[i][j][3]
* vy) + S[i][j][3];
            Ut[i][j][4] = -(Ux[i][j][4] * u + U[i][j][4] * ux) - (Uy[i][j][4] * v + U[i][j][4] * vy
+ py) + S[i][j][4];
```

```
                        Ut[i][j][5] = -(Ux[i][j][5] + px)*u - (U[i][j][5] + p)*ux - (Uy[i][j][5] + py)*v -
(U[i][j][5] + p)*vy + S[i][j][5];

                        VOF1t=Ut[i][j][0];
                        rout=Ut[i][j][1]+Ut[i][j][2];
                        ut=Ut[i][j][3]/rou-U[i][j][3]/rou/rou*rout;
                        vt=Ut[i][j][4]/rou-U[i][j][4]/rou/rou*rout;
                        gamat=-(1.0/(GAMA1-1)-1.0/(GAMA2-1))*(gama-1)*(gama-1)*VOF1t;
                        pait=(GAMA1*PAI1/(GAMA1-1)-GAMA2*PAI2/(GAMA2-
1))*VOF1t*(gama-1)/gama+pai/(gama-1)/gama*gamat;
                        pt=gamat*(U[i][j][5]-0.5*rou*(u*u+v*v))+(gama-1)*(Ut[i][j][5]-
0.5*rout*(u*u+v*v)-rou*(u*ut+v*vt))-gamat*pai-gama*pait;
                        Ft[i][j][1]=Ut[i][j][1]*u+U[i][j][1]*ut;
                        Ft[i][j][2]=Ut[i][j][2]*u+U[i][j][2]*ut;
                        Ft[i][j][3]=Ut[i][j][3]*u+U[i][j][3]*ut+pt;
                        Ft[i][j][4]=Ut[i][j][4]*u+U[i][j][4]*ut;
                        Ft[i][j][5]=(Ut[i][j][5]+pt)*u+(U[i][j][5]+p)*ut;

                        Gt[i][j][1]=Ut[i][j][1]*v+U[i][j][1]*vt;
                        Gt[i][j][2]=Ut[i][j][2]*v+U[i][j][2]*vt;
                        Gt[i][j][3]=Ut[i][j][3]*v+U[i][j][3]*vt;
                        Gt[i][j][4]=Ut[i][j][4]*v+U[i][j][4]*vt+pt;
                        Gt[i][j][5]=(Ut[i][j][5]+pt)*v+(U[i][j][5]+p)*vt;

                        if (is_symm&&y>0.1*dy)
                        {
                                for (k = 0; k < Neq - 1; k++) St[i][j][k] = -(Ut[i][j][k] * v + U[i][j][k] *
vt) / y;
                                St[i][j][5] = -(Ut[i][j][5] + pt)*v / y - (U[i][j][5] + p)*vt / y;
                        }
                        else
                        {
                                for (k = 0; k < Neq; k++) St[i][j][k] = 0;
                        }
                }
        }
```

```
}
void Comput_Ut(double *U, double *Ux, double *Uy, double y,double Curvature,double nx,double ny, double *Qt)
{
   double VOF1,rou,u,v,p,gama,pai,VOF1x,roux,ux,vx,px,gamax,paix,VOF1y,rouy,uy,vy,py,gamay,paiy;
          VOF1=U[0];
          rou=U[1]+U[2];
          u=U[3]/rou;
          v=U[4]/rou;
          gama=1.0/(VOF1/(GAMA1-1)+(1-VOF1)/(GAMA2-1))+1;
          pai=(VOF1*GAMA1*PAI1/(GAMA1-1)+(1-VOF1)*GAMA2*PAI2/(GAMA2-1))*(gama-1)/gama;
          p=(gama-1)*(U[5]-0.5*rou*(u*u+v*v))-gama*pai;
          VOF1x=Ux[0];
          roux=Ux[1]+Ux[2];
          ux=Ux[3]/rou-U[3]/rou/rou*roux;
          vx=Ux[4]/rou-U[4]/rou/rou*roux;
          gamax=-(1.0/(GAMA1-1)-1.0/(GAMA2-1))*(gama-1)*(gama-1)*VOF1x;
          paix=(GAMA1*PAI1/(GAMA1-1)-GAMA2*PAI2/(GAMA2-1))*VOF1x*(gama-1)/gama+pai/(gama-1)/gama*gamax;
          px=gamax*(U[5]-0.5*rou*(u*u+v*v))+(gama-1)*(Ux[5]-0.5*roux*(u*u+v*v)-rou*(u*ux+v*vx))-gamax*pai-gama*paix;
          VOF1y=Uy[0];
          rouy=Uy[1]+Uy[2];
          uy=Uy[3]/rou-U[3]/rou/rou*rouy;
          vy=Uy[4]/rou-U[4]/rou/rou*rouy;
          gamay=-(1.0/(GAMA1-1)-1.0/(GAMA2-1))*(gama-1)*(gama-1)*VOF1y;
          paiy=(GAMA1*PAI1/(GAMA1-1)-GAMA2*PAI2/(GAMA2-1))*VOF1y*(gama-1)/gama+pai/(gama-1)/gama*gamay;
          py=gamay*(U[5]-0.5*rou*(u*u+v*v))+(gama-1)*(Uy[5]-0.5*rouy*(u*u+v*v)-rou*(u*uy+v*vy))-gamay*pai-gama*paiy;
          if (is_symm&&y > 0.1*dy)
          {
                    Qt[0] = -u*Ux[0] - v*Uy[0];
                    Qt[1] = -(Ux[1] * u + U[1] * ux) - (Uy[1] * v + U[1] * vy) - U[1] * v / y;
                    Qt[2] = -(Ux[2] * u + U[2] * ux) - (Uy[2] * v + U[2] * vy) - U[2] * v / y;
```

```
		Qt[3] = -(Ux[3] * u + U[3] * ux + px) - (Uy[3] * v + U[3] * vy) - U[3] * v / y-
2*SurfaceTension*Curvature*nx;
		Qt[4] = -(Ux[4] * u + U[4] * ux) - (Uy[4] * v + U[4] * vy + py) - U[4] * v / y -
2*SurfaceTension*Curvature*ny;
		Qt[5] = -(Ux[5] + px)*u - (U[5] + p)*ux - (Uy[5] + py)*v - (U[5] + p)*vy - (U[5]+p) * v /
y;
	}
	else
	{
		Qt[0] = -u*Ux[0] - v*Uy[0];
		Qt[1] = -(Ux[1] * u + U[1] * ux) - (Uy[1] * v + U[1] * vy);
		Qt[2] = -(Ux[2] * u + U[2] * ux) - (Uy[2] * v + U[2] * vy);
		Qt[3] = -(Ux[3] * u + U[3] * ux + px) - (Uy[3] * v + U[3] * vy) -
SurfaceTension*Curvature*nx;
		Qt[4] = -(Ux[4] * u + U[4] * ux) - (Uy[4] * v + U[4] * vy + py) -
SurfaceTension*Curvature*ny;
		Qt[5] = -(Ux[5] + px)*u - (U[5] + p)*ux - (Uy[5] + py)*v - (U[5] + p)*vy;
	}
}

bool Rotated_HLLC_RiemannSolver(double QL[Neq],double QR[Neq],double FHLLC[Neq],double
GridNormalx,double GridNormaly)
{
	int k,m;
	double
gamaL,paiL,alphaL,rouL,uL,vL,pL,EL,aL,SL,qL,rL,gamaR,paiR,alphaR,rouR,uR,vR,pR,ER,aR,SR,qR,rR,
FL[Neq],FR[Neq];
	double S_star,ratioL_star,QL_star[Neq],ratioR_star,QR_star[Neq];
	double cBar,DeltaU,DeltaV,DeltaUV,Lamda[2],RotatedNormal[2][2];
	alphaL=QL[0];
	rouL=QL[1]+QL[2];
	uL=QL[3]/rouL;
	vL=QL[4]/rouL;
	gamaL=1.0/(alphaL/(GAMA1-1)+(1-alphaL)/(GAMA2-1))+1;
	paiL=(alphaL*GAMA1*PAI1/(GAMA1-1)+(1-alphaL)*GAMA2*PAI2/(GAMA2-1))*(gamaL-
1)/gamaL;
```

```
EL=QL[5];
pL=(gamaL-1)*(QL[5]-0.5*rouL*(uL*uL+vL*vL))-gamaL*paiL;
if (gamaL <= SmallNumber || rouL <= RHOMIN || (pL + paiL) <= PMIN)          return false;
aL=sqrt(gamaL*(pL+paiL)/rouL);
alphaR=QR[0];
rouR=QR[1]+QR[2];
uR=QR[3]/rouR;
vR=QR[4]/rouR;
gamaR=1.0/(alphaR/(GAMA1-1)+(1-alphaR)/(GAMA2-1))+1;
paiR=(alphaR*GAMA1*PAI1/(GAMA1-1)+(1-alphaR)*GAMA2*PAI2/(GAMA2-1))*(gamaR-
1)/gamaR;
ER=QR[5];
pR=(gamaR-1)*(QR[5]-0.5*rouR*(uR*uR+vR*vR))-gamaR*paiR;
if (gamaR<=SmallNumber||rouR <= RHOMIN || (pR + paiR) <= PMIN) return false;
aR=sqrt(gamaR*(pR+paiR)/rouR);
DeltaU=uR-uL;
DeltaV=vR-vL;
DeltaUV=sqrt(DeltaU*DeltaU+DeltaV*DeltaV);
cBar=0.5*(aL+aR);
if(DeltaUV<=1e-3*cBar)
{
        RotatedNormal[0][0]=GridNormalx;
        RotatedNormal[0][1]=GridNormaly;
        RotatedNormal[1][0]=-RotatedNormal[0][1];
        RotatedNormal[1][1]=RotatedNormal[0][0];//0--nx,1--ny
}
else
{
        RotatedNormal[0][0]=DeltaU/DeltaUV;
        RotatedNormal[0][1]=DeltaV/DeltaUV;
        RotatedNormal[1][0]=-RotatedNormal[0][1];
        RotatedNormal[1][1]=RotatedNormal[0][0];
}
for(k=0;k<Neq;k++)
{
        FHLLC[k]=0;
```

```
	}
	for(m=0;m<2;m++)
	{
		Lamda[m]=GridNormalx*RotatedNormal[m][0]+GridNormaly*RotatedNormal[m][1];
		if(Lamda[m]==0)
		{
			continue;
		}
		else
		{
			if(Lamda[m]<0)
			{
				RotatedNormal[m][0]=-RotatedNormal[m][0];
				RotatedNormal[m][1]=-RotatedNormal[m][1];
			}
			qL=uL*RotatedNormal[m][0]+vL*RotatedNormal[m][1];
			rL=-uL*RotatedNormal[m][1]+vL*RotatedNormal[m][0];
			qR=uR*RotatedNormal[m][0]+vR*RotatedNormal[m][1];
			rR=-uR*RotatedNormal[m][1]+vR*RotatedNormal[m][0];
			SL=MIN(qL-aL,qR-aR);
			SR=MAX(qL+aL,qR+aR);
			FL[1]=QL[1]*qL;
			FL[2]=QL[2]*qL;
			FL[3]=QL[3]*qL+pL*RotatedNormal[m][0];
			FL[4]=QL[4]*qL+pL*RotatedNormal[m][1];
			FL[5]=(EL+pL)*qL;
			FR[1]=QR[1]*qR;
			FR[2]=QR[2]*qR;
			FR[3]=QR[3]*qR+pR*RotatedNormal[m][0];
			FR[4]=QR[4]*qR+pR*RotatedNormal[m][1];
			FR[5]=(ER+pR)*qR;
			S_star=(pR-pL+rouL*qL*(SL-qL)-rouR*qR*(SR-qR))/(rouL*(SL-qL)-
rouR*(SR-qR));
			ratioL_star=(SL-qL)/(SL-S_star);
			QL_star[1]=ratioL_star*QL[1];
			QL_star[2]=ratioL_star*QL[2];
```

```
                    QL_star[3]=ratioL_star*rouL*(S_star*RotatedNormal[m][0]-
rL*RotatedNormal[m][1]);

        QL_star[4]=ratioL_star*rouL*(S_star*RotatedNormal[m][1]+rL*RotatedNormal[m][0]);
                    QL_star[5]=ratioL_star*rouL*(EL/rouL+(S_star-qL)*(S_star+pL/rouL/(SL-
qL)));
                    ratioR_star=(SR-qR)/(SR-S_star);
                    QR_star[1]=ratioR_star*QR[1];
                    QR_star[2]=ratioR_star*QR[2];
                    QR_star[3]=ratioR_star*rouR*(S_star*RotatedNormal[m][0]-
rR*RotatedNormal[m][1]);

        QR_star[4]=ratioR_star*rouR*(S_star*RotatedNormal[m][1]+rR*RotatedNormal[m][0]);
                    QR_star[5]=ratioR_star*rouR*(ER/rouR+(S_star-qR)*(S_star+pR/rouR/(SR-
qR)));
                    if(SL>=0)
                    {
                        for(k=1; k<Neq; k++)
                        {
                            FHLLC[k]=FHLLC[k]+fabs(Lamda[m])*FL[k];
                        }
                    }
                    else
                    {
                        if(SR<=0)
                        {
                            for(k=1; k<Neq; k++)
                            {
                                FHLLC[k]=FHLLC[k]+fabs(Lamda[m])*FR[k];
                            }
                        }
                        else
                        {
                            if(S_star>=0)
                            {
                                for(k=1; k<Neq; k++)
```

```
                                                {

            FHLLC[k]=FHLLC[k]+fabs(Lamda[m])*(FL[k]+SL*(QL_star[k]-QL[k]));
                                                }
                                        }
                                        else
                                        {
                                                for(k=1; k<Neq; k++)
                                                {

            FHLLC[k]=FHLLC[k]+fabs(Lamda[m])*(FR[k]+SR*(QR_star[k]-QR[k]));
                                                }
                                        }
                                }
                        }
                }
            }
            return true;
}
double WBAP_Limiter(double Theta1,double Theta2)
{
   double FinalValue;
   if(Theta1<=SmallNumber||Theta2<=SmallNumber)FinalValue=0;
   else
FinalValue=(WBAP_Parameter_n+1.0/Theta1+1.0/Theta2)/(WBAP_Parameter_n+1.0/Theta1/Theta1+1.0/
Theta2/Theta2);
   return FinalValue;
}
double WeightedAverage(double GradientL,double GradientR)
{
            double Gradient;
            Gradient=(pow(fabs(GradientL),2)*GradientR+pow(fabs(GradientR),2)*GradientL)
                        /(pow(fabs(GradientL),2)+pow(fabs(GradientR),2)+SmallNumber);
            return Gradient;
}//
```

```
void CESE_2DTimeMarching(double ***U_old, double ***Ux_old, double ***Uy_old,
                         double ***U_new,double ***Ux_new,double ***Uy_new,double dt,int IsHalf)
{
   int i,j,k,I,J;
          bool signalL, signalR, signalD, signalU;
          double y, U_LD[Neq], U_LU[Neq], U_RD[Neq], U_RU[Neq], F_LD, F_LU, F_RD, F_RU, G_DL,
G_DR, G_UL, G_UR, S_LD, S_LU, S_RD, S_RU;
   //fluxes through the surfaces of CE
   double
ULDR[Neq],ULDU[Neq],URDL[Neq],URDU[Neq],URUL[Neq],URUD[Neq],ULUR[Neq],ULUD[Neq];
   double FC_L[Neq],FC_D[Neq],FC_R[Neq],FC_U[Neq];//fluxes through the inner boundaries
   //R-right;L-left;U-upper;D-Down
   double UI,VI,UxC,UyC,UxL[Neq],UxR[Neq],UyL[Neq],UyR[Neq],UtL[Neq],UtR[Neq];
   double Theta1,Theta2,DUDa[Neq],DUDb[Neq],Ratioa,Ratiob;

#pragma omp parallel for private(j,k,I,J,signalL, signalR, signalD,
signalU,y,U_LD,U_LU,U_RD,U_RU,F_LD,F_LU,F_RD,F_RU,G_DL,G_DR,G_UL,G_UR, S_LD, S_LU,
S_RD,
S_RU,ULDR,ULDU,URDL,URDU,URUL,URUD,ULUR,ULUD,FC_L,FC_D,FC_R,FC_U,UI,VI,UxC,U
yC,UxL,UxR,UyL,UyR,UtL,UtR,Theta1,Theta2,DUDa,DUDb,Ratioa,Ratiob)
   for(i=IsHalf;i<=Nx;i++)
   {
      for(j=IsHalf;j<=Ny;j++)
      {
         I=i+1-IsHalf;
         J=j+1-IsHalf;

         for(k=0;k<Neq;k++)
         {
            U_LD[k]=U_old[I-1][J-1][k]+Ux_old[I-1][J-1][k]*dx/4+Uy_old[I-1][J-1][k]*dy/4;
            U_RD[k]=U_old[I][J-1][k]-Ux_old[I][J-1][k]*dx/4+Uy_old[I][J-1][k]*dy/4;
            U_RU[k]=U_old[I][J][k]-Ux_old[I][J][k]*dx/4-Uy_old[I][J][k]*dy/4;
            U_LU[k]=U_old[I-1][J][k]+Ux_old[I-1][J][k]*dx/4-Uy_old[I-1][J][k]*dy/4;
         }
```

```
for(k=0;k<Neq;k++)
{
	UyC=2*(U_LU[k]-U_LD[k])/dy;
	if(fabs(Uy_old[I-1][J-1][k])<=SmallNumber)    UyL[k]=Uy_old[I-1][J-
1][k];
	else
	{
		Theta1=UyC/Uy_old[I-1][J-1][k];
		Theta2=Uy_old[I-1][J][k]/Uy_old[I-1][J-1][k];
		UyL[k]=Uy_old[I-1][J-1][k]*WBAP_Limiter(Theta1,Theta2);
	}


	if(fabs(Uy_old[I-1][J][k])<=SmallNumber) UyR[k]=Uy_old[I-1][J][k];
	else
	{
		Theta1=UyC/Uy_old[I-1][J][k];
		Theta2=Uy_old[I-1][J-1][k]/Uy_old[I-1][J][k];
		UyR[k]=Uy_old[I-1][J][k]*WBAP_Limiter(Theta1,Theta2);
	}

}
y = (J - 1)*dy - 0.5*(1-IsHalf)*dy;
Comput_Ut(U_LD, Ux_old[I - 1][J - 1], UyL, y, Curv[I - 1][J - 1], Normx[I - 1][J
- 1], Normy[I - 1][J - 1], UtL);
y = J*dy - 0.5*(1 - IsHalf)*dy;
Comput_Ut(U_LU, Ux_old[I - 1][J], UyR, y, Curv[I - 1][J], Normx[I - 1][J],
Normy[I - 1][J], UtR);
for(k=0;k<Neq;k++)
{
	ULDU[k]=U_LD[k]+UyL[k]*dy/4+UtL[k]*dt/4;
	ULUD[k]=U_LU[k]-UyR[k]*dy/4+UtR[k]*dt/4;
}
signalL=Rotated_HLLC_RiemannSolver(ULDU,ULUD,FC_L,0,1);
```

```
for(k=0;k<Neq;k++)
{
	UyC=2*(U_RU[k]-U_RD[k])/dy;
	if(fabs(Uy_old[I][J-1][k])<=SmallNumber) UyL[k]=Uy_old[I][J-1][k];
	else
	{
		Theta1=UyC/Uy_old[I][J-1][k];
		Theta2=Uy_old[I][J][k]/Uy_old[I][J-1][k];
		UyL[k]=Uy_old[I][J-1][k]*WBAP_Limiter(Theta1,Theta2);
	}

	if(fabs(Uy_old[I][J][k])<=SmallNumber) UyR[k]=Uy_old[I][J][k];
	else
	{
		Theta1=UyC/Uy_old[I][J][k];
		Theta2=Uy_old[I][J-1][k]/Uy_old[I][J][k];
		UyR[k]=Uy_old[I][J][k]*WBAP_Limiter(Theta1,Theta2);
	}
}
y = (J - 1)*dy - 0.5*(1 - IsHalf)*dy;
Comput_Ut(U_RD, Ux_old[I][J - 1], UyL, y, Curv[I][J - 1], Normx[I][J - 1],
Normy[I][J - 1], UtL);
y = J *dy - 0.5*(1 - IsHalf)*dy;
Comput_Ut(U_RU, Ux_old[I][J], UyR, y, Curv[I][J], Normx[I][J], Normy[I][J],
UtR);
for(k=0;k<Neq;k++)
{
	URDU[k]=U_RD[k]+UyL[k]*dy/4+UtL[k]*dt/4;
	URUD[k]=U_RU[k]-UyR[k]*dy/4+UtR[k]*dt/4;
}
signalR=Rotated_HLLC_RiemannSolver(URDU,URUD,FC_R,0,1);


for(k=0;k<Neq;k++)
{
	UxC=2*(U_RD[k]-U_LD[k])/dx;
```

```
                    if(fabs(Ux_old[I-1][J-1][k])<=SmallNumber)    UxL[k]=Ux_old[I-1][J-
1][k];
                    else
                    {
                        Theta1=UxC/Ux_old[I-1][J-1][k];
                        Theta2=Ux_old[I][J-1][k]/Ux_old[I-1][J-1][k];
                        UxL[k]=Ux_old[I-1][J-1][k]*WBAP_Limiter(Theta1,Theta2);
                    }

                    if(fabs(Ux_old[I][J-1][k])<=SmallNumber) UxR[k]=Ux_old[I][J-1][k];
                    else
                    {
                        Theta1=UxC/Ux_old[I][J-1][k];
                        Theta2=Ux_old[I-1][J-1][k]/Ux_old[I][J-1][k];
                        UxR[k]=Ux_old[I][J-1][k]*WBAP_Limiter(Theta1,Theta2);
                    }

                }
                y = (J - 1)*dy - 0.5*(1 - IsHalf)*dy;
                Comput_Ut(U_LD, UxL, Uy_old[I - 1][J - 1], y, Curv[I - 1][J - 1], Normx[I - 1][J
- 1], Normy[I - 1][J - 1], UtL);
                Comput_Ut(U_RD, UxR, Uy_old[I][J - 1], y, Curv[I][J - 1], Normx[I][J - 1],
Normy[I][J - 1], UtR);
                for(k=0;k<Neq;k++)
                {
                    ULDR[k]=U_LD[k]+UxL[k]*dx/4+UtL[k]*dt/4;
                    URDL[k]=U_RD[k]-UxR[k]*dx/4+UtR[k]*dt/4;
                }
                signalD=Rotated_HLLC_RiemannSolver(ULDR,URDL,FC_D,1,0);


                for(k=0;k<Neq;k++)
                {
                    UxC=2*(U_RU[k]-U_LU[k])/dx;
                    if(fabs(Ux_old[I-1][J][k])<=SmallNumber) UxL[k]=Ux_old[I-1][J][k];
                    else
```

```
                {
                        Theta1=UxC/Ux_old[I-1][J][k];
                        Theta2=Ux_old[I][J][k]/Ux_old[I-1][J][k];
                        UxL[k]=Ux_old[I-1][J][k]*WBAP_Limiter(Theta1,Theta2);
                }

                if(fabs(Ux_old[I][J][k])<=SmallNumber) UxR[k]=Ux_old[I][J][k];
                else
                {
                        Theta1=UxC/Ux_old[I][J][k];
                        Theta2=Ux_old[I-1][J][k]/Ux_old[I][J][k];
                        UxR[k]=Ux_old[I][J][k]*WBAP_Limiter(Theta1,Theta2);
                }
        }
        y = J *dy - 0.5*(1 - IsHalf)*dy;
        Comput_Ut(U_LU, UxL, Uy_old[I - 1][J], y, Curv[I - 1][J], Normx[I - 1][J],
Normy[I - 1][J], UtL);
        Comput_Ut(U_RU, UxR, Uy_old[I][J], y, Curv[I][J], Normx[I][J], Normy[I][J],
UtR);
        for(k=0;k<Neq;k++)
        {
                ULUR[k]=U_LU[k]+UxL[k]*dx/4+UtL[k]*dt/4;
                URUL[k]=U_RU[k]-UxR[k]*dx/4+UtR[k]*dt/4;
        }
        signalU=Rotated_HLLC_RiemannSolver(ULUR,URUL,FC_U,1,0);


    for(k=1;k<Neq;k++)
    {

      F_LD=F[I-1][J-1][k]+Fy[I-1][J-1][k]*dy/4+Ft[I-1][J-1][k]*dt/4;
      F_LU=F[I-1][J][k]-Fy[I-1][J][k]*dy/4+Ft[I-1][J][k]*dt/4;
      F_RD=F[I][J-1][k]+Fy[I][J-1][k]*dy/4+Ft[I][J-1][k]*dt/4;
      F_RU=F[I][J][k]-Fy[I][J][k]*dy/4+Ft[I][J][k]*dt/4;

      G_DL=G[I-1][J-1][k]+Gx[I-1][J-1][k]*dx/4+Gt[I-1][J-1][k]*dt/4;
```

```
G_DR=G[I][J-1][k]-Gx[I][J-1][k]*dx/4+Gt[I][J-1][k]*dt/4;
G_UL=G[I-1][J][k]+Gx[I-1][J][k]*dx/4+Gt[I-1][J][k]*dt/4;
G_UR=G[I][J][k]-Gx[I][J][k]*dx/4+Gt[I][J][k]*dt/4;

S_LD = S[I - 1][J - 1][k] + Sx[I - 1][J - 1][k] * dx / 4 + Sy[I - 1][J - 1][k]
* dy / 4 + St[I - 1][J - 1][k] * dt / 4;
S_LU = S[I - 1][J][k] + Sx[I - 1][J][k] * dx / 4 - Sy[I - 1][J][k] * dy / 4 +
St[I - 1][J][k] * dt / 4;
S_RD = S[I][J - 1][k] - Sx[I][J - 1][k] * dx / 4 + Sy[I][J - 1][k] * dy / 4
+ St[I][J - 1][k] * dt / 4;
S_RU = S[I][J][k] - Sx[I][J][k] * dx / 4 - Sy[I][J][k] *  dy / 4 + St[I][J][k]
* dt / 4;


U_LD[k] = U_LD[k] + dt / dx*(F_LD - FC_D[k]) + dt / dy*(G_DL -
FC_L[k])+  dt / 2 * S_LD;
U_RD[k] = U_RD[k] + dt / dx*(FC_D[k] - F_RD) + dt / dy*(G_DR -
FC_R[k]) + dt / 2 * S_RD;
U_RU[k] = U_RU[k] + dt / dx*(FC_U[k] - F_RU) + dt / dy*(FC_R[k] -
G_UR) + dt / 2 * S_RU;
U_LU[k] = U_LU[k] + dt / dx*(F_LU - FC_U[k]) + dt / dy*(FC_L[k] -
G_UL) + dt / 2 * S_LU;
U_new[i][j][k]=0.25*(U_LD[k]+U_RD[k]+U_LU[k]+U_RU[k]);
if (signalL&&signalR&&signalD&&signalU)
{
    UxL[k] = 2 * (U_RD[k] - U_LD[k]) / dx;
    UxR[k] = 2 * (U_RU[k] - U_LU[k]) / dx;
    UxC = 0.5*(UxL[k] + UxR[k]);
    if (fabs(UxC) <= SmallNumber) Ux_new[i][j][k] = UxC;
    else
    {
        Theta1 = UxL[k] / UxC;
        Theta2 = UxR[k] / UxC;
        Ux_new[i][j][k]    =    UxC*WBAP_Limiter(Theta1,
Theta2);

    }
```

```
                                        UyL[k] = 2 * (U_LU[k] - U_LD[k]) / dy;
                                        UyR[k] = 2 * (U_RU[k] - U_RD[k]) / dy;
                                        UyC = 0.5*(UyL[k] + UyR[k]);
                                        if (fabs(UyC) <= SmallNumber) Uy_new[i][j][k] = UyC;
                                        else
                                        {
                                                Theta1 = UyL[k] / UyC;
                                                Theta2 = UyR[k] / UyC;
                                                Uy_new[i][j][k]    =    UyC*WBAP_Limiter(Theta1,
Theta2);
                                        }
                                }
                                else Ux_new[i][j][k] = Uy_new[i][j][k] = 0;
                                DUDa[k]=costheta*Ux_new[i][j][k]+sintheta*Uy_new[i][j][k];
                                DUDb[k]=sintheta*Uy_new[i][j][k]-costheta*Ux_new[i][j][k];
        }
                        UI=U_new[i][j][3]/(U_new[i][j][1]+U_new[i][j][2]);
                        VI=U_new[i][j][4]/(U_new[i][j][1]+U_new[i][j][2]);
                        if(UI>=0)
                        {
                                FC_D[0]=UI*ULDR[0];
                                FC_U[0]=UI*ULUR[0];
                        }
                        else
                        {
                                FC_D[0]=UI*URDL[0];
                                FC_U[0]=UI*URUL[0];
                        }
                        if(VI>=0)
                        {
                                FC_L[0]=VI*ULDU[0];
                                FC_R[0]=VI*URDU[0];
                        }
                        else
                        {
```

```
        FC_L[0]=VI*ULUD[0];
        FC_R[0]=VI*URUD[0];
}

F_LD=UI*(U_old[I-1][J-1][0]+Uy_old[I-1][J-1][0]*dy/4+Ut[I-1][J-1][0]*dt/4);
F_LU=UI*(U_old[I-1][J][0]-Uy_old[I-1][J][0]*dy/4+Ut[I-1][J][0]*dt/4);
F_RD=UI*(U_old[I][J-1][0]+Uy_old[I][J-1][0]*dy/4+Ut[I][J-1][0]*dt/4);
F_RU=UI*(U_old[I][J][0]-Uy_old[I][J][0]*dy/4+Ut[I][J][0]*dt/4);

G_DL=VI*(U_old[I-1][J-1][0]+Ux_old[I-1][J-1][0]*dx/4+Ut[I-1][J-1][0]*dt/4);
G_DR=VI*(U_old[I][J-1][0]-Ux_old[I][J-1][0]*dx/4+Ut[I][J-1][0]*dt/4);
G_UL=VI*(U_old[I-1][J][0]+Ux_old[I-1][J][0]*dx/4+Ut[I-1][J][0]*dt/4);
G_UR=VI*(U_old[I][J][0]-Ux_old[I][J][0]*dx/4+Ut[I][J][0]*dt/4);



U_LD[0] = U_LD[0] + dt / dx*(F_LD - FC_D[0]) + dt / dy*(G_DL - FC_L[0]);
U_RD[0] = U_RD[0] + dt / dx*(FC_D[0] - F_RD) + dt / dy*(G_DR - FC_R[0]);
U_RU[0] = U_RU[0] + dt / dx*(FC_U[0] - F_RU) + dt / dy*(FC_R[0] - G_UR);
U_LU[0] = U_LU[0] + dt / dx*(F_LU - FC_U[0]) + dt / dy*(FC_L[0] - G_UL);
U_new[i][j][0] = 0.25*(U_LD[0] + U_RD[0] + U_LU[0] + U_RU[0]);
if (signalL&&signalR&&signalD&&signalU)
{
        UxL[0] = 2 * (U_RD[0] - U_LD[0]) / dx;
        UxR[0] = 2 * (U_RU[0] - U_LU[0]) / dx;
        UxC = 0.5*(UxL[0] + UxR[0]);
        if (fabs(UxC) <= SmallNumber) Ux_new[i][j][0] = UxC;
        else
        {
                Theta1 = UxL[0] / UxC;
                Theta2 = UxR[0] / UxC;
                Ux_new[i][j][0] = UxC*WBAP_Limiter(Theta1, Theta2);
        }

        UyL[0] = 2 * (U_LU[0] - U_LD[0]) / dy;
        UyR[0] = 2 * (U_RU[0] - U_RD[0]) / dy;
        UyC = 0.5*(UyL[0] + UyR[0]);
```

```
            if (fabs(UyC) <= SmallNumber) Uy_new[i][j][0] = UyC;
            else
            {
                        Theta1 = UyL[0] / UyC;
                        Theta2 = UyR[0] / UyC;
                        Uy_new[i][j][0] = UyC*WBAP_Limiter(Theta1, Theta2);
            }
}
else Ux_new[i][j][0] = Uy_new[i][j][0] =0;
DUDa[0]=costheta*Ux_new[i][j][0]+sintheta*Uy_new[i][j][0];
DUDb[0]=sintheta*Uy_new[i][j][0]-costheta*Ux_new[i][j][0];

if(fabs(DUDa[0])<=SmallNumber)
{
            Ratioa=1.0;
            DUDa[0]=0;
}
else
{
            Theta1=fabs(MAXIMUM-U_new[i][j][0])/(fabs(DUDa[0])*DeltaL/2);
            Theta2=fabs(MINIMUM-U_new[i][j][0])/(fabs(DUDa[0])*DeltaL/2);
            Ratioa=MIN(1,MIN(Theta1,Theta2));
}

if(fabs(DUDb[0])<=SmallNumber)
{
            Ratiob=1.0;
            DUDb[0]=0;
}
else
{
            Theta1=fabs(MAXIMUM-U_new[i][j][0])/(fabs(DUDb[0])*DeltaL/2);
            Theta2=fabs(MINIMUM-U_new[i][j][0])/(fabs(DUDb[0])*DeltaL/2);
            Ratiob=MIN(1,MIN(Theta1,Theta2));
}
```

```
                        for(k=0;k<Neq;k++)
                        {
                                DUDa[k]=Ratioa*DUDa[k];
                                DUDb[k]=Ratiob*DUDb[k];

                                Ux_new[i][j][k]=(DUDa[k]-DUDb[k])/(2*costheta);
                                Uy_new[i][j][k]=(DUDa[k]+DUDb[k])/(2*sintheta);
                        }

        }
    }
}//mesh variables updated in half time step
void Compute_NormAndCurv(double ***U, int IsHalf)
{
        int i, j;
        double NLD, NxLD, NyLD, NRD, NxRD, NyRD, NRU, NxRU, NyRU, NLU, NxLU, NyLU ;
#pragma omp parallel for private(j,NLD, NxLD, NyLD, NRD, NxRD, NyRD, NRU, NxRU, NyRU, NLU,
NxLU, NyLU)
        for (i = 1; i < Nx + IsHalf; i++)
        {
                for (j = 1; j < Ny + IsHalf; j++)
                {
                        if (U[i][j][0]>0.1&&U[i][j][0] < 0.9)
                        {
                                NxLD = (U[i][j][0] + U[i][j - 1][0] - U[i - 1][j][0] - U[i - 1][j - 1][0]) / (2
* dx);
                                NyLD = (U[i][j][0] + U[i- 1][j ][0] - U[i ][j- 1][0] - U[i - 1][j - 1][0]) / (2
* dy);
                                NLD = sqrt(NxLD*NxLD + NyLD*NyLD);
                                NxRD = (U[i+1][j][0] + U[i+1][j - 1][0] - U[i ][j][0] - U[i ][j - 1][0]) / (2
* dx);
                                NyRD = (U[i+1][j][0] + U[i ][j][0] - U[i+1][j - 1][0] - U[i ][j - 1][0]) / (2
* dy);
                                NRD = sqrt(NxRD*NxRD + NyRD*NyRD);
                                NxRU = (U[i + 1][j+1][0] + U[i + 1][j][0] - U[i][j+1][0] - U[i][j][0]) / (2
* dx);
```

```
                                        NyRU = (U[i + 1][j+1][0] + U[i][j+1][0] - U[i + 1][j ][0] - U[i][j ][0]) /
(2 * dy);
                                        NRU = sqrt(NxRU*NxRU + NyRU*NyRU);
                                        NxLU = (U[i][j+1][0] + U[i][j ][0] - U[i - 1][j+1][0] - U[i - 1][j][0]) / (2
* dx);
                                        NyLU = (U[i][j+1][0] + U[i - 1][j+1][0] - U[i][j ][0] - U[i - 1][j ][0]) / (2
* dy);
                                        NLU = sqrt(NxLU*NxLU + NyLU*NyLU);
                                        Normx[i][j] = 0.25*(NxLD + NxRD + NxRU + NxLU);
                                        Normy[i][j] = 0.25*(NyLD + NyRD + NyRU + NyLU);
                                        Curv[i][j] = (NxRD / NRD + NxRU / NRU - NxLD / NLD - NxLU /
NLU) / (2 * dx) + (NyLU / NLU + NyRU / NRU - NyLD / NLD - NyRD / NRD) / (2 * dy);
                                }
                                else Normx[i][j] = Normy[i][j] = Curv[i][j] = 0;
                        }
                }
#pragma omp parallel for
                for (j = 1; j < Ny + IsHalf; j++)
                {
                        Normx[0][j] = Normx[1][j];
                        Normy[0][j] = Normy[1][j];
                        Curv[0][j] = Curv[1][j];
                        Normx[Nx + IsHalf][j] = Normx[Nx + IsHalf - 1][j];
                        Normy[Nx + IsHalf][j] = Normy[Nx + IsHalf - 1][j];
                        Curv[Nx + IsHalf][j] = Curv[Nx + IsHalf - 1][j];
                }
#pragma omp parallel for
                for (i = 0; i <= Ny + IsHalf; i++)
                {
                        Normx[i][0] = Normx[i][1];
                        if (IsHalf) Normy[i][0] = -Normy[i][1];
                        else Normy[i][0] = 0;
                        Curv[i][0] = Curv[i][1];
                        Normx[i][Ny + IsHalf] = Normx[i][Ny + IsHalf - 1];
                        Normy[i][Ny + IsHalf] = Normy[i][Ny + IsHalf - 1];
                        Curv[i][Ny + IsHalf] = Curv[i][Ny + IsHalf - 1];
```

```
	}

}

void WriteAsciiData(char *filename,double ***U)
{
	int i,j;
	FILE *fp;
	double rou,alpha,u,v,p,gama,pai;
	printf("WriteAsciiData...");
	fp=fopen(filename,"w+");
	fprintf(fp,"TITLE
	=\"Dataset\"\nVARIABLES=\"x\"\"y\"\"roug\"\"roul\"\"rou\"\"alpha\"\"u\"\"v\"\"p\"\"Curv\"\"Nx\
"\"Ny\"");
	fprintf(fp,"ZONE T=\"Zone 1\"\nI=%d J=%d K=%d ZONETYPE=Ordered\n",Ny+1,Nx+1,1);
	fprintf(fp,"DATAPACKING=POINT\n");
	for(i=0;i<=Nx;i++)
	{
		for(j=0;j<=Ny;j++)
		{
			alpha=U[i][j][0];
			rou=U[i][j][1]+U[i][j][2];
			u=U[i][j][3]/rou;
			v=U[i][j][4]/rou;
			gama=1.0/(alpha/(GAMA1-1)+(1-alpha)/(GAMA2-1))+1;
			pai=(alpha*GAMA1*PAI1/(GAMA1-1)+(1-alpha)*GAMA2*PAI2/(GAMA2-
1))*(gama-1)/gama;
			p=(gama-1)*(U[i][j][5]-0.5*rou*(u*u+v*v))-gama*pai;
			fprintf(fp,
"%20f%20f%20.10e%20.10e%20.10e%20.10e%20.10e%20.10e%20.10e%20.10e%20.10e\n",
i*dx, j*dy, U[i][j][1], U[i][j][2], rou, alpha, u, v, p, Curv[i][j], -SurfaceTension*Curv[i][j] * Normx[i][j], -
SurfaceTension*Curv[i][j] * Normy[i][j]);
		}
	}
	fclose(fp);
	printf("Done!\n");
```

```
}
//Output Ascii results
void WriteBinaryData(char* filename,double ***U,double PresentTime)
{
	int i,j,k;
	FILE	*fp;
	double x,y,rou,alpha,u,v,p,gama,pai;
	double DD;
	int II;
	float FF,ZONEMARKER,EOHMARKER;
	char CC[128];
	double MAX_roug,MAX_roul,MAX_rou,MAX_alpha,MAX_u,MAX_v,MAX_p;
	double MIN_roug,MIN_roul,MIN_rou,MIN_alpha,MIN_u,MIN_v,MIN_p;
	printf("WriteBinaryData...");
	ZONEMARKER=299.0;
	EOHMARKER =357.0;
	MAX_roug=MAX_roul=MAX_rou=MAX_alpha=MAX_u=MAX_v=MAX_p=-1e100;
	MIN_roug=MIN_roul=MIN_rou=MIN_alpha=MIN_u=MIN_v=MIN_p=1e100;
	for(i=0;i<=Nx;i++)
	{
		for(j=0;j<=Ny;j++)
		{
			alpha=U[i][j][0];
			rou=U[i][j][1]+U[i][j][2];
			u=U[i][j][3]/rou;
			v=U[i][j][4]/rou;
			gama=1.0/(alpha/(GAMA1-1)+(1-alpha)/(GAMA2-1))+1;
			pai=(alpha*GAMA1*PAI1/(GAMA1-1)+(1-alpha)*GAMA2*PAI2/(GAMA2-
1))*(gama-1)/gama;
			p=(gama-1)*(U[i][j][5]-0.5*rou*(u*u+v*v))-gama*pai;
			MAX_roug=MAX(MAX_roug,U[i][j][1]);
			MIN_roug=MIN(MIN_roug,U[i][j][1]);
			MAX_roul=MAX(MAX_roul,U[i][j][2]);
			MIN_roul=MIN(MIN_roul,U[i][j][2]);
			MAX_rou=MAX(MAX_rou,rou);
			MIN_rou=MIN(MIN_rou,rou);
```

```
                        MAX_alpha=MAX(MAX_alpha,U[i][j][0]);
                        MIN_alpha=MIN(MIN_alpha,U[i][j][0]);
                        MAX_u=MAX(MAX_u,u);
                        MIN_u=MIN(MIN_u,u);
                        MAX_v=MAX(MAX_v,v);
                        MIN_v=MIN(MIN_v,v);
                        MAX_p=MAX(MAX_p,p);
                        MIN_p=MIN(MIN_p,p);
                }
        }
         /*
    * Open the file and write the tecplot datafile
    * header information
    */

        fp=fopen(filename,"wb");
        strcpy(CC,"#!TDV111");
        fwrite(CC,sizeof(char),8,fp);//Magic number, Version number, version information
        II=1;
        fwrite(&II,sizeof(int),1,fp);//Integer value of 1;This is used to determine the byte order of the reader,
relative to the writer
        II=0;
        fwrite(&II,sizeof(int),1,fp);//FileType
        strcpy(CC,"DataBase");
        for(i=0;i<9;i++)
        {
                if(i==8)II=0;
                else II=CC[i];
                fwrite(&II,sizeof(int),1,fp);
        }
        //Title
        II=9;
        fwrite(&II,sizeof(int),1,fp);//number of variable
        strcpy(CC,"X");
        for(i=0;i<2;i++)
        {
```

```
            if(i==1)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"Y");
for(i=0;i<2;i++)
{
            if(i==1)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"roug");
for(i=0;i<5;i++)
{
            if(i==4)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"roul");
for(i=0;i<5;i++)
{
            if(i==4)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"rou");
for(i=0;i<4;i++)
{
            if(i==3)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"alpha");
for(i=0;i<6;i++)
{
            if(i==5)II=0;
```

```
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"u");
for(i=0;i<2;i++)
{
            if(i==1)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"v");
for(i=0;i<2;i++)
{
            if(i==1)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}
strcpy(CC,"p");
for(i=0;i<2;i++)
{
            if(i==1)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}//variable name
fwrite(&ZONEMARKER,sizeof(float),1,fp);
strcpy(CC,"ZONE1");
for(i=0;i<6;i++)
{
            if(i==5)II=0;
            else II=CC[i];
            fwrite(&II,sizeof(int),1,fp);
}//Zone name
II=-1;
fwrite(&II,sizeof(int),1,fp);//parent zone
//Zero-based zone number within this datafile to which this zone is a child.
II=-2;
```

```
fwrite(&II,sizeof(int),1,fp);//StrandID: -2 = pending strand ID for assignment by Tecplot
//          -1 = static strand ID
//          0 <= N < 32700 valid strand ID
fwrite(&PresentTime,sizeof(double),1,fp);//solution time
II=-1;
fwrite(&II,sizeof(int),1,fp);//zone color (set to -1 if you want Tecplot to determine).
II=0;
fwrite(&II,sizeof(int),1,fp);//ZoneType 0=ORDERED,       1=FELINESEG,
// 2=FETRIANGLE,    3=FEQUADRILATERAL,
//4=FETETRAHEDRON,  5=FEBRICK,
//6=FEPOLYGON,     7=FEPOLYHEDRON
II=1;
fwrite(&II,sizeof(int),1,fp);//data packing 0-block 1-point
II=0;
fwrite(&II,sizeof(int),1,fp);//Specify Var Location. 0 = Don't specify, all data is located at the nodes.
1 = Specify
II=0;
fwrite(&II,sizeof(int),1,fp);//Are raw local 1-to-1 face neighbors supplied? (0=FALSE 1=TRUE).
II=0;
fwrite(&II,sizeof(int),1,fp);//Number of miscellaneous user-defined face neighbor connections
(value >= 0).
II=Nx+1;
fwrite(&II,sizeof(int),1,fp);//IMAX
II=Ny+1;
fwrite(&II,sizeof(int),1,fp);//JMAX
II=1;
fwrite(&II,sizeof(int),1,fp);//KMAX
II=0;
fwrite(&II,sizeof(int),1,fp);//1=Auxiliary name/value pair to follow     0=No more Auxiliar
name/value pairs
fwrite(&EOHMARKER,sizeof(float),1,fp);
fwrite(&ZONEMARKER,sizeof(float),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
```

```
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);
II=1;
fwrite(&II,sizeof(int),1,fp);//variable data format, 1=Float, 2=Double, 3=LongInt, 4=ShortInt, 5=Byte, 6=Bit
II=0;
fwrite(&II,sizeof(int),1,fp);//Has passive variables: 0 = no, 1 = yes.
II=0;
fwrite(&II,sizeof(int),1,fp);//Has variable sharing 0 = no, 1 = yes
II=-1;
fwrite(&II,sizeof(int),1,fp);//Zone number to share connectivity list with (-1 = no sharing)
DD=Lx_S;
fwrite(&DD,sizeof(double),1,fp);
DD=Lx_E;
fwrite(&DD,sizeof(double),1,fp);
DD=Ly_S;
fwrite(&DD,sizeof(double),1,fp);
DD=Ly_E;
fwrite(&DD,sizeof(double),1,fp);
fwrite(&MIN_roug,sizeof(double),1,fp);
fwrite(&MAX_roug,sizeof(double),1,fp);
fwrite(&MIN_roul,sizeof(double),1,fp);
fwrite(&MAX_roul,sizeof(double),1,fp);
fwrite(&MIN_rou,sizeof(double),1,fp);
fwrite(&MAX_rou,sizeof(double),1,fp);
fwrite(&MIN_alpha,sizeof(double),1,fp);
```

```
fwrite(&MAX_alpha,sizeof(double),1,fp);
fwrite(&MIN_u,sizeof(double),1,fp);
fwrite(&MAX_u,sizeof(double),1,fp);
fwrite(&MIN_v,sizeof(double),1,fp);
fwrite(&MAX_v,sizeof(double),1,fp);
fwrite(&MIN_p,sizeof(double),1,fp);
fwrite(&MAX_p,sizeof(double),1,fp);
// Compressed list of min/max pairs for each non-shared and non-passive
//variable. For each non-shared and non-passive variable (as specified above)
for(j=0;j<=Ny;j++)
{
        for(i=0;i<=Nx;i++)
        {
                x=Lx_S+i*dx;
                y=Ly_S+j*dy;
                alpha=U[i][j][0];
                rou=U[i][j][1]+U[i][j][2];
                u=U[i][j][3]/rou;
                v=U[i][j][4]/rou;
                gama=1.0/(alpha/(GAMA1-1)+(1-alpha)/(GAMA2-1))+1;
                pai=(alpha*GAMA1*PAI1/(GAMA1-1)+(1-alpha)*GAMA2*PAI2/(GAMA2-
1))*(gama-1)/gama;
                p=(gama-1)*(U[i][j][5]-0.5*rou*(u*u+v*v))-gama*pai;
                FF=float(x);
                fwrite(&FF,sizeof(float),1,fp);
                FF=float(y);
                fwrite(&FF,sizeof(float),1,fp);
                FF=float(U[i][j][1]);
                fwrite(&FF,sizeof(float),1,fp);
                FF=float(U[i][j][2]);
                fwrite(&FF,sizeof(float),1,fp);
                FF=float(rou);
                fwrite(&FF,sizeof(float),1,fp);
                FF=float(U[i][j][0]);
                fwrite(&FF,sizeof(float),1,fp);
                FF=float(u);
```

```
                    fwrite(&FF,sizeof(float),1,fp);
                    FF=float(v);
                    fwrite(&FF,sizeof(float),1,fp);
                    FF=float(p);
                    fwrite(&FF,sizeof(float),1,fp);
                }
            }
            fclose(fp);
            printf("Done!\n");
}
//Output Binary results
void CESE_Solver()
{
            int i,j,NumberofStep;
            FILE *fp;
            char key,sFName[1024];
            double x,AA,XX,ZZ,ComputingTime,dt,MaxAlpha,MinAlpha;
            do
            {
                    printf("Please select:\n[1] New Computation\n[2] Restart an Old Computation\n");
                    key=_getch();
            }while(!(key=='1'||key=='2'));
            printf("\nYour choice is [%c]\n\n",key);
            if(key=='1')
            {
                    ComputingTime=0;
                    NumberofStep=0;
                    Initializer();
                    InitialMass1=(U1[0][0][1]+U1[0][Ny][1]+U1[Nx][0][1]+U1[Nx][Ny][1])*dx*dy/4;
                    InitialMass2=(U1[0][0][2]+U1[0][Ny][2]+U1[Nx][0][2]+U1[Nx][Ny][2])*dx*dy/4;
                    InitialEnergy=(U1[0][0][5]+U1[0][Ny][5]+U1[Nx][0][5]+U1[Nx][Ny][5])*dx*dy/4;
                    for(i=1;i<Nx;i++)
                    {
                            InitialMass1=InitialMass1+(U1[i][0][1]+U1[i][Ny][1])*dx*dy/2;
                            InitialMass2=InitialMass2+(U1[i][0][2]+U1[i][Ny][2])*dx*dy/2;
                            InitialEnergy=InitialEnergy+(U1[i][0][5]+U1[i][Ny][5])*dx*dy/2;
```

```
        }
        for(j=1;j<Ny;j++)
        {
                InitialMass1=InitialMass1+(U1[0][j][1]+U1[Nx][j][1])*dx*dy/2;
                InitialMass2=InitialMass2+(U1[0][j][2]+U1[Nx][j][2])*dx*dy/2;
                InitialEnergy=InitialEnergy+(U1[0][j][5]+U1[Nx][j][5])*dx*dy/2;
        }
        for(i=1;i<Nx;i++)
        {
                for(j=1;j<Ny;j++)
                {
                        InitialMass1=InitialMass1+U1[i][j][1]*dx*dy;
                        InitialMass2=InitialMass2+U1[i][j][2]*dx*dy;
                        InitialEnergy=InitialEnergy+U1[i][j][5]*dx*dy;
                }
        }
        fp=fopen("Conservation Property.txt","w+");
        fprintf(fp,"%e\t%e\t%e\t%e\t%e\t%e\n",ComputingTime,0.0,0.0,0.0,0.0,0.0);
        fclose(fp);
}
else
{
        printf("Input the data file name:\n");
        scanf("%s",sFName);
        restart(sFName,ComputingTime,NumberofStep);
}
Compute_NormAndCurv(U1, 0);
sprintf(sFName,"result_t=%.2fus.plt",ComputingTime*1e6);
if(OutputFileType==0) WriteAsciiData(sFName,U1);
else WriteBinaryData(sFName,U1,ComputingTime);
while(ComputingTime<time)
{
        if(NumberofStep%30==0)
        {
                AA=XX=ZZ=0;
           for(i=0;i<=Nx;i++)
```

```
                {
                        x=Lx_S+i*dx;
                   for(j=0;j<=Ny;j++)
                        {
                                AA=AA+(1-U1[i][j][0]);
                            XX=XX+U1[i][j][2]*x;
                            ZZ=ZZ+U1[i][j][2];
                        }
                }
                    AA=AA*dx*dy;
                XX=XX/ZZ;
                if(ComputingTime==0)fp=fopen("centroid.txt","w");
                else fp=fopen("centroid.txt","a");
                fprintf(fp,"%e\t%e\t%e\n",ComputingTime,XX,AA);
                fclose(fp);
            }
            dt=ComputingTimeStep(U1);
            ComputingTime=ComputingTime+dt;
            NumberofStep++;
            printf("N=%d,dt=%e,t=%e\n",NumberofStep,dt,ComputingTime);
            ComputeFluxesAndDerivatives(U1,Ux1,Uy1,0);
            CESE_2DTimeMarching(U1,Ux1,Uy1,U2,Ux2,Uy2,dt,1);
            BoundaryTreatment();
            Compute_NormAndCurv(U2, 1);
ComputeFluxesAndDerivatives(U2,Ux2,Uy2,1);
CESE_2DTimeMarching(U2,Ux2,Uy2,U1,Ux1,Uy1,dt,0);
            Compute_NormAndCurv(U1, 0);
            if(NumberofStep%StoreFreq==0)
            {
                    sprintf(sFName,"result_t=%.2fus.plt",ComputingTime*1e6);
                    if(OutputFileType==0) WriteAsciiData(sFName,U1);
                    else WriteBinaryData(sFName,U1,ComputingTime);
                    restoreall("restoreall.dat",ComputingTime,NumberofStep);
            }
            MaxAlpha=0;
            MinAlpha=1.0;
```

```
for(i=0;i<=Nx;i++)
{
	for(j=0;j<=Ny;j++)
	{
		MaxAlpha=MAX(MaxAlpha,U1[i][j][0]);
		MinAlpha=MIN(MinAlpha,U1[i][j][0]);
	}
}
TotalMass1=(U1[0][0][1]+U1[0][Ny][1]+U1[Nx][0][1]+U1[Nx][Ny][1])*dx*dy/4;
TotalMass2=(U1[0][0][2]+U1[0][Ny][2]+U1[Nx][0][2]+U1[Nx][Ny][2])*dx*dy/4;
TotalEnergy=(U1[0][0][5]+U1[0][Ny][5]+U1[Nx][0][5]+U1[Nx][Ny][5])*dx*dy/4;
for(i=1;i<Nx;i++)
{
	TotalMass1=TotalMass1+(U1[i][0][1]+U1[i][Ny][1])*dx*dy/2;
	TotalMass2=TotalMass2+(U1[i][0][2]+U1[i][Ny][2])*dx*dy/2;
	TotalEnergy=TotalEnergy+(U1[i][0][5]+U1[i][Ny][5])*dx*dy/2;
}
for(j=1;j<Ny;j++)
{
	TotalMass1=TotalMass1+(U1[0][j][1]+U1[Nx][j][1])*dx*dy/2;
	TotalMass2=TotalMass2+(U1[0][j][2]+U1[Nx][j][2])*dx*dy/2;
	TotalEnergy=TotalEnergy+(U1[0][j][5]+U1[Nx][j][5])*dx*dy/2;
}
for(i=1;i<Nx;i++)
{
	for(j=1;j<Ny;j++)
	{
		TotalMass1=TotalMass1+U1[i][j][1]*dx*dy;
		TotalMass2=TotalMass2+U1[i][j][2]*dx*dy;
		TotalEnergy=TotalEnergy+U1[i][j][5]*dx*dy;
	}
}
TotalMass1=TotalMass1/InitialMass1-1.0;
TotalMass2=TotalMass2/InitialMass2-1.0;
TotalEnergy=TotalEnergy/InitialEnergy-1.0;
fp=fopen("Conservation Property.txt","a+");
```

```
        fprintf(fp,"%e\t%e\t%e\t%e\t%e\t%e\n",ComputingTime,TotalMass1,TotalMass2,TotalEnergy,Mi
nAlpha,1-MaxAlpha);
                fclose(fp);
        }
        if(OutputFileType==0) WriteAsciiData("FinalResult.plt",U1);
        else WriteBinaryData("FinalResult.plt",U1,ComputingTime);
}

int main()
{
        AllocateMemory();
        omp_set_num_threads(NumberOfThreads);
   CESE_Solver();
        FreeMemory();
}
```