Mesoscale modeling and direct simulation of explosively dispersed granular materials

by

Huangrui Mo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Dr. Luc Bauwens Professor, Mechanical and Manufacturing Engineering University of Calgary
Supervisors	Dr. Fue-Sang Lien Professor, Mechanical and Mechatronics Engineering University of Waterloo
	Dr. Fan Zhang Senior Scientist, Advanced Energetics Group Defence Research and Development Canada
	Dr. Duane Cronin Professor, Mechanical and Mechatronics Engineering University of Waterloo
Internal Members	Dr. Cecile Devaud Professor, Mechanical and Mechatronics Engineering University of Waterloo
	Dr. Jean-Pierre Hickey Professor, Mechanical and Mechatronics Engineering University of Waterloo
Internal-External Member	Dr. Lilia Krivodonova Professor, Applied Mathematics University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Permissions

Figure 5.11 was reprinted with kind permission from Springer Nature: International Symposium on Shock Waves, Rodriguez V., Saurel R., Jourdan G., and Houas L. (2017), "Study on the jet formation during dispersal of solid particles by shock and blast waves", Volume 1, Page 747, Figure 3.

Figures 1.4 and 1.5 are property of the Government of Canada, and were used with permission from Defence Research and Development Canada (DRDC).

Chapter 2 contains content of an article published by Springer Nature. Mo, H., Lien, F.-S., Zhang, F., and Cronin, D., "A numerical framework for the direct simulation of dense particulate flow under explosive dispersal", Shock Waves (2018), Volume 28, Issue 3, Page 559-577. This article was published with the permission of the Minister of National Defence, and is protected by copyright and all rights are held exclusively by Her Majesty the Queen in Right of Canada.

Chapter 3 contains content of a preprint released on arXiv. Mo, H., Lien, F.-S., Zhang, F., and Cronin, D., "A simple field function for solving complex and dynamic fluid-solid system on Cartesian grid", arXiv:1702.02474.

Chapter 4 is part of an article published by Wiley. Mo, H., Lien, F.-S., Zhang, F., and Cronin, D., "An immersed boundary method for solving compressible flow with arbitrarily irregular and moving geometry", International Journal for Numerical Methods in Fluids (2018), Volume 88, Issue 5, Page 239-563. This article was published with the permission of the Minister of National Defence, and is protected by copyright and all rights are held exclusively by Her Majesty the Queen in Right of Canada.

Abstract

Explosively dispersed granular materials frequently exhibit macroscale coherent particle clustering and jetting structures. The underlying mechanism is of significant interest to study instability and mixing in high-speed gas-solid flows but remains unclear, primarily attributed to the complex mesoscale multiphase interactions involved in the dispersal process. In order to advance the understanding of particle clustering and jetting instabilities, this thesis establishes a numerical framework for solving interface-resolved gas-solid flows with non-deforming bodies that are able to move, contact, and collide. The developed framework is implemented to create a computational solver and then verified using a variety of gas-solid flow problems at different geometric scales. Employing the developed framework and solver, this thesis further studies the particle clustering and jetting instabilities in explosively dispersed granular materials.

A Cartesian, 3D, high-resolution, parallelized, gas-solid flow solver is created with the capability of tackling shocked flow conditions, irregular and moving geometries, and multibody collisions. The underlying numerical framework integrates operator splitting for partitioned fluid-solid interaction in the time domain, 2nd/3rd order strong stability-preserving Runge–Kutta methods and 3rd/5th order weighted essentially nonoscillatory schemes for high-resolution tempo-spatial discretization, the front-tracking method for evolving phase interfaces, a new field function developed for facilitating the solution of complex and dynamic fluid-solid systems on Cartesian grids, a new collision model developed for deterministic multibody contact and collision with parameterized coefficients of restitution and friction, and a new immersed boundary method developed for treating arbitrarily irregular and moving boundaries. The developed framework and solver are able to accurately, efficiently, and robustly solve coupled fluid-fluid, fluid-solid, and solid-solid interactions with flow conditions ranging from subsonic to hypersonic states.

Employing the developed framework and solver, direct simulations that capture interface-resolved multiphase interactions and deterministic mesoscale granular dynamics are conducted to investigate particle clustering and jetting instabilities. A random sampling algorithm is employed to generate stochastic payload morphologies with randomly distributed particle positions and sizes. Through solving and analyzing cases that cover a set of stochastic payloads, burster states, and coefficients of restitution, a valid statistical dissipative property of the framework in solving explosively dispersed granular materials with respect to Gurney velocity is demonstrated. The predicted surface expansion velocities can extend the time range of the velocity scaling law with regard to Gurney energy in the Gurney theory from the steady-state termination phase to the unsteady evolution phase. When considering the mean surface expansion velocities, the maximum error of the unsteady velocity scaling law is about 0.792% among the investigated Gurney energies. In addition, a dissipation analysis of the current discrete modeling of granular payloads suggests that incorporating the effects of porosity can enhance the prediction of Gurney velocity for explosively dispersed granular payloads. On the basis of direct simulations, an explanation for particle clustering and jetting instabilities is proposed to increase the understanding of established experimental observations in the literature. Results suggest that the development of internal sliding and colliding lines in the shock-compacted granular payload can be critical to the subsequent fracture pattern of the payload. Particle clusters manifested through payload fracture are then maintained by local pressure gradient between surrounding and interstitial flows as well as by dissipative inter-grain collisions. The existence of stable clusters introduce a more non-equilibrium momentum distribution in the overall payload, exhibiting as a form of clustering instability.

Under the current assumptions of non-deformable grains, the mesoscale granular dynamics largely depends on the payload morphology as a result of packing methods. Different payload morphologies can develop varied sliding and colliding lines, which lead to a corresponding pattern for payload fracturing and particle clustering. With the rapid development of high-performance computing technology, future direct simulations on stochastic payloads with significantly increased domain sizes, number of particles, and solution times are expected to lead to a better understanding of the flow instability in explosively dispersed granular payloads. It is suggested that statistics collected from a large number of mesoscale computations based on random payload morphologies can potentially evolve into a macroscopic theory of multiphase flow instability for particle clustering and jetting phenomena widely observed in many areas involving dense gas-solid flows.

Acknowledgements

I owe my deepest gratitude and appreciation to my doctoral advisors: Dr. Fue-Sang Lien, Dr. Fan Zhang, and Dr. Duane Cronin. Dr. Fue-Sang Lien has always been supportive, approachable, and helpful throughout my doctoral study. His encouragement and understanding helped me go through the difficulties and created the space for me to develop research ideas. Working with Dr. Fan Zhang has been the most amazing experience in my life. I have always been fascinated by his insights on physics and the ability to instantly and accurately identify the key problems based on a set of fragmented information. His keen and open-minded guidance inspired my interests in the field of my doctoral study, shaped my critical thinking, and challenged me to be a higher level thinker. It was an enlightening and enriching experience to collaborate with Dr. Duane Cronin. Whenever I needed advice, he was ready and patient to help. He was always kind to teach me and willing to share his experience and vision on how to be a professional, rigorous, and persuasive researcher. The moments I interacted with and the knowledge I learned from my outstanding advisors will be remembered by me throughout the rest of my life.

I am grateful to Natural Sciences and Engineering Research Council of Canada (NSERC), Defence Research and Development Canada (DRDC), and Waterloo CFD Engineering Consulting Inc (WATCFD) for the financial support of this research project. This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca) and Compute/Calcul Canada.

I would like to thank the official members of my examining committee for their efforts in reviewing my thesis and providing helpful suggestions. In addition, I want to express my gratitude to Dr. Jean-Pierre Hickey for his kind help with my teaching practice, to Dr. Cecile Devaud for her helpful comments on my comprehensive examination report, to Dr. Lilia Krivodonova for her thoughtful teaching on numerical solutions of partial differential equations, and to Dr. Luc Bauwens for taking his time to come and attend my examination in person. I also want to thank my colleagues in the Energy Research Center for their support and discussions and the faculty and staff in the Mechanical and Mechatronics Engineering department for their assistance and help throughout my doctoral study. Finally, I am indebted to my family for their continuous support and understanding with my pursuit of scientific research.

Table of Contents

List of Figures

List of Tables

Nomenclature

Characters

Symbol	Description	Unit
R	specific gas constant	$\mathrm{m}^2 \cdot \mathrm{s}^{-2} \cdot \mathrm{K}^{-1}$
C_v	specific heat capacity at constant volume	$\mathrm{m}^2 \cdot \mathrm{s}^{-2} \cdot \mathrm{K}^{-1}$
C_p	specific heat capacity at constant pressure	$\mathrm{m}^2 \cdot \mathrm{s}^{-2} \cdot \mathrm{K}^{-1}$
e_T	specific total energy	$\mathrm{m}^2 \cdot \mathrm{s}^{-2}$
e	specific internal energy	$\mathrm{m}^2 \cdot \mathrm{s}^{-2}$
h_T	specific total enthalpy	$\mathrm{m}^2 \cdot \mathrm{s}^{-2}$
h	specific enthalpy	$\mathrm{m}^2 \cdot \mathrm{s}^{-2}$
k	thermal conductivity	$\mathrm{kg} \cdot \mathrm{m} \cdot \mathrm{s}^{-3} \cdot \mathrm{K}^{-1}$
T	temperature	K
t	time	s
p	thermodynamic pressure	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-2}$
$\boldsymbol{f}_\mathrm{b}$	body force	$\mathrm{kg} \cdot \mathrm{m}^{-2} \cdot \mathrm{s}^{-2}$
S	boundary surface	m^2
Ω	spatial domain	m^3
$\boldsymbol{V}$	velocity vector	$\mathrm{m} \cdot \mathrm{s}^{-1}$
u	x component of velocity	$\mathrm{m} \cdot \mathrm{s}^{-1}$
v	y component of velocity	$\mathrm{m} \cdot \mathrm{s}^{-1}$
w	z component of velocity	$\mathrm{m} \cdot \mathrm{s}^{-1}$
c	speed of sound	$\mathrm{m} \cdot \mathrm{s}^{-1}$
$\boldsymbol{x}$	position vector	m
$\mathbf{n}$	unit outward normal vector	1
$\hat{\mathbf{t}}$	unit tangent vector	1
$\tilde{\mathbf{t}}$	unit bitangent vector	1

C_R	coefficient of restitution	1
Re	Reynolds number	1
Pr	Prandtl number	1
Ma	Mach number	1
α	thermal diffusivity	$\mathrm{m}^2 \cdot \mathrm{s}^{-1}$
μ	dynamic viscosity	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$
ν	kinematic viscosity	$\mathrm{m}^2 \cdot \mathrm{s}^{-1}$
γ	heat capacity ratio	1
ρ	density	$\mathrm{kg} \cdot \mathrm{m}^{-3}$
$\boldsymbol{U}$	vector of conservative variables	
$\boldsymbol{F}$	vector of fluxes	
$\boldsymbol{\Phi}$	vector of source terms	
$\boldsymbol{\sigma}$	stress tensor	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-2}$
$\boldsymbol{\tau}$	viscous stress tensor	$\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-2}$
$\boldsymbol{\delta}$	Kronecker tensor	1
$\mathbf{I}$	identity tensor	1

Operators

Symbol	**Description**
O	order of magnitude
Δ	difference
∇	gradient operator

Abbreviations

Acronym	**Description**
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy
CJ	Chapman–Jouguet
EOS	Equation of State
SSP	Strong Stability Preserving
WENO	Weighted Essentially Non-oscillatory
ZND	Zel'dovich–von Neumann–Doering

Chapter 1

Introduction

In many physical problems, such as explosions with explosive charges surrounded by or mixed with solid particles [8, 9], pulse detonation propulsion [10], explosive volcanic eruption [11], near-surface landmine explosion [12, 13], impact crater formation [14], and supernova explosion [15], the impulsive dispersal of granular materials commonly occurs and exhibits interesting particle clustering and jetting behaviors. In particular, when granular materials are explosively dispersed [8, 9], mesoscale perturbations develop within and at the surface of the granular cloud at a timescale of shock propagating through the payload, frequently leading to the formation of macroscale coherent particle clustering and jetting structures with a dual hierarchy at a later time. Influencing the mass concentration and related particle reaction and energy release, this particle clustering and jetting phenomenon is of considerable interest to study instability and mixing in high-speed gas-solid flows. However, the underlying mechanism remains unclear and can be largely inhibited by the complex mesoscale multiphase interactions involved in the dispersal process.

1.1 Particle clustering and jetting phenomenon

The presence of granular scales and the rapid release of intensive driving energy are common characteristics for the formation of coherent particle clustering and jetting structures in different particle dispersal processes. Nonetheless, an explosive dispersal can involve additional features related to detonation and shock waves.

1.1.1 Homogeneous detonation

Consisting of fuel and oxidizer premixed on molecular level, explosives are energetic materials that can rapidly release the contained chemical energy through combustion. Depending on the reaction rate, combustion in explosives can happen in the form of either deflagration or detonation. In deflagration, exothermic chemical reactions occur at the material surface, and thermal conduction transfers the released heat to an adjacent unreacted layer for future ignition. Compared to deflagration that is a subsonic layer-by-layer decomposition, detonation is a supersonic combustion involving a closely coupled leading shock wave and exothermic reaction zone propagating through the material [1]. Attaining a higher energy release rate, detonation is the primary reaction form of high explosives [2].

The simplest model of detonation is based on the Chapman–Jouguet (CJ) detonation theory, which treats a detonation wave as a jump discontinuity separating an upstream state and a downstream thermochemical equilibrium state. The reaction zone is assumed to be infinitesimally thin and has an infinite chemical reaction rate. As a jump discontinuity, the flow crossing the detonation wave satisfies the Rankine–Hugoniot conditions, which represent the integral conservation laws of mass, momentum, and energy.

A more physical interpretation of detonation is the Zel'dovich–von Neumann–Doering (ZND) model. As illustrated in Fig. 1.1, the ZND model represents a detonation wave as a shock discontinuity followed by a finite length reaction zone. The reaction zone has a finite chemical reaction rate and terminates with the CJ equilibrium state known as the sonic plane. In the reference frame attached to the leading shock, the flow in the reaction zone is steady. Therefore, the shock and the reaction zone propagate together with a constant velocity referred to as the detonation velocity V_{D}. The flow following the CJ state is a time-dependent rarefaction wave.

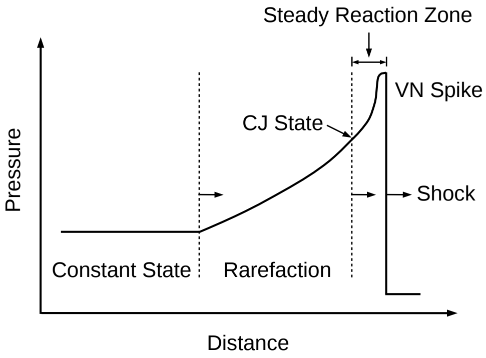


Figure 1.1 One-dimensional ZND detonation model. The schematic diagram is adapted from the work of Fickett and Davis [1].

The structure of a self-sustaining detonation in explosives is further depicted in Fig. 1.2. During a detonation process, the supersonic shock front instantly compresses the explosives into a high-pressure and high-temperature state known as the von Neumann spike and initiates explosives into chemical reaction, in which the main energy transfer occurs in the form of shock wave compression rather than thermal conduction [2]. Meanwhile, the finite length reaction zone, which starts along the leading shock wave and terminates at a sonic plane followed by an unsteady expansion flow, derives energy from the chemical reactions in the explosives and supports the propagating shock front. This close collaboration between shock front and reaction zone causes rapid energy liberation in explosives and generates high-energy gaseous products, which form a fierce expansion flow and are able to do mechanical work on surrounding materials at a remarkable rate.

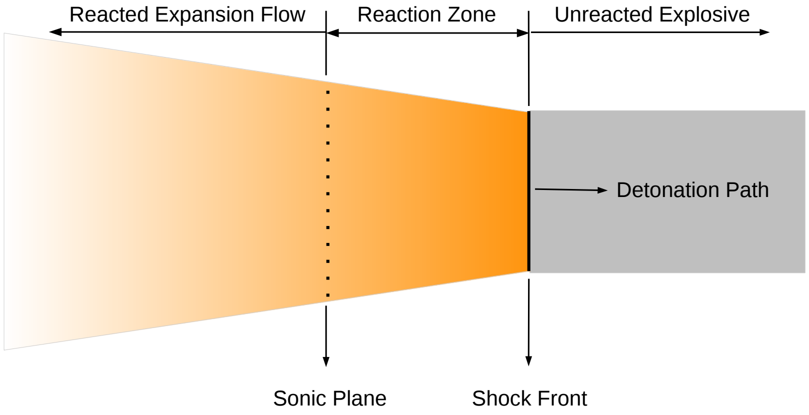


Figure 1.2 A self-sustaining detonation with a coupled leading shock wave and exothermic finite length reaction zone. The schematic diagram is adapted from the work of Bdzil [2].

1.1.2 Heterogeneous detonation

Detonation in explosive materials rapidly releases chemical energy and amasses kinetic and thermal energy. The accumulated energy is then dissipated through a variety of means, such as the formation of pressure waves and thermal radiation. Granular additives such as glass particles or aluminum powders are widely used in explosives to either mitigate or enhance the overall performance [16–19].

Depending on the property of particles, the granular additives can be either inert or energetic. In the former case, particles participate mechanically in the detonation process and act as blast mitigants through absorbing kinetic and thermal energy. When particles are energetic, the timescale of particle ignition and combustion is usually greater than that of explosive detonation, particularly for particles with relatively large sizes. Particles are first dispersed with a significantly increased surface area to volume ratio and then react under the rapid mixing with the detonation products or later with air [18]. The primary heat release from metal particle combustion often occurs behind the propagating detonation reaction zone and has a very limited contribution to sustaining detonation.

Therefore, although energy release from the oxidation of energetic particles can significantly enhance the pressure work and the post-detonation blast wave and hence improve the overall performance of explosives, shock wave dispersion rather than chemical reaction dominates the early-stage interactions between explosives and particles for both inert and energetic particles [17, 18]. The added solid particles introduces inhomogeneities in the explosives, which interact with detonation waves and result in heterogeneous detonation. In general, the process of a heterogeneous detonation involves multiscale physics that comprise microscale, mesoscale, and macroscale interactions.

1.1.2.1 Microscale interaction

At the microscale level, the propagation of detonation shock front causes molecular excitation, vibration, and chemical decomposition and reaction in the condensed explosive, which increase kinetic energy, temperature, pressure and further produce expansion work to support the propagation of the shock front.

The pressure of the detonation shock in high explosives may reach $1.0 \times 10^{10} - 5.0 \times 10^{10}\,\mathrm{Pa}$ with propagation speed ranging from $6.0 \times 10^3 - 9.0 \times 10^3\,\mathrm{m/s}$ [2]. Heating the explosive molecules and initiating chemical reactions, the detonation shock effectively converts the explosives into high-pressure and high-temperature gaseous products, resulting fierce macroscale explosion and expansion behaviors.

1.1.2.2 Mesoscale interaction

While the effects of the structural arrangement of molecules and microstructural imperfections are manifested at the microscale level, and the rapid energy release and expansion processes can be described at the macroscale level with continuum modeling, an intermediate scale, which covers the geometric length scales of $1.0 \times 10^{-7} - 1.0 \times 10^{-3}\,\mathrm{m}$ and the time horizons of $1.0 \times 10^{-6} - 1.0 \times 10^{-3}\,\mathrm{s}$ [3], is required to characterize the heterogeneity of explosives at the granular scale introduced by particle additives and is referred to as the mesoscale.

As illustrated in Fig. 1.3, grain-scale heterogeneities can cause intricate mesoscale physics, including strongly coupled shock-shock, shock-particle, and particle-particle interactions.

The presence of voids, density discontinuities, irregular material interfaces, and other inhomogeneities causes shock reflection, diffraction, and interference patterns, especially at the leading edges of particles, where strong wave reflection happens, and at concave regions of neighboring particles, where shock waves collide. These shock-shock interactions not only influence the detonation front curvature and the propagation behavior of detonation

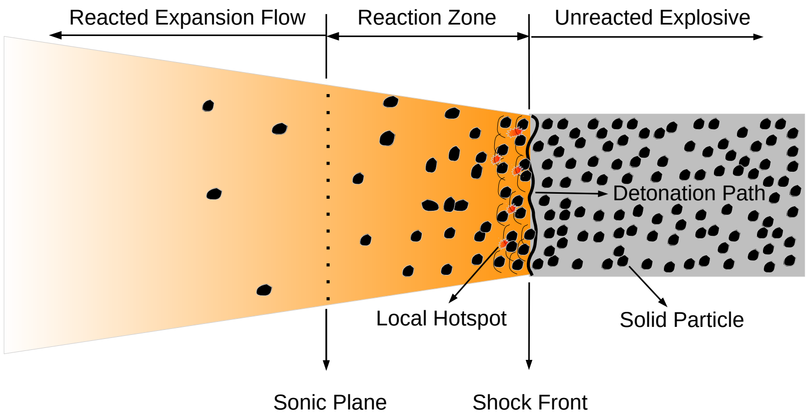


Figure 1.3 Heterogeneous detonation with grain-scale effects introduced by particle additives. The schematic diagram is adapted from the work of Bdzil [2] and Ripley et al. [3]. In the current diagram, the detonation reaction zone length is assumed to be much larger than particle diameter.

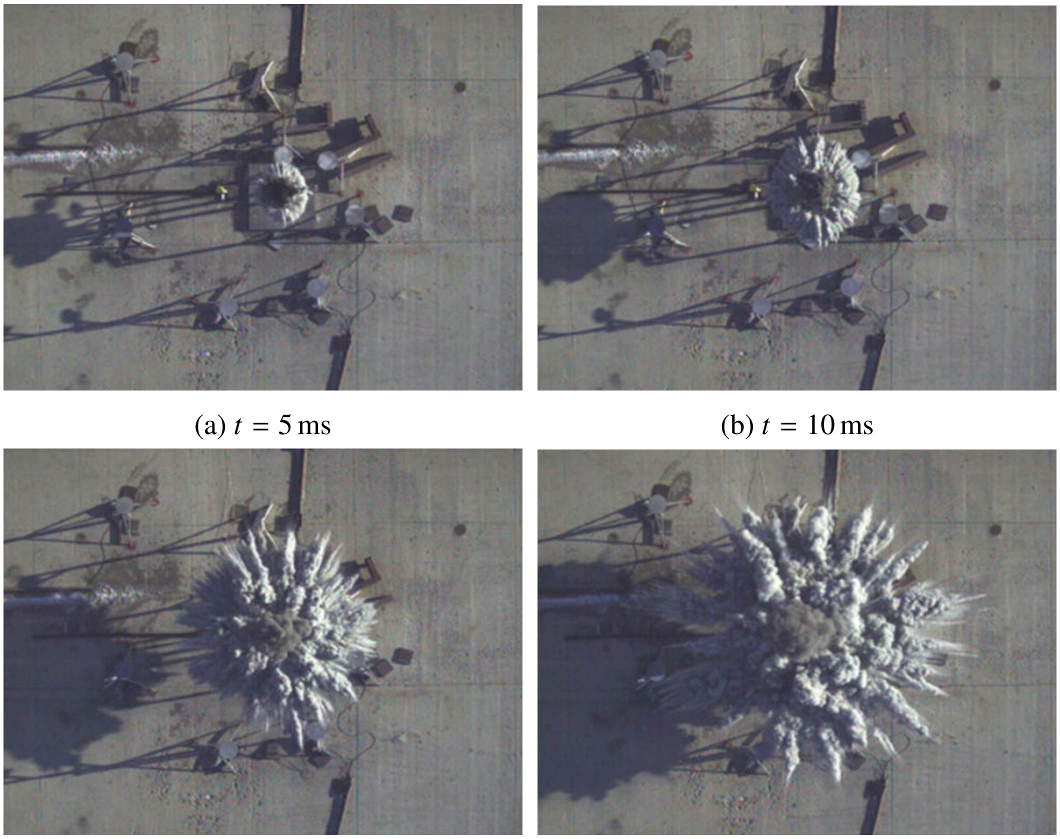
waves, but also produce local hotspots and pressure fluctuations at material interfaces, affecting the stability of detonation and granular payload motion.

When interacting with particles, detonation waves transmit strong pressure waves into particles. Due to the shock impedance differences between explosives and particles, a wave speed mismatch between the shock traveling in explosives and the shock propagating in particles usually presents. The wave reflections and interactions in the detonation flow result in local pressure jumps and velocity discontinuities around particles, which initially and impulsively accelerates the particle material. As the detonation waves propagate over particles, the length of detonation reaction zone may be considerably increased due to the energy transfer from detonation waves to particles. The density ratio of explosives to metal particles and length ratio of detonation reaction zone to particle diameter are important measurable parameters to characterize the effectiveness of energy transfer and the shock propagation behaviors at material interfaces, respectively. The dominance of mesoscale shock-particle interaction changes from phase-frozen shock to shock diffraction at interfaces when the ratio of detonation reaction zone length to particle diameter decreases [3].

Under shock compaction, particles experience collision, agglomeration, deformation, and even fragmentation. The granular payload is first compacted with an increased particle volume fraction. Then, subjecting to a diverging expansion, the particle system undergoes a dense-to-dilute transition characterized by a decreasing particle volume fraction [8].

1.1.2.3 Macroscale cluster and jet formation

The introduced grain-scale heterogeneities lead to coupled mesoscale interactions between the detonation reaction zone, shock waves, and solid particles. Under high rates of momentum and energy transfer resulting from shock compaction and subsequent expansion flow, the mesoscale interactions are effectively enhanced and later are manifested via complex macroscale phenomena as the expansion proceeds. As shown in Fig. 1.4, one important phenomenon is the formation of coherent particle clustering and jetting structures. Influencing the mass concentration and related particle reaction and energy release, the underlying physical mechanism is of significant interest to study flow instability and turbulent mixing in heterogeneous detonation and explosion [18, 20].



(a) $t = 5\,\mathrm{ms}$

(b) $t = 10\,\mathrm{ms}$

(c) $t = 25\,\mathrm{ms}$

(d) $t = 50\,\mathrm{ms}$

Figure 1.4 Particle cluster and jet formation under the explosive dispersal of a cylindrical charge that comprises a central explosive burster surrounded by an annular Aluminum payload. Figure from Zhang et al. [4]. Photographs from Defence Research and Development Canada, courtesy of Dr. Fan Zhang.

1.1.3 Clustering and jetting instability

In experiments concerning the explosive dispersal of liquids [4, 9, 17, 20], dry powders [9, 17, 18, 20, 21], or hybrid liquid-powder mixtures [8, 9, 20, 22] in a stratified [4, 9, 17, 18, 20–22] or premixed [8] burster-payload configuration with a cylindrical [4, 9, 18, 20, 22] or spherical [8, 9, 17, 21] charge shape, a dual hierarchy of particle jets featuring a limited number of large primary jets accompanied by numerous small fine jets, as illustrated in Fig. 1.5, was frequently observed, although different configurations can influence the number and length scales of the formed jets. In general, the fine jets occurred at the outer payload-air interface immediately after the fragmentation of casing, and the primary jets emerged out of the surface of the fine jets at a later time. While the fine jets dissipated through aerodynamic interaction and destroyed by payload expansion as the dispersal proceeds, the primary jets persisted as coherent jetting structures for a longer time. This jetting structure can be developed within an expansion length of several times the initial charge diameter [4].

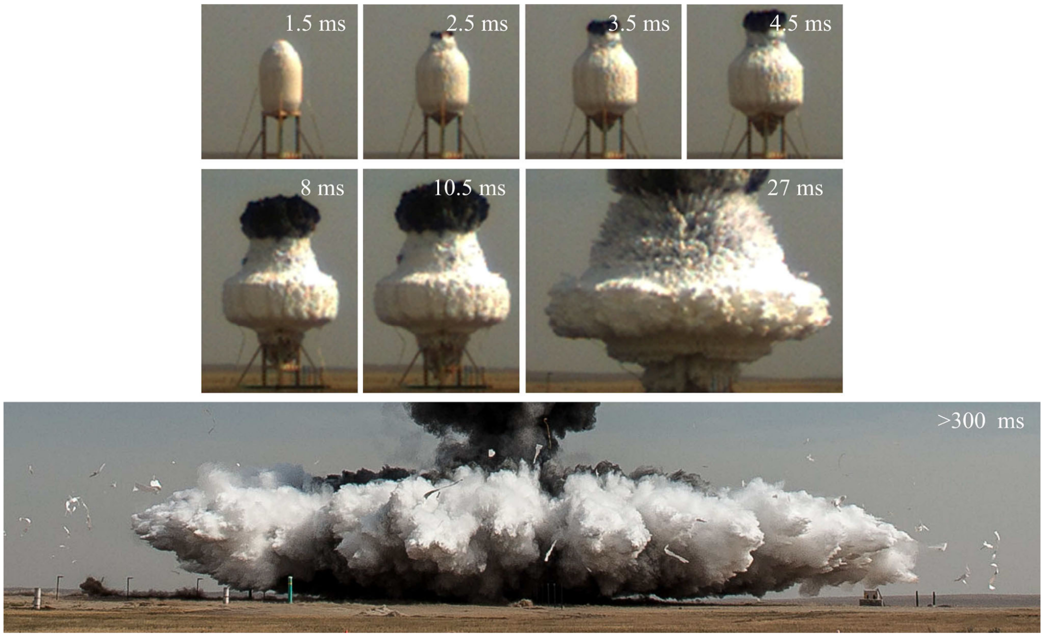


Figure 1.5 A dual hierarchy of particle jets observed in the explosive dispersal of 5, 090 kg gasoline. Figure from Zhang et al. [4]. Photographs from Defence Research and Development Canada, courtesy of Dr. Fan Zhang.

Casing fragmentation at the inner burster-payload and outer payload-air interfaces could enhance the interfacial behaviors and hence the formation of particle clusters and jets, particularly for liquid payloads [4, 22]. Strong shocks generated by detonation can induce plastic deformation and agglomeration of solid grains such as those made of aluminum to form a casing-like structure near the inner burster-payload interface, which subsequently fragments under expansion and introduces interfacial instabilities [23]. Less affected by the compression and plastic deformation, liquids can still reveal granular behaviors after breakup and mixing with air. The formation of coherent particle clustering and jetting structures with a dual hierarchy was also observed in the absence of casings and plastic deformations, such as in experiments on dispersed flour or polystyrene powders using Hele-Shaw cells with blast of moderate overpressures [24–26], in which particle rings were confined between two closely separated parallel plates and dispersed by a central gas burster. This consistency suggests an inherent clustering and jetting instability in impulsively dispersed granular materials.

Compared with 3D spherical or cylindrical settings, quasi-2D Hele-Shaw cell configurations [24–26] have provided better visualization of the dispersal evolution. Nonetheless, extra effects can be introduced by the confining plates, such as wall boundaries, burster fluid tending to escape near the upper plate, and payload might dragged by the lower plate under gravity and plate friction. When considered as the zero-surface-tension limit of a high viscous fluid, granular materials such as spherical glass beads dispersed by a gas burster can exhibit the finger-like branching pattern featured in Saffman–Taylor instabilities (STI) [27], particularly at the early fragmentation stage of the payload. Widely studied via the displacement of a viscous fluid by a less viscous one in Hele-Shaw cells, STI [28] has been linked to the patten formation in granular materials among recent studies [27, 29], in which granular dissipation instead of the classical viscous damping is characterized as the stabilizing mechanism for counterbalancing the destabilizing pressure gradient.

From the perspective of accelerated material interfaces with density disparity and initial interfacial nonuniformity, particle clusters and jets at the inner burster-payload and outer payload-air interfaces can be related to the Rayleigh–Taylor instability (RTI) [30, 31] and Richtmyer–Meshkov instability (RMI) [32, 33], in which RMI can be considered as an impulsive RTI subjected to shock rather than continuous acceleration [34].

RTI and RMI can grow initial surface corrugations with locally misaligned pressure and density gradients in the form of spikes, bubbles, and fingers [34]. One example is supernova explosion generated by the energy release from the gravitational collapse of a star that has exhausted nuclear fuel [15], in which astrophysical-scale RTI fingers develop in the process of the core medium accelerating into the denser interstellar medium [15, 35]. Another example is inertial confinement fusion implosion, in which a fusion target, a spherical pellet consists of layered materials such as an ablator, a pusher, and a fuel core, is compressed to the fuel ignition point by a high-energy source for fusion reactions [36]. RTI, accompanied by RMI, is recognized as the primary cause of the asymmetric rupture of the outer ablator-pusher interface in the inward acceleration phase. The amplified perturbations via RTI will feed through the shell and subsequently seed the development of RTI at the inner pusher-fuel interface when the pellet shell is decelerated by the high pressure building up inside the shell [36, 37].

The macroscale explosive dispersal of granular materials shares many similarities, such as burster-payload interaction, layered material shells, rapid acceleration under intensive energy release, and shock presence, with astrophysical supernova explosions and microscale inertial confinement fusion implosions with an inverse acceleration direction. Through measuring the growth rates of particle jets, recent Hele-Shaw cell studies [26] on blast-dispersed flour rings suggest that the formation of fine jets at the outer payload-air interface can also be seen as a form of RTI. Initiated very early but revealed later, primary jets are only weakly linked to RTI due to the discrepancy in timescales and growth rates [17, 26]. Therefore, a distinct evolution process should be proposed for the primary and fine jets, respectively, in which numerical simulations could better reveal the early time multiphase interactions through visualizing the fluid and solid motions simultaneously, particularly for the complex coupling between gas and granular motions and the nonlinear force-chain networks in granular materials.

1.1.4 Modeling issues and challenges

The formation of particle clusters and jets involves a fluid-solid system that covers multiple temporal and spatial scales. In numerical modeling of a multiphase and multiscale system, a small-scale such as micro/mesoscale model built on the first principle of physical laws can be very accurate but not efficient enough for large systems, while a large-scale such as macroscale model built on continuum mechanics can be efficient but not accurate enough for resolving small-scale structures. Therefore, multiscale modeling that employs a hierarchy of reference frames and physical laws to describe the system on different spatial scales and provide varying levels of details is needed to balance accuracy and efficiency. According to the possible reference frames for the fluid and solid phases, two widely used approaches [38] are Eulerian–Eulerian and Eulerian–Lagrangian descriptions.

An Eulerian–Eulerian approach, referred to as macroscale modeling, treats each phase as a separate flow using a continuum model, in which the constitutive equation for granular flow can be derived from soil mechanics [39], visco-plastic analogy [40], or kinetic theory [41] when considering the granular flow as elastic-quasistatic, visco-plastic, and rapid collisional, respectively, and describes the phase interactions such as drag forces via interphase exchange source terms [38]. Due to the continuum modeling, granular physics such as particle motions and collisions are embedded in volume-averaged field quantities, limiting the ability for resolving grain-scale structures and physics [42]. Meanwhile, an Eulerian–Lagrangian approach describes the fluid and solid phases using a continuum model and a discrete model, respectively. As each solid is tracked as a discrete entity, individual granular behaviors can be modeled in detail.

Depending on whether phase interfaces are resolved in the coupling of fluid-solid interactions, Eulerian–Lagrangian models can be further categorized into non-resolved and resolved models, referred to as semi-mesoscale and mesoscale modeling, respectively. That is, length and model scales herein characterized by molecules, non-resolved grains, resolved grains, and continua are referred to as microscale, semi-mesoscale, mesoscale, and macroscale, respectively. The classification of length scales can be a matter of debate in the literature. When focusing on the modeling analogy to collisional molecular dynamics but ignoring the physical disparity between molecules and grains, the grain-scale is referred to as the microscale in references like [42, 43], in which granular clusters with sizes being 10 – 100 times granular diameters are categorized into mesoscale. In a non-resolved model such as the discrete element model [26, 44, 45], the Eulerian grid is usually several orders of magnitude larger than solid sizes. While solids in the Lagrangian frame describing the solid phase preserve a finite volume and interact through spring-based collision rules, solids in the fluid domain are treated as point sources and sinks of momentum, whose interaction with the fluid phase requires closure laws similar to those in Eulerian–Eulerian models [38].

In an explosive dispersal, in addition to the multiphase and multiscale properties, the fluid-solid system experiences a flow regime with dense-to-dilute transitions characterized by a time-dependent particle volume fraction ϕ, in which the flow regime changes from a granular flow ($\phi > 50\%$) featuring inelastic multibody contact and collisions to a dense gas-granule flow ($1\% \leq \phi \leq 50\%$) characterizing strong gas-granule and granule-granule interactions and further to a dilute gas-granule flow ($\phi < 1\%$) outlining aerodynamic forces and less severe boundary layer interactions among individual particles [8]. In supplying closure laws for Eulerian–Eulerian and non-resolved Eulerian–Lagrangian models, while standard models perform relatively well for dilute gas-granule flows with low-pressure conditions, momentum exchange laws that can well describe flow regimes with dense-to-dilute transitions and shocked flow conditions still undergoes active development through experimental correlations [8, 46, 47], numerical simulations [46, 48, 49], or surrogate models [50].

Adopting a resolved Eulerian–Lagrangian approach can achieve direct simulations that avoid the necessity of correlated closure laws and automatically activate interphase interactions via boundary conditions at phase interfaces. The challenges are then mainly shifted to irregular and moving boundaries, mesh generation and regeneration, computational efficiency and robustness, and particle-particle interactions. Advances in Cartesian-grid methods such as immersed boundary methods [51], immersed interface methods [52, 53], ghost fluid methods [54, 55], and embedded boundary methods [56, 57] have greatly simplified the grid generation and regeneration for treating irregular and moving boundaries using non-body-conformal Cartesian grids [58–62], which enable cost-effective direct simulations for explosive dispersal problems, provided a method that can robustly handle irregular, moving, and colliding geometries with shocked flow conditions.

Under a dense setting, particle interactions can strongly influence the stresses in the fluid-solid mixture [63–65] and generate nonlinear force-chain networks in the granular cloud [66]. With fundamental preconditions like random, binary collision, and sufficient inter-particle distance, kinetic theory has been extended to describe granular systems with non-Maxwellian velocity distribution [41]. Nonetheless, multibody collisions are excluded through assuming being comparably rare and not affecting the statistical properties of the system. However, the dispersal problem concerned herein contains an initial granular system with sustained contacts among inherently dissipative grains, in which multibody collisions are responsible for the transfer of momentum and energy and can no longer be reasonably excluded, introducing challenges in solving collision detection and response. Multibody collision in theory remains as an unsolved problem [67]. A few approximation attempts such as sequential pairwise collision [68] and perfectly inelastic non-contact collision [69] have been explored in interface-resolved multibody collision modeling. Nonetheless, a deterministic multibody contact and collision model can better facilitate the investigations of the particle clustering and jetting problem.

1.2 Motivations and findings

For the purpose of advancing the understanding of particle clustering and jetting dynamics, particularly the governing instability of primary jets, through capturing mesoscale multiphase interactions, this thesis develops a resolved Eulerian–Lagrangian approach to directly simulate gas-solid flow with shock-shock, shock-particle, and particle-particle interactions, in which the Eulerian gird describing the fluid phase is at least one order of magnitude less than particle sizes to resolve phase interfaces, fluid-solid interaction is via enforcing boundary conditions at phase interfaces, and particle-particle interaction is through a deterministic multibody contact and collision model with parameterized coefficients of restitution and friction.

The framework is implemented to create a computational solver and then validated for solving flow with irregular, moving, and colliding granular bodies. Employing the theory of Gurney velocity [70], a valid statistical dissipative property of the framework in solving explosively dispersed granular materials is also demonstrated. Combining direct simulations herein and experimental observations in the literature, an explanation for particle clustering and jetting instabilities that admits a dual structure and concerns the effects of shock compaction, interfacial instabilities, inelastic collisions, interstitial fluid, particle properties, payload morphologies, and burster-payload relations is proposed.

1.3 Novelty and contribution

This thesis concentrates on modeling and solving the coupled and interface-resolved multiphase interactions in gas-solid systems, with the purpose of advancing the understanding of particle clustering and jetting instabilities in explosively dispersed granular materials through direct simulations. The main novelty and contribution are summarized as follows:

1. A field function [71].
    - Facilitates the solution of complex and dynamic fluid-solid systems on Cartesian grids.
    - Provides single-integer-based multidomain node mapping and efficient node remapping.
    - Enables linear-time collision detection and expedient surface force integration.
2. A collision model [71].
    - Adopts exact pairwise collisions with parameterized coefficients of restitution and friction.
    - Achieves deterministic multibody contact and collision response.
3. An immersed boundary method [72].
    - Develops a second-order three-step flow reconstruction scheme.
    - Enforces the Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner.
    - Provides efficient, accurate, and robust boundary treatment for arbitrarily irregular and moving boundaries.
4. A Cartesian, 3D, high-resolution, parallelized, gas-solid flow solver [73].
    - Establishes a numerical framework for the direct simulation of gas-solid flows.
    - Solves coupled and interface-resolved fluid-fluid, fluid-solid, and solid-solid interactions.
    - Addresses shocked flow conditions, irregular and moving geometries, and multibody contact and collisions.
    - Unifies 1D, 2D, and 3D computations with the generation of complex geometric objects via simply positioning components.
5. Advancement in understanding particle clustering and jetting instabilities [74, 75].
    - Designs a set of cases that cover different types of stochastic payloads, burster states, and coefficients of restitution.
    - Demonstrates a valid statistical dissipative property in solving explosively dispersed granular materials with respect to Gurney velocity.

- Extends the time range of the velocity scaling law with regard to Gurney energy in the Gurney theory from the steady-state termination phase to the unsteady evolution phase.
- Proposes an explanation for particle clustering and jetting instabilities to increase the understanding of established experimental observations in the literature.

Being a fundamental problem in many engineering applications, the interface-resolved predictive modeling of gas-solid flows is an active area of research and requires addressing a series of challenges. Therefore, the development of an accurate, efficient, and robust direct simulation numerical framework contributes to numerical methods applied to complex gas-solid flow problems. In addition, the advancement in understanding particle clustering and jet instabilities can potentially contribute to the development and application of a number of engineering problems, including heterogeneous explosives.

1.4 Thesis structure

This thesis is structured as the following:

- Chapter 1 first introduces the particle clustering and jetting phenomenon and its significance. After analyzing the fundamentals and reviewing the existing studies, the motivations and objectives of this thesis are stated and explained. Meanwhile, the novelty and contribution of the study are summarized, and an overview of the thesis structure is provided.
- Chapter 2 describes and justifies the research design and methodology.
- Chapter 3 presents a field function for solving complex and dynamic fluid-solid systems and a collision model.
- Chapter 4 develops an immersed boundary method for treating arbitrarily irregular and moving boundaries.
- Chapter 5 investigates particle clustering and jetting instabilities in explosively dispersed granular materials.
- Chapter 6 draws final conclusions, discusses the limitations of the current study, and proposes areas for future research.
- Appendix A provides a detailed reference for the 3D Navier–Stokes equations, the Jacobian matrices, and their eigendecompositions.
- Appendix B presents some additional code validation of the developed computational solver.

Chapter 2

Research Design and Methodology

To study particle cluster and jet formation via direct simulations, this thesis includes two major scopes: the development of a high-fidelity gas-solid flow solver and the numerical investigation on particle clustering and jetting instabilities in explosively dispersed granular materials.

2.1 Solver development

In resolving the multiscale and multiphase interactions in explosively dispersed granular materials through direct simulation, the principal challenges are associated with fluid-solid interaction, dynamic phase interface, mesh generation and regeneration, multibody contact and collision, complex interfacial condition, and computational efficiency and robustness. Moreover, the presence of high-pressure shock waves further reinforces those challenges. Therefore, in addition to integrating existing numerical techniques, to obtain a capable computational solver also requires developing new mathematical models and numerical methods to bridge the gap between the available tools and remaining problems.

2.1.1 Fluid-solid coupling

A fully coupled fluid-solid interaction algorithm can support the development of a general solver for all the problem domains but usually involves implicit discretization of governing equations and solves large matrix systems by iteration, which imposes stringent conditions on the solution process. In contrast, a partitioned algorithm decouples physical processes in the time domain. Adopting numerical techniques optimized for each problem domain and physical process, a partitioned approach has advantages in flexibility and efficiency [76].

A partitioned fluid-solid interaction algorithm with second-order temporal accuracy, which is obtained via applying Strang splitting [77] to split physical processes, is employed to model the coupling between fluid and solid motions:

$$\boldsymbol{U}^{n+1} = \mathcal{S}_{\mathrm{s}}(\frac{\Delta t}{2})\,\mathcal{S}_{\mathrm{f}}(\frac{\Delta t}{2})\,\mathcal{S}_{\mathrm{f}}(\frac{\Delta t}{2})\,\mathcal{S}_{\mathrm{s}}(\frac{\Delta t}{2})\boldsymbol{U}^{n} \tag{2.1}$$

in which $\boldsymbol{U}^n$ and $\boldsymbol{U}^{n+1}$ denote the solution vectors of physical quantities at time t^n and t^{n+1}, respectively; $\mathcal{S}_{\mathrm{s}}$ and $\mathcal{S}_{\mathrm{f}}$ represent the solution operators of solid dynamics and fluid dynamics, respectively.

Traditionally, to reduce the computational load resulting from a symmetric splitting, the group property of the solution operator, $\mathcal{S}(\Delta t_1)\,\mathcal{S}(\Delta t_2) = \mathcal{S}(\Delta t_1 + \Delta t_2)$, is applied in Strang splitting to concatenate consecutive operators of the same type [76, 77]. Consequently, the algorithmic uniformity is not preserved at the start and end steps of the computation cycle. By observing that a doubled range of the CFL number can be adopted, the present ungrouped form preserves algorithmic uniformity while achieving equivalent overall efficiency for temporal integration or higher efficiency when more than two solution operators are involved.

2.1.2 Governing equations and discretization

2.1.2.1 Fluid dynamics

The motion of fluids is described by the conservative form of the Navier–Stokes equations in Cartesian coordinates:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_i}{\partial x_i} = \frac{\partial \boldsymbol{F}_i^{\mathrm{v}}}{\partial x_i} + \boldsymbol{\Phi} \tag{2.2}$$

The vectors of conservative variables $\boldsymbol{U}$, convective fluxes $\boldsymbol{F}_i$, diffusive fluxes $\boldsymbol{F}_i^{\mathrm{v}}$, and source terms $\boldsymbol{\Phi}$ are as follows:

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho V_j \\ \rho e_{\mathrm{T}} \end{pmatrix},\ \boldsymbol{F}_i = \begin{pmatrix} \rho V_i \\ \rho V_i V_j + p\delta_{ij} \\ (\rho e_{\mathrm{T}} + p)V_i \end{pmatrix},\ \boldsymbol{F}_i^{\mathrm{v}} = \begin{pmatrix} 0 \\ \tau_{ij} \\ k\frac{\partial T}{\partial x_i} + \tau_{il}V_l \end{pmatrix},\ \boldsymbol{\Phi} = \begin{pmatrix} 0 \\ f_j^{\mathrm{b}} \\ f_l^{\mathrm{b}} V_l \end{pmatrix} \tag{2.3}$$

where ρ is the density, $\boldsymbol{V}$ is the velocity, $e_{\mathrm{T}} = e + \boldsymbol{V}\cdot\boldsymbol{V}/2$ is the specific total energy, e is the specific internal energy, p is the thermodynamic pressure, $\boldsymbol{\tau}$ is the viscous stress tensor, T is the temperature, k is the thermal conductivity, $\boldsymbol{f}^{\mathrm{b}}$ represents external body forces such as gravity, i is a free index, j is an enumerator, l is a dummy index.

In order to simplify numerical discretization while preserving the principal physics of fluid motion, the current closure of the system is through supplying the Newtonian fluid relation with the Stokes hypothesis

$$\tau_{ij} = \mu\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i} - \frac{2}{3}(\nabla\cdot\boldsymbol{V})\delta_{ij}\right) \tag{2.4}$$

and the perfect gas law

$$\begin{aligned} p &= \rho R T \\ e &= C_v T \end{aligned} \tag{2.5}$$

in which R is the specific gas constant, and C_v is the specific heat capacity at constant volume.

The use of Navier–Stokes equations for describing fluid motion provides flexibility in the consideration of viscous effects. When considering viscous flows, the dynamic viscosity, μ, is non-zero and can be determined by the Sutherland viscosity law

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{T + C_2} \tag{2.6}$$

where $C_1 = 1.458 \times 10^{-6}\,\mathrm{kg \cdot m^{-1} \cdot s^{-1} \cdot K^{-1/2}}$ and $C_2 = 110.4\,\mathrm{K}$ are two constant coefficients. When omitting flow viscosity, $\mu = 0$ is then adopted to obtain inviscid flows. Since the primary focus is to solve flow involving irregular, moving, and colliding geometries under shocked flow conditions, the modeling of turbulence and the resolving of boundary layers are not specifically concerned in the present study.

The temporal integration is performed via the second-order or third-order strong stability-preserving (SSP) Runge–Kutta method [78, 79]. For instance, the third-order one can be described as the following:

$$\begin{aligned} \boldsymbol{U}^{(1)} &= \mathcal{L}\mathcal{L}\,\boldsymbol{U}^n \\ \boldsymbol{U}^{(2)} &= 3/4\boldsymbol{U}^n + 1/4\,\mathcal{L}\mathcal{L}\,\boldsymbol{U}^{(1)} \\ \boldsymbol{U}^{n+1} &= 1/3\boldsymbol{U}^n + 2/3\,\mathcal{L}\mathcal{L}\,\boldsymbol{U}^{(2)} \\ \mathcal{L}\mathcal{L} &= (\mathbf{I} + \Delta t\,\mathcal{L}) \end{aligned} \tag{2.7}$$

where $\mathbf{I}$ is the identity matrix, operator $\mathcal{L} = \mathcal{L}_x + \mathcal{L}_y + \mathcal{L}_z$, $\mathcal{L}_x$, $\mathcal{L}_y$, and $\mathcal{L}_z$ represent the spatial operators of x, y, and z dimension, respectively.

For the system of conservation laws in multidimensional space, the discretization of spatial operators can be conducted using dimension-by-dimension [78] or dimensional-splitting [77] approximation. While the former preserves temporal accuracy, the latter has a much less severe stability constraint. To guarantee discrete mass conservation, conservative discretization is applied for all the spatial derivatives. Using the x dimension as an example, the flux derivative at a node i is approximated as

$$\left.\frac{\partial \boldsymbol{F}}{\partial x}\right|_i = \frac{1}{\Delta x}\left[\hat{\boldsymbol{F}}_{i+\frac{1}{2}} - \hat{\boldsymbol{F}}_{i-\frac{1}{2}}\right] \tag{2.8}$$

where $\boldsymbol{F}$ represents either the convective flux vector or the diffusive flux vector, $\hat{\boldsymbol{F}}_{i+1/2}$ is a numerical flux at the interface between the discretization interval $\Omega_i = [x_{i-1/2}, x_{i+1/2}]$ and $\Omega_{i+1} = [x_{i+1/2}, x_{i+3/2}]$.

In convective flux discretization, the system of equations is projected onto its characteristic fields to locally decompose the vector system into a set of scalar conservation laws. Then, a flux splitting method, such as Lax–Friedrichs splitting [80] or Steger–Warming splitting [81], is conducted on the scalar fluxes to ensure upwinding property

$$f(u) = f^{+}(u) + f^{-}(u), \ \frac{\mathrm{d}f^{+}(u)}{\mathrm{d}u} \geq 0, \ \frac{\mathrm{d}f^{-}(u)}{\mathrm{d}u} \leq 0 \tag{2.9}$$

where f is a scalar characteristic flux.

Since both the forward flux and backward flux are discretized in conservative form, the discretization of a scalar flux derivative has the form

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{1}{\Delta x}\left[\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right], \ \hat{f}_{i+\frac{1}{2}} = \hat{f}^{+}_{i+\frac{1}{2}} + \hat{f}^{-}_{i+\frac{1}{2}}, \ \hat{f}_{i-\frac{1}{2}} = \hat{f}^{+}_{i-\frac{1}{2}} + \hat{f}^{-}_{i-\frac{1}{2}} \tag{2.10}$$

The third-order or fifth-order weighted essentially nonoscillatory (WENO) scheme [82] is then applied for the reconstruction of numerical fluxes. Since the numerical flux $\hat{f}^{+}_{i+1/2}$ and $\hat{f}^{-}_{i-1/2}$ are symmetric with respect to x_i, only the reconstruction of the former is described herein. The latter can be obtained via replacing all $+$ and $-$ signs in the superscript and subscript of each variable in the equations by the corresponding opposite signs $-$ and $+$, respectively. For the fifth-order WENO scheme, the numerical flux $\hat{f}^{+}_{i+1/2}$ has the following form:

$$\hat{f}^{+}_{i+\frac{1}{2}} = \sum_{n=0}^{N} \omega^{+}_n q^{+}_n(f^{+}_{i+n-N}, \ldots, f^{+}_{i+n}), \ N = (r-1) = 2 \tag{2.11}$$

where

$$\begin{aligned}
q^{+}_0(f^{+}_{i-2}, \ldots, f^{+}_i) &= (2f^{+}_{i-2} - 7f^{+}_{i-1} + 11f^{+}_i)/6 \\
q^{+}_1(f^{+}_{i-1}, \ldots, f^{+}_{i+1}) &= (-f^{+}_{i-1} + 5f^{+}_i + 2f^{+}_{i+1})/6 \\
q^{+}_2(f^{+}_i, \ldots, f^{+}_{i+2}) &= (2f^{+}_i + 5f^{+}_{i+1} - f^{+}_{i+2})/6 \\
\omega^{+}_n &= \frac{\alpha^{+}_n}{\alpha^{+}_0 + \cdots + \alpha^{+}_N}, \ \alpha^{+}_n = \frac{C_n}{(\varepsilon + IS^{+}_n)^2}, \ \varepsilon = 10^{-6} \\
C_0 &= \frac{1}{10}, \ C_1 = \frac{6}{10}, \ C_2 = \frac{3}{10} \\
IS^{+}_0 &= \frac{13}{12}(f^{+}_{i-2} - 2f^{+}_{i-1} + f^{+}_i)^2 + \frac{1}{4}(f^{+}_{i-2} - 4f^{+}_{i-1} + 3f^{+}_i)^2 \\
IS^{+}_1 &= \frac{13}{12}(f^{+}_{i-1} - 2f^{+}_i + f^{+}_{i+1})^2 + \frac{1}{4}(f^{+}_{i-1} - f^{+}_{i+1})^2 \\
IS^{+}_2 &= \frac{13}{12}(f^{+}_i - 2f^{+}_{i+1} + f^{+}_{i+2})^2 + \frac{1}{4}(3f^{+}_i - 4f^{+}_{i+1} + f^{+}_{i+2})^2
\end{aligned} \tag{2.12}$$

where r is the number of candidate stencils, q_n are the r-th order approximations of $\hat{f}_{i+1/2}$ on the candidate stencils $S_n = (x_{i+n-N}, \ldots, x_{i+n})$, ω_n are the actual weights of q_n, which are determined by the smoothness of solution in the candidate stencils S_n, as measured by

IS_n, and C_n are optimal weights to ensure that the convex combination of q_n converges to a $(2r-1)$-th order approximation of $\hat{f}_{i+1/2}$ on the undivided stencil $S = (x_{i-N}, \ldots, x_{i+N})$ in smooth regions.

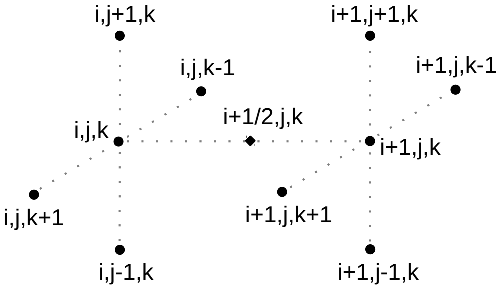


Figure 2.1 A schematic diagram illustrating interfacial diffusive flux reconstruction.

The second-order central difference discretization is employed for diffusive fluxes. A conservative discretization of the diffusive fluxes involves consecutive differentiation, which may lead to an even-odd decoupling issue [83, 84]. Therefore, the reconstruction function should be carefully devised. In this work, the interfacial flux $\hat{\boldsymbol{F}}^{\mathrm{v}}_{i+1/2}$ is reconstructed on the discretized space $[i, i+1]\times[j-1, j+1]\times[k-1, k+1]$, as illustrated in Fig. 2.1. Let ϕ denote a physical quantity in $\boldsymbol{F}^{\mathrm{v}}$, in order to avoid even-odd decoupling resulting from applying consecutive derivative discretization, the following reconstructions can be adopted:

$$\begin{aligned}
\phi_{i+\frac{1}{2},j,k} &= \frac{\phi_{i,j,k} + \phi_{i+1,j,k}}{2} \\
\left.\frac{\partial \phi}{\partial x}\right|_{i+\frac{1}{2},j,k} &= \frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x} \\
\left.\frac{\partial \phi}{\partial y}\right|_{i+\frac{1}{2},j,k} &= \frac{\phi_{i,j+1,k} + \phi_{i+1,j+1,k} - \phi_{i,j-1,k} - \phi_{i+1,j-1,k}}{4\Delta y} \\
\left.\frac{\partial \phi}{\partial z}\right|_{i+\frac{1}{2},j,k} &= \frac{\phi_{i,j,k+1} + \phi_{i+1,j,k+1} - \phi_{i,j,k-1} - \phi_{i+1,j,k-1}}{4\Delta z}
\end{aligned} \tag{2.13}$$

Computation acceleration The conservative discretization of fluxes enables an acceleration technique that can reduce nearly half of the computational work by using the fact that the same interfacial flux is shared by the two neighboring cells or nodes

$$\hat{\boldsymbol{F}}_{s-\frac{1}{2}} = \hat{\boldsymbol{F}}_{(s-1)+\frac{1}{2}} \tag{2.14}$$

where flux $\boldsymbol{F}$ is either a convective flux or a diffusive flux, and s is one of the i, j, k index.

Let $\mathcal{L}_s$ denote the spatial operator in the s dimension for a multidimensional problem with dimension-by-dimension approximation, and let $\mathcal{S}_s$ denote the solution operator of the subproblem

$$\frac{\partial \boldsymbol{U}}{\partial t} = \mathcal{L}_s \boldsymbol{U}^{(s)} \tag{2.15}$$

for a multidimensional problem with dimensional-splitting. When computing $\mathcal{L}_s$ or $\mathcal{S}_s$, if the s index is arranged as the innermost loop of the spatial sweep, then only the interfacial flux $\hat{\boldsymbol{F}}_{s+1/2}$ is always required to be computed, while $\hat{\boldsymbol{F}}_{s-1/2}$ can be inherited from the previous node $(s-1)$ unless the node $(s-1)$ is a boundary node. If $(s-1)$ is a boundary node, then $\hat{\boldsymbol{F}}_{s-1/2}$ requires to be computed. When the type of each computational node is classified in advance, the implementation of this acceleration technique is straightforward, and the computational overhead of the implementation is one single conditional statement.

2.1.2.2 Solid dynamics

The motion of solids is governed by the equation system comprising the Newton's second law of translational motion and the Euler equations of rotational motion:

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{\Phi},\ \boldsymbol{U} = \begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{x}_{\mathrm{c}} \\ \mathbf{I}_{\mathrm{c}}\boldsymbol{\omega} \\ \boldsymbol{\theta} \end{pmatrix},\ \boldsymbol{\Phi} = \begin{pmatrix} \frac{1}{m} \int\limits_{\partial\Omega} \mathbf{n} \cdot (-p\mathbf{I} + \boldsymbol{\tau})\,\mathrm{d}S + \boldsymbol{g} \\ \boldsymbol{V} \\ \int\limits_{\partial\Omega} (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}}) \times [\mathbf{n} \cdot (-p\mathbf{I} + \boldsymbol{\tau})]\,\mathrm{d}S \\ \boldsymbol{\omega} \end{pmatrix} \tag{2.16}$$

where $\boldsymbol{x}$ is the position vector of spatial points, Ω is the spatial domain occupied by a solid, $\boldsymbol{x}_{\mathrm{c}}$ is the position vector of the solid centroid, $\boldsymbol{\theta}$ is the orientation (vector of Euler angles) of the solid, $\boldsymbol{V}$ and $\boldsymbol{\omega}$ are the translational and angular velocities of the solid, respectively, m is the mass of the solid, $\mathbf{I}_{\mathrm{c}}$ is the moment of inertia matrix, $\mathbf{n}$ is the unit outward surface normal vector, p and $\boldsymbol{\tau}$ are the pressure and viscous stress tensor field exerted on the solid surface via fluid, respectively, and $\boldsymbol{g}$ is the body force per unit mass, such as gravitational acceleration, exerted by external fields.

The time integration of the ordinary differential equation system is performed via a second-order Runge–Kutta scheme:

$$\begin{aligned} \boldsymbol{k}_1 &= \boldsymbol{\Phi}(t^n, \boldsymbol{U}^n) \\ \boldsymbol{k}_2 &= \boldsymbol{\Phi}(t^n + \Delta t, \boldsymbol{U}^n + \Delta t \boldsymbol{k}_1) \\ \boldsymbol{U}^{n+1} &= \boldsymbol{U}^n + \Delta t(\boldsymbol{k}_1 + \boldsymbol{k}_2)/2 \end{aligned} \tag{2.17}$$

2.1.3 Interface description and evolution

Interface description, interface evolution, and interface boundary treatment are three fundamental problems for solving flow involving dynamic phase interfaces [85–87]. A variety of interface description methods, which largely determine the choice of interface evolution algorithm, have been successfully implemented in computing multiphase flows. These methods are frequently categorized into two groups: interface-capturing methods and interface-tracking methods.

In interface-capturing methods, such as volume of fluid methods [85] and level-set methods [88], the information of an interface, for instance, its spatial position, is embedded in and later can be reconstructed from a field function. The field function is advected either on an Eulerian grid [88, 89] or a Lagrangian grid [90, 91] to evolve moving interfaces. In interface-tracking methods, such as front-tracking methods [92], an interface is always explicitly described and evolved by a Lagrangian grid. The application of an Eulerian level-set method and a Lagrangian front-tracking method for the description of an interface in a computational domain is illustrated in Fig. 2.2.

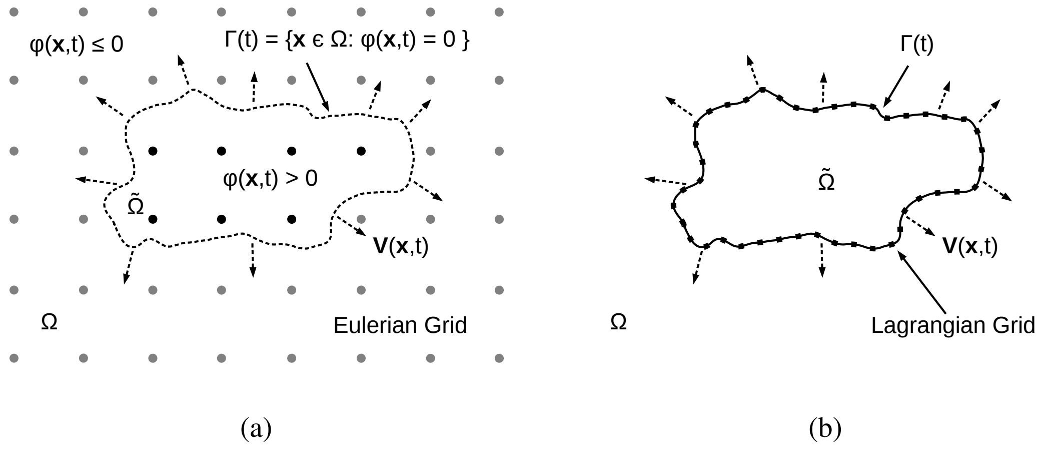


Figure 2.2 Schematic diagrams of interface description methods. (a) Eulerian level-set method. (b) Lagrangian front-tracking method. [Nomenclature: Ω, a computational domain; Γ, an interface in the computational domain; $\tilde{\Omega}$, the interface enclosed open region; φ, a scalar field function.]

In a volume of fluid method, the volume fraction of one material phase in each computational cell is adopted as the field function and is generally advected on an Eulerian grid to identify and evolve the phase interface. A geometrical reconstruction using the volume fraction is employed to rebuild the phase interface. While volume of fluid methods possess good properties such as volume conservation, exactly locating the interfacial points and computing interfacial quantities, such as surface normal or surface tension, are generally considered difficult.

In a level-set method, a node-based smooth scalar function whose zero-contour represents one phase interface is employed as the field function and is either advected on an Eulerian grid [88, 89] or a Lagrangian grid [90, 91]. For instance, in a computational domain Ω, the interface $\Gamma(t) = \partial\tilde{\Omega}$ of an open region $\tilde{\Omega} \subset \Omega$ can be embedded as the zero level-set of a signed distance function [93, 94], $\varphi(\boldsymbol{x}, t)$: $\mathbb{R}^3 \rightarrow \mathbb{R}$,

$$\Gamma(t) = \{\boldsymbol{x} \in \Omega : \ \varphi(\boldsymbol{x}, t) = 0\} \tag{2.18}$$

where the level-set function $\varphi(\boldsymbol{x}, t)$ has the following properties:

$$\begin{cases} \varphi(\boldsymbol{x}, t) > 0, & \boldsymbol{x} \in \tilde{\Omega} \\ \varphi(\boldsymbol{x}, t) \leq 0, & \boldsymbol{x} \notin \tilde{\Omega} \end{cases} \tag{2.19}$$

Suppose $\boldsymbol{x}(t)$ is a point trajectory on $\Gamma(t)$ evolving with $\boldsymbol{V} = \dot{\boldsymbol{x}}(t)$, then,

$$\varphi(\boldsymbol{x}(t), t) = 0 \tag{2.20}$$

Differentiate with respect to t [95], yields,

$$\frac{\partial \varphi}{\partial t} + \boldsymbol{V} \cdot \nabla \varphi = 0, \ t > 0; \ \varphi(\boldsymbol{x}, 0) = \varphi_0(\boldsymbol{x}) \tag{2.21}$$

A level-set method provides convenient normal calculation and easy treatment for geometry with topological changes. However, accurate computation of interfacial physics is not straightforward, and mass conservation is generally not preserved.

A front-tracking method explicitly tracks interfaces by individual Lagrangian grids such as triangulated meshes and provides minimum numerical diffusion [92, 96]. However, as interfaces are tracked by connected marker points, restructuring the deforming front for resolution maintenance or accounting for topological changes will inevitably modify the connectivity of maker points. The modification of topological connectivity involves complex operations and is the primary disadvantage of a front-tracking method [92].

In the currently developed flow solver that mainly concerns non-deformable solids, interface description and evolution are conducted using the front-tracking method [92]. Therefore, interfaces are explicitly tracked via individual Lagrangian grids such as triangulated facets, and the evolution of interfaces is governed by the laws of solid motion.

2.1.4 Multidomain node mapping and multibody collision

Complex and dynamic fluid-solid systems arise in many applications such as fluidized bed [38, 97], blood flow [51], particle-added explosives [8, 98], aerodynamic flow [99–101], and computer graphics [68, 102]. These systems usually involve interactions that comprise the coupled motions of solid and fluid flow. When a predictive modeling approach that resolves the fluid-solid interfaces is employed, in addition to addressing the coupled fluid-solid and solid-solid interactions, one inherent challenge can be the numerical discretization complicated by a set of irregular and moving geometries.

The development of Cartesian-grid methods [51–54, 56, 103] has provided a feasible way to enforce boundary conditions at phase interfaces that are not conforming to the computational grid, which can effectively simplify the grid generation for irregular geometries and grid regeneration for moving geometries [60, 104]. In addition, a solid object immersed in the computational grid can be described by the STereoLithography (STL) representation, which approximates the object as a closed triangulated surface and is a standard format for rapid prototyping and computer-aided design (CAD) systems. When equipped with a suitable Cartesian grid-based numerical framework, the STL represented solid can be directly inputted into the numerical solver without the need of CAD/CFD geometric translations to effectively enable automated mesh generation and numerical solution [105–107].

In applications involving multiple independent and irregular solid bodies, as each immersed body occupies a corresponding spatial region, the computational domain is therefore segmented into a set of subdomains. The subsequent implementation of numerical discretization and boundary conditions requires a node map that correctly classifies the type and region of computational nodes and identifies domain boundaries [98, 106, 108]. In addition, when the solids move in space, the requirement for node remapping also arises during the solution process. For problems without the need of differentiating a solid domain from the other solid domains, a binary node map that distinguishes fluid and solid domains through flagging the nodes inside any solid as 0 and nodes outside the solids as 1, or vice versa, has been popularly utilized to aid the enforcement of boundary conditions at solid boundaries [61, 106, 108, 109]. When further considering interactions such as collisions among solids or different material properties and/or boundary conditions for some solid domains, uniquely tracking and identifying each solid domain are then necessary, for which a possible solution can be employing an individual binary node map for each solid.

In computing a fluid-solid system with dense solids, collisions among solids can exert a strong influence on the stresses in the fluid-solid mixture [63]. Therefore, collision modeling can be an essential element. Models based on experimental correlations [63, 110, 111] often have limited description for flow with dense-to-dilute transitions. Short-range repulsive-force collisions [97, 112] usually consider the position of solids and omit size and shape effects. An interface-resolved collision model can provide a more comprehensive description of collision dynamics and thus a wider applicability. Nonetheless, additional challenges from collision detection and response are introduced and need to be addressed.

For each object in a solid system, collision detection is to determine the collision status of the current solid with regard to other solids, which can include colliding or non-colliding state, the list of colliding solids, and the line of impact for each colliding pair. Collision response is to solve the post-collision velocity under a detected collision status. In a system of moving solids, simultaneous multibody collisions are much less common than binary collisions. Nonetheless, when the system contains dense solids with spherical geometries or with sustained contacts among solids to transmit impulses, the presence of multibody collisions can increase greatly. Remaining as an unsolved problem [67], multibody collision is often approximated by sequential pairwise collisions [41, 68, 113]. Due to sequential collision, a temporal priority can be introduced into pairwise collisions in the process of choosing which pair collides first, and the collision process may no longer be deterministic.

In collision detection, when the surfaces of solids are only explicitly represented by triangulated meshes, checking every solid against every other solid can be inefficient if the number of solids is large and the geometry is complex. Considerable research has been devoted to optimizing the problem with strategies focusing on hierarchical object representation, orientation-based pruning criteria, spatial partitioning schemes, and distance computation algorithms [68, 114–116]. A multilevel algorithm that integrates temporal coherence exploitation, pairwise pruning, and exact pairwise collision-detection techniques can effectively reduce the collision-detection operations for a dense solid system with convex shapes but can still be expensive for non-convex objects [115, 117]. Employing implicit surfaces defined by field functions such as signed distance functions has shown success in collision-related modeling [61, 68, 94]. While triangulated surface representation is used for normal calculation and the determination of the line of impact, a signed distance function per solid object can map the point-inclusion state for each computational node with regard to the solid object. As a result, using the layer of nodes that is nearest to the zero isocontour of a signed distance function as sample points, the collision status of a solid object can be determined by testing the values of the sample points with regard to the signed distance functions of other solids [68]. Since the surface resolution of implicitly defined surfaces is proportional to the grid resolution for field functions, high accuracy can be obtained when a well-resolved grid is employed.

Considering that using a binary node-mapping function for a multidomain problem usually requires one field function per object and can consume memory that is proportional to the number of involved solids, the development of a field function that can compress the multidomain node mapping information into a single field to reduce memory requirement can be useful. To facilitate the solution of complex and dynamic fluid-solid systems on Cartesian grids, an integer-type field function that solves multidomain node mapping is proposed in Chapter 3. For a Cartesian-grid-discretized computational domain segmented by a set of solid bodies, compared with a binary node-mapping function that produces node mapping information using one field function per subdomain, the proposed field function can uniquely track all the subdomains with multiple layers of interfacial nodes using only one integer field in total. As a result, the present field function enables single-integer-based multidomain node mapping, efficient node remapping, linear-time collision detection, and expedient surface force integration.

2.1.5 Interface boundary treatment

Fluid-solid interaction involving complex geometric settings is an active field of research and development, owing to the ubiquitous presence of fluid-solid interaction phenomena and the great difficulty in tackling those problems via mathematical modeling. To solve flow involving irregular and moving geometries, one of the main challenges is related to enforcing boundary conditions.

In recent years, Cartesian grid-based boundary treatment methods such as embedded boundary methods [56, 57], ghost fluid methods [54, 55], immersed interface methods [52, 53], and immersed boundary methods [51] have gained increasing popularity in interface boundary treatment [61, 62, 118–122]. As an attractive alternative to the arbitrary Lagrangian–Eulerian method [123], in which a body-conformal grid following the movement of phase interfaces is employed, Cartesian grid-based methods are able to solve problems with complex interfaces on a fixed Cartesian grid, as illustrated in Fig. 2.3. Benefiting from the use of a Cartesian gird, grid generation is greatly simplified, and the per-grid-point computation exempts from operations associated with grid topology or transformations. Since the main data structures for numerical computation are simple arrays, the memory requirement is largely reduced, and high-order spatial discretization schemes are easy to implement [60].

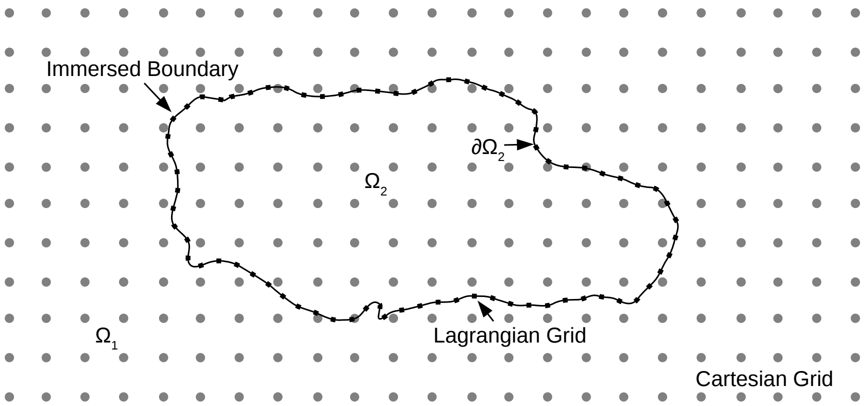


Figure 2.3 A schematic diagram illustrating an immersed boundary in a Cartesian grid. [Nomenclature: Ω_1, domain of phase 1; Ω_2, domain of phase 2; $\partial\Omega_2$, the boundary of phase 2.]

Extensions of the immersed boundary method, which was originally introduced by Peskin [51], have been continuously developed to improve the numerical properties of the method, particularly in aspects related to interface resolution, stability constraints, mass conservation, computational efficiency and robustness [60, 104, 124].

To simplify numerical discretization and relax stability constraints, Mohd-Yusof [125] and Fadlun et al. [58] proposed the direct forcing immersed boundary method, in which boundary forces are implicitly imposed via flow reconstruction. Balaras [126] later improved the reconstruction procedure of the direct forcing approach and applied to large-eddy simulations. Integrating ideas from the ghost fluid method [54, 55] and the direct forcing method [58], Tseng and Ferziger [59] systematically developed a polynomial reconstruction-based ghost-cell immersed boundary method to further increase implementation flexibility while maintaining sharp interfaces. Kapahi et al. [61] proposed a least squares interpolation approach and applied to solving impact problems. Employing the adaptive mesh refinement technique for resolving boundary layers, Brehm et al. [84] developed a locally stabilized immersed boundary method and applied to simulating the laminar to turbulent transition process on no-slip walls.

For a direct forcing immersed boundary method, its robustness highly depends on the numerical stability and stencil adaption capability of the interpolation method used [59, 61, 127]. Polynomial reconstruction-based methods involve constructing linear systems on neighboring stencils of the interpolated node. When the stencil nodes are not well distributed in space, the resulting linear systems may suffer from numerical singularities [59, 127]. Additionally, a fixed minimum number of stencil nodes is needed to avoid under-determined linear systems. Therefore, special treatments are required when strongly concave or convex interfaces exist [61, 127]. To enhance numerical stability and stencil adaption capability, the idea of using inverse distance weighting interpolation was firstly introduced by Tseng and Ferziger [59], and a hybrid Taylor series expansion / inverse distance weighting approach was later developed by Gao et al. [127] for flow with no-slip walls.

In addition to numerical stability and stencil adaption capability, being able to enforce different types of boundary conditions in a straightforward and consistent manner is another vital factor in obtaining an efficient, accurate, and robust immersed boundary method, since a variety of boundary conditions are required to be repeatedly enforced on numerical boundaries. For instance, in solving Navier–Stokes equations, constant temperature at a wall and velocity at a no-slip wall have Dirichlet boundary conditions, pressure at a wall and temperature at an adiabatic wall have Neumann boundary conditions, and velocity at a slip wall has a type of Cauchy boundary conditions.

Excluding the Dirichlet boundary conditions in which boundary values are determined and known, the enforcement of other types of boundary conditions, particularly the Cauchy type of boundary conditions, for immersed boundaries demands considerable efforts [121, 122, 128]. Kempe et al. [121] devised a numerical implementation of the slip-wall boundary conditions in the context of immersed boundary methods. However, the realization is not straightforward due to its complexity [121]. In addition, most direct forcing immersed boundary methods require constructing and solving a designated linear system for each type of boundary conditions. Therefore, to enforce a variety of boundary conditions in a straightforward and consistent manner is beneficial but can be challenging.

Employing the direct forcing approach, Chapter 4 develops an immersed boundary method via devising a second-order three-step flow reconstruction scheme. The developed method can enforce the Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner, and is able to provide efficient, accurate, and robust boundary treatment for solving flow with arbitrarily irregular and moving geometries on Cartesian grids.

2.2 Clustering and jetting instability investigation

After being validated for solving gas-solid flow with irregular, moving, and colliding non-deformable granular bodies, the developed interface-resolved direct simulation framework is then applied to the investigation of particle clustering and jetting instabilities in explosively dispersed granular materials. In order to closely resemble realistic packing of payloads, the numerical investigation considers granular payloads with stochastic morphologies. A set of cases that cover a variety of randomly generated stochastic payloads, burster states, and coefficients of restitution for pairwise collisions are designed, solved, and analyzed in Chapter 5 to help understand the particle clustering and jetting dynamics in explosively dispersed granular materials.

Chapter 3

A Field Function for Solving Complex and Dynamic Fluid-Solid Systems

3.1 Introduction

Modeling complex and dynamic fluid-solid systems requires the simulation of coupled multiphase interactions. In this chapter, a simple field function is presented for facilitating the solution of fluid-solid systems on Cartesian grids with interface-resolved fluid-solid and solid-solid interactions. For a Cartesian-grid-discretized computational domain segmented by a set of solid bodies, the proposed field function can uniquely track all the subdomains with multiple layers of interfacial nodes using only one integer field in total. As a result, the present field function enables single-integer-based multidomain node mapping, efficient node remapping, linear-time collision detection, and expedient surface force integration. Implementation algorithms for the field function and its associated functionalities are presented. Equipped with a deterministic multibody collision model, numerical experiments involving complex and dynamic fluid-solid systems solved via immersed boundary treatments are conducted to validate and demonstrate the applicability of the proposed field function.

3.2 Method development

3.2.1 Field function description

As illustrated in Fig. 3.1, for a set of solids represented by triangulated polyhedrons $\{\Omega_p : p = 1, \ldots, P\}$ and distributed in a spatial domain Ω, an additional subdomain Ω_0 can be introduced as

$$\Omega_0 = \{\boldsymbol{x} \in \Omega : \boldsymbol{x} \notin \cup_{p=1}^{P} \Omega_p\} \tag{3.1}$$

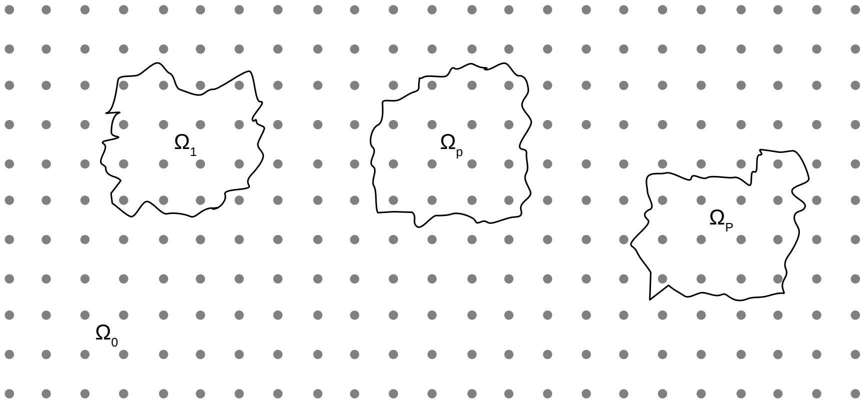


Figure 3.1 A schematic diagram illustrating a set of polyhedron domains distributed in a spatial domain discretized by a Cartesian grid.

When a Cartesian grid $I \times J \times K$ is used to discretize the spatial domain Ω, it is necessary to build a node map for defining the spatial relation between computational nodes and the individual polyhedral subdomains. In this study, a two-component integer-type field function is developed to classify each node with regard to Ω_m and also to identify R layers of interfacial nodes for each Ω_m, $m = 0, \dots, P$:

$$\mathbf{\Phi} = \{(\phi, \varphi) : \phi \in \{0, \dots, P\}, \varphi \in \{0, \dots, R\}\} \tag{3.2}$$

in which ϕ is the domain identifier and is determined by the point inclusion state:

$$\phi_{i,j,k} = m, \text{ if } \boldsymbol{x}_{i,j,k} \in \Omega_m \tag{3.3}$$

and φ is the interfacial layer identifier and is determined by the existence of a heterogeneous node (i', j', k'), that is, a node with a different ϕ value, in the range $L(r)$ surrounding the processing node (i, j, k):

$$\varphi_{i,j,k} = \begin{cases} \min\{r\}, & \text{if } \exists\, \phi_{i',j',k'} \neq \phi_{i,j,k} \text{ in range L(r)} \\ 0, & \text{if } r > R \end{cases} \tag{3.4}$$

here $\boldsymbol{x}_{i,j,k}$ is the position vector of the node (i, j, k), and R represents the maximum layers of interfacial nodes that needs to be identified. The range $L(r)$ is the size of numerical stencils used for the spatial discretization of the governing equations at the node (i, j, k). Therefore, the range $L(r)$ needs to be adapted according to the involved differential operators and spatial schemes. For example, the discretization of non-mixed and mixed derivatives requires $L(r)$ to be the line-type stencils in Eq. (3.5) and the plane-type stencils in Eq. (3.6), respectively. Note that the determination of the last three criteria in Eq. (3.6) is to maximize the overlap between the first three criteria and the last three ones.

$$\begin{cases} |i'-i| = r,\ |j'-j| = 0,\ |k'-k| = 0 & \text{or} \\ |i'-i| = 0,\ |j'-j| = r,\ |k'-k| = 0 & \text{or} \\ |i'-i| = 0,\ |j'-j| = 0,\ |k'-k| = r & \end{cases} \tag{3.5}$$

$$\begin{cases} |i'-i| = r,\ |j'-j| = 0,\ |k'-k| = 0 & \text{or} \\ |i'-i| = 0,\ |j'-j| = r,\ |k'-k| = 0 & \text{or} \\ |i'-i| = 0,\ |j'-j| = 0,\ |k'-k| = r & \text{or} \\ |i'-i| = r-1,\ |j'-j| = r-1,\ |k'-k| = 0 & \text{or} \\ |i'-i| = r-1,\ |j'-j| = 0,\ |k'-k| = r-1 & \text{or} \\ |i'-i| = 0,\ |j'-j| = r-1,\ |k'-k| = r-1 & \end{cases} \tag{3.6}$$

3.2.2 Single-integer-based multidomain node mapping

The field function $\mathbf{\Phi}(\phi, \varphi)$ can produce a complete node map for the numerical discretization of complex fluid-solid systems. As shown in Fig. 3.2, for each node (i, j, k), $\phi_{i,j,k}$ and $\varphi_{i,j,k}$ can provide the subdomain and interfacial state, respectively.

In general, when Ω_m is a solution domain, two approaches are available to compute the solutions in Ω_m. 1) a non-ghost-cell approach, in which $\mathbf{\Phi}_{i,j,k}(\phi = m, \varphi = 0)$ represents a normal solution node, and $\mathbf{\Phi}_{i,j,k}(\phi = m, \varphi > 0)$ represents a node locating at the numerical boundary of Ω_m. 2) a ghost-cell approach, in which $\mathbf{\Phi}_{i,j,k}(\phi = m, \varphi \geq 0)$ represents a normal solution node, and $\mathbf{\Phi}_{i,j,k}(\phi \neq m, \varphi > 0)$ with a node $\mathbf{\Phi}_{i',j',k'}(\phi = m, \varphi \geq 0)$ existing in the range $L(r)$ of the node (i, j, k) represents a node locating at the numerical boundary of Ω_m.

When the space occupied by Ω_m changes in time, interfacial nodes of another domain Ω_n can change their corresponding domain and become interfacial nodes of Ω_m. In a non-ghost-cell approach, the newly joined interfacial nodes of Ω_m always become boundary nodes, whose values will be constructed by boundary treatment. However, in a ghost-cell approach, the newly joined interfacial nodes of Ω_m directly become normal solution nodes of the solution domain Ω_m. Therefore, a special treatment, such as a reconstruction for the values of the newly joined nodes from the values of the original normal solution nodes of Ω_m is required to deal with these newly joined nodes.

For a computational domain segmented by a set of solid bodies, the proposed field function $\mathbf{\Phi}(\phi, \varphi)$ uniquely identifies all the subdomains with multiple layers of interfacial nodes. Therefore, it is straightforward to apply designated governing equations, constitutive models, numerical schemes, and boundary conditions for each subdomain. In the practical implementation, the two-component $\mathbf{\Phi}(\phi, \varphi)$ can be mapped onto a single scalar $\Phi = \phi + \varphi * (P+1)$, from which individual components can be extracted through $\varphi = \Phi \bmod (P+1)$ and $\phi = \Phi - \varphi * (P+1)$, respectively. As an integer-type field function that can be stored on a single grid, the proposed field function herein enables single-integer-based multidomain node mapping and consumes memory that is independent of the number of represented objects. For instance, assume that the number of nodes for defining a field function is M and the number of represented objects is P, the estimated memory consumption for using the proposed field function is then about $1/P$ of that for using a binary node-mapping function. This low-memory requirement can be particularly useful for applications involving a large set of solid bodies, such as in solving fluid-solid systems with thousands of colliding particles.

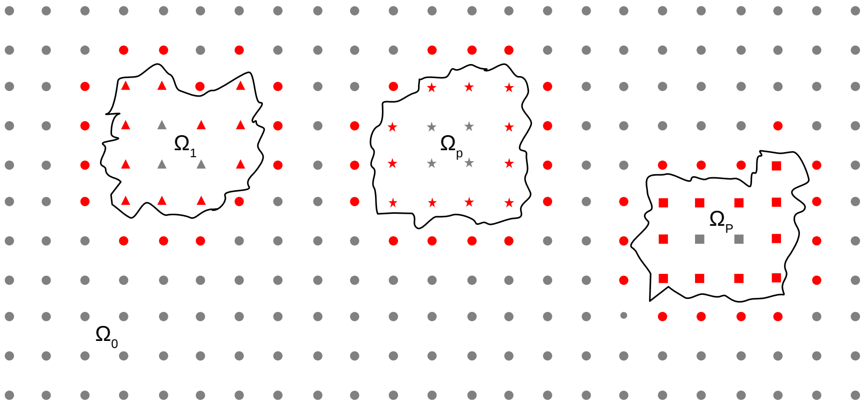


Figure 3.2 A schematic diagram illustrating the field function $\mathbf{\Phi}(\phi, \varphi)$ for multidomain node mapping. The value of ϕ is represented by *shapes*: $\circ = 0$; $\triangle = 1$; $\star = p$; $\square = P$. The value of φ is represented by *colors*: ● = 0; ● = 1. $R = 1$ is adopted here for clarification.

3.2.3 Efficient node remapping

During the solution process, when the positions of polyhedrons keep changing, the requirement for node remapping arises. As illustrated in Fig. 3.3, the field function $\mathbf{\Phi}(\phi, \varphi)$ enables efficient node remapping: From time t^n to t^{n+1}, suppose the domain occupied by Ω_p changing from Ω_p^n to Ω_p^{n+1}. When the computational time step size is restricted by a stability condition such as the Courant–Friedrichs–Lewy (CFL) condition [129], the boundary of Ω_p will correspondingly have restricted travelling distance. If the stability condition restricts the value of travelling distance to no more than one grid size, and the maximum number of identified interfacial layers has $R \geq 2$, it is safe to assume that a node (i, j, k) with $\mathbf{\Phi}_{i,j,k}(\phi = p, \varphi = 0)$ in Ω_p^n will remain in Ω_p^{n+1}, and then only the interfacial nodes need to be reset. As a result of this temporal coherence, nodes with $\mathbf{\Phi}_{i,j,k}(\phi = p, \varphi = 0)$ can be exempted from future point-inclusion tests. As these non-interfacial nodes constitute the major fraction of the computational nodes in a practical grid, this exemption from the expensive point-inclusion test can significantly reduce the costs of node remapping to obtain an acceleration rate inversely proportional to the surface-to-volume ratio of polyhedrons. An efficient node remapping algorithm adopting the described temporal coherence is proposed in the present study as the following:

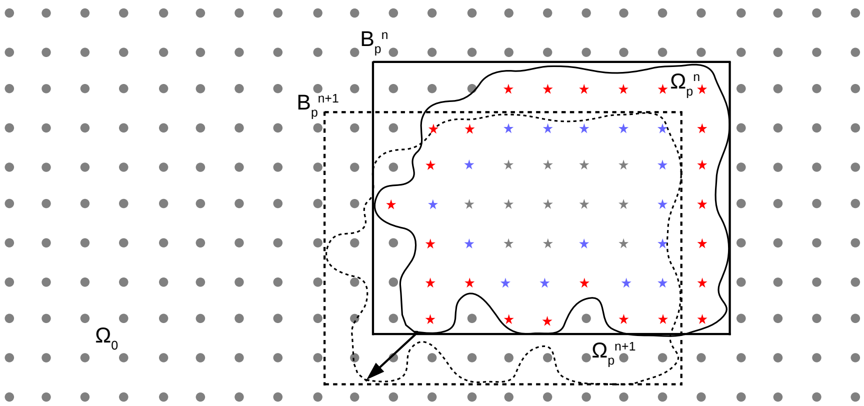


Figure 3.3 A schematic diagram illustrating the field function $\mathbf{\Phi}(\phi, \varphi)$ for efficient node remapping. The value of ϕ is represented by *shapes*: $\circ = 0$; $\star = p$. The value of φ is represented by *colors*: ● = 0; ● = 1; ● = 2. $R = 2$ is adopted, and the interfacial nodes of Ω_0 are not shown for clarification.

1. *Initialization*. Sweep each node (i, j, k) in $I \times J \times K$: if $\varphi_{i,j,k} > 0$, set $\mathbf{\Phi}_{i,j,k}(\phi, \varphi) = (0, 0)$.

2. *Compute the domain identifier ϕ*. Sweep each polyhedron Ω_p in $\{\Omega_p : p = 1, \ldots, P\}$:

   (a) Find the bounding box $B_p = [I_{\min}, I_{\max}] \times [J_{\min}, J_{\max}] \times [K_{\min}, K_{\max}]$.

   (b) Sweep each node (i, j, k) in B_p: if $\phi_{i,j,k} = 0$, do point-in-polyhedron test for the node (i, j, k) over Ω_p to determine the value of $\phi_{i,j,k}$ using Eq. (3.3).

3. *Compute the interfacial layer identifier φ*. Sweep each node (i, j, k) in $I \times J \times K$: determine the value of $\varphi_{i,j,k}$ using Eq. (3.4).

In the described algorithm, the point-in-polyhedron test for the node (i, j, k) over Ω_p is a point-inclusion test with regard to a single polyhedron. A variety of established methods, such as the ray-crossing methods [130], angular methods [131], winding number methods [132], and signed distance methods [94], are available. The angle weighted pseudonormal signed distance computation method [93] is employed herein for a balance of efficiency and robustness. Meanwhile, it finds the closest point, the distance to surface, and the surface normal for a computational node with regard to the solid geometry, which is essential for implementing a Cartesian grid-based boundary treatment method.

The proposed algorithm herein effectively solves the node classification and boundary identification issues simultaneously for a Cartesian grid segmented by a set of polyhedrons in space, which involves a set of points together with a set of polyhedrons and represents a generalized point-in-polyhedron problem. In addition, since $\mathbf{\Phi}(\phi, \varphi) = (0, 0)$ is true initially, the presented algorithm successfully unifies the procedures of the initial multidomain node mapping and the subsequent node remapping for moving objects. This unification can simplify the code structure and reduce the complexity of implementation.

3.2.4 Linear-time collision detection

As captured in Fig. 3.4, employing the field function $\mathbf{\Phi}(\phi, \varphi)$, polyhedrons $\{\Omega_n\}$ colliding with Ω_p can be detected efficiently via sweeping through nodes (i, j, k) with $\mathbf{\Phi}_{i,j,k}(\phi = p, \varphi = 1)$ in the bounding box of Ω_p to detect nodes (i', j', k') with $\mathbf{\Phi}_{i',j',k'}(\phi \neq p, \varphi = 1)$ in the range $|i - i'| \leq 1$, $|j - j'| \leq 1$, and $|k - k'| \leq 1$.

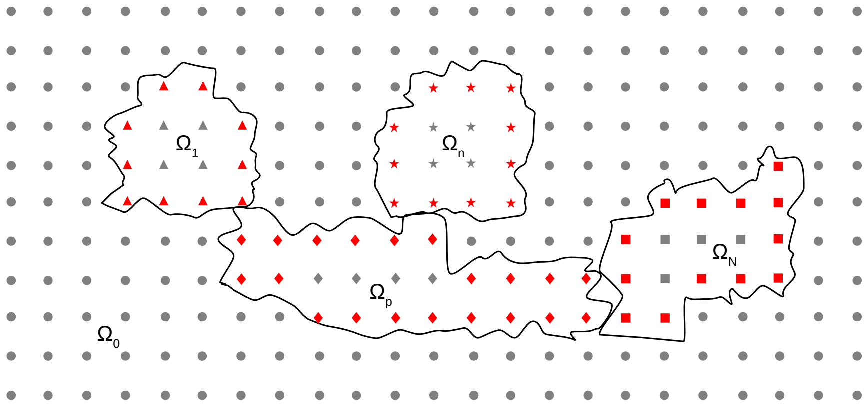


Figure 3.4 A schematic diagram illustrating the field function $\mathbf{\Phi}(\phi, \varphi)$ for collision detection. The value of ϕ is represented by *shapes*: ○ = 0; △ = 1; ★ = n; □ = N; ◇ = p. The value of φ is represented by *colors*: ● = 0; ● = 1. $R = 1$ is adopted, and the interfacial nodes of Ω_0 are not shown here for clarification.

Comparing to explicit surfaces such as triangulated meshes, implicit surfaces, such as the binary node-mapping functions and the proposed integer field function herein, for collision detection have algorithm efficiency and complexity that are much less sensitive to geometric complexity. This is because collision detection for implicit surfaces uses point-wise data comparison rather than facet-wise distance calculation as the basic operator and is independent of the number of facets and the level of concavity.

On a given Cartesian grid, let $\mathcal{S}_{\mathrm{cd}}$ represent the collision detection operator for an interfacial node (i, j, k) in a polyhedron Ω_p. $\mathcal{S}_{\mathrm{cd}}$ based on a binary node mapping involves fetching and comparing of about P floating-point or integer data scattered in P field functions, while $\mathcal{S}_{\mathrm{cd}}$ based on $\mathbf{\Phi}$ requires fetching and comparing of about 2 integer data per spatial dimension locally located on one Cartesian grid. Therefore, the algorithm complexity of collision detection for P polyhedrons can be $\mathrm{O}(P^2)$ for using a binary node mapping but only $\mathrm{O}(P)$ for using the proposed field function $\mathbf{\Phi}$ herein.

During Ω_p colliding with Ω_n, it is possible that several geometric elements, such as vertices, edges, and faces, of Ω_p and Ω_n can be in contact simultaneously. In addition, one element of Ω_p could come into contact with a few elements of Ω_n at the same time. This multicontact issue imposes challenges in finding the line of impact for collision modeling.

Approximating the line of impact via the proposed field function $\mathbf{\Phi}(\phi, \varphi)$ can then provide an alternative perspective to the approximation via explicit surfaces defined by triangulated meshes. Suppose C nodes (i_c, j_c, k_c), $c = 1, \ldots, C$, in Ω_p satisfying $\mathbf{\Phi}_{i_c,j_c,k_c}(\phi = p, \varphi = 1)$, and each (i_c, j_c, k_c) comes with D neighboring nodes (i'_d, j'_d, k'_d), $d = 1, \ldots, D$, such that $\mathbf{\Phi}_{i'_d,j'_d,k'_d}(\phi = n, \varphi = 1)$, the suggested approximation of the line of impact between Ω_p and Ω_n is defined as

$$\mathbf{e}_{pn} = \frac{\sum_{c=1}^{C} \sum_{d=1}^{D} [(i'_d - i_c)\mathbf{e}_1 + (j'_d - j_c)\mathbf{e}_2 + (k'_d - k_c)\mathbf{e}_3]}{|\sum_{c=1}^{C} \sum_{d=1}^{D} [(i'_d - i_c)\mathbf{e}_1 + (j'_d - j_c)\mathbf{e}_2 + (k'_d - k_c)\mathbf{e}_3]|} \tag{3.7}$$

where $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ are the unit vectors in the x, y, and z directions, respectively. This approximation of the line of impact via the field function can simplify the multicontact problem with adequate accuracy, as to be demonstrated in the numerical experiments later.

3.2.5 Deterministic collision response

Assume N polyhedrons $\{\Omega_n\}$, $n = 1, \ldots, N$, colliding with Ω_p simultaneously, the pre- and post-collision velocity of Ω_p as $\boldsymbol{V}_p$ and $\boldsymbol{V}'_p$, respectively, and the pre-collision velocity of Ω_n as $\boldsymbol{V}_n$. A collision model that approximates multibody collision without introducing temporal priority and also extends the applicability of the model in [73] to a wider range of multibody collision scenarios is employed herein:

1. Conduct the n-th pairwise collision (Ω_p, Ω_n) with the pre-collision velocity $(\boldsymbol{V}_p, \boldsymbol{V}_n)$ to solve the corresponding post-collision velocity $\boldsymbol{V}'_{p,n}$ and the velocity change $\Delta\boldsymbol{V}'_{p,n}$ of Ω_p:

$$\Delta\boldsymbol{V}'_{p,n} = \boldsymbol{V}'_{p,n} - \boldsymbol{V}_p = -\frac{m_n}{m_p + m_n}(1 + C_{\mathrm{R}})(\boldsymbol{V}_{pn} \cdot \mathbf{e}_{pn})\mathbf{e}_{pn} - C_{\mathrm{f}}[\boldsymbol{V}_{pn} - (\boldsymbol{V}_{pn} \cdot \mathbf{e}_{pn})\mathbf{e}_{pn}] \tag{3.8}$$

in which $\boldsymbol{V}_{pn} = (\boldsymbol{V}_p - \boldsymbol{V}_n)$, C_{R} is the coefficient of restitution in the normal direction ($C_{\mathrm{R}} = 0$, $0 < C_{\mathrm{R}} < 1$, and $C_{\mathrm{R}} = 1$ corresponds to perfectly inelastic collision, partially inelastic collision, and elastic collision, respectively), C_f represents the effect of sliding friction, m_p and m_n are the mass of Ω_p and Ω_n, respectively.

2. Next, approximate the post-collision velocity of Ω_p under the multibody collision via a vector summation of the pre-collision velocity and velocity changes:

$$\boldsymbol{V}'_p = \boldsymbol{V}_p + \sum_{n=1}^{N} \Delta \boldsymbol{V}'_{p,n} \tag{3.9}$$

3. Apply the above procedures to each Ω_p, $p = 1, \ldots, P$, in the solid system to obtain a post-collision velocity $\boldsymbol{V}'_p$ after multibody collision.
4. Update the velocity state of each Ω_p through replacing the pre-collision velocity $\boldsymbol{V}_p$ by the post-collision velocity $\boldsymbol{V}'_p$.

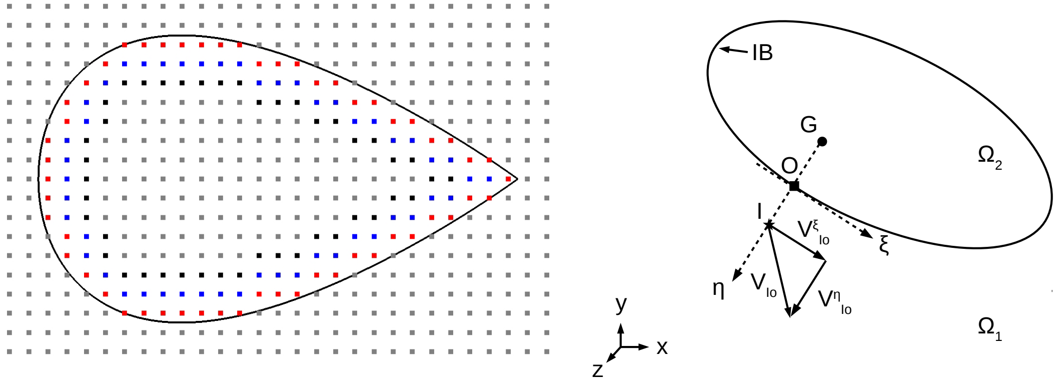
Avoiding any temporal priority in each polyhedron and each collision, the proposed multibody collision algorithm based on exact pairwise collision is deterministic and can be parallelized.

3.2.6 Expedient surface force integration

In the interface-resolved modeling of fluid-solid interactions, surface force integration is an essential part. The proposed field function $\boldsymbol{\Phi}(\phi, \varphi)$ can aid the surface force integration for irregular solids immersed in a Cartesian grid.

Employing the proposed field function and its implementation algorithm, the computed three inner interfacial layers of an irregular solid immersed in a Cartesian grid are shown in Fig. 3.5a. It can be observed that the $\varphi = 1$ layer conforms with the solid boundary very closely, in which the distance discrepancy reduces with mesh refinement and is in the interval $[0, \Delta s)$, where $\Delta s = \max(\Delta_x, \Delta_y, \Delta_z)$. Therefore, for a non-ghost-cell approach, the integration of surface forces can be conducted on the $\varphi = 2$ layer in the fluid domain, of which the values of flow variables are known and the distance discrepancy with solid boundary is in the interval $[\Delta s, 2\Delta s)$. Meanwhile, for a ghost-cell approach, the integration of surface forces can be properly conducted on the corresponding point layer formed by the image points of the $\varphi = 2$ nodes of the solid.

Here, the surface force integration in the ghost-cell approach is discussed and applied in this study, as the method can be easily transformed for a non-ghost-cell approach, whose surface force integration is more straightforward. By exploring the relation between the ghost node G, the boundary point O, and the image point I, the surface force integration in a ghost-cell approach can be simplified as discussed below.



(a)

(b)

Figure 3.5 Diagrams of applying the field function $\mathbf{\Phi}(\phi, \varphi)$ for surface force integration over immersed boundaries. (a) Computational results showing the interfacial layer identifier φ for an irregular solid immersed in a Cartesian grid. The value of φ is represented by *colors*: ● = 0; ● = 1; ● = 2; ● = 3. (b) A schematic diagram illustrating wall shear stress calculation. [Nomenclature: G, ghost node; O, boundary point; I, image point; Ω_1, fluid domain; Ω_2, solid domain; IB, immersed boundary.]

Wall pressure Pressure at the wall can generally be approximated via the zero normal gradient assumption

$$\left.\frac{\partial p}{\partial n}\right|_O = 0 \tag{3.10}$$

Therefore,

$$p_O = p_I = p_G \tag{3.11}$$

Thus, the pressure component in the surface stress vector is

$$-p_G\mathbf{n} \tag{3.12}$$

Wall shear stress As illustrated in Fig. 3.5b, suppose a natural coordinate system, $\eta - \xi$, is placed at the boundary point O and is located in the plane defined by the normal vector $(\mathbf{n}\times\boldsymbol{V}_{IO})$, where $\boldsymbol{V}_{IO} = \boldsymbol{V}_I - \boldsymbol{V}_O$ is the relative velocity of the image point I to the boundary point O. Then, the wall shear stress is defined as

$$\boldsymbol{\tau}_\mathrm{w} \equiv \boldsymbol{\tau}(\eta = 0) = \mu \left.\frac{\partial \boldsymbol{V}^{\xi}_{IO}}{\partial \eta}\right|_{\eta=0} \tag{3.13}$$

A second-order central differencing approximation gives

$$\boldsymbol{\tau}_\mathrm{w} = \mu\frac{\boldsymbol{V}^{\xi}_{IO} - \boldsymbol{V}^{\xi}_{GO}}{2||\boldsymbol{x}_I - \boldsymbol{x}_O||} \tag{3.14}$$

By using $\boldsymbol{V}_{IO} = -\boldsymbol{V}_{GO}$, $||\boldsymbol{x}_I - \boldsymbol{x}_O|| = ||\boldsymbol{x}_G - \boldsymbol{x}_O||$, and $\boldsymbol{V}^{\xi}_{GO} = \boldsymbol{V}_{GO} - (\boldsymbol{V}_{GO} \cdot \mathbf{n})\mathbf{n}$, it gives

$$\boldsymbol{\tau}_{\mathrm{w}} = -\mu \frac{\boldsymbol{V}_{GO} - (\boldsymbol{V}_{GO} \cdot \mathbf{n})\mathbf{n}}{||\boldsymbol{x}_G - \boldsymbol{x}_O||} \tag{3.15}$$

Hence, the surface stress vector at the boundary point O is obtained as

$$\boldsymbol{T}^{(\mathbf{n})}_O = -p_G\mathbf{n} + \boldsymbol{\tau}_{\mathrm{w}} = -p_G\mathbf{n} - \mu \frac{\boldsymbol{V}_{GO} - (\boldsymbol{V}_{GO} \cdot \mathbf{n})\mathbf{n}}{||\boldsymbol{x}_G - \boldsymbol{x}_O||} \tag{3.16}$$

By employing the derived relations, the surface force integration for irregular solids can be directly computed and expressed in the global Eulerian coordinate system without involving coordinate transformation. In addition, both the wall pressure and the wall shear stress are explicitly computed at the ghost node G, which is a computational node with known flow values. As a result, the proposed surface force integration method is straightforward to implement.

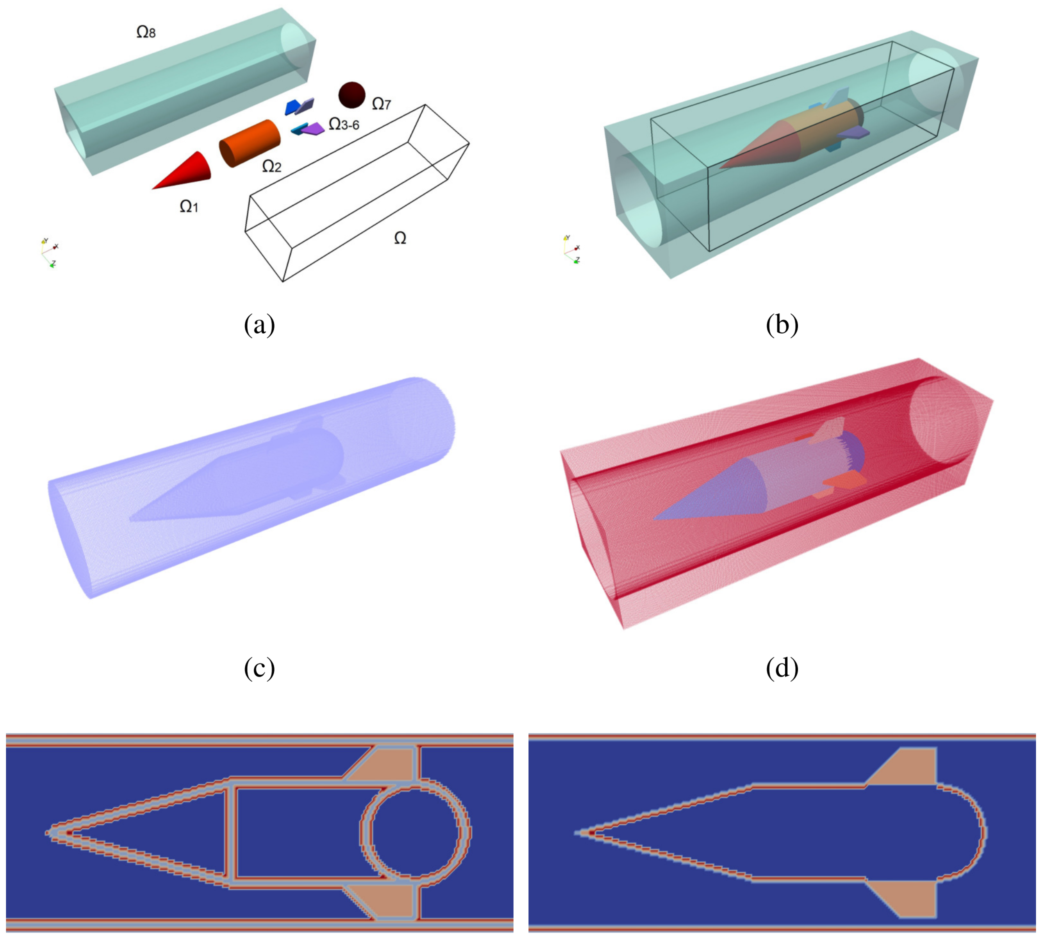
3.3 Numerical experiments

3.3.1 A wind tunnel test constructed via field function

A numerical wind tunnel test is constructed to illustrate the proposed field function for complex multidomain node mapping. As shown in Fig. 3.6a, 8 triangulated polyhedrons $\{\Omega_p : p = 1, \ldots, 8\}$ and a spatial domain Ω are used. Ω_1 is a cone with an opening angle $\theta_1 = 30^\circ$ and height $h_1 = 0.5\,\mathrm{m}$. Ω_2 is a cylinder with length $l_2 = h_1$ and radius $r_2 = h_1 \tan(\theta_1/2)$. Ω_{3-6} are four irregular polyhedrons with an identical geometry, of which the thickness is $w_3 = 0.02\,\mathrm{m}$ and the lengths of the other five edges counting from the shortest one clockwise are $0.01\,\mathrm{m}$, $0.1\sqrt{2}\,\mathrm{m}$, $0.1\,\mathrm{m}$, $0.11\,\mathrm{m}$, and $0.2\,\mathrm{m}$, respectively. Ω_7 is a sphere of radius $r_7 = r_2$. Ω_8 is a polyhedron whose outer profile is a box of size $l_8 \times h_8 \times w_8 = 2.5\,\mathrm{m} \times 0.6\,\mathrm{m} \times 0.6\,\mathrm{m}$ and inner profile is a cylindrical cavity with length l_8 and radius $r_8 = 0.25\,\mathrm{m}$. The spatial domain Ω is a box of size $l \times h \times w = 2.0\,\mathrm{m} \times 0.55\,\mathrm{m} \times 0.55\,\mathrm{m}$.

In order to construct the wind tunnel test shown in Fig. 3.6b, the central axes of Ω_1, Ω_2, Ω_8, and Ω are all positioned along the x-axis. The apex of Ω_1 is at $O = (0, 0, 0)$; the centers of Ω_2, Ω_8, and Ω are located at $(h_1 + l_2/2, 0, 0)$. Ω_{3-6} are shifted to align with the rear end of Ω_2 with the shortest edge being completely immersed into Ω_2, and the center of Ω_7 is at $(h_1 + l_2, 0, 0)$ so that half of Ω_7 is immersed into Ω_2. In addition, the spatial domain Ω is discretized by Cartesian grids.

The proposed field function $\boldsymbol{\Phi}(\phi, \varphi)$ is then applied to building a node map for the numerical discretization of the assembled domain in Fig. 3.6b. As shown in Fig. 3.6c



(a)

(b)

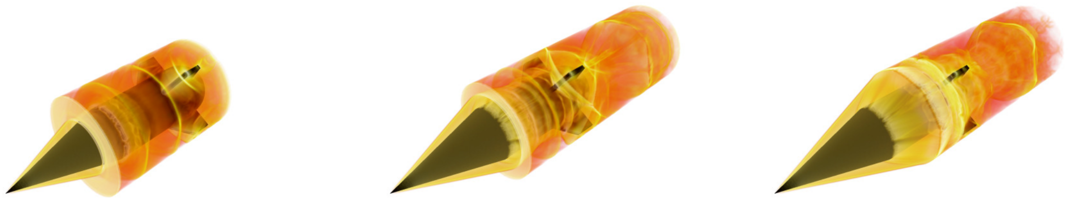
(c)

(d)

(e)

(f)

Figure 3.6 The construction of a numerical wind tunnel test via the field function $\mathbf{\Phi}(\phi, \varphi)$. (a) The employed polyhedrons $\{\Omega_p : p = 1, \ldots, 8\}$ and spatial domain Ω. (b) The constructed wind tunnel test. (c) The computational nodes with $\mathbf{\Phi}_{i,j,k}(\phi = 0, \varphi)$. (d) The computational nodes with $\mathbf{\Phi}_{i,j,k}(8 \geq \phi \geq 1, \varphi)$. (e) A slice plane capturing the layers of interfacial nodes. (f) A slice plane capturing the layers of ghost nodes.



(a) $t = 0.02$ s

(b) $t = 0.03$ s

(c) $t = 0.04$ s

Figure 3.7 The time evolution of the wind tunnel test illustrated by the volume rendering of the numerical Schlieren field solved on the grid $I_3 \times J_3 \times K_3$.

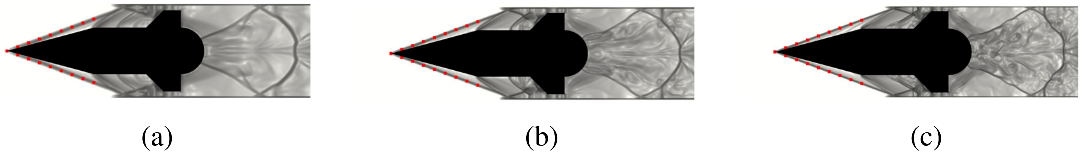


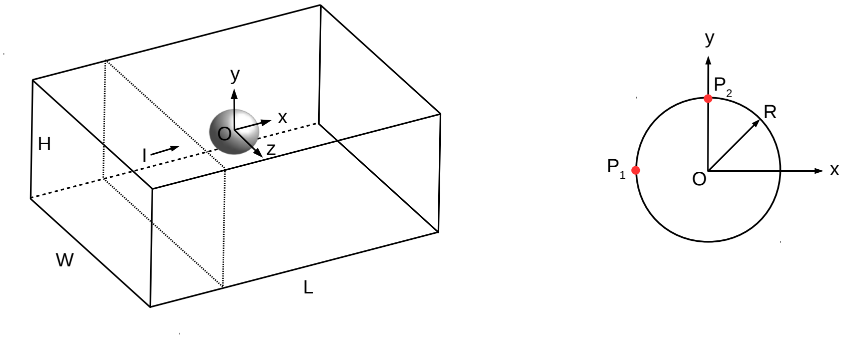
Figure 3.8 The predicted conical shock angles on three different grids at $t = 0.04\,\mathrm{s}$. (a) $I_1 \times J_1 \times K_1$. (b) $I_2 \times J_2 \times K_2$. (c) $I_3 \times J_3 \times K_3$. Lines denoted by square marks represent the analytical solutions.

for nodes in the solution domain Ω_0 and in Fig. 3.6d for nodes in Ω_p, $p = 1, \dots, 8$, the field function $\mathbf{\Phi}(\phi, \varphi)$ can uniquely track all the subdomains. As illustrated via the slice plane in Fig. 3.6e, the field function can also resolve multiple interfacial layers for each subdomain Ω_p, $p = 0, \dots, 8$. In the multidomain node mapping algorithm described in Section 3.2.3, nodes inside the overlapped regions between two polyhedrons will be mapped into the polyhedron that is indexed earlier for node mapping and then be exempted from the point-in-polyhedron tests for the one that is indexed later. For instance, the results in Fig. 3.6d present the case that Ω_2 is indexed earlier than Ω_{3-6} but later than Ω_7 during node mapping. Therefore, the field function $\mathbf{\Phi}(\phi, \varphi)$ enables an automatic boolean treatment for geometric intersections. Due to that Ω_{3-6} are relatively very thin objects, all nodes inside them become interfacial nodes. Since Ω_0 is the desired solution domain, numerical boundary treatment can be conducted at the interfacial nodes of Ω_0 shown in Fig. 3.6e for a non-ghost-cell approach or at the ghost nodes of Ω_0 shown in Fig. 3.6f for a ghost-cell approach. Only related to the outer profile of the assembled object, both the interfacial nodes and ghost nodes of Ω_0 will not be affected by the index sequence of polyhedrons in node mapping.

Three Cartesian grids $I_1 \times J_1 \times K_1 = 350 \times 96 \times 96$, $I_2 \times J_2 \times K_2 = 525 \times 145 \times 145$, and $I_3 \times J_3 \times K_3 = 700 \times 192 \times 192$ are used to discretize Ω. The initial flow condition in Ω is $(\rho_0, u_0, v_0, p_0) = (1.4\,\mathrm{kg/m^3}, 0, 0, 400\,\mathrm{Pa})$ with the speed of sound being $a_0 = 20\,\mathrm{m/s}$. A hypersonic inflow with condition $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (\rho_0, M_\infty a_0, 0, p_0)$ and $M_\infty = 5$ is imposed at the front end of Ω, and an outflow boundary condition is used at the rear end of Ω. The slip wall condition is enforced on the surfaces of Ω_p, $p = 1, \dots, 8$. The evolution of the system is solved to $t = 2l/(M_\infty a_0) = 0.04\,\mathrm{s}$ and is illustrated in Fig. 3.7. The obtained steady conical shocks around the cone region on three grids are captured in Fig. 3.8 and are compared with the analytical solution derived by the Taylor-Maccoll analysis [133]. The predicted shock angles on $I_n \times J_n \times K_n$, $n = 1, 2, 3$, are about 20.526°, 20.487°, and 20.118°, respectively, which agree very well with the analytical solution 20.051° and demonstrate the success of applying the proposed field function for constructing complex fluid-solid systems via multidomain node mapping.

3.3.2 Supersonic shock-sphere interaction

As illustrated in Fig. 3.9, the unsteady drag force and pressure history acting on a sphere with radius $R = 0.04\,\mathrm{m}$ suspended in a $L \times H \times W = 0.5\,\mathrm{m} \times 0.3\,\mathrm{m} \times 0.3\,\mathrm{m}$ test region and impacted by a Mach 1.22 planar incident shock are studied to validate the proposed field function for surface force integration. The center of the sphere overlaps with the center of the test region and is at the origin position $O(0, 0, 0)$, and the incident shock is initially positioned at $x = -1.5R$. The pre-shock and post-shock states are $(\rho_1, u_1, v_1, w_1, p_1) = (1.205\,\mathrm{kg/m^3}, 0, 0, 0, 101325\,\mathrm{Pa})$ and $(\rho_2, u_2, v_2, w_2, p_2) = (1.658\,\mathrm{kg/m^3}, 114.477\,\mathrm{m/s}, 0, 0, 159060\,\mathrm{Pa})$, respectively. The drag coefficient is computed as $C_\mathrm{D} = F_x/(0.5\rho_2 u_2^2 \pi R^2)$, where F_x is the x-component of the total force acting on the sphere.



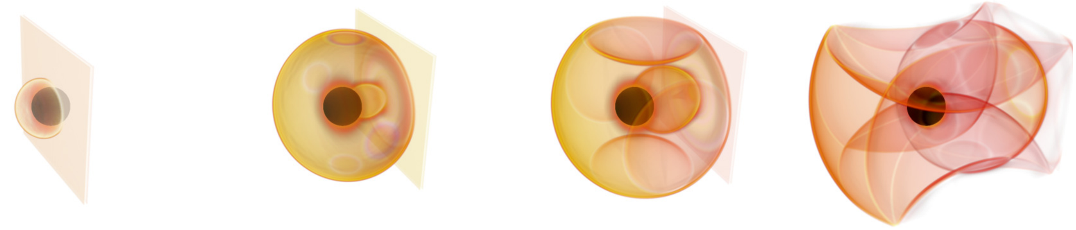
(a)

(b)

Figure 3.9 Schematic diagrams illustrating the shock-sphere interaction problem. (a) The $3D$ problem domain. (b) Pressure probe locations. [Nomenclature: L, domain length; H, domain height; W, domain width; I, incident shock; R, sphere radius; P_1 and P_2, pressure probes at sphere surface.]

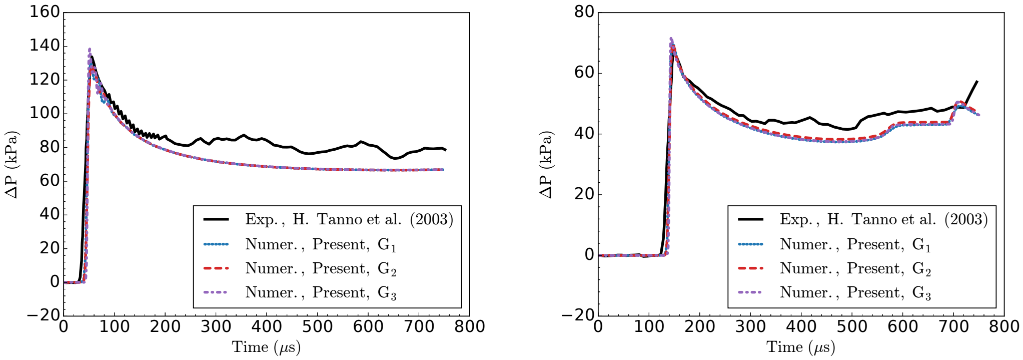
Tanno et al. [134] experimentally measured the drag coefficient and pressure history using a shock tube facility. In addition, employing a $2D$ curvilinear grid and the axisymmetric Navier–Stokes equations, they also numerically simulated the flow with the no-slip wall boundary condition and reported the obtained drag coefficient. In this study, $3D$ Cartesian grids are used. To reduce computational cost, the Euler equations with the slip-wall boundary condition are applied instead, since the viscous effect is very limited in this supersonic flow, and the pressure force dominates the shock-sphere interaction [134]. Three levels of grids, $G_1 = 400 \times 240 \times 240$, $G_2 = 600 \times 360 \times 360$, and $G_3 = 800 \times 480 \times 480$, are used to test grid convergence.

The evolution of the numerical Schlieren field is captured in Fig. 3.10, in which the reflection and diffraction of the shock wave along sphere surface and the formation of wake by shock collision are clearly illustrated. A comparison of the predicted pressure variation $\Delta p = p - p_1$ at the two probe locations with the experimental measurements



(a) $t = 150\,\mu$s (b) $t = 400\,\mu$s (c) $t = 500\,\mu$s (d) $t = 750\,\mu$s

Figure 3.10 The time evolution of shock-sphere interaction illustrated by the volume rendering of the numerical Schlieren field solved on the grid G_1.



(a) (b)

Figure 3.11 Pressure variation over time at the probe locations. (a) Probe P_1. (b) Probe P_2.

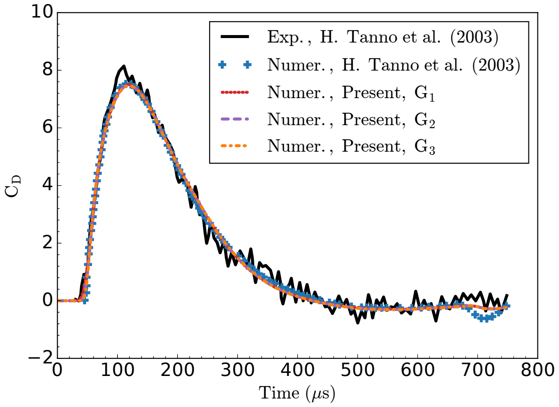


Figure 3.12 Comparison of drag coefficient for shock-sphere interaction.

in reference [134] is shown in Fig. 3.11. For $t \in [0, 160\,\mu\mathrm{s}]$, some minor discrepancies are observed among the solutions obtained from three different grids, and the numerical pressure data closely resemble the experimental one, including both the arrival time and the value of the peak pressure. At the later stage, the numerical and experimental results show moderate discrepancies. A further comparison of the predicted drag coefficient with the experimental and numerical data in reference [134] is depicted in Fig. 3.12. The obtained drag coefficient herein is consistent with the numerical result in [134] and agrees well with the experimental measurement. It is worth noting that, in the reference [134], compared to the pressure measurement, a model configuration less affecting the flow was used in the drag measurement, which might be one of the reasons leading to the different levels of agreement between the numerical and experimental data on pressure and drag coefficient at the late stage of evolution.

3.3.3 A multibody contact and collision system

As illustrated in Fig. 3.13, a fluid-solid system is employed to demonstrate the field function for multibody contact and collision applications. In a $L \times H = [-5D, 5D] \times [-5D, 5D]$ domain with an initial flow state $(\rho_0, u_0, v_0, p_0) = (1.4\,\mathrm{kg/m^3}, 0, 0, 400\,\mathrm{Pa})$, five circular solids with diameter $D = 1\,\mathrm{m}$ are placed with the centers being $C_1(-4D, 0)$, $C_2(0, 4D)$, $C_3(0, 0)$, $C_4(2D, -2D)$, and $C_5(2D + 1/\sqrt{2}D, -2D - 1/\sqrt{2}D)$, respectively. Extending the collision system in [73], a 90°-angled wall with the inner corner locating at $W_6(2.5D + \sqrt{2}D, -2.5D - \sqrt{2}D)$ is added to introduce a multicontact problem between C_5 and W_6. During collision, both C_5 and W_6 will have two contact regions occurring simultaneously. As a difficult problem to solve in rigid-body dynamics [115], the multicontact collision between C_5 and W_6 is used to further demonstrate the collision detection of the proposed field function.

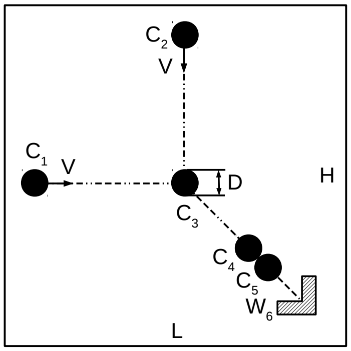


Figure 3.13 A fluid-solid system with analytically solvable multibody contact and collisions.

Initially, C_1 and C_2 move with $V = 50\mathrm{m/s}$ and Mach number $M = 2.5$. C_3, C_4, and

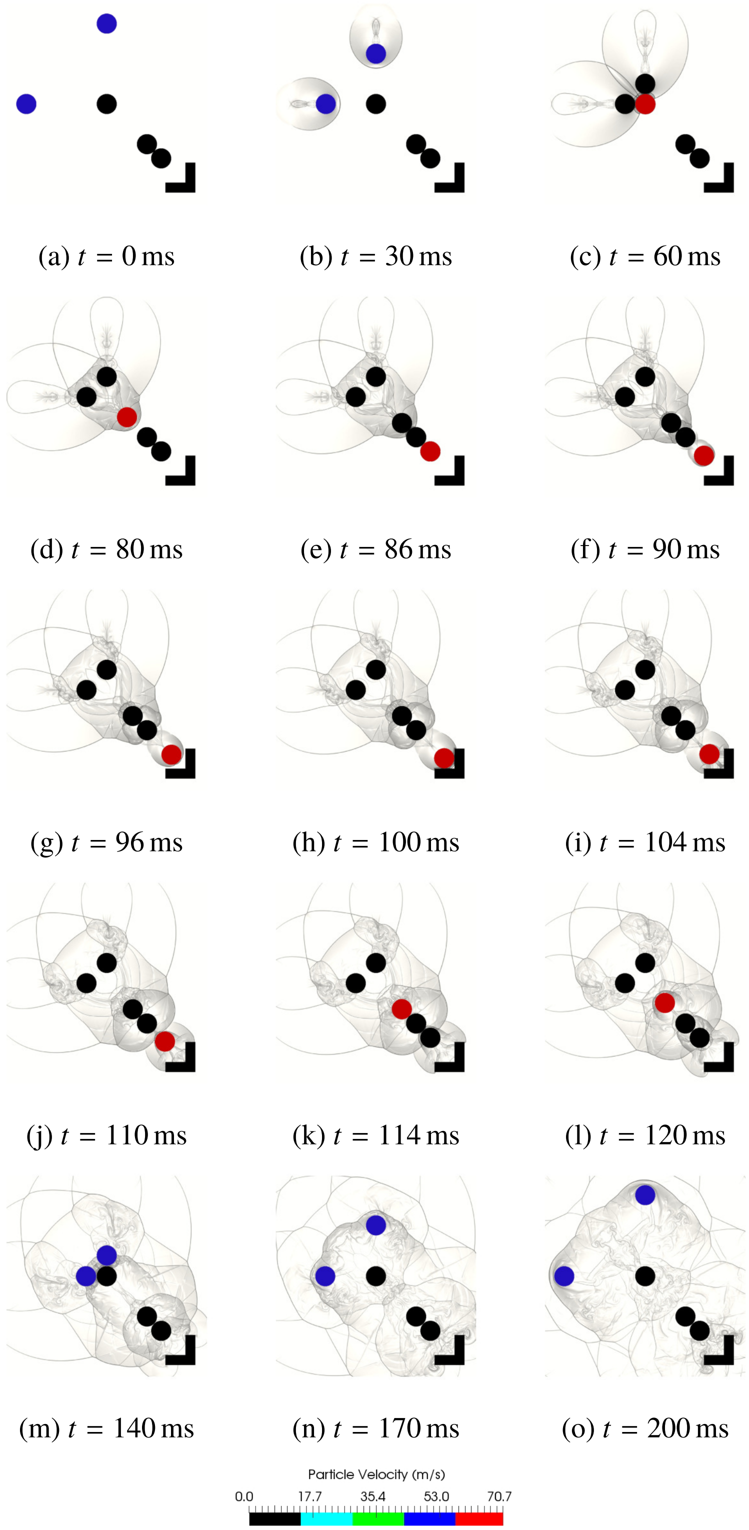
C_5 are stationary, and the wall W_6 is fixed in space. If all the collisions are assumed to be elastic and fluid forces acting on solids are neglected, then the motions of the solids are analytically solvable. The fluid-solid system is solved to $t = 200\,\mathrm{ms}$ on three grids $G_1 = 300 \times 300$, $G_2 = 600 \times 600$, and $G_3 = 1200 \times 1200$, and the evolution is captured in Fig. 3.14 using results obtained on G_3. As shown from Fig. 3.14a to Fig. 3.14c, once colliding with C_3 simultaneously at $t = 60\,\mathrm{ms}$, C_1 and C_2 pass their momentum to C_3. From Fig. 3.14d to Fig. 3.14f, C_3 moves with $(u, v) = (50\,\mathrm{m/s}, 50\,\mathrm{m/s})$ and collides with C_4 at $t = 100 - 10\sqrt{2}\,\mathrm{ms}$. As a result, a colliding chain is formed between C_3, C_4, and C_5 to instantly pass momentum from C_3 to C_5. From Fig. 3.14g to Fig. 3.14i, C_5 collides with the wall W_6 and then bounces back, causing the collision sequence to be inverted. From Fig. 3.14j to Fig. 3.14l, C_5 moves with $(u, v) = (50\,\mathrm{m/s}, 50\,\mathrm{m/s})$ and collides with C_4 at $t = 100 + 10\sqrt{2}\,\mathrm{ms}$ to reform the colliding chain between C_5, C_4, and C_3 and instantly transfer the momentum to C_3. From Fig. 3.14m to Fig. 3.14o, C_3 collides with C_1 and C_2 simultaneously and completely transfer its x-momentum to C_1 and y-momentum to C_2. Then, C_1 and C_2 travel back to their initial positions.

The evolution process involves multibody collisions with momentum transfer at both aligned (Fig. 3.14e and Fig. 3.14k) and angled (Fig. 3.14c and Fig. 3.14m) directions, as well as a multicontact collision between C_5 and W_6 (Fig. 3.14h). During the collisions, solid states can instantly switch between $M = 0$ and $M = 5\sqrt{2}$, introducing computational challenges.

The solutions obtained on G_n, $n = 1, 2, 3$, are compared in Fig. 3.15. The collision dynamics, including collision detections, the lines of impacts, and collision velocities are solved exactly on each grid. Table 3.1 presents the predicted position errors of solid centers on G_n. The maximum position errors of the solid centers happen at C_1 and C_2 and are about 11%, 5.7%, and 2.8% relative to the diameter D for G_1, G_2, and G_3, respectively. The obtained results demonstrate that the proposed field function can successfully facilitate the solution of complex and dynamic fluid-solid system, and the collision detection and response algorithms based on the field function can correctly resolve the multibody contact and collisions present in the current fluid-solid system. Although not being able to solve multibody collisions in general exactly, the current collision model provides a deterministic approach for approximating multibody collision response with parameterized elasticity and friction and effectively supports the validation of the collision detection capability of the proposed field function.

3.3.4 Supersonic wedge penetrating a particle bed

A supersonic wedge penetrating a particle bed is simulated to further demonstrate the applicability of the field function for solving complex and dynamic fluid-solid systems. As illustrated in Fig. 3.16, in a $L \times H = [-0.5D, 13.5D] \times [-3.5D, 3.5D]$ domain, a



(a) $t = 0\,\text{ms}$ (b) $t = 30\,\text{ms}$ (c) $t = 60\,\text{ms}$

(d) $t = 80\,\text{ms}$ (e) $t = 86\,\text{ms}$ (f) $t = 90\,\text{ms}$

(g) $t = 96\,\text{ms}$ (h) $t = 100\,\text{ms}$ (i) $t = 104\,\text{ms}$

(j) $t = 110\,\text{ms}$ (k) $t = 114\,\text{ms}$ (l) $t = 120\,\text{ms}$

(m) $t = 140\,\text{ms}$ (n) $t = 170\,\text{ms}$ (o) $t = 200\,\text{ms}$

Figure 3.14 Numerical solution of a fluid-solid system with analytically solvable multibody contact and collisions.

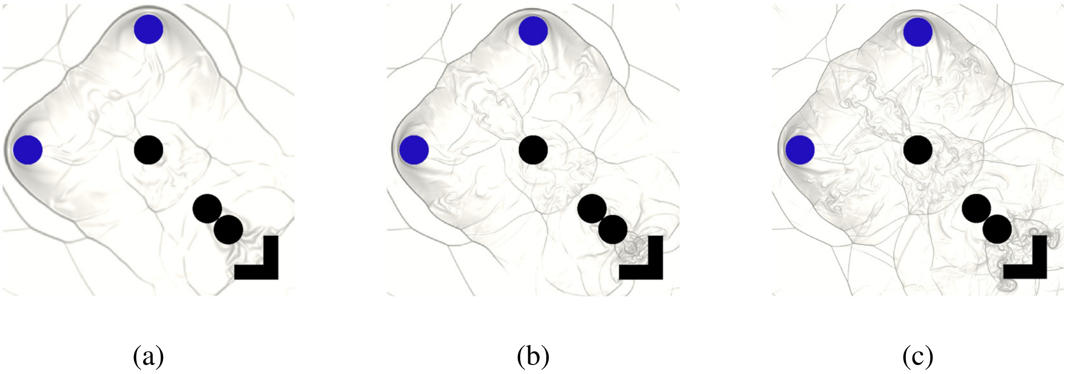

(a) (b) (c)

Figure 3.15 Grid sensitivity study on the solution of the fluid-solid system with multibody contact and collisions. (a) G_1. (b) G_2. (c) G_3.

	C_1	C_2	C_3	C_4	C_5
Exact	$(-4, 0)$	$(0, 4)$	$(0, 0)$	$(2, -2)$	$(2+\frac{1}{\sqrt{2}}, -2-\frac{1}{\sqrt{2}})$
G_1 Error	$(-1.1\text{e}{-1}, 0)$	$(0, 1.1\text{e}{-1})$	$(-3.6\text{e}{-3}, 3.6\text{e}{-3})$	$(0, 0)$	$(1.0\text{e}{-2}, -1.0\text{e}{-2})$
G_2 Error	$(-5.7\text{e}{-2}, 0)$	$(0, 5.7\text{e}{-2})$	$(-7.8\text{e}{-4}, 7.8\text{e}{-4})$	$(3.0\text{e}{-4}, -3.0\text{e}{-4})$	$(2.9\text{e}{-3}, -2.9\text{e}{-3})$
G_3 Error	$(-2.8\text{e}{-2}, 0)$	$(0, 2.8\text{e}{-2})$	$(-6.6\text{e}{-4}, 6.6\text{e}{-4})$	$(-4.0\text{e}{-5}, 4.0\text{e}{-5})$	$(4.6\text{e}{-3}, -4.6\text{e}{-3})$

Table 3.1 The predicted position errors of solid centers at the end of solution time for the fluid-solid system.

wedge with length $D = 1\,\text{m}$ and deflection angle $\theta = 15°$ is horizontally positioned in the domain, and the front vertex of the wedge locates at $O(12D, 0)$. In addition, in the $w \times h = [2D, 4D] \times [-1D, 1D]$ region, 64 identical circular particles with diameter $d = 0.25D$ are tightly packed.

Initially, the gas in the domain has the state $(\rho_0, u_0, v_0, p_0) = (1.4\,\text{kg/m}^3, 0, 0, 400\,\text{Pa})$, in which the speed of sound is $a_0 = 20\,\text{m/s}$. The wedge has a density $\rho_\text{s} = 2700\,\text{kg/m}^3$, a coefficient of restitution $C_R = 0.5$, and an initial velocity $M_\infty = 3$. Particles have a density ρ_s, a coefficient of restitution $C_R = 0.0$, and zero initial velocity. Flow inside the domain is assumed to be inviscid. The slip-wall condition is imposed at the top and bottom domain boundaries as well as at the wedge and particle surfaces, while the outflow boundary condition is enforced at the left and right domain boundaries. The evolution of this fluid-solid system is solved to $t = 0.25\,\text{s}$ on a 2800×1400 Cartesian grid.

During the solution process, before the wedge collides with the particle bed (for $t \leq 4/30\,\text{s}$), the fluid forces acting on the wedge are deactivated such that the wedge can move with a constant supersonic speed. As a result, oblique shock waves generated at the nose of the moving wedge can reach a steady state with a constant shock angle β. The simulated time evolution of the system is captured in Fig. 3.17, in which the lines denoted by square marks represent the analytical solutions of the oblique shock angles at position $(4.5D, 0)$.

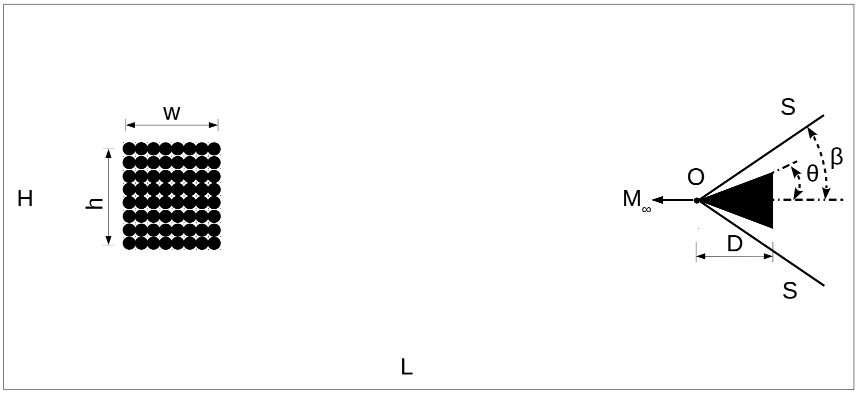
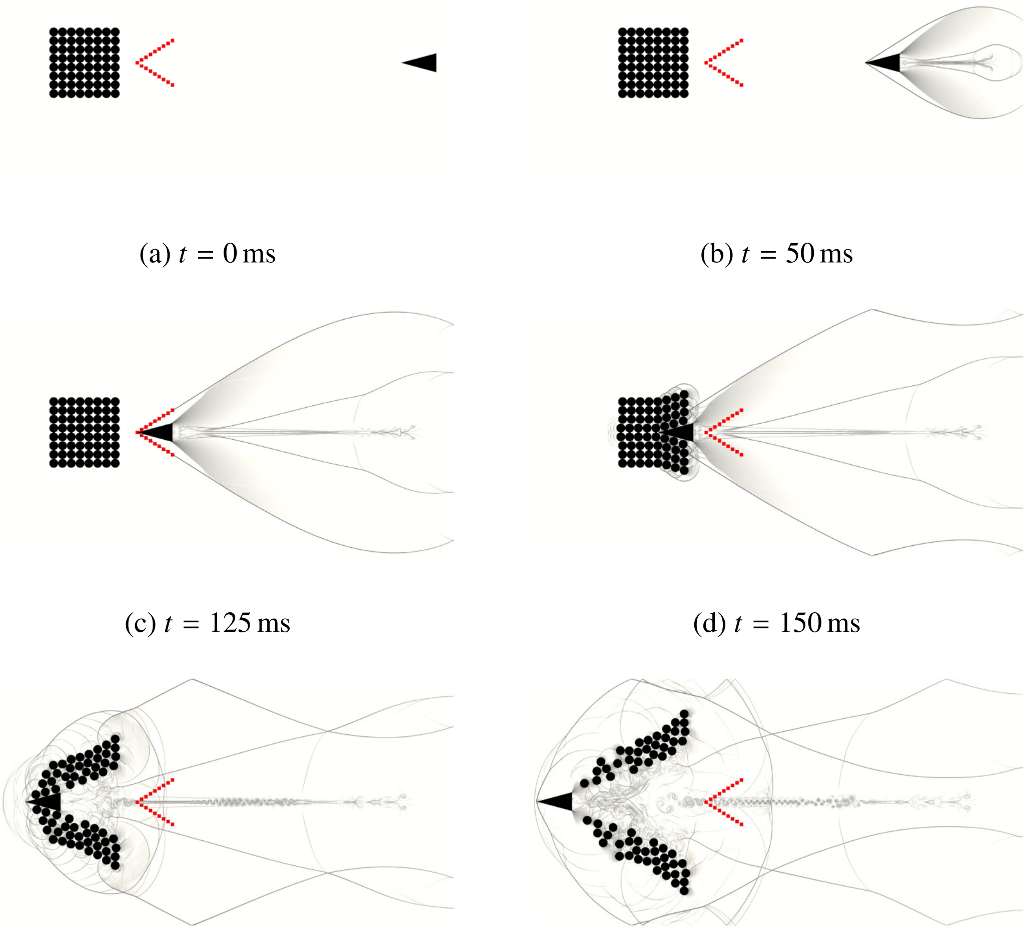


Figure 3.16 A schematic diagram for the supersonic wedge penetrating a particle bed problem. [Nomenclature: M_∞, Mach number of the moving wedge; S, oblique shock; θ, deflection angle; β, shock angle; D, length of wedge; O, the front vertex of wedge; L, domain length; H, domain height; w, particle bed width; h, particle bed height.]

As shown in Fig. 3.17c, the predicted oblique shock angle $\beta_n = 32.259°$ agrees very well with the analytical solution $\beta_e = 32.240°$ computed from the $M_\infty - \theta - \beta$ relation [5].

After the wedge collides with the particle bed, a force chain within the contacted particles is created due to the penetrating wedge. This force chain accelerates the particles and fractures the particle bed. The suddenly destabilized particle bed generates strong flow disturbances at the surrounding area, which interact with the wedge generated shocks and waves, forming complex wave diffraction and interference patterns in space. During the wedge penetrating the particle bed, intensive multibody contact and collisions are successfully simulated, and an intuitive dynamic process is captured in the solution. These results illustrate the ability of the proposed field function for facilitating the solution of complex and dynamic fluid-solid systems involving coupled fluid-solid and solid-solid interactions.



(a) $t = 0\,\mathrm{ms}$

(b) $t = 50\,\mathrm{ms}$

(c) $t = 125\,\mathrm{ms}$

(d) $t = 150\,\mathrm{ms}$

(e) $t = 200\,\mathrm{ms}$

(f) $t = 250\,\mathrm{ms}$

Figure 3.17 Time evolution of a supersonic wedge penetrating a particle bed. Lines denoted by square marks represent the analytical solutions of the oblique shocks.

3.4 Conclusion

A new integer-type field function has been developed to facilitate the solution of complex and dynamic fluid-solid systems on Cartesian grids with interface-resolved fluid-solid and solid-solid interactions. The main conclusions are summarized as follows:

- For a Cartesian-grid-discretized computational domain segmented by a set of solid bodies, the proposed field function can uniquely track all the subdomains with multiple layers of interfacial nodes. Employing the proposed field function, it is straightforward to apply designated governing equations, constitutive models, numerical schemes, and boundary conditions for each subdomain.

- Benefiting from the ability to uniquely track all the subdomains with multiple layers of interfacial nodes using only one integer field in total, the proposed field function can enable four types of useful functionalities. The first functionality is to provide single-integer-based multidomain node mapping to minimize memory usage and maximize information set. For instance, in providing node mapping information for P subdomains, the memory consumption for using the proposed field function can be about $1/P$ of that for using a binary node-mapping function. The second functionality is to enable efficient node remapping that employs a temporal coherence to obtain an acceleration rate inversely proportional to the surface-to-volume ratio of subdomains. The third functionality is to perform collision detection for determining collision queues and lines of impacts with linear time complexity, instead of the quadric time complexity required by a binary node-mapping function. The fourth functionality is to facilitate the surface force integration for irregular solids immersed in a Cartesian grid. Easy-to-implement algorithms for the field function and its associated functionalities have also been presented.

- Equipped with a deterministic multibody collision model, the applicability of the developed field function for solving complex and dynamic fluid-solid systems has been validated and demonstrated through a set of numerical experiments, such as a complex wind tunnel test efficiently constructed via the field function to demonstrate multidomain node mapping, supersonic shock-sphere interaction to validate surface force integration, a multibody contact and collision system and a supersonic wedge penetrating a particle bed to demonstrate collision detection and node remapping for dynamic fluid-solid systems. The obtained numerical results are in close agreement with the corresponding published numerical data, experimental observations, or analytical solutions.

Chapter 4

An Immersed Boundary Method for Treating Arbitrarily Irregular and Moving Boundaries

4.1 Introduction

To solve flow involving irregular and moving geometries, one of the main challenges is related to enforcing boundary conditions. In this chapter, an immersed boundary method is developed, validated, and applied. Through devising a second-order three-step flow reconstruction scheme, the proposed method is able to enforce the Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner. Equipped with a fluid-solid coupling framework that integrates high-order temporal and spatial discretization schemes, numerical experiments concerning flow involving stationary and moving objects, convex and concave geometries, no-slip and slip wall boundary conditions, as well as subsonic and supersonic motions are conducted to validate the method. Using analytical solutions, experimental observations, published numerical results, and Galilean transformations, it is demonstrated that the proposed method can provide efficient, accurate, and robust boundary treatment for solving flow with arbitrarily irregular and moving geometries on Cartesian grids. On the basis of the proposed method, the development of a solver that unifies 1D, 2D, and 3D computations and the generation of complex geometric objects via simply positioning components are described. In addition, a surface-normalized absolute flux is proposed for interface sharpness measurement, and an analytically solvable modified vortex preservation problem is developed for a convergence study concerning smooth flow with irregular geometries.

4.2 Method development

4.2.1 A generalized framework

Two- and three-dimensional Cartesian grid-based computational domains with immersed boundaries are illustrated in Fig. 4.1, in which G denotes a ghost node, a computational node that is outside the physical domain and locates at the numerical boundaries, O denotes a boundary point with $\boldsymbol{GO}$ as the outward normal vector, and I is the image point of the ghost node G reflected by the physical boundary.

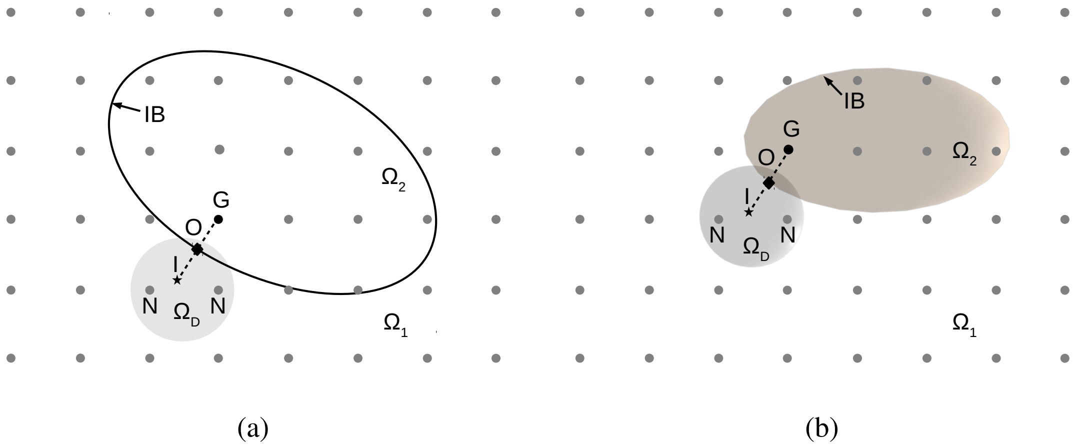


Figure 4.1 Schematic diagrams of Cartesian grid-based computational domains with immersed boundaries. (a) 2D space. (b) 3D Space. [Nomenclature: G, ghost node; O, boundary point; I, image point; Ω_{D}, domain of dependence; N, neighboring fluid nodes in the domain of dependence; Ω_1, fluid domain; Ω_2, solid domain; IB, immersed boundary.]

In a direct forcing immersed boundary method, boundary treatment is mainly about constructing the flow at numerical boundaries. To obtain a proper ghost flow with the existence of physical boundaries effectively admitted, an interpolation approach incorporating the method of images is commonly adopted:

$$\psi_G = 2\psi_O - \psi_I \tag{4.1a}$$

$$\psi_I = f(x_I, y_I, z_I) \tag{4.1b}$$

where ψ denotes a generic flow variable, $f(x, y, z)$ is a local reconstruction function of ψ at the image point I.

As the local reconstruction function will largely determine the numerical properties of the resulting immersed boundary method, a three-step flow reconstruction scheme is developed to achieve efficient, accurate, and robust boundary treatment for arbitrarily irregular and moving boundaries.

4.2.2 A three-step flow reconstruction scheme

In this study, the local reconstruction function is established on the physical boundary conditions at the boundary point O and the known values of ψ at nearby fluid nodes:

$$\psi_I = f(\{\psi_N\}, \psi_O) \tag{4.2}$$

where ψ_O is the value of ψ at the boundary point O, at which the physical boundary conditions are enforced. Except for a Dirichlet boundary condition, the value of ψ_O is unknown and implicitly subject to different types of mathematical constraints, which are the main challenges in developing a local reconstruction function. $\{\psi_N\}$ represents the values of ψ at fluid nodes $\{N\}$ that satisfy

$$d_N = ||\boldsymbol{x}_I - \boldsymbol{x}_N|| \leq R_I \tag{4.3}$$

in which $\boldsymbol{x}_I$ and $\boldsymbol{x}_N$ are the position vectors of the point I and N, respectively. R_I, referred to as the domain of dependence of the point I and illustrated in Fig. 4.1, is the maximum distance from the point I to nearby fluid nodes that are employed for flow reconstruction.

The incorporation of physical boundary conditions in the local reconstruction function of ψ_I gives

$$\begin{aligned}
\lim_{||\boldsymbol{x}_G-\boldsymbol{x}_O||\to 0} \psi_G &= 2\psi_O - \lim_{||\boldsymbol{x}_G-\boldsymbol{x}_O||\to 0} \psi_I \\
&= 2\psi_O - \lim_{||\boldsymbol{x}_I-\boldsymbol{x}_O||\to 0} \psi_I \\
&= 2\psi_O - \lim_{||\boldsymbol{x}_I-\boldsymbol{x}_O||\to 0} f(\{\psi_N\}, \psi_O) \\
&= \psi_O
\end{aligned} \tag{4.4}$$

Therefore, the constructed ψ_G converges to the exact physical boundary conditions when the ghost node G converges to the boundary point O.

For ψ representing a generic field variable, to construct $\psi_I = f(\{\psi_N\}, \psi_O)$ regarding the Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner, a three-step flow reconstruction scheme is proposed.

Prediction step Pre-estimate the value of ψ_I by applying inverse distance weighting on the fluid nodes that locate in the domain of dependence of the image point I:

$$\psi_I^* = \frac{\sum w(d_N)\psi_N}{\sum w(d_N)}, \ d_N = \max(d_N, d_{\text{tiny}}) \text{ and } \ d_N \leq R_I \tag{4.5}$$

where ψ_I^* denotes the predicted value of ψ_I, the weighting function $w(d)$ employs an inverse-power law $1/d^q$ with $q \in \mathbb{N}$, d_N is the distance from the node N to the image point I, $d_{\text{tiny}} = \varepsilon \min(\Delta_x, \Delta_y, \Delta_z)$ is a positive real number to avoid a zero denominator, $\Delta_x, \Delta_y, \Delta_z$

are the mesh sizes in x, y, z directions, respectively. The power $q = 2$, the constant $\varepsilon = 10^{-6}$, and the domain of dependence $R_I = 2\max(\Delta_x, \Delta_y, \Delta_z)$ are adopted in the test cases herein, which are shown to be well adequate for a variety of flow problems, as demonstrated in the numerical experiments.

Physical boundary condition enforcement step Determine the value of ψ_O via the physical boundary conditions that ψ needs to satisfy at the boundary point O and the values of ψ at interior physical domain.

Dirichlet boundary condition If ψ satisfies the Dirichlet boundary condition, the value of ψ_O is exactly determined by the specified boundary condition:

$$\psi_O = g \tag{4.6}$$

where g is a given value or function.

Neumann boundary condition ψ is required to satisfy

$$\left.\frac{\partial \psi}{\partial n}\right|_O = \frac{\partial \psi_O}{\partial n} \tag{4.7}$$

where $\partial \psi_O / \partial n$ is a given value or function.

In other words,

$$\lim_{l \to 0} \frac{\psi(\boldsymbol{x}_O + l\mathbf{n}) - \psi(\boldsymbol{x}_O)}{l} = \frac{\partial \psi_O}{\partial n} \tag{4.8}$$

where $\boldsymbol{x}_O$ and $\mathbf{n}$ are the position vector and the unit outward surface normal vector at the boundary point O, respectively.

Since point I is on the normal direction of point O, it gives

$$\mathbf{n} = \frac{\boldsymbol{x}_I - \boldsymbol{x}_O}{||\boldsymbol{x}_I - \boldsymbol{x}_O||} \tag{4.9}$$

Therefore,

$$\frac{\psi_I - \psi_O}{||\boldsymbol{x}_I - \boldsymbol{x}_O||} - \left.\frac{\partial^2 \psi}{\partial n^2}\right|_O ||\boldsymbol{x}_I - \boldsymbol{x}_O|| + \mathrm{O}(||\boldsymbol{x}_I - \boldsymbol{x}_O||^2) = \frac{\partial \psi_O}{\partial n} \tag{4.10}$$

Due to Eq. (4.1a), the second-order derivative term has

$$\left.\frac{\partial^2 \psi}{\partial n^2}\right|_O = \frac{\psi_I + \psi_G - 2\psi_O}{2||\boldsymbol{x}_I - \boldsymbol{x}_O||^2} + \mathrm{O}(||\boldsymbol{x}_I - \boldsymbol{x}_O||^2) = \mathrm{O}(||\boldsymbol{x}_I - \boldsymbol{x}_O||^2) \tag{4.11}$$

Hence, a second-order accurate approximation of ψ_O can be given as

$$\psi_O = \psi_I - ||\boldsymbol{x}_I - \boldsymbol{x}_O|| \frac{\partial \psi_O}{\partial n} \tag{4.12}$$

Robin boundary condition A linear combination of the value of ψ and its normal derivative on the boundary point O is specified:

$$\alpha\psi_O + \beta \left.\frac{\partial\psi}{\partial n}\right|_O = g \tag{4.13}$$

where α and β are the linear combination coefficients, g is a given value or function.

After approximating the normal derivative with second-order accuracy, it gives

$$\alpha\psi_O + \beta\frac{\psi_I - \psi_O}{||\boldsymbol{x}_I - \boldsymbol{x}_O||} = g \tag{4.14}$$

Then,

$$\psi_O = \frac{\beta\psi_I - ||\boldsymbol{x}_I - \boldsymbol{x}_O||g}{\beta - ||\boldsymbol{x}_I - \boldsymbol{x}_O||\alpha} \tag{4.15}$$

Cauchy boundary condition For illustration purpose, ψ is replaced by the velocity vector $\boldsymbol{V} = (u, v, w)$ that satisfies the slip-wall boundary condition:

$$\begin{aligned} (\boldsymbol{V}\cdot\mathbf{n})|_{\boldsymbol{x}=\boldsymbol{x}_O} &= \boldsymbol{V}_S\cdot\mathbf{n} \\ \left.\frac{\partial(\boldsymbol{V}\cdot\hat{\boldsymbol{t}})}{\partial n}\right|_{\boldsymbol{x}=\boldsymbol{x}_O} &= 0 \\ \left.\frac{\partial(\boldsymbol{V}\cdot\tilde{\boldsymbol{t}})}{\partial n}\right|_{\boldsymbol{x}=\boldsymbol{x}_O} &= 0 \end{aligned} \tag{4.16}$$

where $\mathbf{n}$, $\hat{\mathbf{t}}$, and $\tilde{\mathbf{t}}$ are the unit normal vector, unit tangent vector, and unit bitangent vector at the boundary point O, respectively. $\boldsymbol{V}_S$ is the velocity of the boundary surface at the point O.

After approximating the normal derivative with second-order accuracy, it gives

$$\begin{aligned} u_On_x + v_On_y + w_On_z &= u_Sn_x + v_Sn_y + w_Sn_z \\ u_O\hat{t}_x + v_O\hat{t}_y + w_O\hat{t}_z &= u_I\hat{t}_x + v_I\hat{t}_y + w_I\hat{t}_z \\ u_O\tilde{t}_x + v_O\tilde{t}_y + w_O\tilde{t}_z &= u_I\tilde{t}_x + v_I\tilde{t}_y + w_I\tilde{t}_z \end{aligned} \tag{4.17}$$

Using the orthogonality of the coefficient matrix, $\boldsymbol{V}_O$ is determined as

$$\begin{pmatrix} u_O \\ v_O \\ w_O \end{pmatrix} = \begin{bmatrix} n_x & n_y & n_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}^{\mathrm{T}} \begin{pmatrix} u_Sn_x + v_Sn_y + w_Sn_z \\ u_I\hat{t}_x + v_I\hat{t}_y + w_I\hat{t}_z \\ u_I\tilde{t}_x + v_I\tilde{t}_y + w_I\tilde{t}_z \end{pmatrix} \tag{4.18}$$

The solution equations of ψ_O for different types of boundary conditions now can be unified as

$$\psi_O = C\psi_I + \text{RRHS} \tag{4.19}$$

where the values of the coefficient C and the rest right-hand side RRHS are summarized in Table 4.1.

Table 4.1 Value map of C and RRHS for different types of boundary conditions.

Type	Example form	C	RRHS
Dirichlet	$\psi_O = g$	0	g
Neumann	$\left.\frac{\partial\psi}{\partial n}\right\|_O = \frac{\partial\psi_O}{\partial n}$	1	$-\|\|\boldsymbol{x}_I - \boldsymbol{x}_O\|\|\frac{\partial\psi_O}{\partial n}$
Robin	$\alpha\psi_O + \beta \left.\frac{\partial\psi}{\partial n}\right\|_O = g$	$\frac{\beta}{\beta - \|\|\boldsymbol{x}_I - \boldsymbol{x}_O\|\|\alpha}$	$\frac{-\|\|\boldsymbol{x}_I - \boldsymbol{x}_O\|\|g}{\beta - \|\|\boldsymbol{x}_I - \boldsymbol{x}_O\|\|\alpha}$
Cauchy	$(\boldsymbol{V} \cdot \mathbf{n})\|_{\boldsymbol{x}=\boldsymbol{x}_O} = \boldsymbol{V}_S \cdot \mathbf{n}$ $\left.\frac{\partial(\boldsymbol{V} \cdot \hat{\boldsymbol{t}})}{\partial n}\right\|_{\boldsymbol{x}=\boldsymbol{x}_O} = 0$ $\left.\frac{\partial(\boldsymbol{V} \cdot \tilde{\boldsymbol{t}})}{\partial n}\right\|_{\boldsymbol{x}=\boldsymbol{x}_O} = 0$	$\begin{bmatrix} n_x & n_y & n_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 0 & 0 & 0 \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}$	$\begin{bmatrix} n_x & n_y & n_z \\ \hat{t}_x & \hat{t}_y & \hat{t}_z \\ \tilde{t}_x & \tilde{t}_y & \tilde{t}_z \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} n_x & n_y & n_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \boldsymbol{V}_S$

Correction step Solve the value of ψ_I by adding the boundary point O as a stencil node for the inverse distance weighting of ψ_I:

$$\psi_I = \frac{\sum w(d_N)\psi_N + w(d_O)\psi_O}{\sum w(d_N) + w(d_O)} = \frac{\psi_I^* + \frac{w(d_O)}{\sum w(d_N)}\psi_O}{1 + \frac{w(d_O)}{\sum w(d_N)}} \tag{4.20}$$

in which the repetition of calculations on fluid nodes can be avoided, since the sum of weights and sum of weighted values are already obtained in the prediction step.

Due to the unknown ψ_I in Eq. (4.19), the solution equation of ψ_O is coupled with the solution equation of ψ_I in the correction step. To solve this problem, one method is a synchronous solving approach to solve ψ_O and ψ_I simultaneously:

$$\begin{cases} \psi_O = \ C\psi_I + \text{RRHS} \\ \psi_I = \ \frac{\psi_I^* + \frac{w(d_O)}{\sum w(d_N)}\psi_O}{1 + \frac{w(d_O)}{\sum w(d_N)}} \end{cases} \tag{4.21}$$

The other is an asynchronous solving approach: first, solve ψ_O via approximating the unknown ψ_I with the pre-estimated ψ_I^*; then, solve ψ_I in the correction step.

$$\begin{cases} \psi_O = C\psi_I^* + \text{RRHS} \\ \psi_I = \frac{\psi_I^* + \frac{w(d_O)}{\sum w(d_N)}\psi_O}{1 + \frac{w(d_O)}{\sum w(d_N)}} \end{cases} \tag{4.22}$$

The enforcement of the Dirichlet and trivial Neumann boundary conditions is equivalent in these two approaches. For the other types of boundary conditions, when the asynchronous solving approach is used, the physical boundary condition enforcement step and the correction step can be iteratively implemented. However, numerical tests suggest that the effects of iterative implementation on the overall solution accuracy are insignificant.

The asynchronous solving approach without iterative implementation is adopted and examined herein, since the validity of the synchronous solving approach is established when the validity of the asynchronous solving approach is proved.

4.2.3 Method discussion

4.2.3.1 Advantages of three-step design

As a matrix inversion-free method, the proposed three-step flow reconstruction scheme herein exempts from the potential numerical singularities involved in solving linear systems, which is a challenging issue for polynomial reconstruction-based methods when stencil nodes are not well distributed due to highly irregular or colliding geometries. Meanwhile, the algorithm complexity of matrix inversion for a $n \times n$ matrix, which is one of the procedures in a polynomial reconstruction-based method, is usually $\mathrm{O}(n^3)$, where n is the number of stencil nodes. The number of floating-point operations of the proposed scheme for reconstructing a field variable is about four multiplications, one division, and seven additions/subtractions per stencil node, in which the boundary point can have additionally about eighteen multiplications and twelve additions in the case of Cauchy boundary condition. Therefore, the algorithm complexity is $\mathrm{O}(n)$, and the proposed scheme is a linear-time algorithm.

In addition to functioning as a flow reconstruction method for the image points, the proposed scheme also effectively solves the physical quantities at the boundary point for boundary conditions with unknown boundary values, such as the non-trivial Neumann, Robin, and Cauchy boundary conditions. The three-step design then further enables a consistent treatment for the Dirichlet, Neumann, Robin, and Cauchy boundary conditions with shared weights. As the calculation of weight occupies four multiplications, one division, and five additions/subtractions, this consistency not only simplifies implementation but also brings efficiency through treating velocity, pressure, and temperature fields with unified algorithm and shared weights, while polynomial reconstruction-based methods usually require constructing and solving a designated linear system for each type of boundary conditions.

4.2.3.2 Automatic adaptation to geometric irregularity

The domain of dependence of the proposed three-step flow reconstruction scheme is solely distance-based, and no spatial dimension-related parameters are involved, as well as no stencil structures are imposed in the scheme. These properties can easily facilitate the development of a solver that unifies 1D, 2D, and 3D computations, avoiding the necessity of maintaining separate versions of codes.

To better illustrate the idea, the generation of the rotor for the subsonic rotational flow in the numerical experiments is shown in Fig. 4.2. The solver developed herein employs the equivalence between a 2D space and a 3D space with the zero gradient condition on the collapsed dimension. As shown in Fig. 4.2a, a 2D geometry is extruded with unit thickness on the third dimension to form a 3D geometry in Fig. 4.2b. The unit thickness provides the

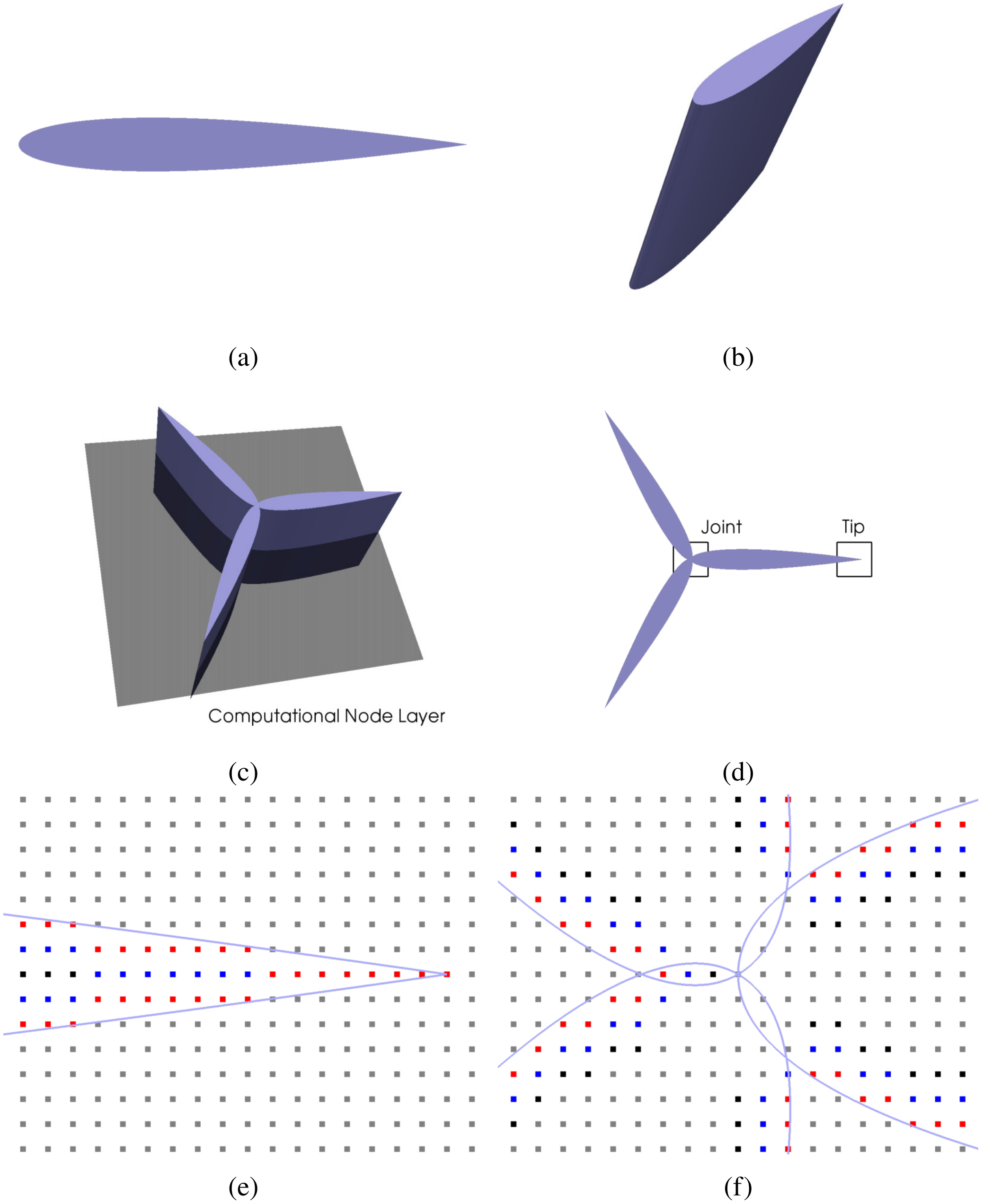


Figure 4.2 Diagrams illustrating ideas related to automatic adaptation to geometric irregularity. (a) A 2D geometry represented by an NACA 0012 airfoil. (b) A 3D geometry generated by extrusion. (c) A 3D complex geometric object generated by component-based assembling and a single computational node layer on the collapsed dimension. The other numerical boundary layers are not shown. (d) The 2D cutting plane for immersed boundary treatment. (e) The node map at the tip region. (f) The node map at the joint region. *Colors* represent the type of nodes: *red*, 1st layer ghost nodes; *blue*, 2nd layer ghost nodes; *black*, 3rd layer ghost nodes; *gray*, fluid or solid nodes for nodes inside the fluid and solid region, respectively. The airfoil profiles are shown in solid curves.

feasibility of directly deriving the perimeter and area of the 2D geometry from the surface area and volume of the 3D geometry.

Benefiting from the robustness of the proposed three-step flow reconstruction scheme, a complex geometric object such as the one in Fig. 4.2c can be generated through assembling its multiple components via positioning while allowing overlapped surfaces, which avoids the cost and cumbersome of producing a single topology. In addition, one computational layer and as many as needed numerical boundary layers, such as three numerical boundary layers on each side for a 7-point-stencil spatial discretization scheme, on the collapsed dimension are used to establish a 3D numerical space, in which the 3D governing equations and the proposed immersed boundary method with 3D spherical domain of dependence are implemented.

When the zero gradient condition is applied to the numerical boundary layers on the collapsed dimension, the 3D governing equations will be computationally equivalent to 2D equations. Since immersed boundary treatment should only be applied to the computational domain, and there is only one computational layer for a 3D space with a collapsed dimension, the implemented 3D spherical domain of dependence for the proposed immersed boundary treatment herein will automatically collapse to a 2D circle. Hence, the method automatically conducts 2D immersed boundary treatment for the 2D cutting plane shown in Fig. 4.2d. Similarly, a 1D space is equivalent to a 3D space with two collapsed dimensions. As a summary, all computations are unified in a 3D space with 3D geometries. The number of computational layers and the applied boundary conditions are used to control the collapse of a specific dimension without modifying the underlying numerical discretization.

The tip and the joint denoted in Fig. 4.2d represent convex and concave regions, respectively. A node map classifying the types of computational nodes is generated using algorithms developed in the reference [71] for facilitating the numerical discretization on a 1200×1200 Cartesian grid. The zoomed tip and joint regions are then shown in Fig. 4.2e and Fig. 4.2f, respectively.

As shown in Fig. 4.2e, for an under-resolved convex region, nodes inside the solid geometry will largely be classified as ghost nodes. In general, due to the convex feature, there will be sufficient neighboring fluid nodes for the corresponding image points, which reduces stencil-related difficulties in flow reconstruction. The tip and the Cartesian grid shown herein are arranged to illustrate a dilemma concerning ghost nodes at the median of a convex angle: each ghost node has two boundary points in 2D space or infinitely many boundary points in 3D space with equal distance but can only store one set of ghost flow. On an under-resolved grid, this dilemma can be alleviated when the median line is not aligned with a node line but placed centrally between two node lines so that there are no shared ghost nodes. Since the underlying cause is the insufficient grid resolution, and this dilemma will be automatically addressed by improving grid resolution, no special treatment is incorporated into the proposed boundary treatment method herein.

As shown in Fig. 4.2f, ghost nodes at a concave region are less affected by that dilemma. However, due to the concave feature, there can be a very limited number of neighboring fluid nodes for the corresponding image points within the domain of dependence. This issue imposes challenges for a polynomial reconstruction-based method, as a fixed number of stencil nodes is needed to avoid under-determined linear systems. On the contrary, the proposed three-step flow reconstruction scheme herein is scalable to the number of stencil nodes and can automatically adapt to a varying number of stencil nodes. Even in the worst situation, in which only one fluid node exists in the domain of dependence of an image point, the proposed method can still be consistently applied. This stencil scalability provides robustness for solving flow with strongly irregular and moving geometries that involve multibody contact and collisions, as demonstrated in the numerical experiment concerning the explosive dispersal of dense particles, in which a significant number of convex and concave regions exist, form, and destroy in the solution process.

4.2.3.3 Fresh node treatment

In solving flow with moving geometries, a ghost node will become a fresh node in the fluid domain when the ghost node moves out from the solid domain after updating the motions of the solids. Therefore, valid field quantities should be reconstructed for these fresh nodes. An idea that consistently treats the image points and fresh nodes is used herein. That is, suppose ψ being a generic flow variable at a fresh node, the proposed three-step flow reconstruction scheme is also applied for the fresh node to reconstruct ψ from the physical boundary conditions at the corresponding boundary point O and the known values of ψ at nearby non-fresh fluid nodes at the same time level. In the numerical experiment involving a supersonic translating wedge, the effectiveness of this approach is demonstrated through comparing the solved solution functionals with analytical solutions as well as via examining the entire solution field using Galilean transformation.

4.3 Numerical experiments

4.3.1 Supersonic flow over a wedge

As illustrated in Fig. 4.3a, when a supersonic flow with Mach number M_∞ passes over a wedge with an adequate deflection angle θ, stationary oblique shock waves with a shock angle β can be created at the nose of the wedge. The $M_\infty - \theta - \beta$ relation can be analytically obtained via a control volume analysis based on conservation laws and has the following form [5]:

$$\tan\theta = \frac{2}{\tan\beta}\frac{M_\infty^2 \sin^2\beta - 1}{M_\infty^2(\gamma + \cos(2\beta)) + 2} \tag{4.23}$$

where γ is the heat capacity ratio.

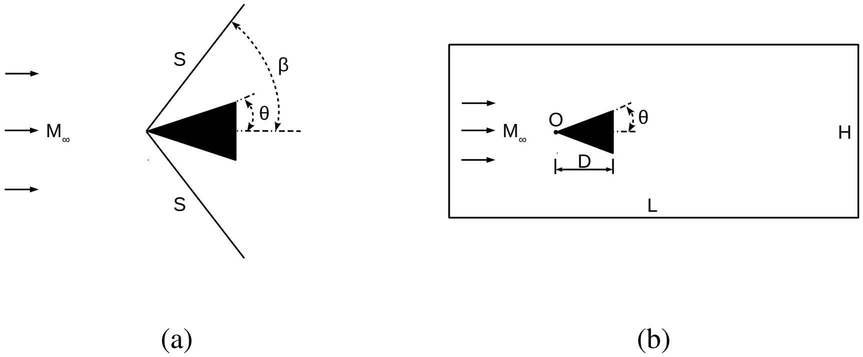
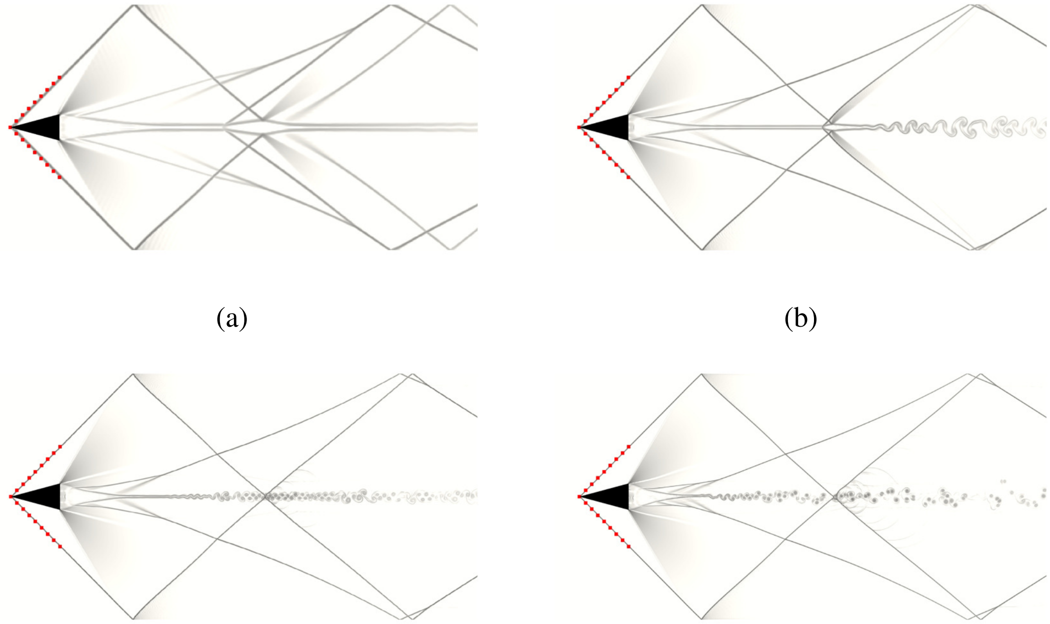


Figure 4.3 Schematic diagrams for the supersonic flow over a wedge problem. (a) Oblique shock relation. (b) Computational configuration. [Nomenclature: M_∞, Mach number of the approaching flow; S, oblique shock; θ, deflection angle; β, shock angle; D, length of wedge; O, the front vertex of wedge; L, domain length; H, domain height.] The schematic diagrams are adapted from the work of Anderson Jr [5].

To validate the proposed method, this supersonic flow over a wedge problem is solved. As illustrated in Fig. 4.3b, in a $L \times H = [-0.5D, 9.5D] \times [-2.5D, 2.5D]$ domain with an initial flow state of $(\rho_0, u_0, v_0, p_0) = (1.4\,\mathrm{kg/m^3}, 40\,\mathrm{m/s}, 0, 400\,\mathrm{Pa})$, in which the speed of sound is $a_0 = 20\,\mathrm{m/s}$, a stationary wedge with $D = 1\,\mathrm{m}$ is positioned at $O(0,0)$ and has the slip-wall boundary condition. The right domain boundary has the outflow condition, and the top and bottom boundaries are treated as slip walls. The left domain boundary has an inflow condition $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (\rho_0, M_\infty a_0, 0, p_0)$, and the inviscid flow is solved to $t = 4L/(M_\infty a_0)$ for obtaining a steady state.

In order to simplify the discussion, notation $M_\infty(\hat{M}) - \theta(\hat{\theta}) - \beta_e(\hat{\beta}_e) - \beta_n(\hat{\beta}_n) - G(\hat{G})$ is used to denote a case with Mach number $\hat{M}$, deflection angle $\hat{\theta}$, analytical shock angle $\hat{\beta}_e$, and numerical shock angle $\hat{\beta}_n$ solved on grid $\hat{G}$. It is worth mentioning that the measurement of numerical shock angles is through manually picking up two points at the center of the computed oblique shock line and then computing the slope angle of that line. Therefore, the presented numerical shock angles herein are subject to sampling errors. To better facilitate the comparison of numerical and analytical solutions, straight lines with slope angles being equal to the analytical shock angles are visualized in the figures of results.

A grid sensitivity study is conducted for case $M_\infty(2) - \theta(15°) - \beta_e(45.344°)$ on a series of successively refined grids, and the results on four chosen grids are shown in Fig. 4.4. As the grid resolution increases, the sharpness of the computed shocks improves accordingly, and the perturbations in the wakes are less dissipated. The generation of oscillating wakes in the inviscid flow may be partially due to the numerical viscosity in the numerical schemes. Nonetheless, the predicted oblique shock angles are in excellent agreement with the analytical solution over a wide range of grid resolution, and Prandtl-Meyer expansion waves are also physically resolved at each rear corner of the wedge. Since good numerical accuracy and shock sharpness can be obtained on the 1200×600 grid, this grid is employed for the investigation of other cases.



(a)

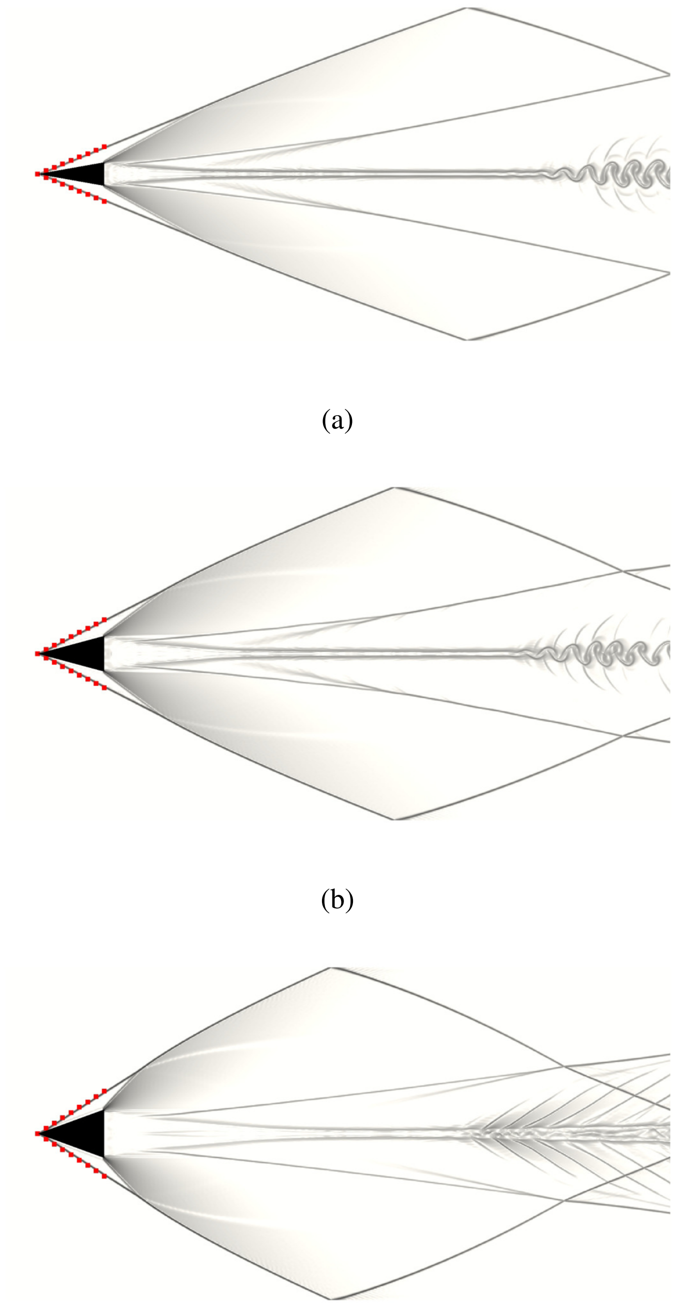
(b)

(c)

(d)

Figure 4.4 Grid sensitivity study of supersonic flow over a wedge for case $M_{\infty}(2) - \theta(15°) - \beta_e(45.344°)$. Lines denoted by square marks represent the analytical solutions. (a) $\beta_n(44.927°) - G(600 \times 300)$. (b) $\beta_n(45.198°) - G(1200 \times 600)$. (c) $\beta_n(45.726°) - G(1800 \times 900)$. (d) $\beta_n(45.352°) - G(2400 \times 1200)$.

The prediction capability on oblique shock relation over different deflection angles and Mach numbers are illustrated in Fig. 4.5 and Fig. 4.6, respectively. Under the same Mach number, a higher deflection angle leads to a higher shock angle. For a fixed deflection angle, as the Mach number increases, the shock angle decreases. Due to the finite length of the wedge, oblique shocks with low shock angles may strongly interfere with the Prandtl-Meyer expansion waves and incline toward the rear corners of the wedge. For the oblique shock angles, the excellent agreement between the numerical and analytical solutions is consistently presented among all the cases, demonstrating the high validity of the proposed method.

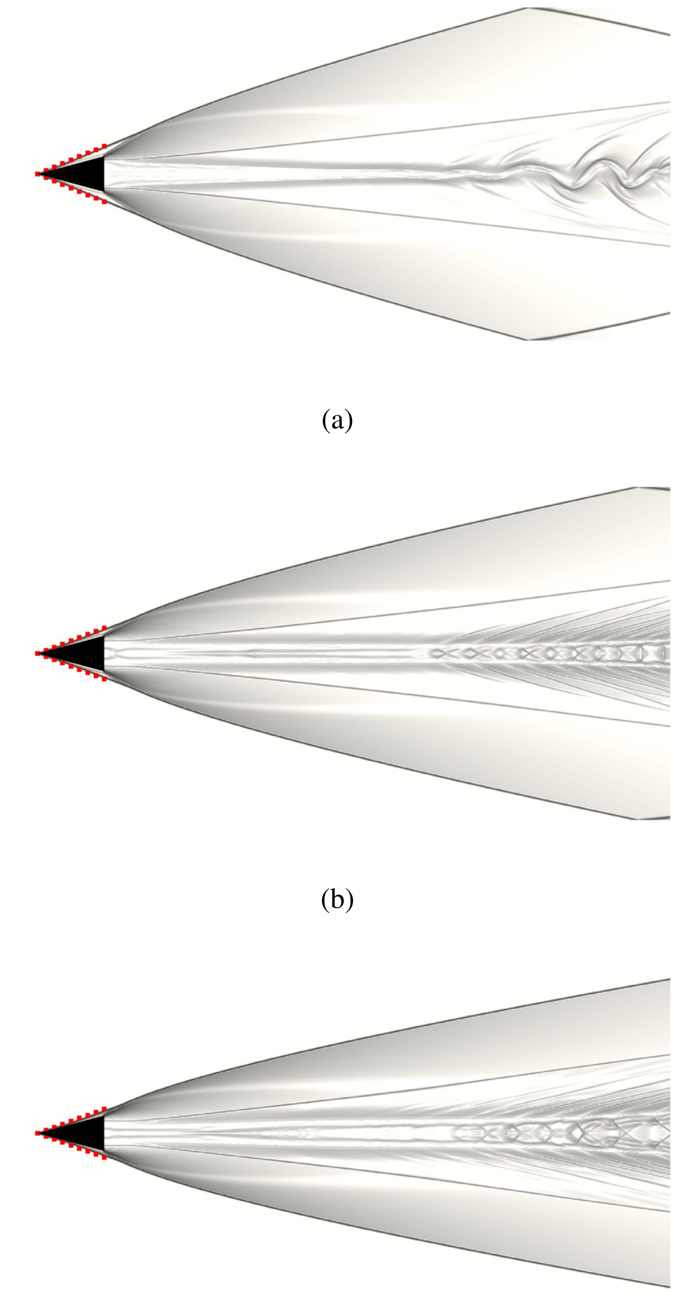


(a)

(b)

(c)

Figure 4.5 Oblique shock relation of supersonic flow over a wedge for case $M_\infty(4) - G(1200 \times 600)$ over different deflection angles. Lines denoted by square marks represent the analytical solutions. (a) $\theta(10°) - \beta_e(22.234°) - \beta_n(22.227°)$. (b) $\theta(15°) - \beta_e(27.063°) - \beta_n(27.325°)$. (c) $\theta(20°) - \beta_e(32.464°) - \beta_n(32.293°)$.



(a)

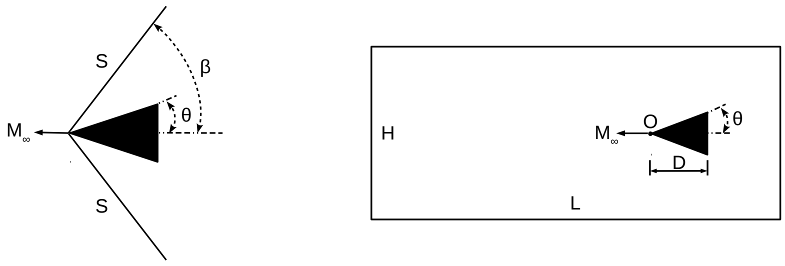
(b)

(c)

Figure 4.6 Oblique shock relation of supersonic flow over a wedge for case $\theta(15°) - G(1200 \times 600)$ over different Mach numbers. Lines denoted by square marks represent the analytical solutions. (a) $M_\infty(6) - \beta_e(22.672°) - \beta_n(22.585°)$. (b) $M_\infty(8) - \beta_e(20.860°) - \beta_n(20.595°)$. (c) $M_\infty(10) - \beta_e(19.942°) - \beta_n(18.930°)$.

4.3.2 Supersonic translating wedge

In order to test the proposed method for solving flow involving moving geometries, the supersonic translating wedge problem is solved, which is a Galilean transformation of the supersonic flow over a wedge problem. As illustrated in Fig. 4.7, in a $L \times H = [-0.5D, 9.5D] \times [-2.5D, 2.5D]$ domain with an initial flow state of $(\rho_0, u_0, v_0, p_0) = (1.4\,\mathrm{kg/m^3}, 0, 0, 400\,\mathrm{Pa})$, in which the speed of sound is $a_0 = 20\,\mathrm{m/s}$, a wedge with $D = 1\,\mathrm{m}$ and $M_\infty = 2$ is initially positioned at $O(8D, 0)$ and has the slip-wall boundary condition. The outflow condition is enforced at the left and right domain boundaries, while the slip-wall condition is imposed at the top and bottom boundaries. To limit the required size of the computational domain, the evolution is solved to $t = 0.8L/(M_\infty a_0)$. Although the most parts of the flow region are still in an unsteady state, and the transient perturbations generated from the sudden motion of the wedge in the initial stationary flow are not advected out of the domain, the oblique shocks are well-developed within the given computational time.



(a) (b)

Figure 4.7 Schematic diagrams for the supersonic translating wedge problem. (a) Oblique shock relation. (b) Computational configuration. [Nomenclature: M_∞, Mach number of the moving wedge; S, oblique shock; θ, deflection angle; β, shock angle; D, length of wedge; O, the front vertex of wedge; L, domain length; H, domain height.] The schematic diagrams are adapted from the work of Anderson Jr [5].

The time evolution of both the supersonic flow over a wedge and the supersonic translating wedge under the condition $M_\infty(2) - \theta(15°) - G(1200 \times 600)$ is captured in Fig. 4.8, in which the dynamic process of oblique shock formation at the wedge nose and Prandtl-Meyer expansion wave generation at the rear corners is clearly depicted.

Main flow features exempted from the extra interpolation error of fresh node treatment, such as the oblique shocks, reflected shocks, and Prandtl-Meyer expansion waves, are well agreed in this Galilean transformation pair. In addition, the predicted wakes are also highly comparable. These well agreed results of this supersonic Galilean transformation pair demonstrate the success of the proposed three-step flow reconstruction scheme for fresh node treatment and the capability of the three-step flow reconstruction-based immersed boundary method herein for solving flow with moving geometries.

To further verify the solution of the supersonic translating wedge problem, a comparison of the numerical and analytical solutions for different deflection angles is shown in Fig. 4.9. The excellent agreement in numerical and analytical solutions demonstrates the validity and accuracy of the method in solving flow with moving geometries.

4.3.3 Shock diffraction over a cylinder

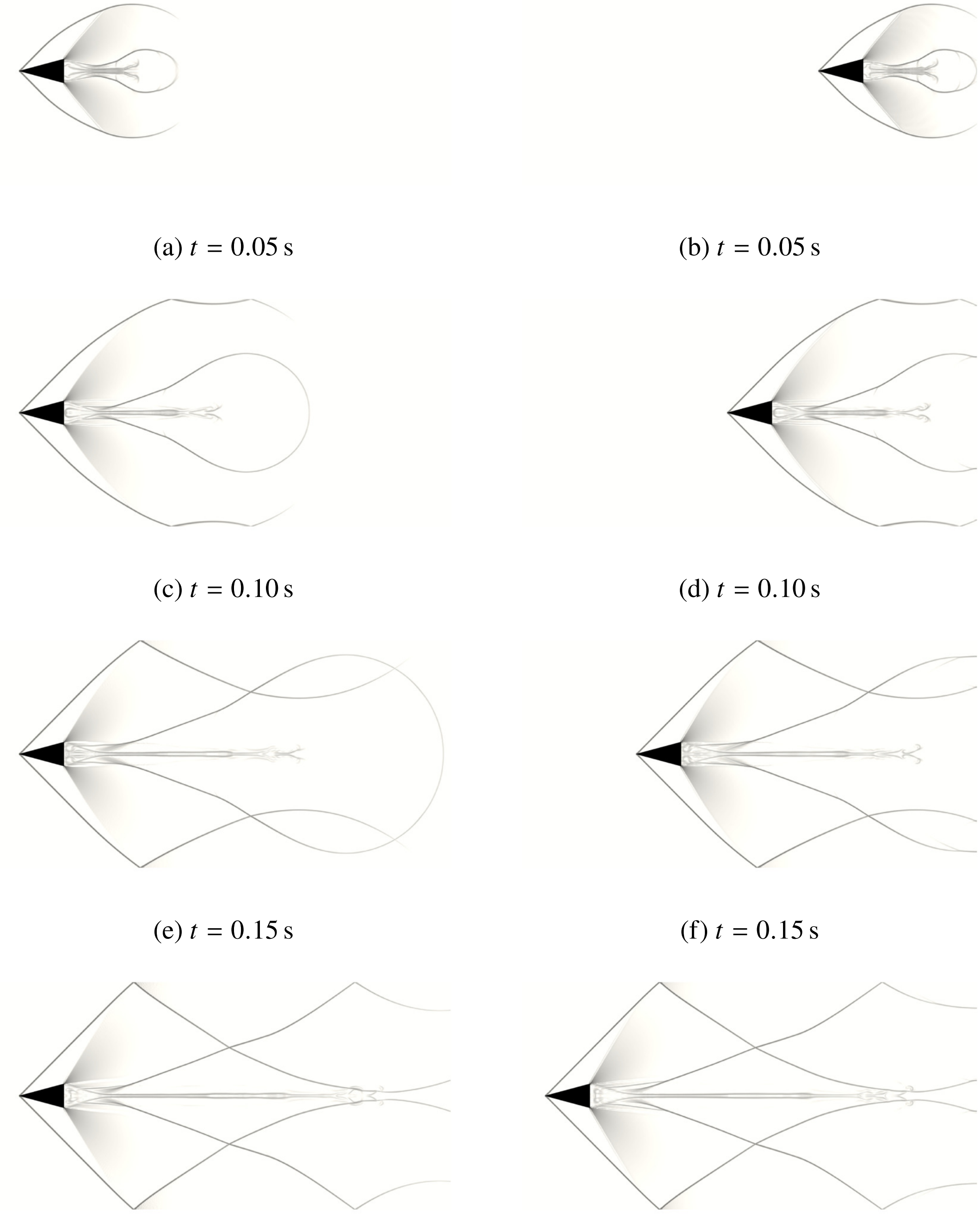
A Mach 2.81 planar shock interacting with a stationary circular cylinder is studied to further evaluate the validity of the developed method. This classical shock diffraction problem has been widely investigated in the literature, including both experimental observations [6, 135, 136] and numerical studies [6, 109, 137].

As a time-dependent process, the interaction between the incident shock and cylinder encompasses complex compressible flow features such as shocks and contact discontinuities. As illustrated in Fig. 4.10, the incident shock initially propagates freely toward the cylinder. Once colliding with the cylinder, the shock reflects as well as diffracts over the convex solid surface with the formation of a curved Mach stem and a slip line at each side of the plane of symmetry. During the evolution, triple points are produced through the intersection of the incident shock, reflected shock, and diffracted shock. At the later stage of evolution, the two diffracting Mach stems collide and form a shock-induced wake at the rear of the cylinder.

In the numerical configuration, a circular cylinder with diameter $D = 1\,\mathrm{m}$ is positioned at the center of a $6D \times 6D$ square domain while an initial shock is positioned $0.5D$ upstream of the cylinder. This computational configuration is similar to Ripley et al. [137] except that a full domain size without symmetric boundary assumption is used herein. The flow is assumed to be inviscid, and the slip-wall boundary condition is enforced at the cylinder surface. The evolution process is solved to $t = 1.0\,\mathrm{s}$.

The computed density contours over a series of grids are shown in Fig. 4.11. An acceptable shock curvature profile can be observed even on the 150×150 grid, which has a grid resolution of about $0.04D$ (25 nodes per diameter). In addition, the plane of symmetry is well preserved over all the employed grids. The numerical Schlieren fields are presented in Fig. 4.12. Compared to the Schlieren photograph measured by Bryson and Gross [135] and the interferometric measurements by Kaca [6] as well as the numerical results in references [109, 137, 138], the slip line, reflected shock, and diffracted shock over the immersed boundary are already resolved very well on the 600×600 grid, which illustrates the high accuracy of the developed method.

As illustrated in Fig. 4.10b, the intersection of the incident shock, reflected shock, and diffracted shock forms a triple point. During the time-dependent evolution process, this



(a) $t = 0.05\,\mathrm{s}$

(b) $t = 0.05\,\mathrm{s}$

(c) $t = 0.10\,\mathrm{s}$

(d) $t = 0.10\,\mathrm{s}$

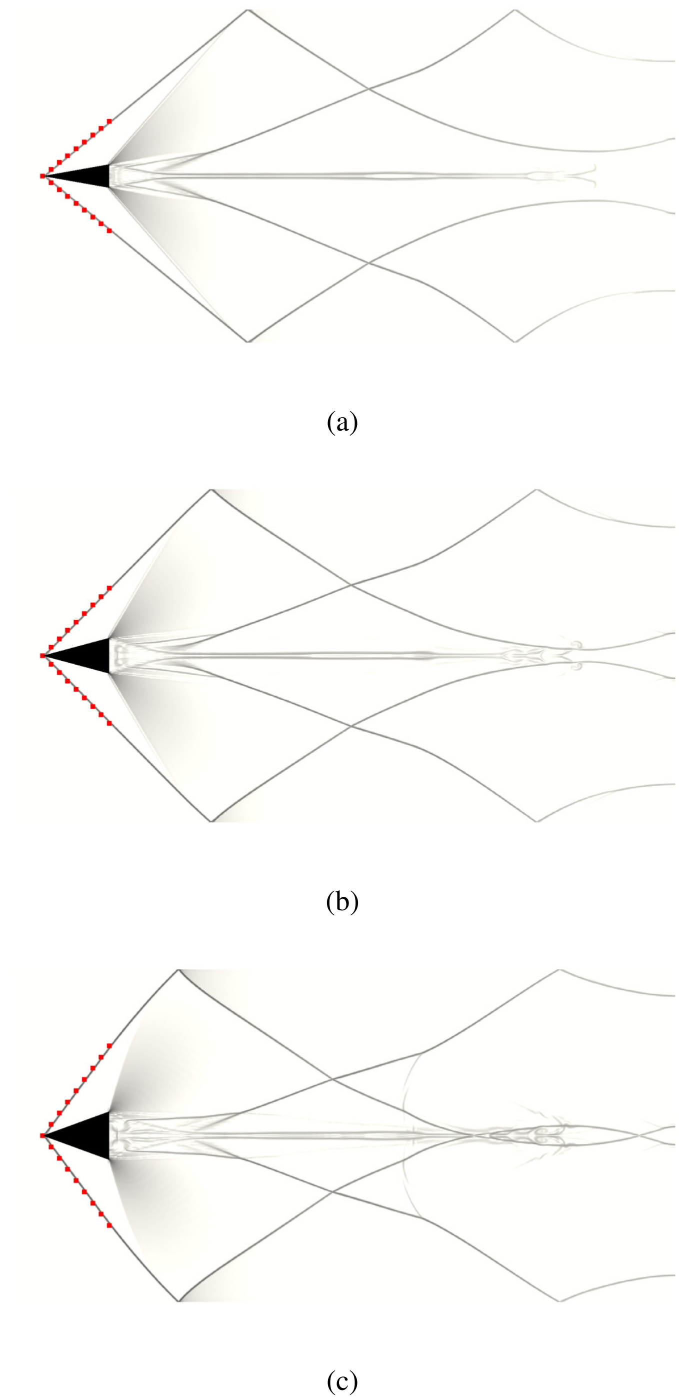
(e) $t = 0.15\,\mathrm{s}$

(f) $t = 0.15\,\mathrm{s}$

(g) $t = 0.20\,\mathrm{s}$

(h) $t = 0.20\,\mathrm{s}$

Figure 4.8 Time evolution of solution for case $M_\infty(2) - \theta(15°) - G(1200 \times 600)$. (a), (c), (e), (g) The supersonic flow over a wedge problem. (b), (d), (f), (h) The supersonic translating wedge problem.



(a)

(b)

(c)

Figure 4.9 Oblique shock relation of supersonic translating wedge for case $M_{\infty}(2) - G(1200 \times 600)$ over different deflection angles. Lines denoted by square marks represent the analytical solutions. (a) $\theta(10°) - \beta_e(39.314°) - \beta_n(39.313°)$. (b) $\theta(15°) - \beta_e(45.344°) - \beta_n(45.034°)$. (c) $\theta(20°) - \beta_e(53.423°) - \beta_n(53.425°)$.

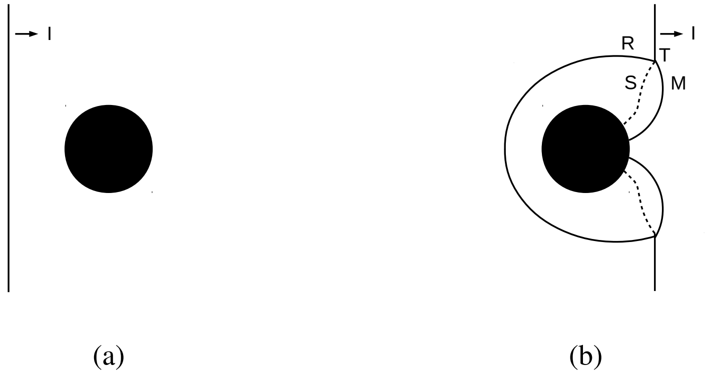


Figure 4.10 Schematic diagrams for a planar shock interacting with a stationary circular cylinder. (a) Initial state. (b) Well-developed diffraction. [Nomenclature: I, incident shock; R, reflected shock; M, Mach stem (diffracted shock); S, slip line (contact discontinuity); T, triple point.] The schematic diagrams are adapted from the work of Kaca [6].

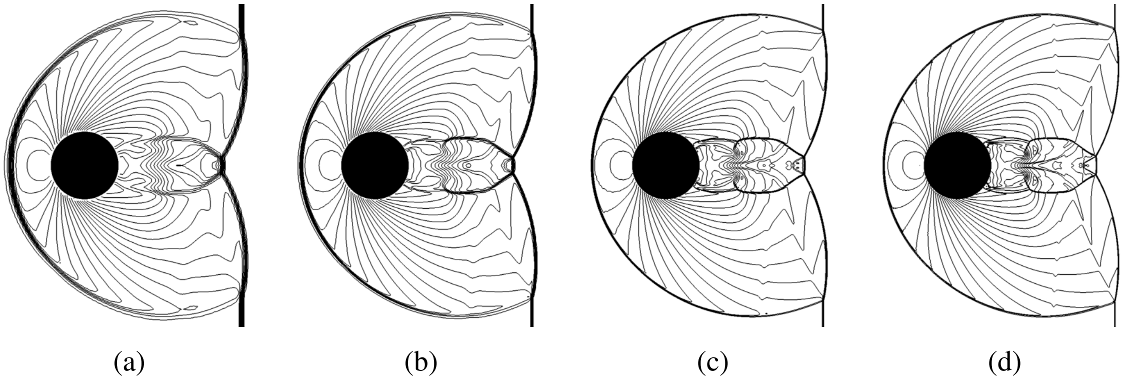


Figure 4.11 Density contour of shock diffraction over a cylinder solved on different grid sizes. (a) 150×150. (b) 300×300. (c) 600×600. (d) 1200×1200.

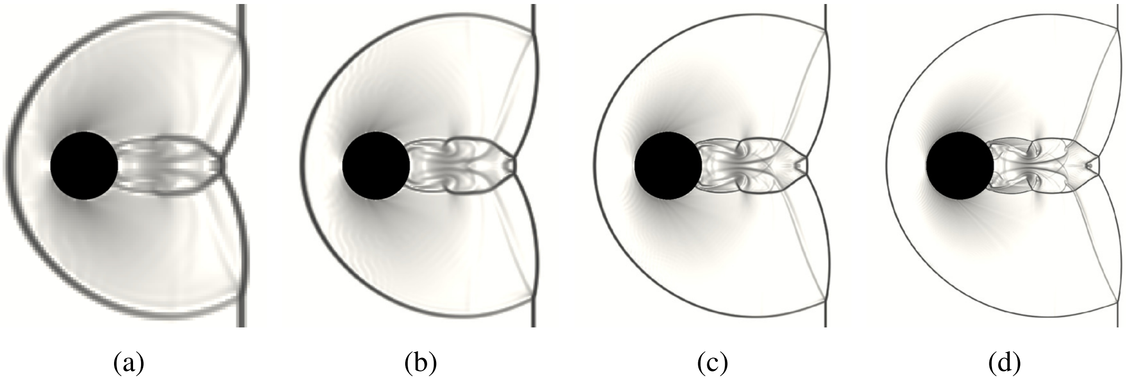


Figure 4.12 Numerical Schlieren of shock diffraction over a cylinder solved on different grid sizes. (a) 150×150. (b) 300×300. (c) 600×600. (d) 1200×1200.

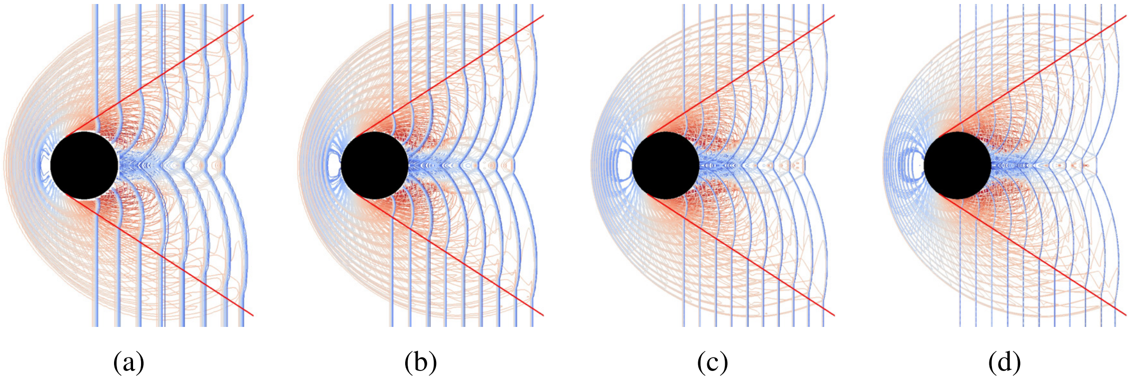


Figure 4.13 Superimposition of density contours showing the predicted propagation path of the triple point (The two straight lines are the 33° tangent lines of the cylinder). (a) 150 × 150. (b) 300 × 300. (c) 600 × 600. (d) 1200 × 1200.

triple point travels in space and produces a triple-point path, as captured in Fig. 4.13. The interferometric measurements of Kaca [6] predict that this triple-point path is tangent to the cylinder at an angle of 33° for Mach numbers in the range of $1.42 - 5.96$.

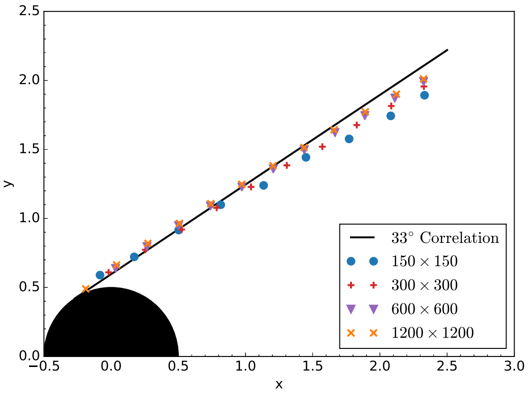


Figure 4.14 Comparison of the predicted triple-point paths with experimental correlation.

The predicted triple-point paths are extracted and plotted in Fig. 4.14. The least squares linear regressions of the predicted triple-point paths on the grids of 150 × 150, 300 × 300, 600 × 600, and 1200 × 1200 nodes are about 28.2°, 29.8°, 30.3°, and 30.9°, respectively. These results, which agree well with the experimental correlation of Kaca [6] and very well with the polynomial reconstruction-based results of Sambasivan and Udaykumar [109], cut-cell method-based results of Ji et al. [138], and unstructured mesh-based results of Ripley et al. [137], further demonstrate the validity of the developed method.

4.3.4 A modified vortex preservation problem

One of the challenges in testing the order of accuracy for an immersed boundary method is the lack of analytically solvable smooth flow with irregular geometries. In order to partially address this challenge, a modified vortex preservation problem adapted from the reference [139] is developed. As shown in Fig. 4.15, in a $L \times H = [-2.5R, 2.5R] \times [-R, R]$ domain with a uniform inviscid background flow $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (1, 1, 1, 1)$, an isentropic vortex initially centered at $O(0, 0)$ with radius $R = 5$ is created by adding the following perturbation to the mean flow

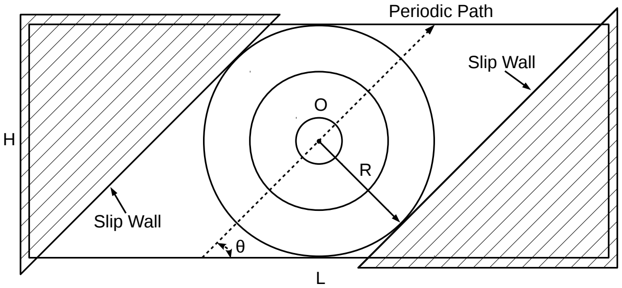


Figure 4.15 A schematic diagram for the modified vortex preservation problem. [Nomenclature: O, vortex center; R, vortex radius; θ, path angle; L, domain length; H, domain height.]

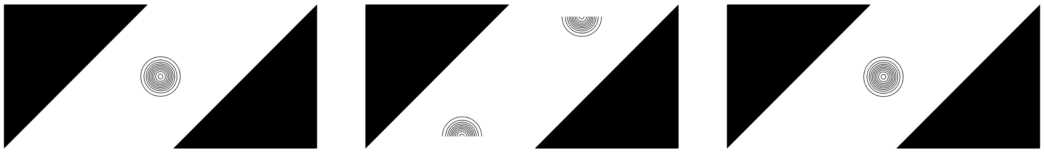
$$
\begin{aligned}
(\delta u, \delta v) &= \frac{\Gamma}{2\pi} e^{0.5(1-r^2)}(-y, x) \\
\delta T &= -\frac{(\gamma - 1)\Gamma^2}{8\gamma\pi^2} e^{(1-r^2)}
\end{aligned}
\tag{4.24}
$$

where the vortex strength $\Gamma = 5$, the ratio of specific heat $\gamma = 1.4$, $r^2 = x^2 + y^2 \le R^2$. Under the isentropic flow condition, the temperature and entropy are defined as $T = p/\rho$ and $s = p/\rho^\gamma$, respectively. When $s = 1$ and $\delta s = 0$ are assumed for the flow, the initial conditions are then taken as follows

$$
u = u_\infty + \delta u, \; v = v_\infty + \delta v, \; T = p_\infty/\rho_\infty + \delta T, \; \rho = T^{\frac{1}{\gamma-1}}, p = T^{\frac{\gamma}{\gamma-1}} \tag{4.25}
$$

Since the perturbation induced by the vortex is weak enough to avoid producing a strongly nonlinear effect, the exact solution of the problem with a specific initial state is the passive convection of the vortex with the mean velocity [139].

In order to incorporate the effects of boundary treatment, two relatively large triangles are positioned at the two horizontal ends of the domain to shape the computational domain into a parallelogram with the length of each edge being $2\sqrt{2}R$. The slip wall boundary condition is enforced on the triangles. The number of mesh cells for each direction is chosen to ensure that $(H/m_y)/(L/m_x) \neq \tan\theta$. Therefore, the two slip walls are not aligned with the grid to manifest the effects of immersed boundary treatment. In addition, each slip wall is tangent to the vortex edge to further manifest the boundary effects. Convected by the chosen background flow, the vortex will propagate at $\theta = 45°$ while slipping along with the two slip walls. When the periodic boundary condition with translation path $(2R, 2R)$ is enforced on the two vertical ends of the domain, and the flow is solved to $t = 10$, the solution should remain unchanged as the time evolves to a complete period of vortex convection, as demonstrated by the computed results shown in Fig. 4.16.



(a) $t = 0$ (b) $t = 5$ (c) $t = 10$

Figure 4.16 Density contour of the propagating isentropic vortex solved on a 320×160 grid.

In order to better examine the order of accuracy of boundary treatment via spatial convergence behavior, the discretization of convective fluxes for this problem is switched to the second-order central difference scheme. The third-order Runge–Kutta scheme with a small CFL coefficient $C_{\text{CFL}} = 0.2$ is used to reduce the influence from temporal discretization errors. The global spatial convergence of the solutions over successively refined grids is shown in Table 4.2. As captured by the results, the solution algorithm is operating closely to its designed order of accuracy measured in L_1, L_2, and L_∞ norms. As the solution of this problem involves the adiabatic condition for the temperature field, zero normal gradient condition for the pressure field, and the slip wall condition, the most complicated type of boundary conditions, for the velocity field, the developed immersed boundary method can therefore preserve the designed second-order accuracy for solving flow with irregular geometries and complex boundary conditions.

$m_x \times m_y$	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
40×20	3.536e−2	–	6.097e−2	–	4.105e−1	–
80×40	9.113e−3	1.956	2.497e−2	1.288	1.997e−1	1.039
160×80	2.034e−3	2.163	6.548e−3	1.931	5.236e−2	1.931
320×160	5.114e−4	1.992	1.640e−3	1.997	1.278e−2	2.035
640×320	1.287e−4	1.990	4.097e−4	2.001	3.119e−3	2.034
1280×640	3.233e−5	1.993	1.024e−4	2.000	7.818e−4	1.996

Table 4.2 Global solution error and convergence rate for the modified vortex preservation problem.

4.3.5 Mass flux examination

Due to using non-body conformal Cartesian grids, mass flux over immersed boundary is a fundamental issue in immersed boundary methods [60], and efforts such as adopting cut-cell approaches have been devoted in existing studies [140–142] in an attempt to alleviate this issue. This section examines the mass flux produced by the proposed method herein on generic Cartesian grids with practical grid sizes.

The streamlines of a Mach 2.81 shock diffracting over different types of particles are presented in Fig. 4.17. The solved streamlines by the developed immersed boundary method are closely aligned with the geometry surfaces, even in the three-dimensional problem where a coarse grid is employed.

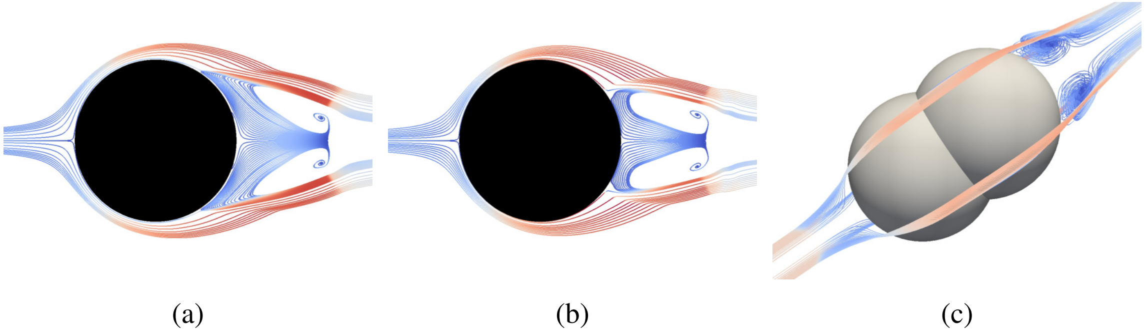


Figure 4.17 Streamlines of a Mach 2.81 shock diffracting over different types of particles with corresponding analytical geometry boundaries presented. (a) Shock diffraction over a cylinder solved on a $6D \times 6D$ domain discretized by a 600×600 grid, no-slip wall. (b) Shock diffraction over a cylinder solved on a $6D \times 6D$ domain discretized by a 600×600 grid, slip wall. (c) Shock diffraction over two partially overlapped spheres solved on a $6D \times 6D \times 6D$ domain discretized by a $250 \times 250 \times 250$ grid, no-slip wall.

For the purpose of quantitatively examining the unphysical flux, the surface-normalized absolute flux over the immersed boundary

$$f_{\text{ibm}} = \frac{1}{S} \int_S |(\boldsymbol{V} - \boldsymbol{V}_S) \cdot \mathbf{n}| \, \mathrm{d}S \tag{4.26}$$

or in a discrete form

$$f_{\text{ibm}} = \frac{1}{N} \sum_{n=1}^{N} |(\boldsymbol{V}_n - \boldsymbol{V}_S) \cdot \mathbf{n}_n| \tag{4.27}$$

is employed as a quantitative measure, in which, N is the number of the first layer ghost nodes, $\boldsymbol{V}$ is the flow velocity at the ghost node, $\boldsymbol{V}_S$ is the velocity of the geometry, $\mathbf{n}$ is the local unit outward surface normal vector.

The proposed f_{ibm} measures the flux at about one grid distance away from the physical boundary without involving interpolation and will overestimate the flux through the geometry. Therefore, f_{ibm} can be a reliable measure of the unphysical flux and hence the interface sharpness.

Table 4.3 The surface-normalized absolute flux f_{ibm} for the supersonic flow over a wedge problem.

Grid	f_{ibm} (m/s)	$f_{\mathrm{ibm}}/(M_\infty a)$
600×300	1.825e−1	0.456%
900×450	1.213e−1	0.303%
1200×600	9.126e−2	0.228%
1800×900	6.099e−2	0.152%
2400×1200	4.557e−2	0.114%

As a problem involves a steady supersonic flow passing a relatively strong convex geometry with the intricate slip-wall boundary condition, the f_{ibm} of the supersonic flow over a wedge problem for the case $M_\infty(2) - \theta(15°) - \beta_e(45.344°)$ is examined and is shown in Table 4.3. For a wide range of grids under the presence of complex shock interactions near the geometry boundary, the proposed method generates very low surface-normalized absolute fluxes over the immersed boundary. For instance, the value of $f_{\mathrm{ibm}}/(M_\infty a_0)$ is about 0.456% for the 600×300 grid. According to the discussed qualitative and quantitative results, the developed immersed boundary method herein retains a very sharp interface and is able to effectively alleviate unphysical flux over physical boundaries when grid resolution is improved.

4.3.6 Subsonic rotational flow

A subsonic rotational flow generated by an accelerating rotor is solved to demonstrate the applicability of the method for fluid-solid systems involving complex geometries. As shown in Fig. 4.18a, the 2D rotor consists of three blades with each blade being the shape of NACA 0012 airfoil, whose chord is $l_c = 1$ m. The rotor is centered in a $6l_c \times 6l_c$ domain discretized by a 1200×1200 grid. The rotor rotates with an initial angular velocity $\omega(t = 0) = 0\,\mathrm{rad/s}$ and a constant angular acceleration $\alpha = 10\pi\,\mathrm{rad/s^2}$. The initial ambient flow state is $(\rho_0, u_0, v_0, p_0) = (1.2047\,\mathrm{kg/m^3}, 0, 0, 101325\,\mathrm{Pa})$, and the no-slip wall boundary condition is enforced on the blades.

The predicted vorticity isocontour at a series of time instants is captured in Fig. 4.18, in which an interesting vortex-induced vortex shedding behavior is observed. As the rotor accelerates, vortices appear at the tips of blades as well as at the sides of the blades. The accelerating tips stretch and deform the generated vortices, causing vortices shedding. The detached vortices soon are entrained by the flow driven by the coming blade, either being merged into the tip vortices of the coming blade or being advected to the front side vortices of the coming blade. The entrained vortices then destabilize the vortex structures they propagate toward, inducing and accelerating the vortex shedding. The above interaction process, combining with the acceleration of the blades, produces a complex and dynamic vortex field showing in the figures.

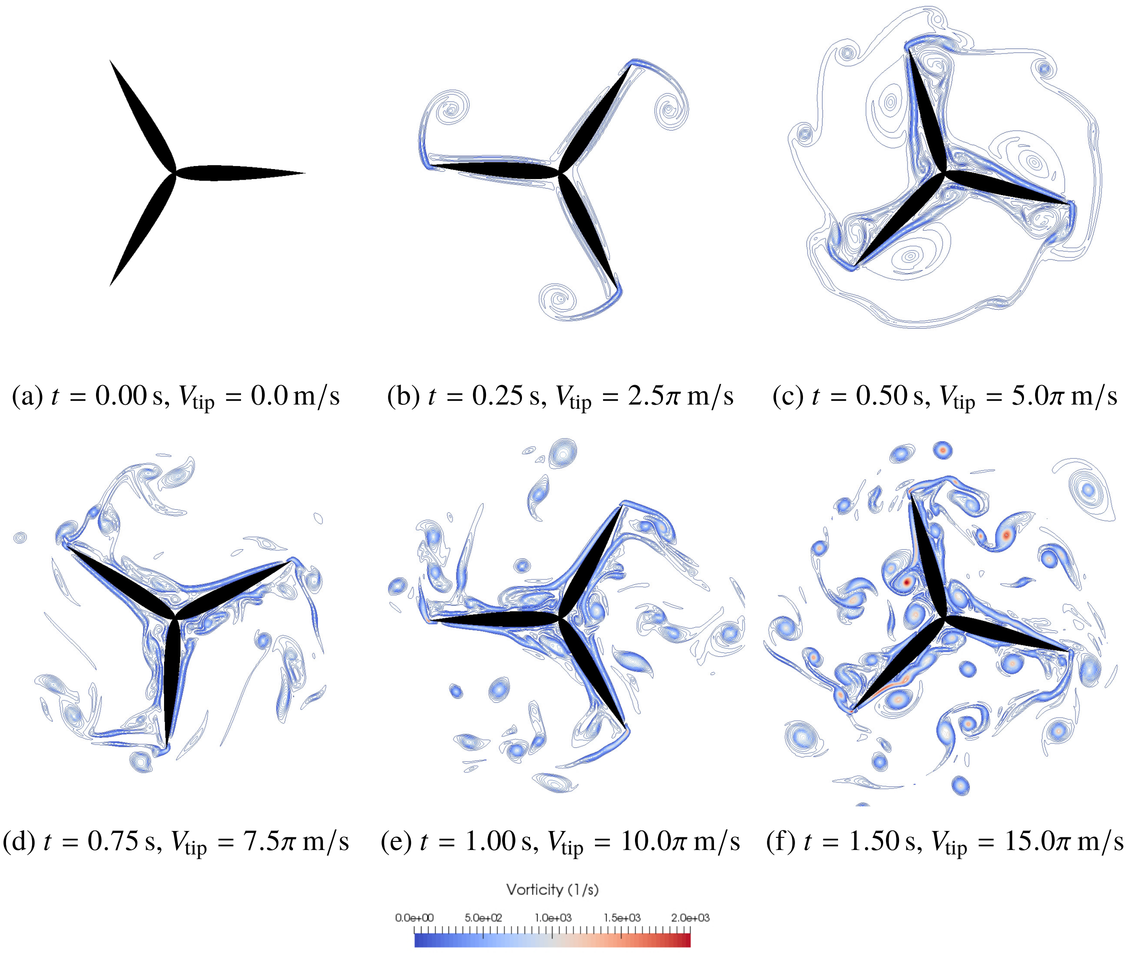
In the employed case setting, the geometry of the rotor is generated by positioning three independent NACA 0012 airfoils at a common center. The rotating system is obtained by specifying the same rotational acceleration for the three blades. Therefore, in simulating flow with complex geometries, employing a robust immersed boundary method enables great simplification for the model generation process. For instance, one can obtain an engineering structure by assembling its components via positioning while allowing overlapped surfaces, which avoids the cost of producing a single topology, as required by many mesh generators such as ANSYS ICEM CFD. In addition, the current grid resolution of the domain is 200 nodes per l_c. Meanwhile, the largest width of the blade is only $0.12l_c$, and the tip of blade is very thin. Consequently, there is only a line of nodes in the region near the tip, whose length is about 5% of the chord. Nonetheless, the proposed method resolves the thin tip region with reasonable sharpness. The overall success of simulating the transient rotational flow illustrates the capability of the presented method for solving general fluid-solid systems.

4.3.7 Explosive dispersal of dense particles

To further demonstrate the robustness of the presented method for solving problems with strongly irregular, moving, and colliding geometries under challenging flow conditions, a dense particle system dispersed by a high-density and high-pressure gas driver is studied.

As shown in Fig. 4.19, in a $L^3 = [-0.5\,\mathrm{m}, 0.5\,\mathrm{m}]^3$ computational domain, a spherical five-layer particulate payload with 2130 particles is initially centered at $O(0, 0)$. The radius of the 0° latitude of the inner-most layer is 0.1 m, and the number of particles centered on this 0° latitude is 24. The entire particulate payload is then generated through varying the radius of particles in each layer such that neighboring particles in each spherical layer are tangent with each other, and neighboring particles in neighboring layers are also tangent with each other as much as possible. A flow state $(\rho_c, u_c, v_c, p_c) = (1000\,\mathrm{kg/m^3}, 0, 0, 1.01325\times10^6\,\mathrm{Pa})$ is initially filled in the center of the particulate payload. The flow state at the rest of the region is set to $(\rho_0, u_0, v_0, p_0) = (1.2047\,\mathrm{kg/m^3}, 0, 0, 1.01325 \times 10^5\,\mathrm{Pa})$. In order to reduce the discrepancy of the timescales between shock propagation and particle acceleration, relatively light particles with a density of $27\,\mathrm{kg/m^3}$ are used.

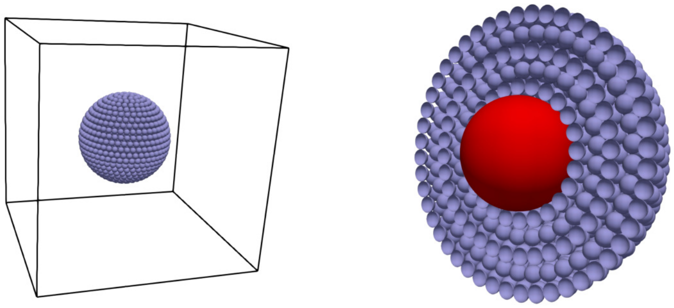
This fluid-solid system is solved to $t = 7\,\mathrm{ms}$ on a 800^3 grid. In the development of the flow solver, one design principle is to store a minimum set of information and to calculate derived quantities on-the-fly. For instance, only five conservative variables at three time



(a) $t = 0.00\,\mathrm{s}$, $V_{\mathrm{tip}} = 0.0\,\mathrm{m/s}$ (b) $t = 0.25\,\mathrm{s}$, $V_{\mathrm{tip}} = 2.5\pi\,\mathrm{m/s}$ (c) $t = 0.50\,\mathrm{s}$, $V_{\mathrm{tip}} = 5.0\pi\,\mathrm{m/s}$

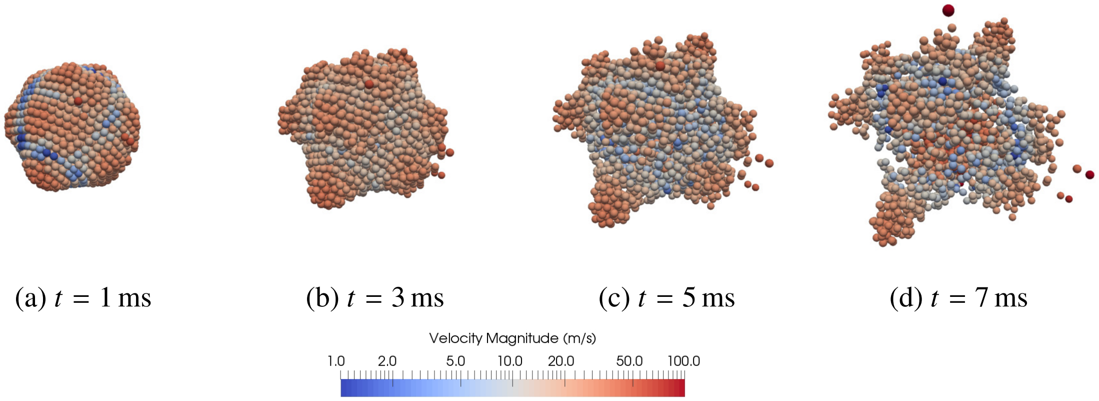
(d) $t = 0.75\,\mathrm{s}$, $V_{\mathrm{tip}} = 7.5\pi\,\mathrm{m/s}$ (e) $t = 1.00\,\mathrm{s}$, $V_{\mathrm{tip}} = 10.0\pi\,\mathrm{m/s}$ (f) $t = 1.50\,\mathrm{s}$, $V_{\mathrm{tip}} = 15.0\pi\,\mathrm{m/s}$

Figure 4.18 Vorticity isocontour at a series of time instants generated by an accelerating rotor. [Nomenclature: V_{tip}, velocity magnitude of the tip of a blade.]



(a) (b)

Figure 4.19 Configuration for explosive dispersal of dense particles. (a) Computational domain. (b) A clip view of the particulate payload.



(a) $t = 1\,\mathrm{ms}$ (b) $t = 3\,\mathrm{ms}$ (c) $t = 5\,\mathrm{ms}$ (d) $t = 7\,\mathrm{ms}$

Figure 4.20 The computed dispersal process of the particle system.

levels are stored for the time integration of the governing equations through a third-order Runge–Kutta scheme, and all the other quantities such as convective fluxes are calculated in progress.

In addition to fifteen float variables, four integers related to multidomain node mapping, ghost node identification, fresh node tracking, and the closest facet linking, respectively, are stored. The purpose of the closest facet linking variable is also to reduce the stored information set. Implementing an immersed boundary method at a ghost node G requires a set of information at the boundary point O such as coordinates, surface velocity, normal and tangent vectors. Instead of storing multiple expensive floats, using an integer variable to link each ghost node G to its closest geometric facet, which is a by-product of the point-in-polyhedron test conducted at the preprocessing step, can save a significant amount of memory while being efficient to derive all the required information through on-the-fly point-to-facet calculations. Furthermore, in order to maintain data locality, a compound data type is used to pack all the float and integer field variables for each node. Using the defined compound data type as element data type, a single linear array is used as the main data structure for improving cache performance. As a result of using a Cartesian grid and the described design principles, the actual runtime memory allocation of this problem with double-precision computation is about 67 gigabytes, while the theoretical requirement of storing fifteen floats and four integers for $807^3 \approx 5.255 \times 10^8$ nodes taking account for the global numerical boundaries is 66.567 gigabytes.

The computed dispersal process of the particle system is presented in Fig. 4.20, which depicts the development of instabilities in the particulate payload under explosive dispersal and the formation of coherent particle clustering and jetting structures. In the solution process, a significant number of convex and concave regions exist, form, and destroy. These successful solutions of the explosive dispersal of a densely packed particle system further demonstrate that the proposed flow reconstruction scheme effectively avoids potential numerical singularities and preserves high robustness in solving flow with strongly irregular, moving, and colliding geometries.

4.4 Conclusion

An immersed boundary method has been developed, validated, and applied. The effectiveness of the method in solving flow with arbitrarily irregular and moving geometries on Cartesian grids has been illustrated through numerical experiments concerning a variety of flow problems. The main properties of the presented method and the primary conclusions from the numerical experiments are summarized below.

Convergence and accuracy The accuracy of the method is established through thorough studies of the supersonic flow over a wedge problem, the supersonic translating wedge problem, and the shock diffraction over a cylinder problem. Employing the analytical $M_\infty - \theta - \beta$ relation of oblique shocks, different cases considering the deflection angle ranging from $\theta = 10°$ to $\theta = 20°$ and the Mach number ranging from $M_\infty = 2$ to $M_\infty = 10$ are tested, and excellent agreement between the numerical and analytical solutions is obtained. In addition, the accuracy of the method for solving flow with moving geometry is demonstrated through comparing the oblique shocks with analytical solutions as well as via examining the entire solution field using Galilean transformation. For the shock diffraction over a cylinder problem, good agreement between the obtained numerical results and experimental observations as well as other published numerical results is achieved. The successful solutions of these test cases demonstrate the validity and accuracy of the proposed method in solving flow involving irregular and moving geometries under challenging flow conditions. In addition, an analytically solvable modified vortex preservation problem has been developed for a convergence study concerning smooth flow with irregular geometries, and the convergence test in L_1, L_2, and L_∞ norms suggests that the developed method can preserve the designed second-order accuracy for solving flow with irregular geometries and complex boundary conditions.

The incorporation of physical boundary conditions in the proposed three-step flow reconstruction scheme leads to the property that the constructed ψ_G converges to the exact physical boundary conditions when the ghost node G converges to the boundary point O. For non-body conformal Cartesian grids, this property is helpful in alleviating the unphysical flux over immersed boundaries. Two- and three-dimensional streamlines of shock-particle interactions with slip and no-slip boundary conditions have qualitatively illustrated that the developed method maintains a very sharp interface. Through an examination of the surface-normalized absolute flux of the supersonic flow over a wedge problem, it has been quantitatively shown that, under the presence of a relatively strong convex geometry with the intricate slip-wall boundary condition and complex shock interactions near immersed boundaries, the developed method produces surface-normalized absolute flux in very low quantities for a wide range of grids.

Uniformity and efficiency The proposed method enforces the Dirichlet, Neumann, Robin, and Cauchy boundary conditions in a straightforward and consistent manner and completely avoids the necessity to solve linear systems. As a result, an arbitrary number of field variables that satisfy different types of boundary conditions, such as velocity, pressure, and temperature, can be efficiently and uniformly treated. In addition, the proposed method easily facilitates the development of a solver that unifies 1D, 2D, and 3D computations as well as the generation of a complex geometric object through assembling its multiple components via positioning while allowing overlapped surfaces to avoid the cost of producing a single topology. The uniformity of the method has been illustrated via the solution of two- and three-dimensional flow problems with no-slip and slip wall boundary conditions. Benefiting from the use of generic Cartesian grids and the linear-time algorithm complexity as well as the matrix inversion-free property of the proposed immersed boundary method, except the explosive dispersal of dense particle problem, all the other presented test cases herein were solved using a single processor.

Robustness and stability The proposed method employs a three-step flow reconstruction scheme that is scalable to the number of stencil nodes and is uniformly valid under a varying number of stencils, even in the worst situation, in which only one fluid node exists in the domain of dependence of an image point. In addition, as demonstrated by the implementation of the 7-point-stencil WENO scheme, which requires 3 ghost node layers, the method can be applied to multiple layers of ghost nodes without imposing extra constraints. This property can greatly facilitate the application of high-order spatial schemes to flow with complex geometries.

Utilizing the three-step reconstruction as well as the convex and extrema-preserving properties of the inverse distance weighting, the proposed method presents strong numerical stability, as demonstrated in the numerical experiments involving challenging flow conditions and dynamic geometries, such as the supersonic translating wedge flow, the subsonic rotational flow, and the explosive dispersal of dense particles. Equipped with suitable discretization schemes, the developed immersed boundary method enables feasible solutions of problems with engineering level of complexity and hence enhances the understanding of physical problems.

Chapter 5

Mesoscale Study on Particle Cluster and Jet Formation

5.1 Introduction

Macroscale coherent particle clustering and jetting structures can be frequently exhibited in explosively dispersed granular materials. Influencing the mass concentration and related particle reaction and energy release, this particle clustering and jetting phenomenon is important to study flow instability and turbulent mixing in heterogeneous detonation and explosion. Primarily due to the complex mesoscale multiphase interactions involved in the dispersal process, the underlying physical mechanism remains unclear. In this chapter, employing the developed framework, direct simulations that capture coupled multiphase interactions and deterministic mesoscale granular dynamics are conducted to investigate particle clustering and jetting instabilities. A random sampling algorithm is employed to generate stochastic payload morphologies with randomly distributed particle positions and sizes. Through solving and analyzing cases that cover a set of stochastic payloads, burster states, and coefficients of restitution, a valid statistical dissipative property of the framework in solving explosively dispersed granular materials with respect to Gurney velocity is demonstrated. The predicted surface expansion velocities can extend the time range of the velocity scaling law with regard to Gurney energy in the Gurney theory from the steady-state termination phase to the unsteady evolution phase. In addition, a dissipation analysis of the current discrete modeling of granular payloads suggests that incorporating the effects of porosity can enhance the prediction of Gurney velocity for explosively dispersed granular payloads. On the basis of direct simulations, an explanation for particle clustering and jetting instabilities is proposed to increase the understanding of established experimental observations in the literature.

5.2 Payload configuration with stochastic morphology

As illustrated in Fig. 5.1, a 2D stratified burster-payload configuration with a cylindrical charge shape is considered. The computational domain is a square region of length l and is characterized by three sub-domains: the burster region Ω_{B}, payload region Ω_{L}, and ambient air region Ω_{A}. The flow states at the payload and ambient air regions are set to $(\rho_a, u_a, v_a, p_a) = (1.2047\,\mathrm{kg/m^3}, 0, 0, 1.01325 \times 10^5\,\mathrm{Pa})$. Since burster detonation happens before payload expansion, and the timescale of the former can be much smaller than that of the latter, the burster in initially condensed phase is approximated by a reacted gas product with constant volume. The initial burster considers three different states as $\mathrm{B}_j : (\rho_b, u_b, v_b, p_b) = (1.0 \times 10^3\,\mathrm{kg/m^3}, 0, 0, p_j)$, where $j = 1, \ldots, 3$, and $p_1 = 0.5 \times 10^5 p_a$, $p_2 = 1.0 \times 10^5 p_a$, and $p_3 = 2.5 \times 10^5 p_a$ with p_a being the pressure of the ambient air.

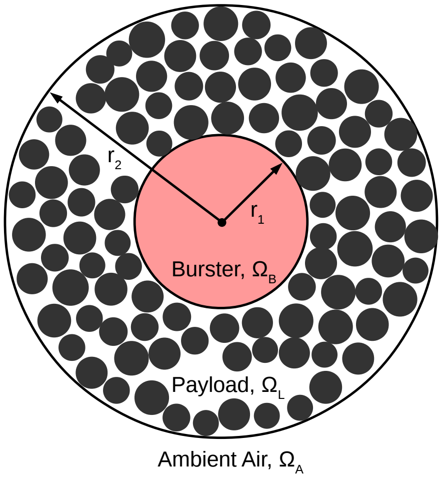


Figure 5.1 A schematic diagram of a 2D stratified burster-payload configuration with a cylindrical charge shape.

In order to focus on granular dynamics, payload casings are omitted, and the payload consists of tightly-packed, inert, rigid, and circular particles. The density of particle is $\rho_s = 2700\,\mathrm{kg/m^3}$. In the current study, the coefficient of restitution C_{R} considers two constant states C_k, $k = 1, 2$, with $\mathrm{C}_1 = 1.0$ for elastic pairwise collision and $\mathrm{C}_2 = 0.5$ for a representative inelastic pairwise collision.

In this study, the morphology of the granular payload that occupies the region Ω_{L} is characterized by the inner radius r_1, outer radius r_2, number of particles n, particle position variable (x, y), particle diameter variable d, particle volume fraction ϕ, and payload to burster mass ratio M/C. For the generation of stochastic payload morphology, the particle position (x, y) and diameter d are considered as the primary random variables.

A random variable X with a continuous univariate distribution can be described by a probability density function (PDF) $f(x)$ such that

$$\mathrm{P}(x_1 \leq X \leq x_2) = \int_{x_1}^{x_2} f(x)\,\mathrm{d}x \tag{5.1}$$

where $\mathrm{P}(x_1 \leq X \leq x_2)$ represents the probability that the random variable X satisfies $x_1 \leq X \leq x_2$, and $f(x)$ is a non-negative function that satisfies the normalization condition

$$\mathrm{P}(-\infty < X < +\infty) = \int_{-\infty}^{+\infty} f(x)\,\mathrm{d}x = 1 \tag{5.2}$$

Subsequently, the probability of $X \leq x$ can be measured by a cumulative distribution function (CDF) $F(x)$:

$$F(x) = \mathrm{P}(X \leq x) = \int_{-\infty}^{x} f(\xi)\,\mathrm{d}\xi \tag{5.3}$$

A random number generator usually produces a uniformly distributed value u_i in $[0, 1]$ for each trial i. To generate a value x_i distributed as a specific $f(x)$ via a random number generator, the inverse transform sampling method [143] that connects x_i with u_i is used:

$$x_i = F^{-1}(u_i) \tag{5.4}$$

in which $F^{-1}(u)$ is the inverse function of $F(x)$.

In the region Ω_L, the particle position (x, y) can be projected to the (r, θ) space, in which r and θ are radius and angle, respectively. As the perimeter of a circle is a linear function of the radius, a random sampling of particle positions with a uniform distribution in Ω_L requires that the PDF of radius r, $f_r(r)$, is a linear function on the support domain $[r_1, r_2]$. That is,

$$f_r(r) = \frac{1}{\int_{r_1}^{r_2} \xi\,\mathrm{d}\xi} r = \frac{2}{r_2^2 - r_1^2} r \tag{5.5}$$

Then, the CDF of radius r has

$$F_r(r) = \frac{2}{r_2^2 - r_1^2} \int_{r_1}^{r} \xi\,\mathrm{d}\xi = \frac{r^2 - r_1^2}{r_2^2 - r_1^2} \tag{5.6}$$

Correspondingly, the inverse CDF of radius r is

$$F_r^{-1}(u) = \sqrt{u(r_2^2 - r_1^2) + r_1^2} \tag{5.7}$$

Therefore, a random sampling of particle positions with a uniform distribution in Ω_L can be described as

$$\begin{aligned} \theta &= u_\theta * 2\pi \\ r &= \sqrt{u_r * (r_2^2 - r_1^2) + r_1^2} \\ x &= r * \cos(\theta) \\ y &= r * \sin(\theta) \end{aligned} \tag{5.8}$$

where u_θ and u_r are random variables with a uniform distribution in $[0, 1]$

When considering particles with finite sizes, the randomly sampled particle position set $\{(x, y)\}$ needs to satisfy a condition that each new trial (x_i, y_i) has a minimum distance of $0.5(d_i + d_j)$ to any previously valid trial (x_j, y_j) as well as of $0.5d_i$ to the boundaries of Ω_L. This distance-constrained random sampling can be seen as a varying-radii type of Poisson-disk sampling [144, 145]. In this study, a rejection sampling that rejects any invalid (x_i, y_i) is used to satisfy the distance constraint. Assuming the particle diameter $d \in [d_\mathrm{min}, d_\mathrm{max}]$, the termination condition of the rejection sampling is chosen as when

$$n_i > N_i,\ N_i = 50(r_2^2 - r_1^2)/(0.5d_\mathrm{min})^2 \tag{5.9}$$

in which n_i is the number of consecutively invalid trials and is reset to $n_i = 0$ after obtaining each valid trial, and N_i represents a heuristic value that is 50 times the area ratio between Ω_L and a particle of diameter d_min.

As the random sampling proceeds, particles with larger diameters are subject to higher rejection probabilities. In order to reduce the occurrence of invalid trials and potentially increase the tightness of the payload under the given termination condition, the particle diameter is randomly sampled as

$$d_i = d_\mathrm{min} + u_d * (d_\mathrm{max} - d_\mathrm{min}) * (1 - \frac{n_i}{N_i})^{0.5} \tag{5.10}$$

where u_d is a random variable with a uniform distribution in $[0, 1]$. That is, the sampled diameter d_i converges to d_min when n_i approaches to N_i. Subsequently, the number of particles n is determined upon the termination of the random sampling process. In addition, the particle volume fraction, ϕ, of the granular payload has

$$\phi = \frac{\sum_{i=1}^{n}(d_i/2)^2}{r_2^2 - r_1^2} \tag{5.11}$$

and the mass ratio, M/C, between the payload mass M and burster mass C has

$$M/C = \frac{\rho_s\phi(r_2^2 - r_1^2)}{\rho_b r_1^2} \tag{5.12}$$

As a multiscale problem, the ratio between the computational domain length l and particle size d is limited by the available computational power. If the minimum number of grids per particle diameter is required to be n_gpd, the total number of grids per spatial dimension n_gps is then about $n_\mathrm{gpd}l/d$. As a result, to ensure a desired n_gpd under a specific n_gps, the particle diameter needs to satisfy $d \geq n_\mathrm{gpd}l/n_\mathrm{gps}$.

For the current random morphologies under an affordable computational power, $l = 0.15$ m, $\max\{n_\mathrm{gps}\} = 7500$, and $\min\{n_\mathrm{gpd}\} = 50$ are assumed. Correspondingly, the particle diameter is assumed to be a random variable distributed in $[d_{\min}, d_{\max}] = [1\,\text{mm}, 2\,\text{mm}]$. The payload region Ω_{L} defined by r_1 and r_2 considers four sizes $\Omega_{\text{L}h} = \{r : r \in [r_1 = 0.01\,\text{m}, r_2 = \hat{r}_h]\}$, $h = 1, \ldots, 4$, in which $\hat{r}_1 = 0.02\,\text{m}$, $\hat{r}_2 = 0.025\,\text{m}$, $\hat{r}_3 = 0.03\,\text{m}$, and $\hat{r}_4 = 0.04\,\text{m}$. For each size $\Omega_{\text{L}h}$ of stochastic payload, three random samples are generated with the i-th random sample of $\Omega_{\text{L}h}$ denoted as $\text{L}_{h.i}$, $i = 1, \ldots, 3$. Payload morphologies for $\Omega_{\text{L}h.1}$ generated by the described sampling algorithm are shown in Fig. 5.2.

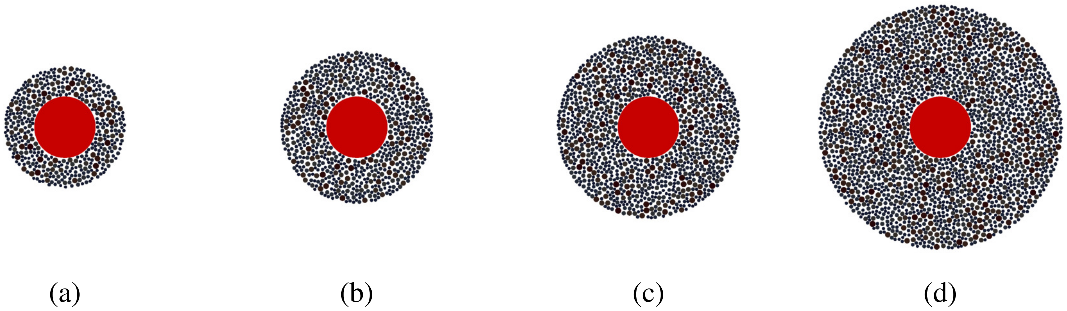
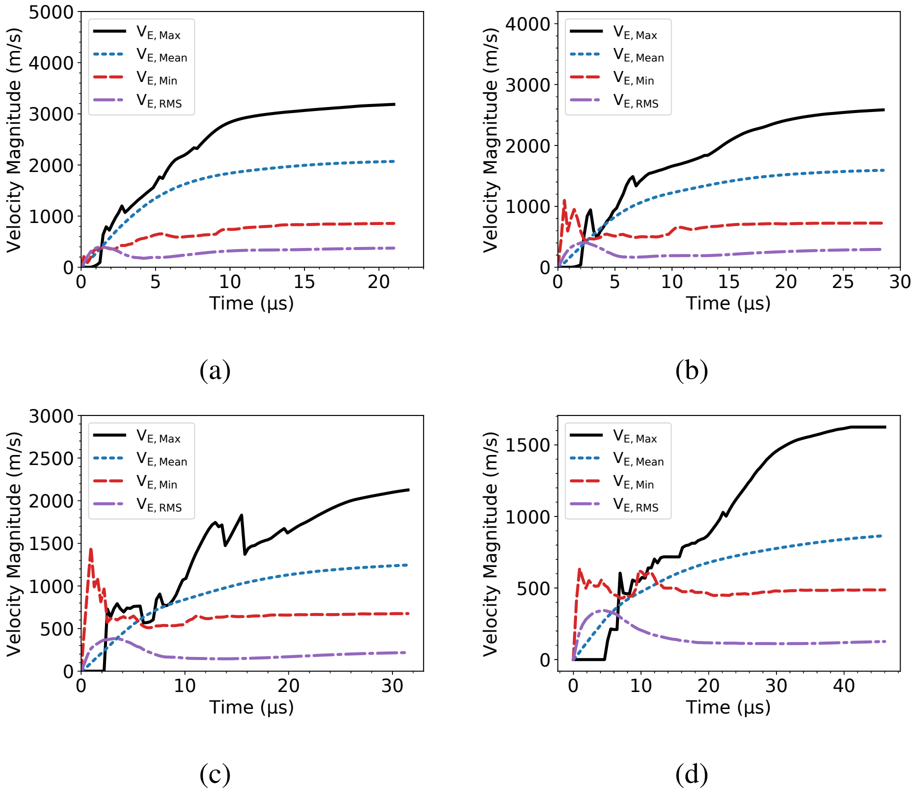


Figure 5.2 Payload morphology samples generated by the random sampling algorithm with rejection sampling. The color of particles is rendered by the particle diameter. (a) $\Omega_{\text{L}1.1}$ with $n = 428$, $M/C = 4.704$, and $\phi = 0.581$. (b) $\Omega_{\text{L}2.1}$ with $n = 742$, $M/C = 8.310$, and $\phi = 0.586$. (c) $\Omega_{\text{L}3.1}$ with $n = 1162$, $M/C = 13.040$, and $\phi = 0.604$. (c) $\Omega_{\text{L}4.1}$ with $n = 2170$, $M/C = 24.700$, and $\phi = 0.610$.

In the following discussions, $\text{L}_{h.i} - \text{B}_j - \text{C}_k$ is used to represent a case with the i-th sample of payload size $\Omega_{\text{L}h}$, burster state B_j, and coefficient of restitution C_k.

Different from a previously published work [75] on deterministic payloads, in this study, the surface expansion velocity V_{E} of a payload is calculated from the velocities of the entire set of particles rather than only the particles near the payload boundaries, since the latter can be strongly affected by the random sampling process for particle positions. Therefore, the current choice can better depict the statistical behaviors of stochastic payloads to make the computational analysis more robust. V_{E} is then analyzed via the maximum $V_{\text{E,Max}}$, mean $V_{\text{E,Mean}}$, and minimum $V_{\text{E,Min}}$ of particle velocities in V_{E} as well as the root-mean-square of velocity fluctuations $V_{\text{E,RMS}}$. Limited by the range of computational domain, at the end of solution time, payloads $\Omega_{\text{L}h}$ with small sizes can reach a nearly constant expansion stage, while payloads with large sizes may still undergo very gradual acceleration but have passed the early phase of rapid acceleration, as captured in Fig. 5.3, in which $V_{\text{E,Max}}$ and $V_{\text{E,Min}}$ consistently record the velocity history of two individual particles that have the maximum and minimum velocity at the end of solution time, respectively.



(a)

(b)

(c)

(d)

Figure 5.3 Time history of surface expansion velocity V_{E} of case $\mathrm{L}_{h.1} - \mathrm{B}_2 - \mathrm{C}_2$, $h = 1, \ldots, 4$. (a) $\mathrm{L}_{1.1} - \mathrm{B}_2 - \mathrm{C}_2$. (b) $\mathrm{L}_{2.1} - \mathrm{B}_2 - \mathrm{C}_2$. (c) $\mathrm{L}_{3.1} - \mathrm{B}_2 - \mathrm{C}_2$. (c) $\mathrm{L}_{4.1} - \mathrm{B}_2 - \mathrm{C}_2$.

5.3 Statistical dissipative behavior

Predicting the expansion speed and subsequent expansion range is useful in quantifying explosive dispersal processes and understanding the clustering of particles. Through assuming a uniform density and a linear velocity gradient in the detonation gas products, an intact payload shell, and axial symmetry, the maximum speed of an explosively expanding solid shell can be modeled by the Gurney equation using 1D theoretical analysis [70]

$$V_{\mathrm{G}} = \sqrt{2E}\left(\frac{M}{C} + \frac{n}{n+2}\right)^{-0.5} \tag{5.13}$$

in which V_{G} is the maximum velocity of the accelerated shell; C is the mass of the burster; M is the mass of the payload shell; n depends on system geometry [146], such as plane $n = 1$, cylinder $n = 2$, and sphere $n = 3$; E is related to the specific total energy of the burster and is referred to as the Gurney energy. Herein, E is determined as the difference between the initial specific total energies of the burster and the ambient air, that is,

$$E = \frac{1}{\gamma - 1}\left(\frac{p_b}{\rho_b} - \frac{p_a}{\rho_a}\right) \tag{5.14}$$

where γ is the heat capacity ratio.

The Gurney theory [70] assumes the payload remaining intact throughout its acceleration. In configurations where metals have low ultimate strains, the solid shell will fracture at relatively small expansion ratios. Subsequently, the detonation gases will stream around the fragments and bypass them to stop the acceleration, resulting in fragment velocities typically about 80% of the value predicted by the Gurney equation [147]. For explosively dispersed dry and fine aluminium powders, Ripley et al. [148] reported an as much as 35% velocity deficit to Gurney velocity using experimental measure. Employing numerical experiments based on a 1D continuum model for explosively dispersed porous shells, Milne [146] proposed an extension to the Gurney equation accounting for velocity reduction resulted from payload porosity

$$\begin{aligned}
&V_{\mathrm{G},\phi} = \sqrt{2E}\left(\frac{M}{C}\frac{1}{\alpha(\rho_0)} + \frac{n}{n+2}\right)^{-0.5} * F(\phi)\\
&\alpha(\rho_0) = a_0\rho_0^{a_1}\\
&F(\phi) = 1 + (a_2\mathrm{e}^{(a_3\phi)} - 0.5) * \log_{10}(\frac{M}{C})
\end{aligned} \tag{5.15}$$

where $V_{\mathrm{G},\phi}$ is the extended Gurney velocity involving porosity; porosity $\phi = \rho/\rho_0$ with ρ and ρ_0 being the bulk density of the porous shell and material density, respectively; constants $a_0 = 0.200$, $a_1 = 0.180$, $a_2 = 0.162$, and $a_3 = 1.127$ for cylindrical charge, and $a_0 = 0.310$, $a_1 = 0.132$, $a_2 = 0.168$, and $a_3 = 1.090$ for spherical charge.

5.3.1 A velocity scaling law with regard to Gurney energy

For both the Gurney equation (Eq. (5.13)) and modified Gurney equation (Eq. (5.15)), under a given M/C and ϕ, the calculated velocity is proportional to $\sqrt{E}$:

$$\frac{\hat{V}_{\mathrm{G}}^{m}}{\hat{V}_{\mathrm{G}}^{n}} = \sqrt{\frac{E_m}{E_n}} \tag{5.16}$$

in which $\hat{V}_{\mathrm{G}}^{m}$ represents the Gurney velocity computed using either Eq. (5.13) or Eq. (5.15) under Gurney energy E_m. Therefore, for reaching a similar expansion range, the solution time is suggested to satisfy an inversely proportional relation:

$$\frac{t_n}{t_m} = \frac{\hat{V}_{\mathrm{G}}^{m}}{\hat{V}_{\mathrm{G}}^{n}} = \sqrt{\frac{E_m}{E_n}} \tag{5.17}$$

As captured in Fig. 5.4, the predicted solutions at the end of solution time well obey the above scaling law. For a given payload with a certain M/C and ϕ, when changing the driver states, the solution times needed to obtain a similar expansion range are inversely proportional to the square root of Gurney energy.

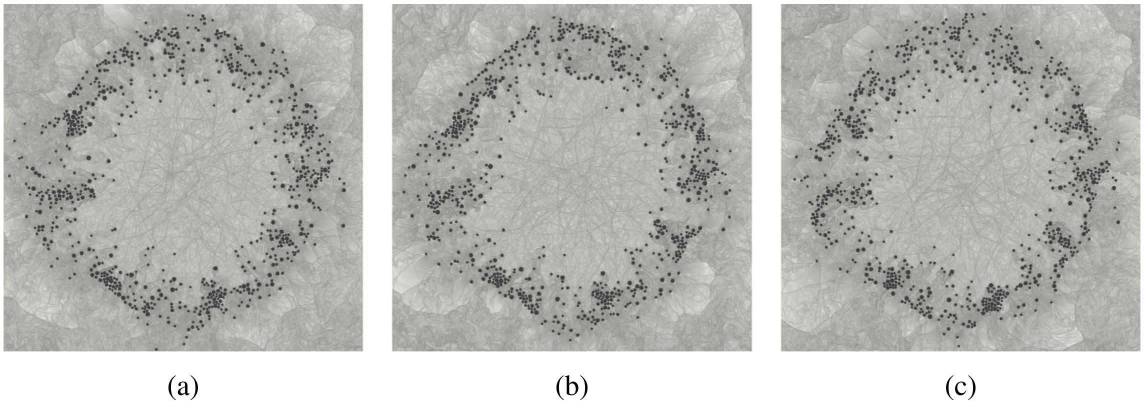

(a) (b) (c)

Figure 5.4 The numerical solutions of $\mathrm{L}_{2.1} - \mathrm{B}_j - \mathrm{C}_2$ that include different Gurney energies E_j and solution times t_j, $j = 1, \ldots, 3$, but reach a similar expansion range. (a) $\mathrm{L}_{2.1} - \mathrm{B}_1 - \mathrm{C}_2$ solved to $t_1 = 40.3\,\mu\mathrm{s}$. (b) $\mathrm{L}_{2.1} - \mathrm{B}_2 - \mathrm{C}_2$ solved to $t_2 = 28.5\,\mu\mathrm{s} \approx t_1\sqrt{E_1/E_2}$. (c) $\mathrm{L}_{2.1} - \mathrm{B}_3 - \mathrm{C}_2$ solved to $t_3 = 18.0\,\mu\mathrm{s} \approx t_1\sqrt{E_1/E_3}$.

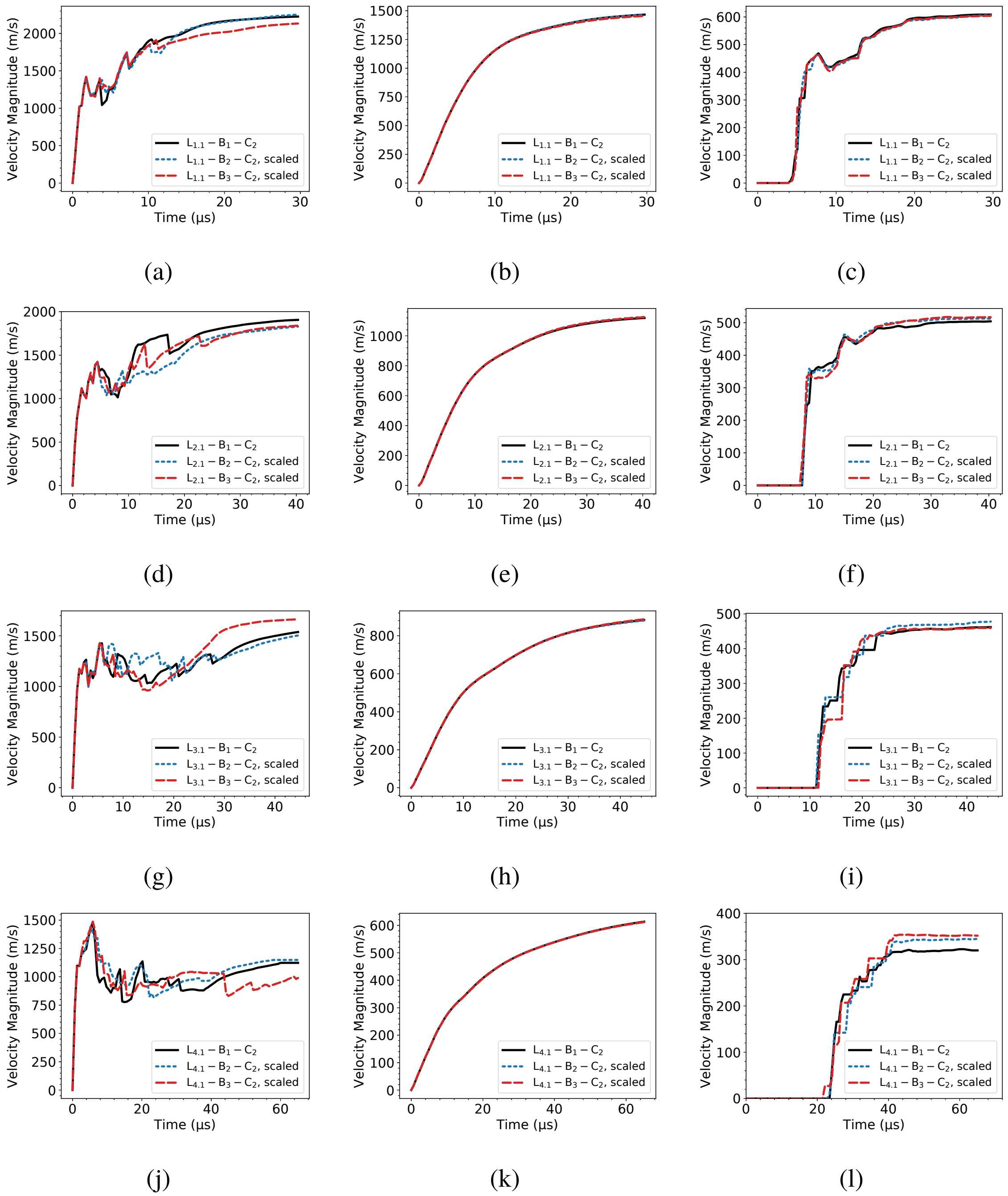
In order to further quantify the velocity scaling law in numerical solutions, for case $\mathrm{L}_{h.i} - \mathrm{B}_j - \mathrm{C}_k$, the solution time t_j and expansion velocity V_{E}^{j} are scaled to $t_j\sqrt{E_j/E_1}$ and $V_{\mathrm{E}}^{j}\sqrt{E_1/E_j}$, respectively. Then, the scaled time histories of V_{E} under different Gurney energies are compared in Fig. 5.5 for a variety of cases. Since the Gurney theory concerns the maximum expansion velocity of the entire solid shell, for the rest of the discussion, $V_{\mathrm{E,Max}}$ and $V_{\mathrm{E,Min}}$ now consistently record the maximum and minimum velocity among the entire set of particles at each time step rather than follow the time history of an individual particle. It can be observed that the predicted expansion velocity can highly preserve the velocity scaling law with regard to Gurney energy not only at the end of solution time but also throughout the solution time, which includes the maximum $V_{\mathrm{E,Max}}$, mean $V_{\mathrm{E,Mean}}$, and minimum $V_{\mathrm{E,Min}}$ and is particularly true for the mean expansion velocity $V_{\mathrm{E,Mean}}$. For instance, the maximum deviation of the scaled $V_{\mathrm{E,Mean}}$ among $\mathrm{L}_{h.1} - \mathrm{B}_j - \mathrm{C}_2$ in Fig. 5.5 happens in the case $\mathrm{L}_{1.1} - \mathrm{B}_2 - \mathrm{C}_2$ and is only 0.792% of that of $\mathrm{L}_{1.1} - \mathrm{B}_1 - \mathrm{C}_2$. This agreement among the scaled expansion velocities in time space extends the time range of the velocity scaling law in the Gurney equation (Eq. (5.13)) and modified Gurney equation (Eq. (5.15)) from the steady-state termination phase to the unsteady evolution phase.

5.3.2 Gurney velocity analysis

Employing the demonstrated velocity scaling law on Gurney energy, the analysis of Gurney velocity can be naturally reduced to only concerning one type of Gurney energy. A comparison between Gurney velocity V_{G}, Gurney velocity with porosity $V_{\mathrm{G},\phi}$, maximum $V_{\mathrm{E,Max}}$, mean $V_{\mathrm{E,Mean}}$, and minimum $V_{\mathrm{E,Min}}$ of the terminal expansion velocity V_{E}, that is, the V_{E} at the end of solution time shown in Fig. 5.5, for $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_k$, $h = 1, \ldots, 4$, $i = 1, \ldots, 3$, $k = 1, 2$, is shown in Fig. 5.6. Comparing $V_{\mathrm{E,Max}}$ with V_{G}, the results suggest that several particles can have velocities close to or even higher than V_{G}. Through animating the solutions in time, it is found that these particles in general are detached from the neighboring particles very early by both inter-grain collisions and fast growing buster fluid jets emerged near them and then are effectively accelerated by the surrounding high pressure gradient generated by the burster jets, causing a much higher velocity than other particles.

Compared to $V_{\mathrm{E,Max}}$ and $V_{\mathrm{E,Min}}$, $V_{\mathrm{E,Mean}}$ can better represent the collective behavior of grains and hence the granular payload. Hence, a comparison between V_{G}, $V_{\mathrm{G},\phi}$, and $V_{\mathrm{E,Mean}}$ is further shown in Fig. 5.7. The predicted $V_{\mathrm{E,Mean}}$ is about 61%-67% of V_{G} and 82%-90% of $V_{\mathrm{G},\phi}$ for $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$ and is about 69%-75% of V_{G} and 86%-110% of $V_{\mathrm{G},\phi}$ for $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$. In addition, the predicted $V_{\mathrm{E,Mean}}$ of each $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$ is consistently lower than that of the corresponding $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$, which indicates a valid dissipation behavior resulting from the change of coefficient of restitution for pairwise collisions.

Comparing $V_{\mathrm{E,Mean}}$ with V_{G} and $V_{\mathrm{G},\phi}$, the results show that $V_{\mathrm{E,Mean}}$ can agree better with $V_{\mathrm{G},\phi}$ than V_{G} for all the cases $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_k$. Considering the difference between the 1D continuum modeling for $V_{\mathrm{G},\phi}$ and the current 2D discrete modeling for $V_{\mathrm{E,Mean}}$, the agreement between $V_{\mathrm{E,Mean}}$ and $V_{\mathrm{G},\phi}$ could be the incidentally agreed overall dissipation between the two different modeling approaches. In the 1D continuum modeling for $V_{\mathrm{G},\phi}$, the dissipation mainly comes from the extra PdV heating of the porous material modeled by the Herrmann P-α equation of state [146]. In the current discrete modeling, the dissipation processes can



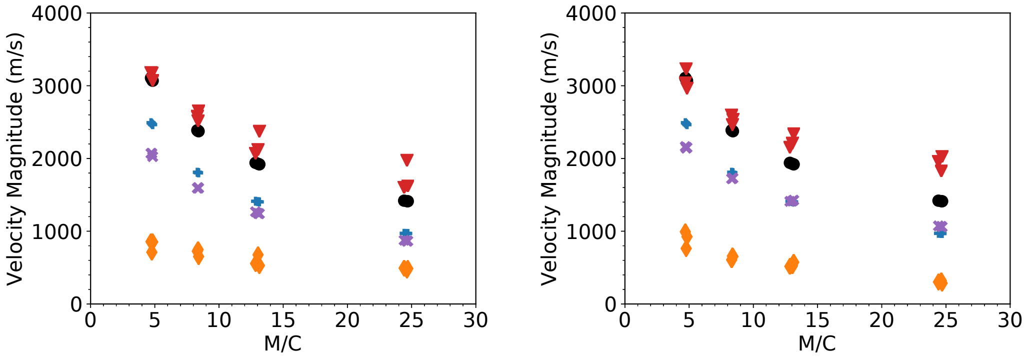
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

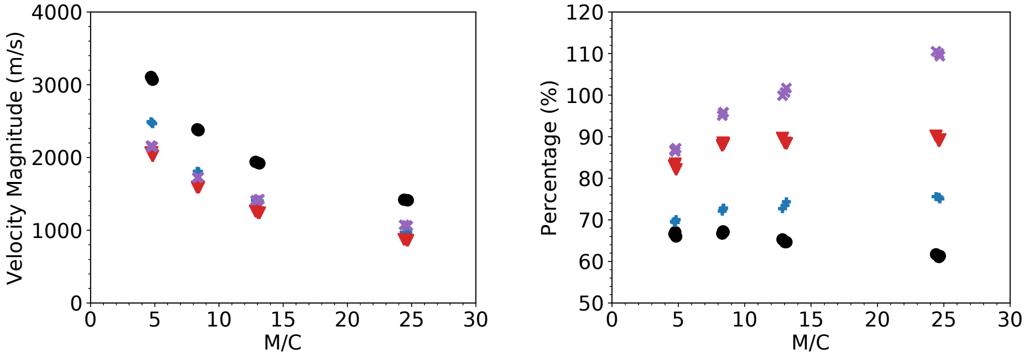
(j) (k) (l)

Figure 5.5 A comparison of the scaled time history of V_{E} for $\mathrm{L}_{h.1}-\mathrm{B}_j-\mathrm{C}_2$ with the scaling $t_j\sqrt{E_j/E_1}$ and $V_{\mathrm{E}}^j\sqrt{E_1/E_j}$, $j = 1,\ldots,3$. (a) $V_{\mathrm{E,Max}}$ of $\mathrm{L}_{1.1}-\mathrm{B}_j-\mathrm{C}_2$. (b) $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{1.1}-\mathrm{B}_j-\mathrm{C}_2$. (c) $V_{\mathrm{E,Min}}$ of $\mathrm{L}_{1.1}-\mathrm{B}_j-\mathrm{C}_2$. (d) $V_{\mathrm{E,Max}}$ of $\mathrm{L}_{2.1}-\mathrm{B}_j-\mathrm{C}_2$. (e) $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{2.1}-\mathrm{B}_j-\mathrm{C}_2$. (f) $V_{\mathrm{E,Min}}$ of $\mathrm{L}_{2.1}-\mathrm{B}_j-\mathrm{C}_2$. (g) $V_{\mathrm{E,Max}}$ of $\mathrm{L}_{3.1}-\mathrm{B}_j-\mathrm{C}_2$. (h) $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{3.1}-\mathrm{B}_j-\mathrm{C}_2$. (i) $V_{\mathrm{E,Min}}$ of $\mathrm{L}_{3.1}-\mathrm{B}_j-\mathrm{C}_2$. (j) $V_{\mathrm{E,Max}}$ of $\mathrm{L}_{4.1}-\mathrm{B}_j-\mathrm{C}_2$. (k) $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{4.1}-\mathrm{B}_j-\mathrm{C}_2$. (l) $V_{\mathrm{E,Min}}$ of $\mathrm{L}_{4.1}-\mathrm{B}_j-\mathrm{C}_2$.



(a) (b)

Figure 5.6 Comparison between V_{G}, $V_{\mathrm{G},\phi}$, $V_{\mathrm{E,Max}}$, $V_{\mathrm{E,Mean}}$, and $V_{\mathrm{E,Min}}$ for $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_k$, $h = 1, \ldots, 4$, $i = 1, \ldots, 3$, $k = 1, 2$, at the end of solution time. (a) $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$. (b) $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$. Legend: ●, V_{G}; +, $V_{\mathrm{G},\phi}$; ▼, $V_{\mathrm{E,Max}}$; ×, $V_{\mathrm{E,Mean}}$; ♦, $V_{\mathrm{E,Min}}$.



(a) (b)

Figure 5.7 Comparison between V_{G}, $V_{\mathrm{G},\phi}$, and $V_{\mathrm{E,Mean}}$ for $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_k$, $h = 1, \ldots, 4$, $i = 1, \ldots, 3$, $k = 1, 2$, at the end of solution time. (a) Primitive values. Legend: ●, V_{G}; +, $V_{\mathrm{G},\phi}$; ▼, $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$; ×, $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$. (b) Value ratios in percentage. Legend: ●, $V_{\mathrm{E,Mean}}/V_{\mathrm{G}} * 100\%$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$; +, $V_{\mathrm{E,Mean}}/V_{\mathrm{G}} * 100\%$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$; ▼, $V_{\mathrm{E,Mean}}/V_{\mathrm{G},\phi} * 100\%$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$; ×, $V_{\mathrm{E,Mean}}/V_{\mathrm{G},\phi} * 100\%$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$.

include inter-grain collisions and the escape of burster fluid, which are physical features that cannot be naturally represented in 1D continuum models for granular materials. Since the $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_2$ is about 61%-67% of V_{G}, while the $V_{\mathrm{E,Mean}}$ of $\mathrm{L}_{h.i} - \mathrm{B}_2 - \mathrm{C}_1$ is about 69%-75% of V_{G}, it is suggested that about 25%-31% dissipation can be roughly attributed to the energy lose resulting from the escape of burster fluid in the current porous payloads. The better agreement between $V_{\mathrm{E,Mean}}$ and $V_{\mathrm{G},\phi}$ and the major dissipation caused by the escape of burster fluid both suggest that incorporating the effects of porosity can enhance the prediction of Gurney velocity for explosively dispersed granular payloads.

In analyzing the surface expansion velocity V_{E}, two critical properties of the Gurney velocity theory, such as V_{G} decreasing with an increase in M/C and V_{G} proportional to $\sqrt{E}$, have been consistently produced in the direct simulations. In addition, the predicted $V_{\mathrm{E,Mean}}$ of inelastic collisions is smaller than that of elastic collisions, obeying the effects of more dissipative collisions. Moreover, the predicted $V_{\mathrm{E,Mean}}$ using a 2D discrete model herein can broadly agree with the $V_{\mathrm{G},\phi}$ concerning porosity ϕ via a 1D continuum model. These findings thus effectively demonstrate a valid statistical dissipative property in the mesoscale model for the granular payload and the subsequent dense gas-solid flow.

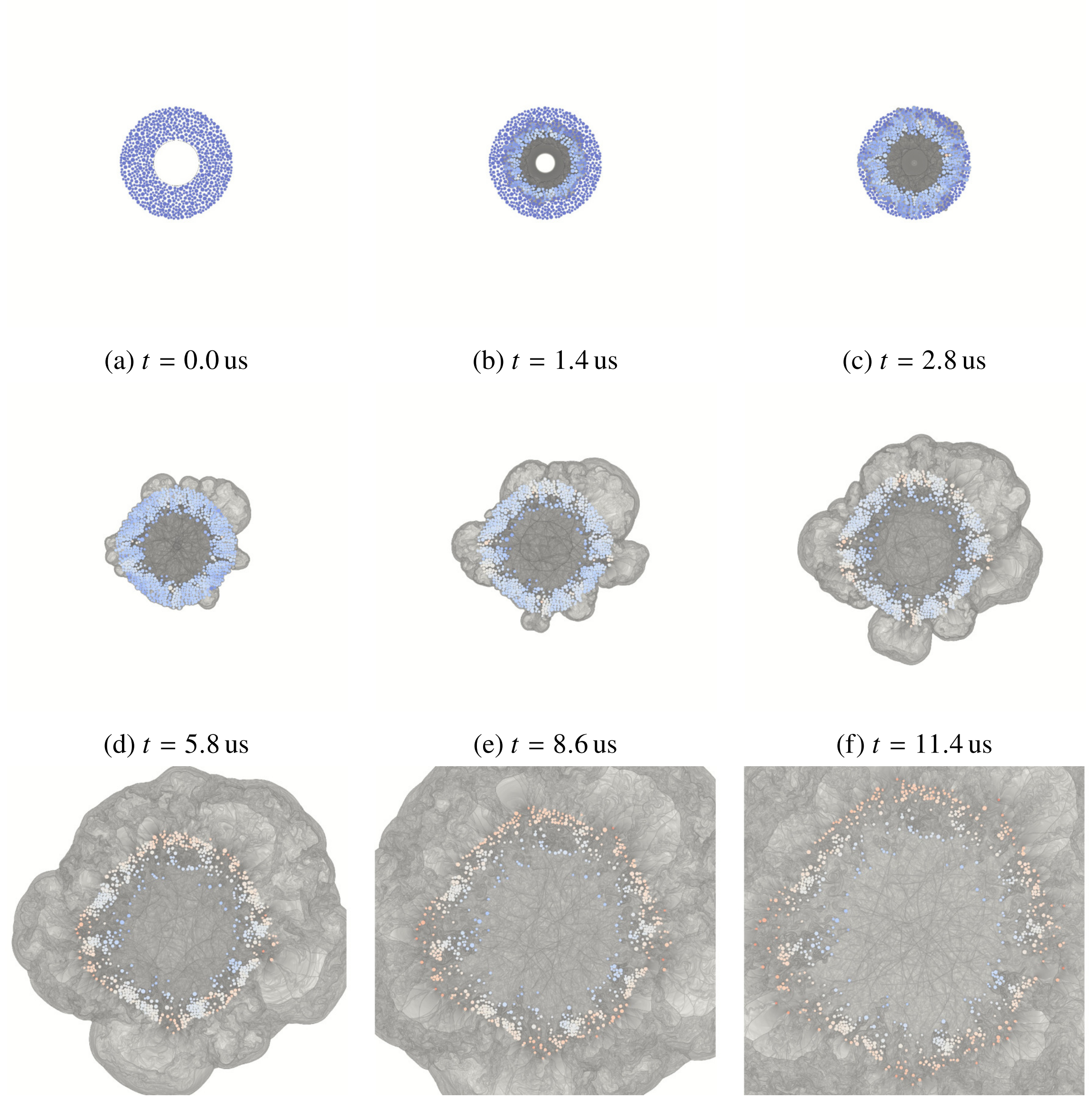
5.4 Particle cluster and jet evolution

The dispersal processes of the case $\mathrm{L}_{2.1} - \mathrm{B}_2 - \mathrm{C}_2$ and $\mathrm{L}_{3.1} - \mathrm{B}_2 - \mathrm{C}_2$ are captured in Fig. 5.8 and Fig. 5.9, respectively. On the basis of the direct simulations, an explanation for particle clustering and jetting instabilities is proposed.

1. Shock compaction phase.

   Once the burster expands in space, the granular payload is radially compacted such that the inner particles move outward to occupy the pore spaces and subsequently collide with nearby particles to build force-chain networks for momentum transfer. As shown from Fig. 5.8a to Fig. 5.8c of $\mathrm{L}_{2.1} - \mathrm{B}_2 - \mathrm{C}_2$ and from Fig. 5.9a to Fig. 5.9c of $\mathrm{L}_{3.1} - \mathrm{B}_2 - \mathrm{C}_2$, until the radial compaction propagates throughout the payload, the outer payload-air interface almost stays stationary.

   At the inner burster-payload interface, shock reflects and diffracts on the irregular surfaces introduced by grains, generating inward density gradients and perturbations. RMI/RTI can then be possibly induced into the burster side, and particles can be entrained by the growing perturbations as well as pushed by the inner particles to form inside clusters and jets, which are then soon disturbed by the subsequent expansion of the converging shocks reflected from the burster center, as shown at the burster region from Fig. 5.8b to Fig. 5.8d of $\mathrm{L}_{2.1} - \mathrm{B}_2 - \mathrm{C}_2$ and from Fig. 5.9b to Fig. 5.9d of $\mathrm{L}_{3.1} - \mathrm{B}_2 - \mathrm{C}_2$.



(a) t = 0.0 us

(b) t = 1.4 us

(c) t = 2.8 us

(d) t = 5.8 us

(e) t = 8.6 us

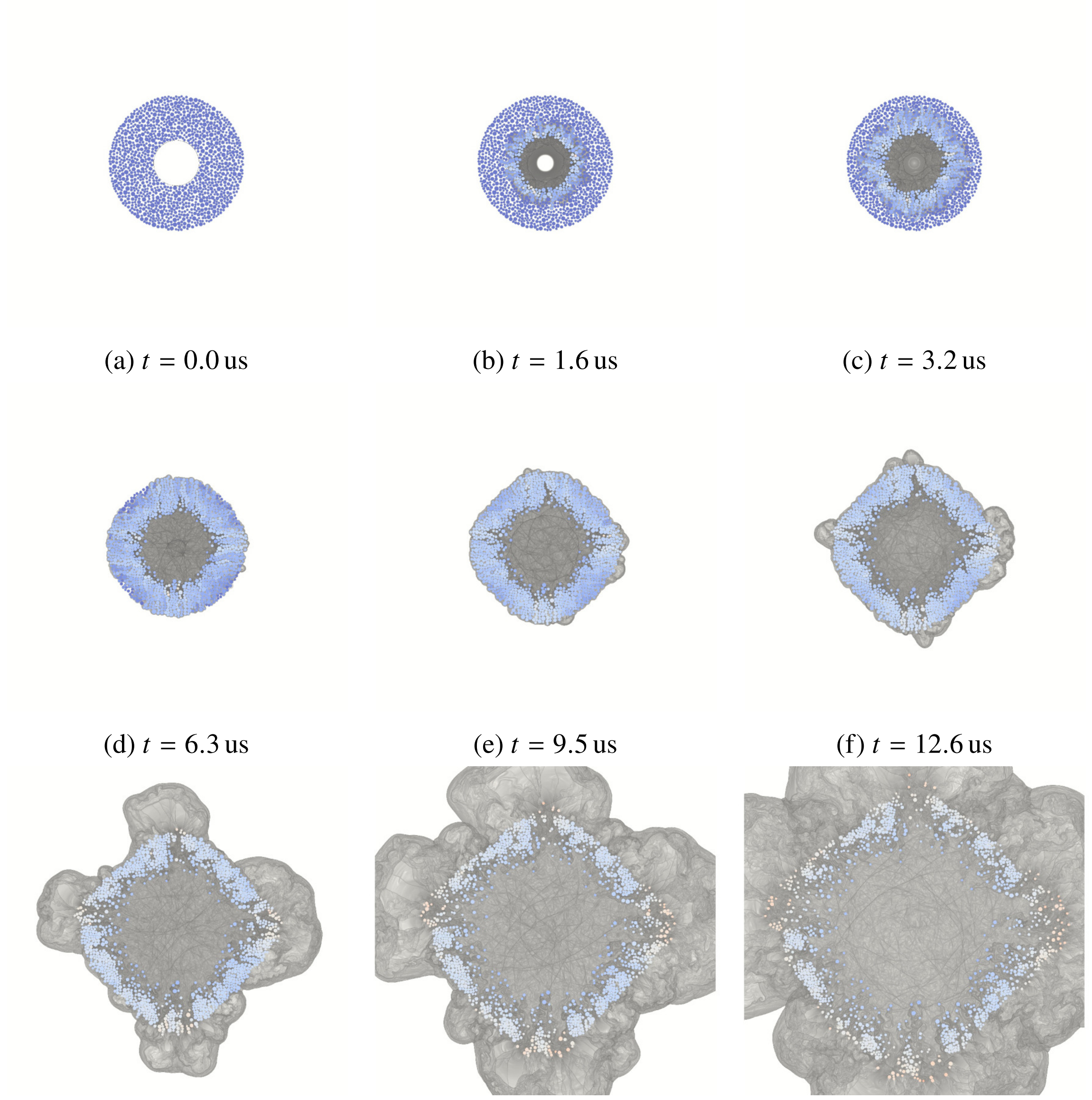
(f) t = 11.4 us

(g) t = 17.1 us

(h) t = 22.8 us

(i) t = 28.5 us

Figure 5.8 The dispersal evolution for case $L_{2.1} - B_2 - C_2$ captured via the superimposition of fluid field rendered by density gradient and particle field rendered by velocity. Density gradient in logarithmic scale: [White, Black] = $[1.0, 1.7 \times 10^7]$. Particle velocity (m/s) in linear scale: [Blue, Red] = [0.0, 3000.0].



(a) t = 0.0 us

(b) t = 1.6 us

(c) t = 3.2 us

(d) t = 6.3 us

(e) t = 9.5 us

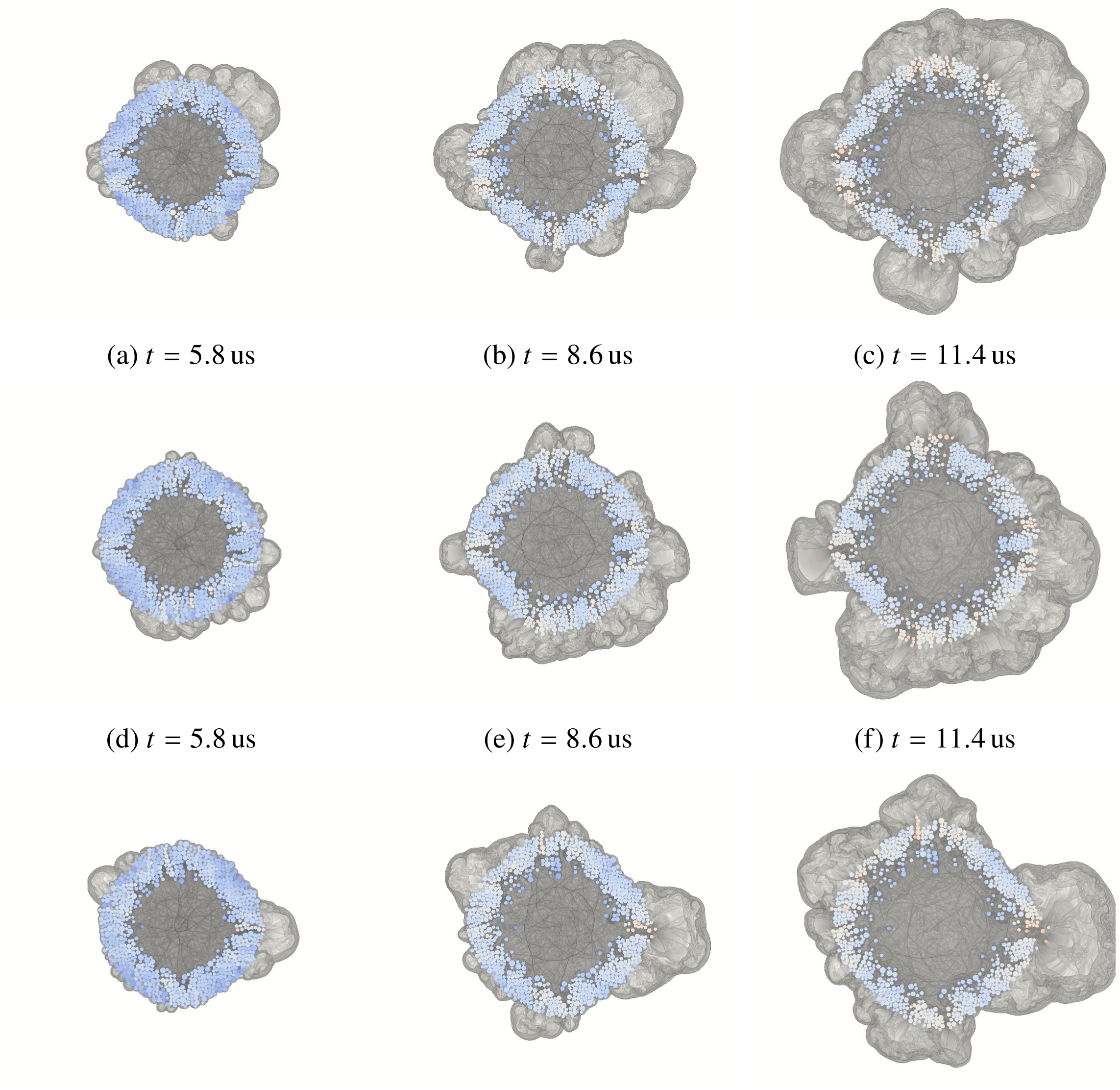
(f) t = 12.6 us

(g) t = 18.9 us

(h) t = 25.2 us

(i) t = 31.5 us

Figure 5.9 The dispersal evolution for case $L_{3.1} - B_2 - C_2$ captured via the superimposition of fluid field rendered by density gradient and particle field rendered by velocity. Density gradient in logarithmic scale: [White, Black] = $[1.0, 1.7 \times 10^7]$. Particle velocity (m/s) in linear scale: [Blue, Red] = $[0.0, 3000.0]$.



(a) t = 5.8 us

(b) t = 8.6 us

(c) t = 11.4 us

(d) t = 5.8 us

(e) t = 8.6 us

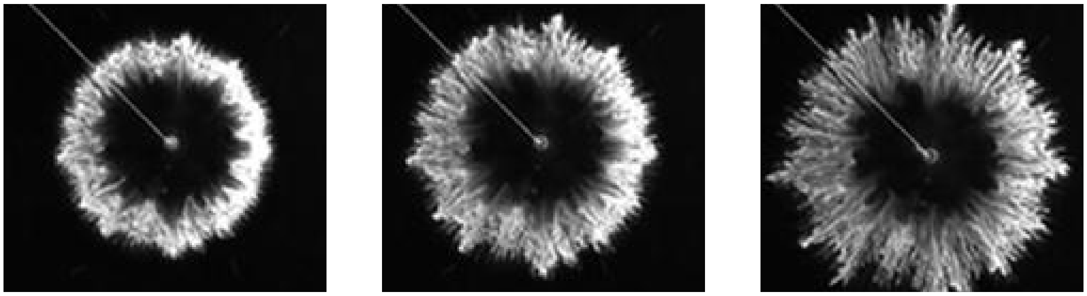
(f) t = 11.4 us

(g) t = 5.8 us

(h) t = 8.6 us

(i) t = 11.4 us

Figure 5.10 The development of internal sliding and colliding lines for cases $L_{2.i} - B_2 - C_2$, $i = 1, \ldots, 3$. (a), (b), (c) $L_{2.1} - B_2 - C_2$. (d), (e), (f) $L_{2.2} - B_2 - C_2$. (g), (h), (i) $L_{2.3} - B_2 - C_2$.



(a) $t = 14.0\,\text{ms}$ (b) $t = 18.0\,\text{ms}$ (c) $t = 24.0\,\text{ms}$

Figure 5.11 The experimental observation of a weak shock dispersed flour powder in a Hele-Shaw cell [7].

2. Jet initiating phase.

   During shock compaction, a large pressure gradient exists between the inside and outside regions of the payload, which accelerates not only the payload but also the burster fluid into the payload.

   (a) Fine jets. The current mesoscale simulations show that, through shock compaction, perturbations at the inner burster-payload interface as well as within the payload can effectively propagate to the outer payload-air interface, as captured from Fig. 5.8b to Fig. 5.8d of $L_{2.1} - B_2 - C_2$ and from Fig. 5.9b to Fig. 5.9d of $L_{3.1} - B_2 - C_2$. An impulsive acceleration can then be imposed on the outer payload-air interface via the force-chain networks in the payload, which could induce macroscale RMI into the ambient air through impulsively growing the initial interfacial irregularities. Under the continuous acceleration sustained by the pressure gradient, it is suggested that the RMI-enhanced perturbations at the outer payload-air interface may further grow as a form of RTI, as observed in [26].

   (b) Primary jets. As shown in Fig. 5.8b of $L_{2.1}-B_2-C_2$ and Fig. 5.9b of $L_{3.1}-B_2-C_2$, at the inner burster-payload interface, shock diffracts at convex grain surfaces and focuses in concave inter-grain pore spaces. The burster fluid entrances the payload through inter-grain pores and initiates payload fractures. However, the payload fracture patten is dominated not by the initial pores at the inner burster-payload interface but mainly by the gas-granule interaction and force-chain propagation within the payload, as further explained below.

   Driven by pressure gradient, burster fluid penetrates the payload through the pore spaces among particles and induce inter-grain collisions. The collisions among particles generate a complex force chain in the granular payload. Within the force-chain networks, internal sliding lines are regions of weak resistance and function as the attraction points for payload fractures and burster fluid, while internal colliding lines are effective for momentum transfer among particles and facilitate the formation of particle clusters. A comparison of the development of sliding and colliding lines for cases $L_{2.i} - B_2 - C_2$, $i = 1, \ldots, 3$, is shown in Fig. 5.10. It can be observed that different payload morphologies can develop varied sliding and colliding lines, which lead to a corresponding pattern for payload fracturing and particle clustering. This development of sliding and colliding lines in a granular payload can also be found in experimental observations, such as in the time evolution of a weak shock dispersed flour powder confined in a Hele-Shaw cell [7], as captured from Fig. 5.11a to Fig. 5.11c.

As the buster fluid penetrates the payload, the particle clusters and their low-pressure interstitial fluid will be contained by the surrounding high-pressure buster fluid. The local pressure gradient between the surrounding flow and interstitial flow can help maintain the stability of the particle clusters and cause more inelastic collisions. The increased collisions effectively induce a more equilibrium momentum distribution in each particle cluster and again maintain the stability of the cluster, as shown from Fig. 5.8e to Fig. 5.8g of $L_{2.1} - B_2 - C_2$ and from Fig. 5.9e to Fig. 5.9g of $L_{3.1} - B_2 - C_2$. The existence of stable clusters introduce a more non-equilibrium momentum distribution in the overall payload, exhibiting as a form of clustering instability.

3. Jet growing and relaxation phase.

The strained expansion of both the burster and payload eventually leads to the complete penetration of buster fluid, as shown in Fig. 5.8e of $L_{2.1} - B_2 - C_2$ and Fig. 5.9g of $L_{3.1} - B_2 - C_2$. The fracture of the payload causes the increase of escaping burster fluid, and particle clusters are accelerated into the ambient air. When subjected to a relatively small growth rate, the early formed fine jets at the outer payload-air interface can be dissipated by aerodynamic interactions or overtaken by the payload front comprising primary clusters or jets during fast expansion. After penetrating a granular payload, the blast front can be highly irregular.

As captured from Fig. 5.8d to Fig. 5.8g of $L_{2.1} - B_2 - C_2$, during the acceleration phase of a particle cluster, the surrounding blast flow shears with the cluster boundaries with the shear direction toward the front region of the cluster, causing nearby particles converge to the shear wake at the cluster front. This shear behavior could be responsible for the finger-like shape of particle jets observed in macroscale experiments and introduce Kelvin–Helmholtz instability (KHI) at cluster boundaries. Note that the resulting KHI has an inverse shear direction with the usual KHI accompanying a RTI, in which the latter KHI leads to a mushroom shape due to adverse shear.

As the blast expansion continues, the velocity difference between the surrounding flows and the particle clusters gradually reduces, and particle clusters have gained most of the momentum from the blast flow. Within the limited M/C ratios studied in the current simulations, it is suggested that the relaxation of surrounding flows weakens the local pressure gradients for maintaining particle clusters and jets, together with aerodynamic interactions, leading to the relaxation of particle clusters and jets, as shown from Fig. 5.8h to Fig. 5.8i of $\mathrm{L}_{2.1}-\mathrm{B}_2-\mathrm{C}_2$ and from Fig. 5.9h to Fig. 5.9i of $\mathrm{L}_{3.1}-\mathrm{B}_2-\mathrm{C}_2$.

In the current dispersal of payloads with relatively small M/C ratios and number of particles, the burst gas escapes through the limited layers of solid particles with less energy loss, causing a quick relaxation of particle clusters and jets. For the dispersal of granular payloads with large M/C ratios and numerous particles, it is suggested that the blast flow subjects to a more rapid energy decay during diverging expansion, while solid particles can largely retain the gained momentum due to inertia and maintain the form of particle clusters or jets for a prolonged period.

5.5 Conclusion

The developed interface-resolved direct simulation framework has been applied to study the gas-solid flow generated by explosively dispersed granular materials. The numerical investigation considers a 2D stratified burster-payload configuration with a cylindrical charge shape. The particle positions and sizes in the granular payloads are obtained by a random sampling algorithm to achieve stochastic payload morphologies. In the current study, a set of cases that cover four sizes of stochastic payloads with three random samples per payload size, three burster states, and two types of coefficients of restitution for pairwise collisions have been solved and analyzed to help understand the particle clustering and jetting dynamics in explosively dispersed granular materials.

Through solving explosively dispersed stochastic payloads with shock-shock, shock-particle, and particle-particle interactions and analyzing the surface expansion velocities, a valid statistical dissipative property of the framework has been demonstrated. The predicted surface expansion velocities can highly preserve the velocity scaling law with regard to Gurney energy not only at the end of solution time but also throughout the solution time, which extends the time range of the velocity scaling law in the Gurney theory [70, 146] from the steady-state termination phase to the unsteady evolution phase. When considering the mean surface expansion velocities, the maximum error of the unsteady velocity scaling law was about 0.792% among the investigated Gurney energies. A quantitative analysis of the dissipation in the current discrete modeling of granular payloads suggests that the escape of burster fluid rather than granular dissipation can be the major factor for the loss of burster energy. As a result, incorporating the effects of porosity can enhance the prediction of Gurney velocity for explosively dispersed granular payloads.

On the basis of direct simulations, an explanation for particle clustering and jetting instabilities has been proposed to increase the understanding of established experimental observations in the literature. According to the direct simulations, the development of internal sliding and colliding lines in the shock-compacted granular payload can be critical to the subsequent fracture pattern of the payload. Particle clusters manifested through payload fracture are then maintained by local pressure gradient between surrounding and interstitial flows as well as by dissipative inter-grain collisions. The existence of stable clusters introduce a more non-equilibrium momentum distribution in the overall payload, exhibiting as a form of clustering instability. Different payload morphologies can develop varied sliding and colliding lines, which lead to a corresponding pattern for payload fracturing and particle clustering.

Chapter 6

Summary and Future Work

6.1 Summary

The explosive dispersal of granular materials involves complex multiphase interactions, such as coupled shock-shock, shock-particle, and particle-particle interactions. Under high rates of momentum and energy transfer resulting from shock compaction and subsequent expansion flow, the mesoscale interactions are effectively enhanced and later are manifested via complex macroscale phenomena as the expansion proceeds. In particular, influencing the mass concentration and related particle reaction and energy release, the formation of coherent particle clustering and jetting structures is of significant interest to study flow instability and turbulent mixing in heterogeneous detonation and explosion.

In order to better understand the dynamics of particle clustering and jetting behaviors, direct simulations that capture mesoscale multiphase interactions were adopted. Due to the difficulties in achieving the direct simulation of explosively dispersed granular materials, such as challenges associated with fluid-solid interaction, dynamic phase interface, mesh generation and regeneration, multibody contact and collision, complex interfacial condition, computational efficiency and robustness, and the presence of high-pressure shock waves, solely integrating existing numerical techniques was not able to obtain a capable computational solver. Therefore, new mathematical models and numerical methods that bridge the gap between the available methods and remaining problems were required. As a result, to study particle cluster and jet formation via direct simulations, this thesis included two major scopes: the development of a high-fidelity gas-solid flow solver and the numerical investigation on particle clustering and jetting instabilities in explosively dispersed granular materials.

A numerical framework that integrates operator splitting for partitioned fluid-solid interaction in the time domain, 2nd/3rd order strong stability-preserving Runge–Kutta methods and 3rd/5th order weighted essentially nonoscillatory schemes for high-resolution tempo-spatial discretization, the front-tracking method for evolving phase interfaces, a new field function developed for facilitating the solution of complex and dynamic fluid-solid systems on Cartesian grids, a new collision model developed for deterministic multibody contact and collision with parameterized coefficients of restitution and friction, and a new immersed boundary method developed for treating arbitrarily irregular and moving boundaries has been developed and validated. On the basis of the developed framework, a Cartesian, 3D, high-resolution, parallelized, gas-solid flow solver has been created with the capability of tackling shocked flow conditions, irregular and moving geometries, and multibody collisions. The developed framework and solver can accurately, efficiently, and robustly solve coupled fluid-fluid, fluid-solid, and solid-solid interactions with flow conditions ranging from subsonic to hypersonic states.

Employing the developed framework and solver, direct simulations that capture coupled multiphase interactions and deterministic mesoscale granular dynamics were conducted to investigate particle clustering and jetting instabilities. The numerical investigation considered a 2D stratified burster-payload configuration with a cylindrical charge shape, in which the particle positions and sizes were generated by a random sampling algorithm to achieve stochastic payload morphologies. Employing the theory of Gurney velocity, a valid statistical dissipative property of the framework in solving explosively dispersed granular materials was demonstrated. The predicted surface expansion velocities highly preserved the Gurney-energy-related velocity scaling law not only at the end of but also throughout the solution time. In addition, a dissipation analysis of the current discrete modeling of granular payloads suggests that incorporating the effects of porosity can enhance the prediction of Gurney velocity for explosively dispersed granular payloads. Using results obtained from direct simulations, an explanation for particle clustering and jetting instabilities was proposed.

6.2 Future work

In the investigation of particle cluster and jet formation, simplifications on the physical model have been made to outline the main features of the physical problem and to make the direct simulations feasible in the current stage. Nonetheless, alleviating the model simplifications could enhance the results and potentially lead to the discovery of new phenomena. Therefore, three main aspects of model improvements are discussed below and can be considered as the primary research topics for future work.

Multibody collision model In the current study, a deterministic multibody collision model built on exact pairwise collision is used. Although providing a deterministic approach for approximating multibody collision response with parameterized elasticity and friction, the current model is not able to solve multibody collisions in general exactly. Therefore, the development of a genuine multibody collision model can be of significant value in the mesoscale modeling of granular materials.

In addition, the computational load of a mesoscale model is lower bounded by the ratio between the largest and smallest resolved scales for the target system. Being computation-intensive, mesoscale modeling is currently not applicable to large-scale practical systems, for which macroscale modeling is still the most viable approach [50]. Therefore, the developed multibody collision model can be adopted to refine binary collision-based kinetic theories for granular materials. The refined kinetic theory can subsequently be used to develop enhanced closure laws for macroscale models covering flow regimes with varying agitation levels and dense-to-dilute transitions.

Deformation and fragmentation When granular materials are explosively dispersed, individual particles are compressed and accelerated by the surrounding high-pressure waves, experiencing collision, agglomeration, deformation, and even fragmentation. In the current study, particles are assumed to be non-deformable bodies. Therefore, in order to enhance the physics involved in direct simulations, future research can consider extending the current model with deformable bodies and fragmentation processes. For instance, the Mie–Gruneisen [149, 150] equation of state and the Johnson-Cook strength model [151] can be adopted to describe solids with deformation and fragmentation behaviors.

Parallelization and large-scale computation In the current investigation of particle cluster and jet formation, the maximum number of particles involved in the dispersal system is at the magnitude of 10^3, which is far less than that in a practical payload and could limit the comparison between numerical results and experimentally observed particle cloud expansions. Therefore, simulations on stochastic payloads with larger sizes and longer run times are to be performed in future studies. To increase the number of particles inevitably demands a higher computational capacity. The parallelization of fluid-solid systems with dense particles imposes great challenges on the data communication and load balancing of MPI-based parallelism, as these systems generally do not conform to the Single Process, Multiple Data pattern of computation. Although being flexible in parallelizing large-scale dispersed granular systems, the current OpenMP-based parallelism restricts the computational resources that can be employed. As a result, a hybrid parallelism that integrates OpenMP and MPI can be considered in the future studies to extend the direct simulations into large scales.

6.3 Concluding remarks

As a fundamental problem in many engineering applications, the interface-resolved predictive modeling of gas-solid flows is an active area of research and requires addressing a series of challenges. Therefore, the development of an accurate, efficient, and robust direct simulation numerical framework contributes to numerical methods applied to complex gas-solid flow problems. In addition, the advancement in understanding particle clustering and jet instabilities can potentially contribute to the development and application of a number of engineering problems, including heterogeneous explosives.

Under the current assumptions of non-deformable grains, the mesoscale granular dynamics largely depends on the payload morphology as a result of packing methods. Different payload morphologies can develop varied sliding and colliding lines, which lead to a corresponding pattern for payload fracturing and particle clustering. With the rapid development of high-performance computing technology, future direct simulations on stochastic payloads with significantly increased domain sizes, number of particles, and solution times are expected to lead to a better understanding of the flow instability in explosively dispersed granular payloads. It is suggested that statistics collected from a large number of mesoscale computations based on random payload morphologies can potentially evolve into a macroscopic theory of multiphase flow instability for particle clustering and jetting phenomena widely observed in many areas involving dense gas-solid flows.

Appendix A

3D Navier–Stokes Equations, Jacobian Matrices, and Eigendecompositions

This chapter provides a detailed reference for the 3D Navier–Stokes equations expressed in a variety of mathematical forms, the Jacobian matrices, and their eigendecompositions. This reference can effectively aid the numerical discretization and code implementation for the numerical solution of Navier–Stokes equations. In order to minimize potential errors related to derivation and typography, the presented equations have been carefully examined both theoretically via mathematical derivation and numerically through code implementation.

A.1 System of equations

The Navier–Stokes equations in differential form can be expressed as a system of partial differential equations that consist of the conservation laws of mass, momentum, and energy [152]:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \\ \frac{\partial (\rho \boldsymbol{V})}{\partial t} + \nabla \cdot (\rho \boldsymbol{V} \boldsymbol{V}) = \nabla \cdot (\boldsymbol{\tau} - p\mathbf{I}) + \boldsymbol{f}^{\mathrm{b}} \\ \frac{\partial (\rho e_{\mathrm{T}})}{\partial t} + \nabla \cdot (\rho e_{\mathrm{T}} \boldsymbol{V}) = \nabla \cdot (k \nabla T) + \nabla \cdot [(\boldsymbol{\tau} - p\mathbf{I}) \cdot \boldsymbol{V}] + \boldsymbol{f}^{\mathrm{b}} \cdot \boldsymbol{V} \end{cases} \tag{A.1}$$

The closure of the system currently considers the Newtonian fluid relation with the Stokes hypothesis

$$\tau_{ij} = \mu \left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i} - \frac{2}{3} (\nabla \cdot \boldsymbol{V}) \delta_{ij} \right) \tag{A.2}$$

and the perfect gas law

$$\begin{aligned} p &= \rho R T \\ e &= C_v T \end{aligned} \tag{A.3}$$

A.2 Vector form

A.2.1 Conservative vector form

Through introducing the vector of conservative variables $\boldsymbol{U}$, the vector of convective fluxes $\boldsymbol{F}_i$, the vector of diffusive fluxes $\boldsymbol{F}_i^{\mathrm{v}}$, and the vector of source terms $\boldsymbol{\Phi}$, the system of partial differential equations of conservation laws can be written into a vector form in a conservative manner

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_x(\boldsymbol{U})}{\partial x} + \frac{\partial \boldsymbol{F}_y(\boldsymbol{U})}{\partial y} + \frac{\partial \boldsymbol{F}_z(\boldsymbol{U})}{\partial z} = \frac{\partial \boldsymbol{F}_x^{\mathrm{v}}(\boldsymbol{U})}{\partial x} + \frac{\partial \boldsymbol{F}_y^{\mathrm{v}}(\boldsymbol{U})}{\partial y} + \frac{\partial \boldsymbol{F}_z^{\mathrm{v}}(\boldsymbol{U})}{\partial z} + \boldsymbol{\Phi} \tag{A.4}$$

$$\boldsymbol{U} = \left(\rho, \rho u, \rho v, \rho w, \rho e_{\mathrm{T}}\right)^{\mathrm{T}}$$

$$\boldsymbol{F}_x(\boldsymbol{U}) = \left(\rho u, \rho uu + p, \rho vu, \rho wu, (\rho e_{\mathrm{T}} + p)u\right)^{\mathrm{T}} = u\boldsymbol{U} + p(0, 1, 0, 0, u)^{\mathrm{T}}$$

$$\boldsymbol{F}_y(\boldsymbol{U}) = \left(\rho v, \rho uv, \rho vv + p, \rho wv, (\rho e_{\mathrm{T}} + p)v\right)^{\mathrm{T}} = v\boldsymbol{U} + p(0, 0, 1, 0, v)^{\mathrm{T}}$$

$$\boldsymbol{F}_z(\boldsymbol{U}) = \left(\rho w, \rho uw, \rho vw, \rho ww + p, (\rho e_{\mathrm{T}} + p)w\right)^{\mathrm{T}} = w\boldsymbol{U} + p(0, 0, 0, 1, w)^{\mathrm{T}}$$

$$\boldsymbol{F}_x^{\mathrm{v}}(\boldsymbol{U}) = \left(0, \tau_{xx}, \tau_{xy}, \tau_{xz}, k\tfrac{\partial T}{\partial x} + \tau_{xx}u + \tau_{xy}v + \tau_{xz}w\right)^{\mathrm{T}}$$

$$\boldsymbol{F}_y^{\mathrm{v}}(\boldsymbol{U}) = \left(0, \tau_{yx}, \tau_{yy}, \tau_{yz}, k\tfrac{\partial T}{\partial y} + \tau_{yx}u + \tau_{yy}v + \tau_{yz}w\right)^{\mathrm{T}}$$

$$\boldsymbol{F}_z^{\mathrm{v}}(\boldsymbol{U}) = \left(0, \tau_{zx}, \tau_{zy}, \tau_{zz}, k\tfrac{\partial T}{\partial z} + \tau_{zx}u + \tau_{zy}v + \tau_{zz}w\right)^{\mathrm{T}}$$

$$\boldsymbol{\Phi} = \left(0, f_x^{\mathrm{b}}, f_y^{\mathrm{b}}, f_z^{\mathrm{b}}, f_x^{\mathrm{b}}u + f_y^{\mathrm{b}}v + f_z^{\mathrm{b}}w\right)^{\mathrm{T}}$$

$$\tau_{ij} = \mu\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i} - \frac{2}{3}(\nabla \cdot \boldsymbol{V})\delta_{ij}\right)$$

$$e_{\mathrm{T}} = e + \frac{1}{2}(u^2 + v^2 + w^2)$$

$$e = C_v T = \frac{p}{(\gamma - 1)\rho}$$

$$p = \rho R T$$

$$C_v = \frac{R}{\gamma - 1}$$

$$C_p = \gamma C_v$$

$$k = \frac{C_p \mu}{Pr}$$

$$c = \sqrt{\gamma \frac{p}{\rho}}$$

A.2.1.1 Dimensional form

To use the equations (Eq. (A.4)) in a dimensional form, the specific gas constant R and the dynamical viscosity μ are dimensional parameters.

A.2.1.2 Dimensionless form

Dimensional variables

$$\tilde{x},\ \tilde{y},\ \tilde{z},\ \tilde{u},\ \tilde{v},\ \tilde{w},\ \tilde{t},\ \tilde{\rho},\ \tilde{T},\ \tilde{p},\ \tilde{e}_{\mathrm{T}},\ \tilde{e},\ \tilde{f}_x^{\mathrm{b}},\ \tilde{f}_y^{\mathrm{b}},\ \tilde{f}_z^{\mathrm{b}}$$

Reference variables

$$L^*,\ U^*,\ \rho^*,\ T^*$$

Nondimensionalized variables

$$x = \frac{\tilde{x}}{L^*},\ y = \frac{\tilde{y}}{L^*},\ z = \frac{\tilde{z}}{L^*},\ u = \frac{\tilde{u}}{U^*},\ v = \frac{\tilde{v}}{U^*},\ w = \frac{\tilde{w}}{U^*}$$

$$t = \frac{\tilde{t}U^*}{L^*},\ \rho = \frac{\tilde{\rho}}{\rho^*},\ T = \frac{\tilde{T}}{T^*},\ p = \frac{\tilde{p}}{\rho^* U^{*2}},\ e_{\mathrm{T}} = \frac{\tilde{e}_{\mathrm{T}}}{U^{*2}},\ e = \frac{\tilde{e}}{U^{*2}}$$

$$f_x^{\mathrm{b}} = \frac{\tilde{f}_x^{\mathrm{b}}}{\rho^* U^{*2}/L^*},\ f_y^{\mathrm{b}} = \frac{\tilde{f}_y^{\mathrm{b}}}{\rho^* U^{*2}/L^*},\ f_z^{\mathrm{b}} = \frac{\tilde{f}_z^{\mathrm{b}}}{\rho^* U^{*2}/L^*}$$

Dimensional parameters

$$\tilde{\mu},\ \tilde{R}$$

Nondimensionalized parameters

$$\mu,\ R,\ C_v,\ C_p,\ c,\ k$$

Dimensionless parameters

$$Re,\ Pr,\ Ma$$

Dimensionless parameters based on reference variables

$$Re_{\mathrm{ref}},\ Ma_{\mathrm{ref}}$$

Constants

$$\gamma$$

While preserving the original form of equations (Eq. (A.4)) after normalization, the following parameters should be adopted for the nondimensionalized governing equations.

$$R = \frac{1}{\gamma Ma_{\mathrm{ref}}^2}$$

$$
\begin{aligned}
Ma_{\text{ref}} &= \frac{U^*}{\sqrt{\gamma \tilde{R} T^*}} \\
\mu &= \frac{1}{Re_{\text{ref}}} \\
Re_{\text{ref}} &= \frac{\rho^* U^* L^*}{\tilde{\mu}}
\end{aligned}
$$

A.2.2 Nonconservative vector-matrix form

The conservative vector form of Navier–Stokes equations can be transformed into a nonconservative vector-matrix form, which is useful for analyzing the mathematical properties of the equations.

Denote the vector of conservative variables and the vector of fluxes as

$$
\boldsymbol{U} = \begin{pmatrix} u_1 \\ \vdots \\ u_m \\ \vdots \\ u_M \end{pmatrix}, \ \boldsymbol{F} = \boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} f_1(u_1, \dots, u_m, \dots, u_M) \\ \vdots \\ f_m(u_1, \dots, u_m, \dots, u_M) \\ \vdots \\ f_M(u_1, \dots, u_m, \dots, u_M) \end{pmatrix} \tag{A.5}
$$

where $\boldsymbol{F}$ represents either a convective flux or a diffusive flux.

The flux component f_m is a multivariable function. After introducing vector notation for the multiple variables, it gives

$$
\mathrm{d}f_m = \frac{\partial f_m}{\partial u_n} \mathrm{d}u_n = (\nabla f_m)^{\mathrm{T}} \mathrm{d}\boldsymbol{U} \tag{A.6}
$$

Since the flux vector $\boldsymbol{F}$ is the vector notation of the set of multivariable functions, it has

$$
\mathrm{d}\boldsymbol{F} = \begin{pmatrix} \mathrm{d}f_1 \\ \vdots \\ \mathrm{d}f_m \\ \vdots \\ \mathrm{d}f_M \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} & \cdots & \frac{\partial f_1}{\partial u_M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial u_1} & \cdots & \frac{\partial f_m}{\partial u_m} & \cdots & \frac{\partial f_m}{\partial u_M} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial u_1} & \cdots & \frac{\partial f_M}{\partial u_m} & \cdots & \frac{\partial f_M}{\partial u_M} \end{pmatrix} \begin{pmatrix} \mathrm{d}u_1 \\ \vdots \\ \mathrm{d}u_m \\ \vdots \\ \mathrm{d}u_M \end{pmatrix} \tag{A.7}
$$

Then, it is straightforward to introduce the Jacobian matrices via vector differentiation

$$
\mathbf{A} = \frac{\mathrm{d}\boldsymbol{F}(\boldsymbol{U})}{\mathrm{d}\boldsymbol{U}}, \ A_{mn} = \frac{\partial F_m}{\partial U_n} \tag{A.8}
$$

A nonconservative vector-matrix form of the governing equations can be obtained as

$$
\frac{\partial \boldsymbol{U}}{\partial t} + \mathbf{A}_x \frac{\partial \boldsymbol{U}}{\partial x} + \mathbf{A}_y \frac{\partial \boldsymbol{U}}{\partial y} + \mathbf{A}_z \frac{\partial \boldsymbol{U}}{\partial z} = \frac{\partial \boldsymbol{F}_x^{\mathrm{v}}(\boldsymbol{U})}{\partial x} + \frac{\partial \boldsymbol{F}_y^{\mathrm{v}}(\boldsymbol{U})}{\partial y} + \frac{\partial \boldsymbol{F}_z^{\mathrm{v}}(\boldsymbol{U})}{\partial z} + \boldsymbol{\Phi} \tag{A.9}
$$

A.3 Jacobian matrices

A.3.1 Transformation

$$\boldsymbol{U} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_{\mathrm{T}} \end{pmatrix}$$

$$\boldsymbol{F}_x(\boldsymbol{U}) = \begin{pmatrix} u_2 \\ \frac{u_2^2}{u_1} + (\gamma - 1)[u_5 - \frac{1}{2}\frac{u_2^2+u_3^2+u_4^2}{u_1}] \\ \frac{u_2 u_3}{u_1} \\ \frac{u_2 u_4}{u_1} \\ \frac{u_5 u_2}{u_1} + (\gamma - 1)[u_5 - \frac{1}{2}\frac{u_2^2+u_3^2+u_4^2}{u_1}]\frac{u_2}{u_1} \end{pmatrix}$$

$$\boldsymbol{F}_y(\boldsymbol{U}) = \begin{pmatrix} u_3 \\ \frac{u_3 u_2}{u_1} \\ \frac{u_3^2}{u_1} + (\gamma - 1)[u_5 - \frac{1}{2}\frac{u_2^2+u_3^2+u_4^2}{u_1}] \\ \frac{u_3 u_4}{u_1} \\ \frac{u_5 u_3}{u_1} + (\gamma - 1)[u_5 - \frac{1}{2}\frac{u_2^2+u_3^2+u_4^2}{u_1}]\frac{u_3}{u_1} \end{pmatrix}$$

$$\boldsymbol{F}_z(\boldsymbol{U}) = \begin{pmatrix} u_4 \\ \frac{u_4 u_2}{u_1} \\ \frac{u_4 u_3}{u_1} \\ \frac{u_4^2}{u_1} + (\gamma - 1)[u_5 - \frac{1}{2}\frac{u_2^2+u_3^2+u_4^2}{u_1}] \\ \frac{u_5 u_4}{u_1} + (\gamma - 1)[u_5 - \frac{1}{2}\frac{u_2^2+u_3^2+u_4^2}{u_1}]\frac{u_4}{u_1} \end{pmatrix}$$

A.3.2 Definition of parameter

$$c = \sqrt{\gamma \frac{p}{\rho}}$$

$$q^2 = u^2 + v^2 + w^2$$

$$e = \frac{c^2}{\gamma(\gamma - 1)}$$

$$e_{\mathrm{T}} = e + \frac{1}{2}q^2 = \frac{c^2}{\gamma(\gamma - 1)} + \frac{1}{2}q^2$$

$$h = e + \frac{p}{\rho} = \frac{c^2}{\gamma - 1} = \gamma e$$

$$h_\mathrm{T} = e_\mathrm{T} + \frac{p}{\rho} = \gamma e + \frac{1}{2}q^2 = \gamma e_\mathrm{T} - \frac{1}{2}q^2(\gamma - 1)$$

$$b = \frac{\gamma - 1}{2c^2}$$

A.3.3 Jacobian matrix of convective flux

$$\mathbf{A} = \frac{\mathrm{d}\boldsymbol{F}(\boldsymbol{U})}{\mathrm{d}\boldsymbol{U}}, \quad A_{ij} = \frac{\partial F_i}{\partial U_j}$$

$$\mathbf{A}_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -u^2 + \frac{\gamma-1}{2}q^2 & 2u - (\gamma-1)u & -(\gamma-1)v & -(\gamma-1)w & \gamma-1 \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ -[\frac{c^2}{\gamma-1} + \frac{2-\gamma}{2}q^2]u & \frac{c^2}{\gamma-1} + \frac{1}{2}q^2 - (\gamma-1)u^2 & -(\gamma-1)uv & -(\gamma-1)uw & \gamma u \end{bmatrix}$$

$$\mathbf{A}_y = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -vu & v & u & 0 & 0 \\ -v^2 + \frac{\gamma-1}{2}q^2 & -(\gamma-1)u & 2v - (\gamma-1)v & -(\gamma-1)w & \gamma-1 \\ -vw & 0 & w & v & 0 \\ -[\frac{c^2}{\gamma-1} + \frac{2-\gamma}{2}q^2]v & -(\gamma-1)vu & \frac{c^2}{\gamma-1} + \frac{1}{2}q^2 - (\gamma-1)v^2 & -(\gamma-1)vw & \gamma v \end{bmatrix}$$

$$\mathbf{A}_z = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -wu & w & 0 & u & 0 \\ -wv & 0 & w & v & 0 \\ -w^2 + \frac{\gamma-1}{2}q^2 & -(\gamma-1)u & -(\gamma-1)v & 2w - (\gamma-1)w & \gamma-1 \\ -[\frac{c^2}{\gamma-1} + \frac{2-\gamma}{2}q^2]w & -(\gamma-1)wu & -(\gamma-1)wv & \frac{c^2}{\gamma-1} + \frac{1}{2}q^2 - (\gamma-1)w^2 & \gamma w \end{bmatrix}$$

A.4 Eigendecompositions

A.4.1 Eigenvalue

$$\mathbf{\Lambda}_x = {\mathbf{R}_x}^{-1}\mathbf{A}_x\mathbf{R}_x = \begin{bmatrix} u-c & & & & \\ & u & & & \\ & & u & & \\ & & & u & \\ & & & & u+c \end{bmatrix} \overset{\mathrm{def}}{=} \begin{bmatrix} \lambda_x^1 & & & & \\ & \lambda_x^2 & & & \\ & & \lambda_x^3 & & \\ & & & \lambda_x^4 & \\ & & & & \lambda_x^5 \end{bmatrix}$$

$$\mathbf{\Lambda}_y = \mathbf{R}_y{}^{-1}\mathbf{A}_y\mathbf{R}_y = \begin{bmatrix} v-c & & & & \\ & v & & & \\ & & v & & \\ & & & v & \\ & & & & v+c \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \lambda_y^1 & & & & \\ & \lambda_y^2 & & & \\ & & \lambda_y^3 & & \\ & & & \lambda_y^4 & \\ & & & & \lambda_y^5 \end{bmatrix}$$

$$\mathbf{\Lambda}_z = \mathbf{R}_z{}^{-1}\mathbf{A}_z\mathbf{R}_z = \begin{bmatrix} w-c & & & & \\ & w & & & \\ & & w & & \\ & & & w & \\ & & & & w+c \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \lambda_z^1 & & & & \\ & \lambda_z^2 & & & \\ & & \lambda_z^3 & & \\ & & & \lambda_z^4 & \\ & & & & \lambda_z^5 \end{bmatrix}$$

A.4.2 Eigenvector

$$\mathbf{R}_x = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ u-c & u & 0 & 0 & u+c \\ v & 0 & 1 & 0 & v \\ w & 0 & 0 & 1 & w \\ h_\mathrm{T} - uc & u^2 - \frac{1}{2}q^2 & v & w & h_\mathrm{T} + uc \end{bmatrix}$$

$$\mathbf{R}_y = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ u & 1 & 0 & 0 & u \\ v-c & 0 & v & 0 & v+c \\ w & 0 & 0 & 1 & w \\ h_\mathrm{T} - vc & u & v^2 - \frac{1}{2}q^2 & w & h_\mathrm{T} + vc \end{bmatrix}$$

$$\mathbf{R}_z = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ u & 1 & 0 & 0 & u \\ v & 0 & 1 & 0 & v \\ w-c & 0 & 0 & w & w+c \\ h_\mathrm{T} - wc & u & v & w^2 - \frac{1}{2}q^2 & h_\mathrm{T} + wc \end{bmatrix}$$

$$\mathbf{L}_x = \mathbf{R}_x{}^{-1} = \begin{bmatrix} \frac{1}{2}bq^2 + \frac{1}{2c}u & -bu - \frac{1}{2c} & -bv & -bw & b \\ -bq^2 + 1 & 2bu & 2bv & 2bw & -2b \\ -bq^2 v & 2bvu & 2bv^2 + 1 & 2bwv & -2bv \\ -bq^2 w & 2bwu & 2bwv & 2bw^2 + 1 & -2bw \\ \frac{1}{2}bq^2 - \frac{1}{2c}u & -bu + \frac{1}{2c} & -bv & -bw & b \end{bmatrix}$$

$$\mathbf{L}_y = \mathbf{R}_y{}^{-1} = \begin{bmatrix} \frac{1}{2}bq^2 + \frac{1}{2c}v & -bu & -bv - \frac{1}{2c} & -bw & b \\ -bq^2 u & 2bu^2 + 1 & 2bvu & 2bwu & -2bu \\ -bq^2 + 1 & 2bu & 2bv & 2bw & -2b \\ -bq^2 w & 2bwu & 2bwv & 2bw^2 + 1 & -2bw \\ \frac{1}{2}bq^2 - \frac{1}{2c}v & -bu & -bv + \frac{1}{2c} & -bw & b \end{bmatrix}$$

$$\mathbf{L}_z = {\mathbf{R}_z}^{-1} = \begin{bmatrix} \frac{1}{2}bq^2 + \frac{1}{2c}w & -bu & -bv & -bw - \frac{1}{2c} & b \\ -bq^2u & 2bu^2+1 & 2bvu & 2bwu & -2bu \\ -bq^2v & 2bvu & 2bv^2+1 & 2bwv & -2bv \\ -bq^2+1 & 2bu & 2bv & 2bw & -2b \\ \frac{1}{2}bq^2 - \frac{1}{2c}w & -bu & -bv & -bw + \frac{1}{2c} & b \end{bmatrix}$$

Due to the important role of eigenvalues, it is useful to present the convective flux as a function of eigenvalues according to the relation:

$$\boldsymbol{F} = \mathbf{A}\boldsymbol{U} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1}\boldsymbol{U}$$

which can be directly used for splitting the convective flux into subvectors associated with specified eigenvalue spectra [81].

$$\boldsymbol{F}_x = \frac{\rho}{2\gamma}\begin{pmatrix} \lambda_x^1 + 2(\gamma-1)\lambda_x^2 + \lambda_x^5 \\ [\lambda_x^1 + 2(\gamma-1)\lambda_x^2 + \lambda_x^5]u + (\lambda_x^5 - \lambda_x^1)c \\ [\lambda_x^1 + 2(\gamma-1)\lambda_x^3 + \lambda_x^5]v \\ [\lambda_x^1 + 2(\gamma-1)\lambda_x^4 + \lambda_x^5]w \\ [\lambda_x^1 + 2(\gamma-1)\lambda_x^2 + \lambda_x^5](\frac{1}{2}q^2) + (\lambda_x^1 + \lambda_x^5)h + (\lambda_x^5 - \lambda_x^1)cu \end{pmatrix}$$

$$\boldsymbol{F}_y = \frac{\rho}{2\gamma}\begin{pmatrix} \lambda_y^1 + 2(\gamma-1)\lambda_y^3 + \lambda_y^5 \\ [\lambda_y^1 + 2(\gamma-1)\lambda_y^2 + \lambda_y^5]u \\ [\lambda_y^1 + 2(\gamma-1)\lambda_y^3 + \lambda_y^5]v + (\lambda_y^5 - \lambda_y^1)c \\ [\lambda_y^1 + 2(\gamma-1)\lambda_y^4 + \lambda_y^5]w \\ [\lambda_y^1 + 2(\gamma-1)\lambda_y^3 + \lambda_y^5](\frac{1}{2}q^2) + (\lambda_y^1 + \lambda_y^5)h + (\lambda_y^5 - \lambda_y^1)cv \end{pmatrix}$$

$$\boldsymbol{F}_z = \frac{\rho}{2\gamma}\begin{pmatrix} \lambda_z^1 + 2(\gamma-1)\lambda_z^4 + \lambda_z^5 \\ [\lambda_z^1 + 2(\gamma-1)\lambda_z^2 + \lambda_z^5]u \\ [\lambda_z^1 + 2(\gamma-1)\lambda_z^3 + \lambda_z^5]v \\ [\lambda_z^1 + 2(\gamma-1)\lambda_z^4 + \lambda_z^5]w + (\lambda_z^5 - \lambda_z^1)c \\ [\lambda_z^1 + 2(\gamma-1)\lambda_z^4 + \lambda_z^5](\frac{1}{2}q^2) + (\lambda_z^1 + \lambda_z^5)h + (\lambda_z^5 - \lambda_z^1)cw \end{pmatrix}$$

Unified form via a direction vector $\boldsymbol{e} = (e_1, e_2, e_3)$

$$\boldsymbol{F} = \frac{\rho}{2\gamma}\begin{pmatrix} \lambda^1 + 2(\gamma-1)\lambda^2 + \lambda^5 \\ [\lambda^1 + 2(\gamma-1)\lambda^2 + \lambda^5]u + (\lambda^5 - \lambda^1)c * e_1 \\ [\lambda^1 + 2(\gamma-1)\lambda^3 + \lambda^5]v + (\lambda^5 - \lambda^1)c * e_2 \\ [\lambda^1 + 2(\gamma-1)\lambda^4 + \lambda^5]w + (\lambda^5 - \lambda^1)c * e_3 \\ [\lambda^1 + 2(\gamma-1)\lambda^2 + \lambda^5](\frac{1}{2}q^2) + (\lambda^1 + \lambda^5)h + (\lambda^5 - \lambda^1)c(u * e_1 + v * e_2 + w * e_3) \end{pmatrix}$$

A.4.3 Vector space

Eigenvector sets $\{\boldsymbol{L}_m\}$ and $\{\boldsymbol{R}^m\}$ of the Jacobian matrix $\mathbf{A}$ are two orthogonal vector spaces in Euclidean space $\mathbb{R}^M$ admitting the fact that

$$\boldsymbol{L}_m \cdot \boldsymbol{R}^n = \delta_m^n, \ \ m, n = 1, \ldots, M$$

The vector $\Delta \boldsymbol{U} = \boldsymbol{U}_\mathrm{R} - \boldsymbol{U}_\mathrm{L}$, which measures the strength of a jump discontinuity comprising a right state $\boldsymbol{U}_\mathrm{R}$ and a left state $\boldsymbol{U}_\mathrm{L}$, belongs to $\mathbb{R}^M$, therefore, it can be decomposed in the vector space $\{\boldsymbol{R}^m\}$:

$$\boldsymbol{U}_\mathrm{R} - \boldsymbol{U}_\mathrm{L} = \sum_{m=1}^{M} \alpha_m \boldsymbol{R}^m$$

where α_m are decomposition or projection coefficients, which can be determined by

$$\alpha_m = \boldsymbol{L}_m \cdot (\boldsymbol{U}_\mathrm{R} - \boldsymbol{U}_\mathrm{L})$$

Define vector $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)^\mathrm{T}$, then equivalently,

$$\boldsymbol{U}_\mathrm{R} - \boldsymbol{U}_\mathrm{L} = \mathbf{R}\boldsymbol{\alpha}$$

Therefore, the jump in flux values has

$$\begin{aligned}
\boldsymbol{F}_\mathrm{R} - \boldsymbol{F}_\mathrm{L} &= \mathbf{A}(\boldsymbol{U}_\mathrm{R} - \boldsymbol{U}_\mathrm{L}) \\
&= \mathbf{A}\mathbf{R}\alpha \\
&= \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^{-1}\mathbf{R}\alpha \\
&= \mathbf{R}\boldsymbol{\Lambda}\alpha \\
&= \sum_{m=1}^{M} \lambda_m \alpha_m \boldsymbol{R}^m
\end{aligned}$$

Appendix B

ArtraCFD: Additional Code Validation

In numerical solutions, errors are introduced in a variety of phases such as physical modeling, geometric representation, computer floating-point arithmetic, iteration, and numerical discretization. This chapter presents some additional validation of the developed computational fluid dynamics code, particularly concerning the vortex preservation problem and standard shock tube problems. The objective is to further demonstrate the validity and accuracy of the computational solver and its implementation. A serial version of the developed solver with about ten thousand lines of code is released as an open source project named as ArtraCFD and is available on Github [153].

In the following discussions, for the purpose of clarity, the notation "RKn-WENOm-XX" is used to represent a solution solved by the n-th order Runge–Kutta scheme for time integration, m-th order WENO scheme for spatial discretization, and "XX" for flux splitting with "XX=LF" and "XX=SW" representing the Lax–Friedrichs [80] and Steger–Warming [81] splitting, respectively.

B.1 Convergence test on smooth flow

B.1.1 Vortex preservation problem

The vortex preservation test case [139], which evolves an isentropic vortex in an inviscid uniform two-dimensional flow, is utilized to test the convergence behavior of the code for solving system of conservation laws in multidimensional space.

The problem is built on a computational domain given by $[-5R, 5R] \times [-5R, 5R]$ and is governed by the Euler equations. On a uniform background flow $(\rho_\infty, u_\infty, v_\infty, p_\infty)$, an isentropic vortex initially centered at $(x, y) = (0, 0)$ with radius $R = 5$ is created by adding the following perturbation to the mean flow

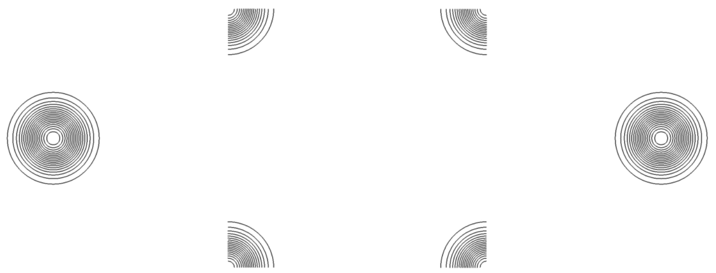
$$\begin{aligned}(\delta u, \delta v) &= \frac{\Gamma}{2\pi} e^{0.5(1-r^2)}(-y, x) \\ \delta T &= -\frac{(\gamma - 1)\Gamma^2}{8\gamma\pi^2} e^{(1-r^2)}\end{aligned} \tag{B.1}$$

where the vortex strength $\Gamma = 5$, the ratio of specific heat $\gamma = 1.4$, $r^2 = x^2 + y^2 \leq R^2$.

Under isentropic flow condition, the temperature and entropy are defined as $T = p/\rho$ and $s = p/\rho^\gamma$, respectively. When $s = 1$ and $\delta s = 0$ are assumed for the flow, the initial conditions are taken as follows

$$u = u_\infty + \delta u,\ v = v_\infty + \delta v,\ T = p_\infty/\rho_\infty + \delta T,\ \rho = T^{\frac{1}{\gamma-1}}, p = T^{\frac{\gamma}{\gamma-1}} \tag{B.2}$$

The background flow is chosen as $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (1, 1, 1, 1)$ such that the vortex propagates at 45° to the grid line, which amplifies the opportunity for manifesting the effects of multidimensional propagation. The solution is solved to $t = 10$, and the CFL coefficient is set to $C_{\text{CFL}} = 0.2$. Periodic boundary conditions are used for boundary treatment of both directions. Since the perturbation induced by the vortex is weak enough to avoid producing a strongly nonlinear effect, the exact solution of the problem with a specific initial state is the passive convection of the vortex with the mean velocity. Therefore, the solution should remain unchanged when time evolves to complete periods of vortex convection, as demonstrated by the computed results shown in Fig. B.1.



(a) $t = 0$ (b) $t = 5$ (c) $t = 10$

Figure B.1 Density contour of the propagating isentropic vortex solved by RK3-WENO5-LF on a 100 × 100 grid at different time instants.

RK3-WENO3-LF and RK3-WENO5-LF with dimension-by-dimension or dimensional-splitting approximation are tested and are shown in Table B.1, Table B.2, Table B.3, and Table B.4, respectively. For the successively refined grids used herein, the chosen CFL number is small enough to produce a temporal step size that holds down the errors from temporal discretization and avoids affecting the spatial accuracy. As captured by the results, the code is operating closely to its designed accuracy. In addition, the observed global convergence rates of RK3-WENO5-LF agrees with the results in [139]. The current results present very similar accuracy for the dimension-by-dimension and dimensional-splitting approaches for solving multidimensional problems.

$N \times N$	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
25×25	2.164e−2	–	5.048e−2	–	3.855e−1	–
50×50	1.149e−2	0.9	2.819e−2	0.8	2.015e−1	0.9
75×75	5.281e−3	1.9	1.315e−2	1.9	8.889e−2	2.0
100×100	2.651e−3	2.4	6.814e−3	2.3	4.793e−2	2.1
150×150	9.221e−4	2.6	2.519e−3	2.5	2.031e−2	2.1

Table B.1 Global convergence of RK3-WENO3-LF via dimension-by-dimension approximation.

$N \times N$	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
25×25	2.164e−2	–	5.048e−2	–	3.855e−1	–
50×50	1.149e−2	0.9	2.819e−2	0.8	2.014e−1	0.9
75×75	5.281e−3	1.9	1.315e−2	1.9	8.889e−2	2.0
100×100	2.651e−3	2.4	6.813e−3	2.3	4.792e−2	2.1
150×150	9.220e−4	2.6	2.519e−3	2.5	2.031e−2	2.1

Table B.2 Global convergence of RK3-WENO3-LF via dimensional-splitting approximation.

$N \times N$	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
25×25	6.609e−3	–	1.521e−2	–	1.091e−1	–
50×50	4.677e−4	3.8	1.195e−3	3.7	1.680e−2	2.7
75×75	9.536e−5	3.9	2.281e−4	4.1	2.577e−3	4.6
100×100	3.349e−5	3.6	6.909e−5	4.2	7.150e−4	4.5
150×150	1.088e−5	2.8	1.659e−5	3.5	9.985e−5	4.9

Table B.3 Global convergence of RK3-WENO5-LF via dimension-by-dimension approximation.

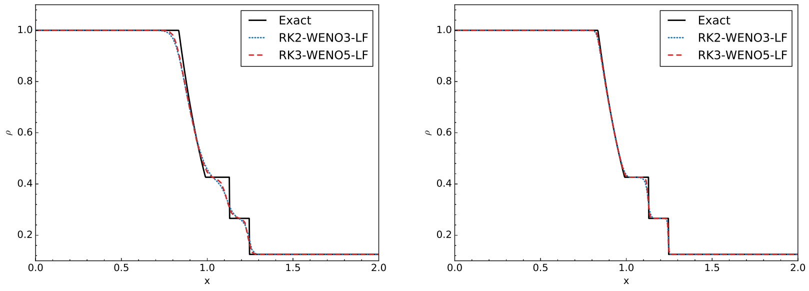
$N \times N$	L_1 error	L_1 order	L_2 error	L_2 order	L_∞ error	L_∞ order
25×25	6.614e−3	–	1.522e−2	–	1.092e−1	–
50×50	4.682e−4	3.8	1.195e−3	3.7	1.677e−2	2.7
75×75	9.520e−5	3.9	2.278e−4	4.1	2.571e−3	4.6
100×100	3.337e−5	3.6	6.891e−5	4.2	7.109e−4	4.5
150×150	1.084e−5	2.8	1.652e−5	3.5	9.757e−5	4.9

Table B.4 Global convergence of RK3-WENO5-LF via dimensional-splitting approximation.

B.2 Test cases with shocked flow

Sod's problem [154]

$$
\begin{array}{llll}
\rho = 1; & u = 0; & p = 1 & \text{if } 0 \le x < 1 \\
\rho = 0.125; & u = 0; & p = 0.1 & \text{if } 1 < x \le 2
\end{array}
\tag{B.3}
$$
$$
t \in [0, 0.14], \quad C_{\text{CFL}} = 0.6
$$

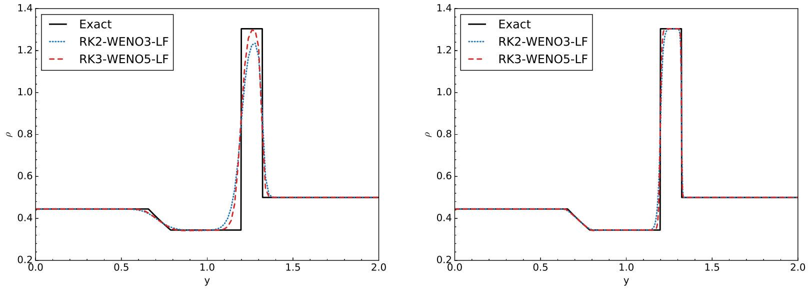


(a) (b)

Figure B.2 Numerical solutions of Sod's problem. (a) $n = 100$. (b) $n = 500$.

Lax's problem [82]

$$
\begin{array}{llll}
\rho = 0.445; & u = 0.698; & p = 3.528 & \text{if } 0 \le x < 1 \\
\rho = 0.5; & u = 0; & p = 0.571 & \text{if } 1 < x \le 2
\end{array}
\tag{B.4}
$$
$$
t \in [0, 0.13], \quad C_{\text{CFL}} = 0.6
$$

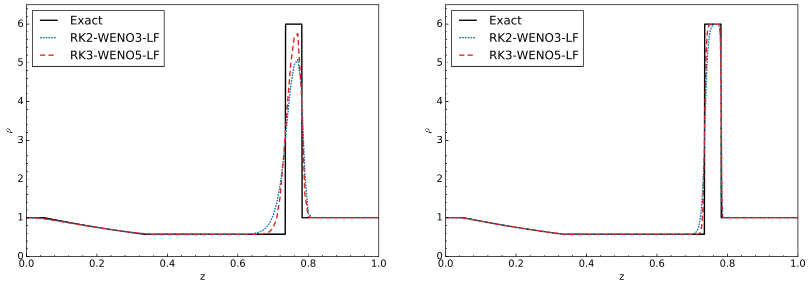


(a) (b)

Figure B.3 Numerical solutions of Lax's problem. (a) $n = 100$. (b) $n = 500$.

Woodward and Colella problem [155]

$$\begin{aligned} &\rho = 1; \quad u = 0; \quad p = 1000 \quad \text{if } 0 \le x < 0.5 \\ &\rho = 1; \quad u = 0; \quad p = 0.01 \quad \text{if } 0.5 < x \le 1 \\ &t \in [0, 0.012], \quad C_{\text{CFL}} = 0.6 \end{aligned} \tag{B.5}$$



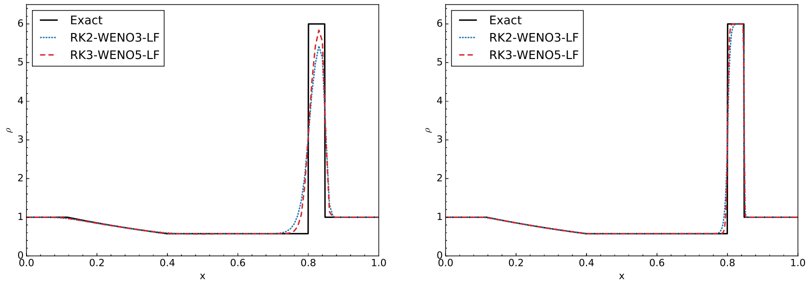
(a)

(b)

Figure B.4 Numerical solutions of Woodward and Colella problem. (a) $n = 100$. (b) $n = 500$.

Stationary contact [155]

$$\begin{aligned} &\rho = 1; \quad u = -19.59745; \quad p = 1000 \quad \text{if } 0 \le x < 0.8 \\ &\rho = 1; \quad u = -19.59745; \quad p = 0.01 \quad \text{if } 0.8 < x \le 1 \\ &t \in [0, 0.012], \quad C_{\text{CFL}} = 0.6 \end{aligned} \tag{B.6}$$



(a)

(b)

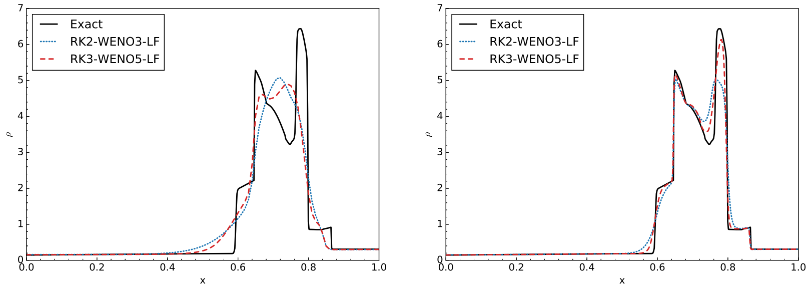
Figure B.5 Numerical solutions of stationary contact problem. (a) $n = 100$. (b) $n = 500$.

Remarks Exact solutions are computed by the exact Riemann solver provided in Toro [155].

Two interacting blast waves [82, 156]

$$\boldsymbol{U}(x,0) = \begin{cases} \boldsymbol{U}_{\mathrm{L}}, & \text{if } 0 < x < 0.1 \\ \boldsymbol{U}_{\mathrm{M}}, & \text{if } 0.1 < x < 0.9 \\ \boldsymbol{U}_{\mathrm{R}}, & \text{if } 0.9 < x < 1 \end{cases}, \quad \begin{cases} (\rho_{\mathrm{L}}, u_{\mathrm{L}}, p_{\mathrm{L}}) = (1, 0, 1000) \\ (\rho_{\mathrm{M}}, u_{\mathrm{M}}, p_{\mathrm{M}}) = (1, 0, 0.01) \\ (\rho_{\mathrm{R}}, u_{\mathrm{R}}, p_{\mathrm{R}}) = (1, 0, 100) \end{cases} \tag{B.7}$$

$$t \in [0, 0.038], \quad C_{\mathrm{CFL}} = 0.6$$



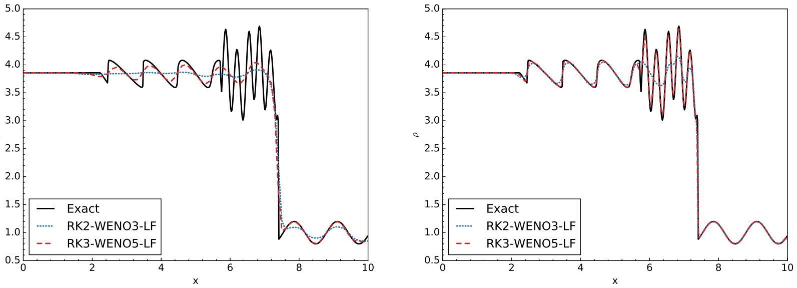
(a) (b)

Figure B.6 Numerical solutions of two interacting blast waves. (a) $n = 100$. (b) $n = 500$. [The "exact" solution is computed by RK3-WENO5-LF with 4096 nodes.

Shu and Osher problem [157]

$$\begin{aligned} &\rho = 3.857143; && u = 2.629369; && p = 10.333333 && \text{if } 0 \le x < 1 \\ &\rho = 1 + 0.2\sin(5x); && u = 0; && p = 1 && \text{if } 1 < x \le 10 \end{aligned} \tag{B.8}$$

$$t \in [0, 1.8], \quad C_{\mathrm{CFL}} = 0.6$$



(a) (b)

Figure B.7 Numerical solutions of Shu and Osher problem. (a) $n = 100$. (b) $n = 500$. [The "exact" solution is computed by RK3-WENO5-LF with 4096 nodes.]

2D Riemann problem [158]

$$
\begin{aligned}
&\rho = 1; \quad u = -0.75; \quad v = -0.5; \quad p = 1 \quad \text{if } 0.5 < x < 1,\ 0.5 < y < 1\\
&\rho = 2; \quad u = -0.75; \quad v = 0.5; \quad p = 1 \quad \text{if } 0 < x < 0.5,\ 0.5 < y < 1\\
&\rho = 1; \quad u = 0.75; \quad v = 0.5; \quad p = 1 \quad \text{if } 0 < x < 0.5,\ 0 < y < 0.5\\
&\rho = 3; \quad u = 0.75; \quad v = -0.5; \quad p = 1 \quad \text{if } 0.5 < x < 1,\ 0 < y < 0.5\\
&\qquad\qquad t \in [0, 0.23], \quad C_{\mathrm{CFL}} = 0.6
\end{aligned}
\tag{B.9}
$$

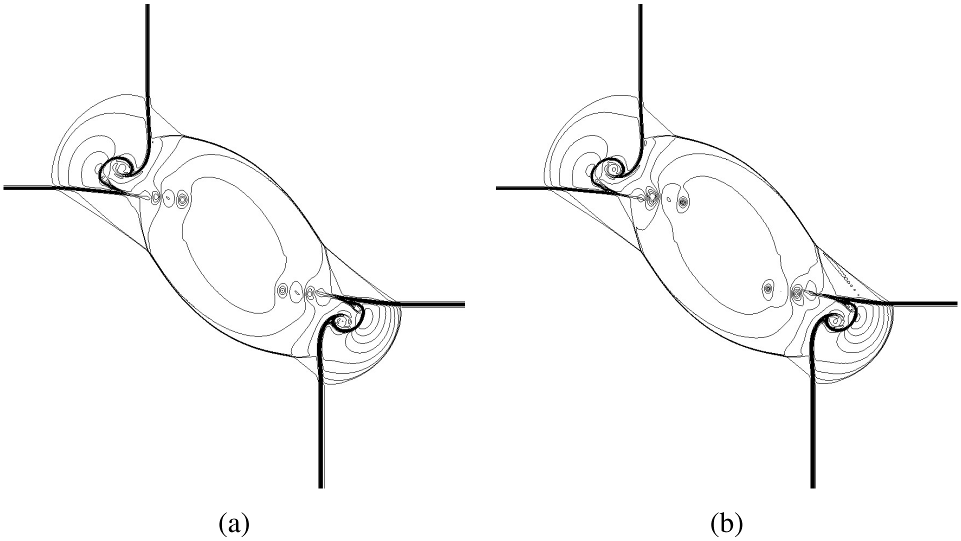


Figure B.8 Density contour of 2D Riemann problem with configuration 1. (a) RK3-WENO5-LF. (b) RK3-WENO5-SW.

References

[1] W. Fickett, C. Davis, Detonation, University of California Press, Berkeley, 1979.

[2] J. B. Bdzil, High-explosives performance, Los Alamos Ser. Basic Appl. Sci. 28 (2003) 96.

[3] R. Ripley, F. Zhang, F.-S. Lien, Acceleration and heating of metal particles in condensed matter detonation, Proc. R. Soc. London, Ser. A 468 (2142) (2012) 1564–1590.

[4] F. Zhang, R. Ripley, A. Yoshinaka, C. Findlay, J. Anderson, B. von Rosen, Large-scale spray detonation and related particle jetting instability phenomenon, Shock Waves 25 (3) (2015) 239–254.

[5] J. D. Anderson Jr, Fundamentals of Aerodynamics, Tata McGraw-Hill Education, New York, 2010.

[6] J. Kaca, An interferometric investigation of the diffraction of a planar shock wave over a semicircular cylinder, Tech. Rep., University of Toronto Institute for Aerospace Studies, 269, 1988.

[7] V. Rodriguez, R. Saurel, G. Jourdan, L. Houas, Study on the jet formation during dispersal of solid particles by shock and blast waves, in: 30th Int. Symp. Shock Waves, Springer, 745–748, 2017.

[8] F. Zhang, D. Frost, P. Thibault, S. Murray, Explosive dispersal of solid particles, Shock Waves 10 (6) (2001) 431–443.

[9] D. L. Frost, Y. Grégoire, O. Petel, S. Goroshin, F. Zhang, Particle jet formation during explosive dispersal of solid particles, Phys. Fluids 24 (9) (2012) 091109.

[10] G. Roy, S. Frolov, A. Borisov, D. Netzer, Pulse detonation propulsion: challenges, current status, and future perspective, Prog. Energy Combust. Sci. 30 (6) (2004) 545–672.

[11] V. Kedrinskiy, Hydrodynamic aspects of explosive eruptions of volcanoes: simulation problems, Shock Waves 18 (6) (2009) 451–464.

[12] S. Rigby, S. Fay, S. Clarke, A. Tyas, J. Reay, J. Warren, M. Gant, I. Elgy, Measuring spatial pressure distribution from explosives buried in dry Leighton Buzzard sand, Int. J. Impact Eng. 96 (2016) 89–104.

[13] S. Rigby, S. Fay, A. Tyas, S. Clarke, J. Reay, J. Warren, M. Gant, I. Elgy, Influence of particle size distribution on the blast pressure profile from explosives buried in saturated soils, Shock Waves 28 (3) (2018) 613–626.

[14] J. R. Royer, E. I. Corwin, A. Flior, M.-L. Cordero, M. L. Rivers, P. J. Eng, H. M. Jaeger, Formation of granular jets observed by high-speed X-ray radiography, Nature Physics 1 (3) (2005) 164.

[15] A. Burrows, Supernova explosions in the universe, Nature 403 (6771) (2000) 727.

[16] M. Baer, J. Nunziato, A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials, Int. J. Multiph. Flow 12 (6) (1986) 861–889.

[17] A. Milne, C. Parrish, I. Worland, Dynamic fragmentation of blast mitigants, Shock Waves 20 (1) (2010) 41–51.

[18] F. Zhang, Metalized heterogeneous detonation and dense reactive particle flow, AIP Conf. Proc. 1426 (1) (2012) 27–34.

[19] P.-H. Chiu, K. L. Olney, D. J. Benson, C. Braithwaite, A. Collins, V. F. Nesterenko, Dynamic fragmentation of Al-W granular rings with different mesostructures, J. Appl. Phys. 121 (4) (2017) 045901.

[20] F. Zhang, A. Yoshinaka, R. Ripley, Hybrid detonation waves in metalized explosive mixtures, in: Proc. 14th Int. Detonation Symp., 11–16, 2010.

[21] K. Xue, F. Li, C. Bai, Explosively driven fragmentation of granular materials, Eur. Phys. J. E 36 (8) (2013) 95.

[22] C.-H. Bai, Y. Wang, K. Xue, L.-F. Wang, Experimental study of detonation of large-scale powder–droplet–vapor mixtures, Shock Waves (2018) 1–13.

[23] D. L. Frost, J. Loiseau, S. Goroshin, F. Zhang, A. Milne, A. Longbottom, Fracture of explosively compacted aluminum particles in a cylinder, AIP Conf. Proc. 1793 (1) (2017) 120019.

[24] V. Rodriguez, R. Saurel, G. Jourdan, L. Houas, Solid-particle jet formation under shock-wave acceleration, Phys. Rev. E 88 (6) (2013) 063011.

[25] V. Rodriguez, R. Saurel, G. Jourdan, L. Houas, External front instabilities induced by a shocked particle ring, Phys. Rev. E 90 (4) (2014) 043013.

[26] K. Xue, K. Du, X. Shi, Y. Ganb, C. Baia, Dual hierarchical particle jetting of a particle ring undergoing radial explosion, Soft Matter 14 (2018) 4422–4431.

[27] X. Cheng, L. Xu, A. Patterson, H. M. Jaeger, S. R. Nagel, Towards the zero-surface-tension limit in granular fingering instability, Nat. Phys. 4 (3) (2008) 234.

[28] P. Saffman, G. Taylor, The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid, Proc. R. Soc. Lond. A 245 (1242) (1958) 312–329.

[29] Ø. Johnsen, R. Toussaint, K. J. Måløy, E. G. Flekkøy, Pattern formation during air injection into granular materials confined in a circular Hele-Shaw cell, Phys. Rev. E 74 (1) (2006) 011301.

[30] R. Lord, Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density, Proc. Lond. Math. Soc. 14 (1883) 170–177.

[31] G. I. Taylor, The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I, Proc. R. Soc. Lond. A 201 (1065) (1950) 192–196.

[32] R. D. Richtmyer, Taylor instability in shock acceleration of compressible fluids, Comm. Pure Appl. Math. 13 (2) (1960) 297–319.

[33] E. Meshkov, Instability of the interface of two gases accelerated by a shock wave, Fluid Dyn. 4 (5) (1969) 101–104.

[34] H.-J. Kull, Theory of the Rayleigh-Taylor instability, Phys. Rep. 206 (5) (1991) 197–325.

[35] C.-Y. Wang, R. A. Chevalier, Instabilities and clumping in type IA supernova remnants, Astrophys. J. 549 (2) (2001) 1119.

[36] J. D. Lindl, R. L. McCrory, E. M. Campbell, Progress toward ignition and burn propagation in inertial confinement fusion, Phys. Today 45 (9) (1992) 32–40.

[37] J. Lindl, Development of the indirect-drive approach to inertial confinement fusion and the target physics basis for ignition and gain, Phys. Plasmas 2 (11) (1995) 3933–4024.

[38] M. A. van der Hoef, M. van Sint Annaland, N. Deen, J. Kuipers, Numerical simulation of dense gas-solid fluidized beds: a multiscale modeling strategy, Annu. Rev. Fluid Mech. 40 (2008) 47–70.

[39] C. E. Brennen, Fundamentals of Multiphase Flow, Cambridge University Press, 2005.

[40] P. Jop, Y. Forterre, O. Pouliquen, A constitutive law for dense granular flows, Nature 441 (7094) (2006) 727.

[41] N. V. Brilliantov, T. Pöschel, Kinetic Theory of Granular Gases, Oxford University Press, 2010.

[42] K. Agrawal, P. N. Loezos, M. Syamlal, S. Sundaresan, The role of meso-scale structures in rapid gas–solid flows, J. Fluid Mech. 445 (2001) 151–185.

[43] S. Tenneti, S. Subramaniam, Particle-resolved direct numerical simulation for gas-solid flow model development, Annu. Rev. Fluid Mech. 46 (2014) 199–230.

[44] Y. Tsuji, T. Kawaguchi, T. Tanaka, Discrete particle simulation of two-dimensional fluidized bed, Powder Technol. 77 (1) (1993) 79–87.

[45] C. Stewart, S. Balachandar, T. P. McGrath, Soft-sphere simulations of a planar shock interaction with a granular bed, Phys. Rev. Fluids 3 (3) (2018) 034308.

[46] Y. Ling, J. Wagner, S. Beresh, S. Kearney, S. Balachandar, Interaction of a planar shock wave with a dense particle curtain: Modeling and experiments, Phys. Fluids 24 (11) (2012) 113301.

[47] E. P. DeMauro, J. L. Wagner, S. J. Beresh, P. A. Farias, Unsteady drag following shock wave impingement on a dense particle curtain measured using pulse-burst PIV, Phys. Rev. Fluids 2 (6) (2017) 064301.

[48] Y. Mehta, T. Jackson, J. Zhang, S. Balachandar, Numerical investigation of shock interaction with one-dimensional transverse array of particles in air, J. Appl. Phys. 119 (10) (2016) 104901.

[49] T. McGrath, J. S. Clair, S. Balachandar, Modeling compressible multiphase flows with dispersed particles in both dense and dilute regimes, Shock Waves 28 (3) (2018) 533–544.

[50] O. Sen, N. J. Gaul, K. Choi, G. Jacobs, H. Udaykumar, Evaluation of multifidelity surrogate modeling techniques to construct closure laws for drag in shock-particle interactions, J. Comput. Phys. 371 (2018) 434–451.

[51] C. S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.

[52] R. J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (3) (1997) 709–735.

[53] J. A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, J. Comput. Phys. 163 (2) (2000) 489–528.

[54] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (2) (1999) 457–492.

[55] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, J. Comput. Phys. 175 (1) (2002) 200–224.

[56] K. Wang, A. Rallu, J.-F. Gerbeau, C. Farhat, Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid–structure interaction problems, Int. J. Numer. Methods Fluids 67 (9) (2011) 1175–1206.

[57] K. Wang, P. Lea, C. Farhat, A computational framework for the simulation of high-speed multi-material fluid–structure interaction problems with dynamic fracture, Int. J. Numer. Methods Eng. 104 (7) (2015) 585–623.

[58] E. A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.

[59] Y.-H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2) (2003) 593–623.

[60] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[61] A. Kapahi, S. Sambasivan, H. Udaykumar, A three-dimensional sharp interface Cartesian grid method for solving high-speed multimaterial impact, penetration and fragmentation problems, J. Comput. Phys. 241 (2013) 308–332.

[62] H. Uddin, R. Kramer, C. Pantano, A Cartesian-based embedded geometry technique with adaptive high-order finite differences for compressible flow around complex geometries, J. Comput. Phys. 262 (2014) 379–407.

[63] O. R. Walton, Numerical simulation of inelastic, frictional particle-particle interactions, Particulate Two-Phase Flow 25 (1993) 884–911.

[64] Y. Ling, A. Haselbacher, S. Balachandar, Importance of unsteady contributions to force and heating for particles in compressible flows: Part 1: Modeling and analysis for shock–particle interaction, Int. J. Multiph. Flow 37 (9) (2011) 1026–1044.

[65] T. Kempe, B. Vowinckel, J. Fröhlich, On the relevance of collision modeling for interface-resolving simulations of sediment transport in open channel flow, Int. J. Multiph. Flow 58 (2014) 214–235.

[66] S. R. Nagel, Experimental soft-matter science, Rev. Mod. Phys. 89 (2) (2017) 025002.

[67] A. Ivanov, On multiple impact, J. Appl. Math. Mech. 59 (6) (1995) 887–902.

[68] E. Guendelman, R. Bridson, R. Fedkiw, Nonconvex rigid bodies with stacking, ACM Trans. Graph. 22 (3) (2003) 871–878.

[69] T. Xu, F.-S. Lien, H. Ji, F. Zhang, Formation of particle jetting in a cylindrical shock tube, Shock Waves 23 (6) (2013) 619–634.

[70] R. W. Gurney, The Initial Velocities of Fragments from Bombs, Shell, Grenades., Tech. Rep., Army Ballistic Research Laboratory, 1943.

[71] H. Mo, F.-S. Lien, F. Zhang, D. Cronin, A simple field function for solving complex and dynamic fluid-solid system on Cartesian grid, arXiv:1702.02474 .

[72] H. Mo, F.-S. Lien, F. Zhang, D. Cronin, An immersed boundary method for solving compressible flow with arbitrarily irregular and moving geometry, Int. J. Numer. Methods Fluids 88 (5) (2018) 239–263.

[73] H. Mo, F.-S. Lien, F. Zhang, D. Cronin, A numerical framework for the direct simulation of dense particulate flow under explosive dispersal, Shock Waves 28 (3) (2018) 559–577.

[74] H. Mo, F.-S. Lien, F. Zhang, D. Cronin, Direct simulation of explosive dispersal of dense particles, in: Proc. 24th Int. Symp. Mil. Aspects Blast Shock, Halifax, Canada, 2016.

[75] H. Mo, F.-S. Lien, F. Zhang, D. Cronin, A mesoscale study on explosively dispersed granular materials, in: Proc. 16th Int. Detonation Symp., Maryland, United States, 2018.

[76] D. Goldman, T. J. Kaper, Nth-order operator splitting schemes and nonreversible systems, SIAM J. Numer. Anal. 33 (1) (1996) 349–367.

[77] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (3) (1968) 506–517.

[78] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471.

[79] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.

[80] P. D. Lax, Hyperbolic systems of conservation laws and the mathematical theory of shock waves, Comm. Pure Appl. Math. 10 (4) (1957) 537–566.

[81] J. L. Steger, R. Warming, Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods, J. Comput. Phys. 40 (2) (1981) 263–293.

[82] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1).

[83] S. Pirozzoli, Numerical methods for high-speed flows, Annu. Rev. Fluid Mech. 43 (2011) 163–194.

[84] C. Brehm, C. Hader, H. F. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, J. Comput. Phys. 295 (2015) 475–504.

[85] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225.

[86] R. Saurel, O. Lemetayer, A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation, J. Fluid Mech. 431 (2001) 239–271.

[87] A. Prosperetti, G. Tryggvason, Computational Methods for Multiphase Flow, Cambridge University Press, 2009.

[88] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.

[89] S. Osher, R. P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2) (2001) 463–502.

[90] D. Enright, R. P. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (1) (2002) 83–116.

[91] S. E. Hieber, P. Koumoutsakos, A Lagrangian particle level set method, J. Comput. Phys. 210 (1) (2005) 342–367.

[92] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2) (2001) 708–759.

[93] J. A. Baerentzen, H. Aanaes, Signed distance computation using the angle weighted pseudonormal, IEEE Trans. Vis. Comput. Graph. 11 (3) (2005) 243–253.

[94] M. W. Jones, J. A. Bærentzen, M. Sramek, 3D distance fields: a survey of techniques and applications, IEEE Trans. Vis. Comput. Graph. 12 (4) (2006) 581–599.

[95] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (2) (1999) 410–438.

[96] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct Numerical Simulations of Gas-Liquid Multiphase Flows, Cambridge University Press, 2011.

[97] R. Glowinski, T. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2) (2001) 363–426.

[98] P. Das, O. Sen, G. Jacobs, H. Udaykumar, A sharp interface Cartesian grid method for viscous simulation of shocked particle-laden flows, Int. J. Comput. Fluid Dyn. 31 (6-8) (2017) 269–291.

[99] S. Murman, M. J. Aftosmis, M. Berger, Simulations of 6-DOF motion with a Cartesian method, in: 41st AIAA Aerosp. Sci. Meet. Exhib., 1246, 2003.

[100] G. R. Anderson, M. Nemec, M. J. Aftosmis, Aerodynamic shape optimization benchmarks with error control and automatic parameterization, in: 53rd AIAA Aerosp. Sci. Meet., 1719, 2015.

[101] J. J. Chiew, M. J. Aftosmis, Efficient simulation of multi-rotor vehicles with low Reynolds number propellers, in: 2018 AIAA Appl. Aerodyn. Conf., 4119, 2018.

[102] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, R. Fedkiw, Two-way coupling of fluids to rigid and deformable solids and shells, ACM Trans. Graph. 27 (3) (2008) 46.

[103] A. Gilmanov, F. Sotiropoulos, E. Balaras, A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, J. Comput. Phys. 191 (2) (2003) 660–669.

[104] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, Prog. Aerosp. Sci. 65 (2014) 1–21.

[105] M. J. Aftosmis, M. J. Berger, J. E. Melton, Robust and efficient Cartesian mesh generation for component-based geometry, AIAA J. 36 (6) (1998) 952–960.

[106] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (3) (2003) 331–347.

[107] S. Murman, M. J. Aftosmis, M. Nemec, Automated parameter studies using a Cartesian method, in: 22nd AIAA Appl. Aerodyn. Conf. Exhib., 5076, 2004.

[108] Y. Zhang, C. Zhou, An immersed boundary method for simulation of inviscid compressible flows, Int. J. Numer. Methods Fluids 74 (11) (2014) 775–793.

[109] S. K. Sambasivan, H. Udaykumar, Ghost fluid method for strong shock interactions Part 2: immersed solid boundaries, AIAA J. 47 (12) (2009) 2923–2937.

[110] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiph. Flow 25 (5) (1999) 755–794.

[111] P. Moreno, F. Bombardelli, 3D numerical simulation of particle-particle collisions in saltation mode near stream beds, Acta Geophys. 60 (6) (2012) 1661–1688.

[112] H. H. Hu, N. A. Patankar, M. Zhu, Direct numerical simulations of fluid–solid systems using the Arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2) (2001) 427–462.

[113] D. Gidaspow, Multiphase Flow and Fluidization: Continuum and Kinetic Theory Descriptions, Academic Press, 1994.

[114] C. M. Lin, S. Gottschalk, Collision detection between geometric models: a survey, IMA Conf. Math. Surf. 1 (1998) 602–608.

[115] P. Jiménez, F. Thomas, C. Torras, 3D collision detection: a survey, Comput. Graph. 25 (2) (2001) 269–285.

[116] I. Wald, S. Boulos, P. Shirley, Ray tracing deformable scenes using dynamic bounding volume hierarchies, ACM Trans. Graph. 26 (1) (2007) 6.

[117] C. Ericson, Real-Time Collision Detection, CRC Press, 2004.

[118] X. Wang, W. K. Liu, Extended immersed boundary method using FEM and RKPM, Comput. Methods Appl. Mech. Eng. 193 (12) (2004) 1305–1321.

[119] Y. Mori, C. S. Peskin, Implicit second-order immersed boundary methods with boundary mass, Comput. Methods Appl. Mech. Eng. 197 (25) (2008) 2049–2067.

[120] C.-C. Liao, C.-A. Lin, Simulations of natural and forced convection flows with moving embedded object using immersed boundary method, Comput. Methods Appl. Mech. Eng. 213 (2012) 58–70.

[121] T. Kempe, M. Lennartz, S. Schwarz, J. Fröhlich, Imposing the free-slip condition with a continuous forcing immersed boundary method, J. Comput. Phys. 282 (2015) 183–209.

[122] S. Schwarz, T. Kempe, J. Fröhlich, An immersed boundary method for the simulation of bubbles with varying shape, J. Comput. Phys. 315 (2016) 124–149.

[123] C. Hirt, A. A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.

[124] C. S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.

[125] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, CTR-Annual Research Briefs-Stanford Univ./NASA Ames .

[126] E. Balaras, modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Comput. Fluids 33 (3) (2004) 375–404.

[127] T. Gao, Y.-H. Tseng, X.-Y. Lu, An improved hybrid Cartesian/immersed boundary method for fluid-solid flows, Int. J. Numer. Methods Fluids 55 (12) (2007) 1189–1211.

[128] R. Crockett, P. Colella, D. T. Graves, A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, J. Comput. Phys. 230 (7) (2011) 2451–2469.

[129] R. Courant, K. O. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, Math. Ann. 100.

[130] J. o'Rourke, Computational Geometry in C, Cambridge University Press, 1998.

[131] P. C. P. Carvalho, P. R. Cavalcanti, Point in polyhedron testing using spherical polygons, Graphics Gems V (1995) 42–49.

[132] E. Haines, Point in polygon strategies, Graphics Gems IV 994 (1994) 24–26.

[133] G. I. Taylor, J. Maccoll, The air pressure on a cone moving at high speeds. - I, Proc. R. Soc. Lond. A 139 (838) (1933) 278–297.

[134] H. Tanno, K. Itoh, T. Saito, A. Abe, K. Takayama, Interaction of a shock with a sphere suspended in a vertical shock tube, Shock Waves 13 (3) (2003) 191–200.

[135] A. Bryson, R. Gross, Diffraction of strong shocks by cones, cylinders, and spheres, J. Fluid Mech. 10 (01) (1961) 1–16.

[136] T. Bazhenova, L. Gvozdeva, M. Nettleton, Unsteady interactions of shock waves, Prog. Aerosp. Sci. 21 (1984) 249–331.

[137] R. Ripley, F.-S. Lien, M. Yovanovich, Numerical simulation of shock diffraction on unstructured meshes, Comput. Fluids 35 (10) (2006) 1420–1431.

[138] H. Ji, F.-S. Lien, E. Yee, Numerical simulation of detonation using an adaptive Cartesian cut-cell method combined with a cell-merging technique, Comput. Fluids 39 (6) (2010) 1041–1057.

[139] D. S. Balsara, C.-W. Shu, Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy, J. Comput. Phys. 160 (2) (2000) 405–452.

[140] H. Ji, F.-S. Lien, E. Yee, A robust and efficient hybrid cut-cell/ghost-cell method with adaptive mesh refinement for moving boundaries on irregular domains, Comput. Methods Appl. Mech. Eng. 198 (3) (2008) 432–448.

[141] A. Mark, B. G. van Wachem, Derivation and validation of a novel implicit second-order accurate immersed boundary method, J. Comput. Phys. 227 (13) (2008) 6660–6680.

[142] J. H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J. Comput. Phys. 230 (19) (2011) 7347–7363.

[143] W. Hörmann, J. Leydold, Continuous random variate generation by fast numerical inversion, ACM Trans. Model. Comput. Simul. 13 (4) (2003) 347–362.

[144] R. Bridson, Fast Poisson disk sampling in arbitrary dimensions., ACM SIGGRAPH Sketches (2007) 22.

[145] M. S. Ebeida, A. A. Davidson, A. Patney, P. M. Knupp, S. A. Mitchell, J. D. Owens, Efficient maximal Poisson-disk sampling, ACM Trans. Graph. 30 (4) (2011) 49.

[146] A. M. Milne, Gurney analysis of porous shells, Propellants Explos. Pyrotech. 41 (4) (2016) 665–671.

[147] P. W. Cooper, Explosives Engineering, Wiley, 1996.

[148] R. Ripley, L. Donahue, F. Zhang, Jetting instabilities of particles from explosive dispersal, AIP Conf. Proc. 1426 (1) (2012) 1615–1618.

[149] G. Mie, Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen, Ann. Phys. 330 (3) (1908) 377–445.

[150] E. Grüneisen, Theorie des festen Zustandes einatomiger Elemente, Ann. Phys. 344 (12) (1912) 257–306.

[151] G. R. Johnson, W. H. Cook, Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures, Eng. Fract. Mech. 21 (1) (1985) 31–48.

[152] C. B. Laney, Computational Gasdynamics, Cambridge University Press, 1998.

[153] H. Mo, ArtraCFD: A Computational Fluid Dynamics Solver, URL `https://github.com/mohuangrui/ArtraCFD`, 2014.

[154] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1) (1978) 1–31.

[155] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, vol. 16, Springer, Berlin, 1999.

[156] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1) (1984) 115–173.

[157] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys. 83 (1) (1989) 32–78.

[158] P. D. Lax, X.-D. Liu, Solution of two-dimensional Riemann problems of gas dynamics by positive schemes, SIAM J. Sci. Comput. 19 (2) (1998) 319–340.