# Stony Brook University

# Numerical coupling and simulation of point-mass system with the turbulent fluid flow

A Dissertation Presented

by

**Zheng Gao**

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

**Doctor of Philosophy**

in

**Applied Mathematics and Statistics**

Stony Brook University

**May 2017**

**Stony Brook University**

The Graduate School

**Zheng Gao**

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

**Xiaolin Li - Dissertation Advisor**
**Professor, Department of Applied Mathematics and Statistics**

**James Glimm - Chairperson of Defense**
**Professor, Department of Applied Mathematics and Statistics**

**Yangang Liu - Outside Member**
**Scientist, Environmental and Climate Sciences Department,**
**Brookhaven National Laboratory**

**Richard D. Charles - Outside Member**
**Engineer, Natick Soldier RD&E Center, U.S. Army RDECOM**

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

# Numerical coupling and simulation of point-mass system with the turbulent fluid flow

by

**Zheng Gao**

**Doctor of Philosophy**

in

**Applied Mathematics and Statistics**

Stony Brook University

**2017**

A computational framework that combines the Eulerian description of the turbulence field with a Lagrangian point-mass ensemble is proposed in this dissertation. Depending on the Reynolds number, the turbulence field is simulated using Direct Numerical Simulation (DNS) or eddy viscosity model. In the meanwhile, the particle system, such as spring-mass system and cloud droplets, are modeled using the ordinary differential system, which is stiff and hence poses a challenge to the stability of the entire system. This computational framework is applied to the numerical study of parachute deceleration and cloud microphysics. These two distinct problems can be uniformly

modeled with Partial Differential Equations (PDEs) and Ordinary Differential Equations (ODEs), and numerically solved in the same framework. For the parachute simulation, a novel porosity model is proposed to simulate the porous effects of the parachute canopy. This model is easy to implement with the projection method and is able to reproduce Darcy's law observed in the experiment. Moreover, the impacts of using different versions of $k$-$\epsilon$ turbulence model in the parachute simulation has been investigated and concludes that the standard and Re-Normalisation Group (RNG) model may overestimate the turbulence effects when Reynolds number is small while the Realizable model has a consistent performance with both large and small Reynolds number. For another application, cloud microphysics, the cloud entrainment-mixing problem is studied in the same numerical framework. Three sets of DNS are carried out with both decaying and forced turbulence. The numerical result suggests a new way parameterize the cloud mixing degree using the dynamical measures. The numerical experiments also verify the negative relationship between the droplets number concentration and the vorticity field. The results imply that the gravity has fewer impacts on the forced turbulence than the decaying turbulence. In summary, the proposed framework can be used to solve a physics problem that involves turbulence field and point-mass system, and therefore has a broad application.

*Key Words:* computational fluid dynamics, fluid-structure interactions, particle system dynamics

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

I would like to express my sincere gratitude to my adviser, Prof. Xiaolin Li. He is the best advisor, who gave me inspiring suggestions on research and patient guidance towards my Ph.D. degree; He is my family, who support me for every decision I made; He is a good friend, with whom I can share any happiness and sorrow. I feel so proud to be his student and enjoy all the five-year time spent with him.

I really appreciate the encouragement and valuable technical suggestions from Prof. James Glimm, Dr. Yangang Liu and Dr. Richard D. Charles. They are the most outstanding researchers in their field. I feel so excited and grateful to have them in my defense committee.

Many thanks to my colleagues, Dr. Yan Li, Dr. Yijing Hu, Dr. Saurabh Joglekar, Dr. Qiangqiang Shi, Dr. Yiyang Yang, Xiaolei Chen, Jingfang Qu, Muye Chen, Tengbo Yang, Liang Zhang, Siwu Liu, Minshen Chen, Chenxiao Xu and Brandon Ballentine.

During my Ph.D. career, I got a lot of support from my parents, family members and close friends. I want to dedicate this thesis to all the people I love and these people knowing I am not perfect but still love me.

This work was supported by the U.S. Department of Energy's Atmospheric System Research (ASR) and Earth System Modeling (ESM) programs; and the US Army Research Office under the award W911NF1410428 and the ARO-DURIP Grant W911NF-15-1-0403.

# Chapter 1

# Background

Fluid-structure interaction (FSI) problems in general are often too complex to solve analytically and so they have to be analyzed by means of experiments or numerical simulation. Although the maturity of the computational fluid dynamics enables numerical simulation of fluid structure interaction, considering turbulence flow in these problems adds more difficulties and challenges. The numerical models for solving FSI problems usually involve solving several equations describing the behaviors of fluid and structures. Sometimes the equations of the scalar field, such as temperature or the concentration of chemical species, also need to be considered as a significant component of the system. In this chapter, three different categories of equations for incompressible fluid, scalar field and particle dynamics are introduced in the mathematics point of view.

## 1.1 Navier-Stokes equation and turbulent flow

It is common to view turbulent flows in both our daily life, such as smoke from a chimney, water in a river, and engineering fields, such as flows around airplanes, or mixing of fuel and air in engines. In these observations, the flow is unsteady, irregular and chaotic. The motion of every eddy or droplet is unpredictable [114]. In another word, the velocity of turbulent flows varies irregularly in both position and time. The velocity field is denoted in mathematics by $\mathbf{U}(\mathbf{x}, t)$, where $\mathbf{x}$ is the position and $t$ is time. Another important feature of turbulence is its ability to transport and mix fluid much more effectively than a comparable laminar flow. The intensity of turbulent flow is often characterized by a single non-dimensional parameter, Reynolds number[117]:

$$Re = UL/\nu \tag{1.1}$$

where $U$ and $L$ are characteristic velocity and length scale of the flow, and $\nu$ is the kinematic viscosity of the fluid. According to Reynolds's experiment, the flow is laminar if $Re$ is less than about 2300, and becomes turbulent if $Re$ exceeds 4000.

A classic way to study fluid dynamics is to treat the fluids as continuous media, based on the continuum hypothesis, which reconciles the discrete molecular nature of fluids with the continuum view. Since the length and time scales of the molecular motion are extremely small compared with human scales, the fluid's velocity $\mathbf{U}(\mathbf{x}, t)$ is defined as the average velocity of the molecules within a spherical region of volume $V$ centered on the point $\mathbf{x}$. Ac-

cording to the conservation law of mass and momentum, the flow of constant property Newtonian fluids is determined by the following partial differential equations (PDE):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \tag{1.2}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{U} \tag{1.3}$$

where $\nu$ is the kinematic viscosity and $p$ is the pressure field. The above equations are usually called the Navier-Stokes equation. In this paper, we only consider constant density flows, and therefore Eq. (1.2) degenerates to the divergence-free condition [13]:

$$\nabla \cdot \mathbf{U} = 0 \tag{1.4}$$

The boundary conditions are essential for solving the above equations. At a stationary solid wall with unit normal $\mathbf{n}$, the boundary conditions are the impermeability condition [13]

$$\mathbf{n} \cdot \mathbf{U} = 0 \tag{1.5}$$

and the no-slip condition [13]

$$\mathbf{U} - \mathbf{n}(\mathbf{n} \cdot \mathbf{U}) = 0 \tag{1.6}$$

which together yield

$$\mathbf{U} = 0 \tag{1.7}$$

It is sometimes reasonable to assume the hypothetical case of an infinite domain, which is defined to have the periodic boundary condition. This condition is useful for approximating a large system by using a small unit cell [13].

It is true that the fluid motion of laminar and turbulent flows can be determined by the Navier-Stokes equation. It appears that if one can solve the Eq. (1.3) and Eq. (1.4), either analytically or numerically, then the behavior of the fluid flow can be predicted. However, solving the Navier-Stokes equations remains an immensely challenging problem. Firstly, until now no one has ever been able to prove that smooth solution always exist, or that if they do exist, they have bounded energy per unit mass. This in called the Navier-Stokes existence and smoothness problem [89] and has not been solved yet. On the other hand, although many well-understood numerical methods have existed for a long time and can be used to solve the PDEs like Eq. (1.3) and Eq. (1.4), massive computing resources are always needed to resolve all scales in a turbulent fluid: from energy containing scale to inertial sub-range, and down to the scale of viscous dissipation [114]. Failing to resolve the complete range of the length scales may result in unphysical numerical results. To obtain solutions for moderately high Reynolds numbers using the brutal-force approach, Directed Numerical Simulation (DNS) [114], requires weeks of computing time on today's largest supercomputers.

An alternative approach introduced by Osborne Reynolds in the late

19th century is to ignore the details of the turbulent flow at each instant and, instead, to regard the flow as a superposition of mean and fluctuating parts. According to the Reynolds decomposition [114], the velocity $\mathbf{U}(\mathbf{x}, t)$ is decomposed into its mean $\langle \mathbf{U}(\mathbf{x}, t) \rangle$ and the fluctuation $\mathbf{u}(\mathbf{x}, t)$:

$$\mathbf{U}(\mathbf{x}, t) = \langle \mathbf{U}(\mathbf{x}, t) \rangle + \mathbf{u}(\mathbf{x}, t) \tag{1.8}$$

Taking the mean of the divergence-free condition Eq. (1.4) and momentum equation Eq. (1.3) yields the Reynolds Averaged Navier-Stokes(RANS) equations [118]:

$$\nabla \cdot \langle \mathbf{U} \rangle = 0 \tag{1.9}$$

$$\frac{\partial \langle U_j \rangle}{\partial t} + \langle \mathbf{U} \rangle \cdot \nabla \langle U_j \rangle = -\frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_j} + \nu \nabla^2 \langle U_j \rangle - \frac{\partial \langle u_i u_j \rangle}{\partial x_i} \tag{1.10}$$

It is obvious that the Reynolds equations Eq. (1.10) resembles the Navier-Stokes equation Eq. (1.3) except the term of Reynolds stresses $\frac{\partial \langle u_i u_j \rangle}{\partial x_i}$ [114], which plays a crucial role to distinguish the behaviors of $\mathbf{U}(\mathbf{x}, t)$ and $\langle \mathbf{U}(\mathbf{x}, t) \rangle$. The equation Eq. (1.10) is incomplete since the Reynolds stresses appear as unknowns and are need to be determined by a turbulence model. The turbulent-viscosity models are based on the turbulent-viscosity hypothesis [122]:

$$\langle u_i u_j \rangle = \frac{2}{3} k \delta_{ij} - \nu_T \left( \frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right) \tag{1.11}$$

where $k = 1/2 \langle u_i u_i \rangle$ is the turbulent kinetic energy, $\delta_{ij}$ equals one when $i = j$ and zero otherwise. Given the turbulent eddy viscosity $\nu_T$ [114], Eq. (1.11) provides a convenient closure to the Reynolds equations. Although the accu-

racy of this hypothesis is poor for many flows [114], this approach is widely accepted as an adequate approximation and has been applied to many studies involving turbulent flows. The turbulent viscosity $\nu_T(\mathbf{x}, t)$ can be determined in many ways through defining different turbulence models, such as algebraic models [10], one-equation models [130] and two-equation models [154]. In consideration of accuracy and efficiency, the RANS model is used only if the Reynolds number is extremely large and hence the computational cost can not be handled by the current computational resources. In this paper, DNS is used to study the homogeneous turbulence flow in a very small domain while RANS is adopted to study the problem with a much larger scale.

## 1.2 Convection-diffusion-reaction equation

A scalar field, such as temperature, water vapor or the concentration of a chemical species, is often accompanied with the turbulent velocity $\mathbf{U}(\mathbf{x}, t)$ in a real physics problem. In addition, the turbulence kinetic energy $k$ [114] and energy dissipation rate $\varepsilon$ [114], which are related to the turbulent viscosity, can also be regarded as scalar fields. In a constant-property flow, the transport equation for a scalar field $\Phi$ is:

$$\frac{\partial \Phi}{\partial t} + \mathbf{U} \cdot \nabla \Phi = D \nabla^2 \Phi + \mathbf{f}(\mathbf{x}, t) \tag{1.12}$$

where $D$ is the diffusivity and $\mathbf{f}(\mathbf{x}, t)$ is the source or sink term. The scalar field may not be passive since its value can take effects on the fluid flow. Eq. (1.12) is much simpler than the Navier-Stokes equation, but has many applications.

In the rest of this section, we list all the convection-diffusion-reaction equations encountered in this paper and the detailed explanation will be given in later chapters.

In the study of cloud microphysics, we have the transport equation of water vapor mixing ratio $q_v$ and temperature field $T$ as below:

$$\partial_t T + (\mathbf{U} \cdot \nabla) T = \frac{L_h}{c_p} C_d + \mu_T \nabla^2 T \tag{1.13}$$

$$\partial_t q_v + (\mathbf{U} \cdot \nabla) q_v = -C_d + \mu_v \nabla^2 q_v \tag{1.14}$$

where $L_h$ is the latent heat of water vapor condensation, $c_p$ is the specific heat at constant pressure, $\mu_T = \mu_v$ are the molecular diffusivity for temperature and water vapor, respectively and assumes to be equal. The condensation rate $C_d$ denotes the rate of exchange between liquid water and water vapor, and hence can be treated as a source term to the temperature field and sink term to the water vapor field. These equations describe the water and energy exchange during the phase transition process between the states of liquid and vapor. Heat energy absorbed by liquid water during evaporation loosens chemical bonds between water molecules, so the molecules break free and become gaseous water vapor, while the condensation proceeds in the opposite way. The detailed explanation of these equations will be given in the later chapters for corresponding topics.

Another set of scalar fields appears in the RANS equation, which requires a formula for the turbulent viscosity to complete the model. To compute the

turbulent viscosity $\nu_T$, the most widely used complete turbulence model is the $k-\varepsilon$ model proposed by Jones and Launder[71]. The $k-\varepsilon$ model consists of two transport equations for $k$ and $\varepsilon$:

$$\frac{\partial k}{\partial t} + \nabla \cdot (k\mathbf{U} - (\nu + \frac{\nu_T}{\delta_k})\nabla k) = P_k - \varepsilon \tag{1.15}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon\mathbf{U} - (\nu + \frac{\nu_T}{\delta_\varepsilon})\nabla \varepsilon) = \frac{\varepsilon}{k}(C_1 P_k - C_2\varepsilon) \tag{1.16}$$

where $P_k = \frac{\nu_T}{2}|\nabla\mathbf{U} + \nabla\mathbf{U}^T|^2$ is the production of turbulent kinetic energy. In summary, the equations given above are all specific examples of convection-diffusion-reaction equations, and can be solved using similar numerical scheme in a unified PDE framework. Again, the detailed explanation and improvements to these equations can be found in the later chapters.

## 1.3 Ordinary differential system for particle dynamics

Particles are objects that have mass, position, and velocity, and respond to forces, but that have no spatial extent [155]. Due to its simple structure, particles are by far the easiest objects to simulate. In spite of simplicity, particles have a wide range of applications. For example, an elastic membrane can be built by connecting particles with simple damped springs; a group of cloud droplets in their early life can be simulated with millions of particles. The motion of a Newtonian particle is governed by the familiar first order

ordinary differential equations (ODE) for position $\mathbf{x}$ and velocity $\mathbf{v}$:

$$\dot{\mathbf{v}} = \mathbf{f}/m \tag{1.17}$$

$$\dot{\mathbf{x}} = \mathbf{v} \tag{1.18}$$

where the force $\mathbf{f}$ is a function of $\mathbf{x}$ and $t$, $m$ is the mass of the particle. A system of $n$ particles is described by $n$ copies of the equation, concatenated to form a $6n$-long vector. The whole system can be treated as a point moving through $6n$-space, which is also called the phase space [155]. An index $i$ is attached to each variable standing for the motion of the $i$-th particle, for example $\mathbf{x}_i$, $\mathbf{v_i}$, $\mathbf{f}_i$ and $m_i$. In the simple case when $\mathbf{f}_i$ only relies on $\mathbf{x}_i$ and $\mathbf{v}_i$, then one particle does not have direct effects on the other particles. This is the case in this simulation of cloud droplets. If the collision and coalescence are neglected, the motion of the particles in the system are independent on each other. Sometimes, the case is not that simple since one particle may have direct or indirect connection with other particles, that is $\mathbf{f}_i$ is a function of the whole phase space $\mathbf{x}$ and $\mathbf{v}$ instead of $\mathbf{x}_i$ and $\mathbf{v}_i$. These particle systems, for example spring-mass system, are handled by maintaining their original topological structure, so that the directly connected particles can be fetched immediately. In a more complicated situation when collision is considered, one particle not only interacts with its direct neighbors, but may also collide with other particles and environment. This requires to find all the potentially collision pairs for each particle in the system. Instead of the brutal force method, whose time complexity is $O(N^2)$ and $N$ is the number of particles in

the system, using some auxiliary data structures, such as hash map [143] and hierarchy tree [150, 17], can speed up the collision detection to $O(NlogN)$. In the rest of this section, some examples of the particles system are introduced.

First, let's look at a simple example of independent particle system, which describes the motion of a small droplet in the turbulence environment:

$$\dot{\mathbf{x}} = \mathbf{v} \tag{1.19}$$

$$\dot{\mathbf{v}} = \frac{1}{\tau_p}[\mathbf{U} - \mathbf{v}] + \mathbf{g} \tag{1.20}$$

Here $\mathbf{x}$ is the droplet position coordinate; $\mathbf{v}$ is the droplet velocity; $\mathbf{U}$ is the velocity of fluid field; $\mathbf{g}$ is the gravitational acceleration; $\tau_p = 2\rho_l R^2/(9\rho_0\nu)$ is the finite particle response time, which measures the droplet inertial effects. In the extreme case when $\tau_p$ is set to be zero, Eq. (1.20) becomes $\mathbf{v} = \mathbf{U}$, which implies that the droplets will exactly follow the turbulent flows. The last term in Eq. (1.20) is called the sedimentation term that accounts for the effect of gravity on droplets motion. Eq. (1.20) is appropriate if the Reynolds number based on the relative velocity between the particle and fluid is significantly less than one [44]. The particle diameter is also assumed to be smaller than the Kolmogorov microscale $\eta$ [114], the smallest length scales of the turbulent flow field. During condensation, direct interactions between droplets are negligible because their sizes are too small comparing with the average distance between two droplets. The Lagrangian trajectory of a particle in the turbulent flow is displayed in Fig. 1.1.

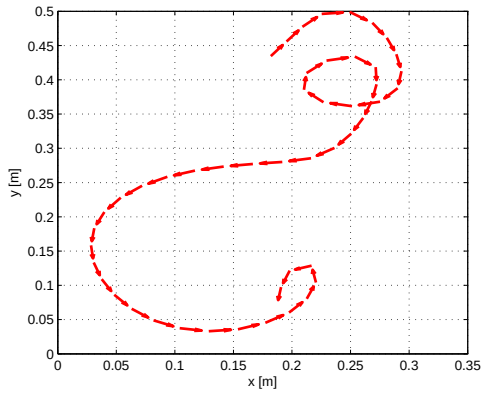


Figure 1.1: Trajectory of an independent particle in turbulence.

An example of the connected particle system is the spring mass model [14], which can effectively simulate the behavior of elastic material, such as cloth[11], muscle [108] and skin[50] etc. A simulation of parachute in turbulent flow using spring-mass model is displayed in Fig. 1.2. In mathematics, the velocity $\mathbf{v}$ and position $\mathbf{x}$ of $n$ particles are determined by:

$$\dot{\mathbf{x}} = \mathbf{v} \tag{1.21}$$

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{f}_i + \mathbf{f}_d + \mathbf{f}_e) \tag{1.22}$$

where $\mathbf{f}_i = -\frac{\partial W}{\partial \mathbf{x}_i}$ is the internal force restoring the surface deformation and $W$ is a scalar function of $\mathbf{x}$ describing the cloth internal energy; diagonal matrix $\mathbf{M} \in \mathbb{R}^{3n\times 3n}$ represents the mass distribution of the cloth; $\mathbf{f}_d$ is the damping function that will be discussed later; $\mathbf{f}_e$ is the external force (such as air-drag, gravity, contact constrains, etc.) acting on the cloth. This model is well defined if the energy function $W$ for each mass point is specified. In fact, the elastic

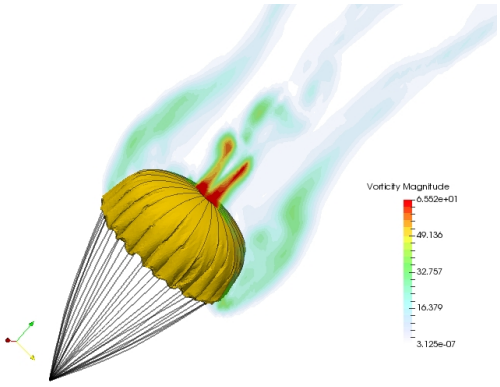


Figure 1.2: Simulation of parachute in turbulent flow using spring-mass system.

membrane can be discretized into triangles and then the potential energy in a deformed triangle consists of two parts: the energy of three tensile springs that prevent edges from stretching; the energy of three angular springs that prevent any change of vertex angles. Intuitively, the formula of $W$ for the $i$-th particle should include all the particles connecting to it. The specific form of $W$ will be introduced in later chapter.

In the next chapter, the numerical methods for solving Navier-Stokes equation, convection-diffusion-reaction equation and particle system are introduced. In fact, these equations can be solved using various numerical methods, which have their own advantages and disadvantages comparing with each other. These will be discussed in details in the next chapter.

# Chapter 2

# Numerical Method

The success of the computational fluid dynamics highly depends on the underlying numerical method. Until now, various numerical methods have been proposed to solve the partial differential equation and ordinary differential equation. However, instead of applying a general approach, some special techniques are usually required for a specific problem. In this chapter, the numerical methods for Navier-Stokes equation, advection-diffusion-reaction equation and ordinary differential equation are introduced. In the meanwhile, the criteria of their convergence and stability are also discussed.

## 2.1 Projection method for incompressible Navier-Stokes equation

A major difficulty for the numerical simulation of incompressible flows is that the velocity and the pressure are coupled by the incompressibility constraint [59]. In 1960s, Chorin and Temam proposed the fraction-step method in their work [140, 31] to overcome the difficulty in time-dependent viscous

incompressible flow. In their method, the velocity and pressure are decoupled at each time step, and consequently only a sequence of decoupled elliptic equation are needed to solve. This makes the projection method very efficient for large scale numerical simulation.

We consider an incompressible fluid in a $d = 2, 3$-dimensional bounded domain $\Omega \subset \Re^d$ during the time interval $(0, T)$. Let $\Delta t_i > 0$ be the $i$-th time step size and use notation $\mathbf{w}^n$ to represent a numerical approximation to $\mathbf{w}(t^n)$, where $t^n = \sum_{i=1}^{n} \Delta t_i$. Following the numerical schemes proposed in [18], we have the second-order, time-discrete semi-implicit forms of Eq. (1.3) and Eq. (1.4):

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \nabla p^{n+1/2} = -[(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+1/2} + \frac{\nu}{2}\nabla^2(\mathbf{u}^{n+1} + \mathbf{u}^n) \tag{2.1}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{2.2}$$

with boundary condition

$$B(\mathbf{u}^{n+1}) = 0 \tag{2.3}$$

where $[(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+1/2}$ represents the convective derivative term at time level $t^{n+1/2}$, and can be computed explicitly[76]; $B(\cdot)$ is the boundary condition of $u$. It is clear that Eq. (2.1) and Eq. (2.2) are coupled together due to the appearance of $p$, and hence difficult to solve directly. A fractional step procedure overcome this difficulty by firstly solving Eq. (2.1) while ignoring the pressure gradient term (pressure-free projection method), and then projecting the solution onto the space of divergence-free fields to obtain $\mathbf{u}^{n+1}$. The

pressure-Poisson version of projection method consists of the following steps:

Step 1: Compute the tentative velocity

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + [(\mathbf{u} \cdot \nabla_h)\mathbf{u}]^{n+1/2} = \nu \Delta_h (\mathbf{u}^* + \mathbf{u}^n) \tag{2.4}$$

$$B(\mathbf{u}^*) = 0, \tag{2.5}$$

Step 2: Projection step

$$\frac{1}{\rho}\Delta_h p^{n+1/2} = \nabla_h \cdot \mathbf{u}^* / \Delta t \tag{2.6}$$

Step 3: Update new velocity

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla_h p^{n+1/2} \tag{2.7}$$

A convenient choice to discretize $\mathbf{u}$ and $p$ in space is to use finite difference method on a regular computational grid [13]. Let $\mathbf{u}^n_{ijk}$ represent the numerical solution of velocity field at grid node $\mathbf{x}_{i,j,k} = [L_x + (i + 0.5)\Delta x, L_y + (j + 0.5)\Delta y, L_z + (k + 0.5)\Delta z]$ at time $t^n$ and an analogous definition holds for the pressure $p^n_{ijk}$ ($i = 0, 1, 2, ..., N_x - 1$, $j = 0, 1, 2, ..., N_y - 1$, $k = 0, 1, 2, ..., N_z - 1$). The vector operators are discretized using central difference scheme:

$$\nabla_h \cdot \mathbf{u} = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x} + \frac{v_{i,j+1,k} - v_{i,j-1,k}}{2\Delta y} + \frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta z} \tag{2.8}$$

$$\nabla_h p = [\frac{p_{i+1,j,k} - p_{i-1,j,k}}{2\Delta x}, \frac{p_{i,j+1,k} - p_{i,j-1,k}}{2\Delta y}, \frac{p_{i,j,k+1} - p_{i,j,k-1}}{2\Delta z}] \tag{2.9}$$

$$\begin{aligned} [lll]\Delta_h p &= \frac{p_{i+1,j,k} + p_{i-1,j,k} - 2p_{i,j,k}}{\Delta x^2} + \frac{p_{i,j+1,k} + p_{i,j-1,k} - 2p_{i,j,k}}{\Delta y^2} && (2.10) \\ &+ \frac{p_{i,j,k+1} + p_{i,j,k-1} - 2p_{i,j,k}}{\Delta z^2} && (2.11) \end{aligned}$$

This method is appealing since it is second order in both time and space. It also prohibits errors in the pressure gradient, which could accumulate in time [59]. In Eq. (2.1), the nonlinear term $adv(\mathbf{u}) = [(\mathbf{u} \cdot \nabla)\mathbf{u}]^{n+1/2}$ at half time step is approximated by extrapolation of the results at previous steps, that is $adv^{n+1/2} = (1 + \Delta t_n/(2\Delta t_{n-1}))adv^n - \Delta t_n/(2\Delta t_{n-1})adv^{n-1}$. Since this hyperbolic term is explicitly applied to the equation 2.1, a total variation diminishing (TVD) scheme [60] is always desirable. Ignoring the notation of time, the first order upwind scheme for adv$(u_i)$ can be written as:

$$\text{adv}(u_i) = u_i^+ u_x^- + u_i^- u_x^+ + v_i^+ u_y^- + v_i^- u_y^+ + w_i^+ u_z^- + w_i^- u_z^+ \tag{2.12}$$

where $u_i^+ = \max(u_i, 0)$ and $u_i^- = \min(u_i, 0)$; $u_x^+ = (u_{i+1} - u_i)/\Delta x$ and $u_x^- = (u_i - u_{i-1})/\Delta x$. A fifth order weighted essential non-oscillatory (WENO) scheme is another popular class of finite difference scheme for numerically approximate solutions of hyperbolic conservation laws due to its high order accuracy in smooth regions and essentially non-oscillatory transition for dis-

continuities [69]. Its third order version was first introduced in [95], and then extended to arbitrary order accurate in [67]. According to the WENO scheme [67], the computation of the hyperbolic term $\text{adv}(u) = uu_x + vu_y + wu_z$ can be generalized to the computation of the derivative of a flux function $f(u)$, which can be either linear (as the last two terms $vu_y$ and $wu_z$) or nonlinear (the first term written as $(u^2/2)_x$). For a general flux $f(u)$, its derivative with respect to $x$ can be written as

$$\frac{\partial f(u)}{\partial x_j} = \frac{1}{\Delta x}(f(u(x_{i+1/2})) - f(u(x_{i-1/2}))) \tag{2.13}$$

The flux $f(u(x_{i+1/2}))$ can be replaced by a monotone numerical flux $\hat{f}(u^-_{i+1/2}, u^+_{i+1/2})$, where $u^-_{i+1/2}$ and $u^+_{i+1/2}$ are reconstructed by fifth-order WENO method:

$$u^-_{i+1/2} = \sum_j \omega_j p_j(x_{j+1/2}) \tag{2.14}$$

where

$$p_0(x_{i+1/2}) = \frac{1}{3}u_{i-2} - \frac{7}{6}u_{i-1} + \frac{11}{6}u_i \tag{2.15}$$

$$p_1(x_{i+1/2}) = -\frac{1}{6}u_{i-1} + \frac{5}{6}u_i + \frac{1}{3}u_{i+1} \tag{2.16}$$

$$p_2(x_{i+1/2}) = \frac{1}{3}u_i + \frac{5}{6}u_{i+1} - \frac{1}{6}u_{i+2} \tag{2.17}$$

$$\omega_j = \frac{\bar{\omega}_j}{\sum_j \bar{\omega}_j}, \quad \bar{\omega}_j = \frac{\gamma_j}{\sum_j (\epsilon + \beta_j)^2} \tag{2.18}$$

and

$$\gamma_0 = \frac{1}{10}$$
$$\gamma_1 = \frac{6}{10}$$
$$\gamma_2 = \frac{3}{10}$$
$$\beta_0 = \frac{13}{12}(u_{i-2} - 2u_{i-1} + u_i)^2 + \frac{1}{4}(3u_{i-2} - 4u_{i-1} + u_i)^2$$
$$\beta_1 = \frac{13}{12}(u_{i-1} - 2u_i + u_{i+1})^2 + \frac{1}{4}(3u_{i-1} - u_{i+1})^2$$
$$\beta_2 = \frac{13}{12}(u_i - 2u_{i+1} + u_{i+2})^2 + \frac{1}{4}(u_i - 4u_{i+1} + u_{i+2})^2$$

The reconstruction to $u^-_{i+1/2}$ is mirror symmetric with respect to $x_i$ of the above procedure. The WENO scheme has been generalized to various types of schemes, and have been applied to various fields including computational fluid dynamics.

## 2.2 Numerical method for advection-diffusion-reaction equation

The convection-diffusion-reaction equation describes the flow of heat, concentration of chemical species where there is both convection, diffusion, and reaction. The numerical scheme used to solve the advection-diffusion-reaction equation is very similar to the numerical method for the Navier-Stokes equation introduced in the last section. Suppose the Crank-Nicolson scheme [34] is still used as the time integration method, then the numerical scheme for

solving Eq. (1.12) is:

$$\frac{\Phi^{n+1}-\Phi^{n}}{\Delta t}+[(\mathbf{U}\cdot\nabla)\Phi]^{n+1/2}=\frac{D}{2}\nabla^{2}(\Phi^{n+1}+\Phi^{n})+f^{n} \tag{2.19}$$

with boundary condition

$$B(\Phi^{n+1})=0 \tag{2.20}$$

As stated before, the advection term $[(\mathbf{U}\cdot\nabla)\Phi]^{n+1/2}$ can be approximated by $(1+\Delta t_n/(2\Delta t_{n-1}))[(\mathbf{U}\cdot\nabla)\Phi]^{n}-\Delta t_n/(2\Delta t_{n-1})[(\mathbf{U}\cdot\nabla)\Phi]^{n-1}$ and computed explicitly using upwind or WENO scheme introduced in the previous section. The Crank Nicolson scheme is second order accurate in time and space, and it is also unconditionally stable. Due to these features, Crank Nicolson scheme is the preferable method for solving the scalar field.

Different from the pure advection-diffusion equation, the reaction term adds more challenges. In many cases, the physics problem requires the underlying scalar field to be positive. For example, the concentration of a chemical species cannot be negative; the turbulence kinetic energy and dissipation rate should always be larger or equal to zero. However, due to numerical error, the existence of sink term in Eq. (1.12) can easily push the solution below zero in one time step. To avoid the loss of positivity of these variables is not an easy task. Simply set a lower bound to these variable does not always work, since this strongly depends on the limitation strategy. Another alternative is to solve for the logarithms of the original variable, but one then deals with modified set of equations involving exponentials of the unknowns. A simple numerical trick is to look at the linearized equation, and then set a lower limitation for

the coefficients. The principle of this approach is that one never touches the solution, just the coefficients of the linearized equations. Take Eq. (2.21) for turbulence kinetic energy as an example. The existence of $-\varepsilon$ on the right hand side of the equation may result in loss of positivity. Therefore, a possible modification for this equation is:

$$\frac{\partial k}{\partial t} + \nabla \cdot (k\mathbf{U} - (\nu + \frac{\nu_T}{\delta_k})\nabla k) = P_k - \gamma_k k \tag{2.21}$$

where $\gamma_k = \max(0, \varepsilon/k)$. If Eq. (2.21) is solved with implicit method, the solution must keeps its positivity. Similar technique can be easily generalized to other reaction equations.

## 2.3 Numerical method for ordinary differential system

The equations for the particle system are essentially ordinary differential equations (ODEs), and hence most numerical scheme for ODEs can be applied, such as linear multistep methods [23], Runge-Kutta methods [23] and boundary value methods [20]. However, the difficulties of solving an ODE lies on its stiffness, that is including some terms that can lead to rapid variation in the solution, and these terms may cause numerical unstable in some situations. The behavior of numerical methods on stiff problem can be analyzed by applying these methods to the test equation

$$y' = ky \tag{2.22}$$

where $y(0) = 1$ and $k \in \mathcal{C}$. Let $h$ denotes the time step and $z = hk$, by induction we have $y_n = (\Phi(z))^n y_0$. The function $\Phi$ is called the stability function [23] and the region of $\{z \in \mathcal{C} || \Phi(z)| < 1\}$ is called the absolute stability region [23]. The numerical method is called A-stable if the region of absolute stability contains the set $\{z \in \mathcal{C} | \Re(z) < 0\}$. The typical methods for solving ODE: $y' = f(t, y)$ are listed below.

1. Runge-Kutta methods are a family of numerical methods used in temporal discretization for the approximation solutions of ODEs. They use intermediate steps to enhance the accuracy and can be either implicit and explicit. The general Runge-Kutta methods take the form

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i \tag{2.23}$$

$$k_i = f(t_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} k_j) \tag{2.24}$$

   A Runge-Kutta method can be determined by putting all the coefficients in the Butcher tableau[23]:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \\ & b_1^* & b_2^* & \cdots & b_s^* \end{array} \tag{2.25}$$

The coefficients $b_i$ and $b_i^*$ are used to construct two methods, one with order $p$ and one with order $p-1$. This produces an estimate of the local truncation error of a single Rung-Kutta step and can be used to control the step size.

$$e_{n+1} = h \sum_{i=1}^{s} (b_i - b_i^*) k_i \tag{2.26}$$

The stability function of a Runge-Kutta method is:

$$\Phi(z) = \frac{det(I - zA + zeb^T)}{det(I - zA)} \tag{2.27}$$

where $e$ denotes an unity vector. The frequently used Runge-Kutta method are listed below using Butcher tableau:

Classic fourth-order explicit Runge-Kutta method:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 0 \\ 2/3 & -1/3 & 1 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 \\ \hline & 1/8 & 3/8 & 3/8 & 1/8 \end{array} \tag{2.28}$$

Note that this method does not have a step control parameters. It is also not suitable for the solution of stiff equations due to its bounded region of absolute stability. It is known that explicit Runge-Kutta methods can never be A-stable. It has been proved that Gauss-Legendre method with $s$ stage has order $2s$ and is A-stable.

2. Multistep methods [23] attempt to achieve efficiency by using the information from previous steps. A general linear multistep method takes the following form

$$y_{k+1} = \sum_{i=1}^{m} \alpha_i y_{k+1-i} + h \sum_{i=0}^{m} \beta_i f\left(t_{k+1-i}, y_{k+1-i}\right) \tag{2.29}$$

where $\{\alpha_i\}$ and $\{\beta_i\}$ are determined by polynomial interpolation.

The Backward differentiation formulas (BDF) methods are implicit methods with $\beta_1 = \cdots = \beta_m = 0$. These methods are especially used for the solution of stiff differential equations. A second order A-stable BDF method takes the form:

$$y_{n+1} - \frac{4}{3} y_n + \frac{1}{3} y_{n-1} = \frac{2}{3} h f(t_{n+1}, y_{n+1}) \tag{2.30}$$

It has been proved that explicit methods can never be A-stable. An implicit multistep method can only be A-stable if their order is at most 2, while an A-stable Runge-Kutta method can have arbitrarily high order [51]. The difficulty of exceeding second order of an A-stable multistep methods is called Dahlquist barrier [36, 35].

3. Boundary value methods are the third way between linear multistep and Runge-Kutta methods proposed in a few years ago [20]. These class of methods transform the original initial value problem into an equivalent boundary value problem using linear multistep formula (LMF), and is able to overcome the major drawback of the linear multistep methods:

Dahlquist barriers. It is known that even the original LMF is unstable, the corresponding boundary value method can be 0-stable and A-stable, provided appropriate boundary conditions are given.

In the rest of the chapters, the proposed computational platform is applied to two applications: parachute simulation and DNS of cloud entrainment mixing. Since both of the two problems can be described by PDEs and ODEs, they can be universally solved in the same computational framework even if the problems come from distinct fields.

# Chapter 3

# Parachute Simulation

Parachute simulation is a complex system coupling many aspects such as elastic mechanics and fluid dynamics. Therefore, a desirable implementation should include decoupled modules handling fluid dynamics, elastic mechanics as well as their interactions. We develop a novel particle-based cloth model to mimic the in-plane energy by introducing the concept of angular stiffness. This model is then coupled with fluid dynamics through impulse method [33]. In order to better simulate the turbulence effects, a zero-equation turbulence model [10] is replaced with a two-equation model [157]. We also proposed a new porosity model for the fabric surface to consider the porosity effects. In addition, a new collision handling function is developed to efficiently and robustly treat the collision among fabric surface, rigid body and suspension lines.

## 3.1 Model Description

The mathematics model of the parachute simulation follows the framework in Chapter 1. The aerodynamic behavior of a turbulent incompressible fluid is governed by the RANS equations for the mean velocity and pressure. We still consider the $k-\varepsilon$ family as the turbulence model, which automatically calculate the turbulence length scale [154]. In standard $k-\varepsilon$ model, the eddy viscosity [114] is defined as

$$\nu_T = C_\mu \frac{k^2}{\varepsilon}, \tag{3.1}$$

where $k$ is the turbulence kinetic energy and $\varepsilon$ is the dissipation rate. To compute $k$ and $\varepsilon$, two additional convection-diffusion-reaction equations are needed:

$$\frac{\partial k}{\partial t} + \nabla \cdot (kU - (\nu + \frac{\nu_T}{\delta_k})\nabla k) = P_k - \varepsilon \tag{3.2}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon U - (\nu + \frac{\nu_T}{\delta_\varepsilon})\nabla \varepsilon) = \frac{\varepsilon}{k}(C_1 P_k - C_2 \varepsilon) \tag{3.3}$$

where $P_k = 0.5\nu_T |\nabla U + \nabla U^T|^2$ is the production of turbulent kinetic energy. For the standard $k-\varepsilon$ model, the default values of the involved empirical constants are: $C_\mu = 0.09$, $C_1 = 1.44$, $C_2 = 1.92$, $\delta_k = 1.0$, $\delta_\varepsilon = 1.3$. Although simple and efficient, the standard model is unable to capture the effects of smaller scales of motion due to its single turbulence length scale. In order to account for the different scales of motion, a mathematical technique called Re-Normalization Group (RNG) method [157] is used to derive a turbulence model

similar to the standard one, resulting in a modified form of the $\varepsilon$ equation:

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon U - (\nu + \frac{\nu_T}{\delta_\varepsilon})\nabla \varepsilon) = \frac{\varepsilon}{k}(C_1 P_k - C_2^* \varepsilon) \tag{3.4}$$

$$C_2^* = C_2 + \frac{C_\mu \eta^3 (1 - \eta/\eta_0)}{1 + \beta \eta^3} \tag{3.5}$$

where $\eta = kS/\varepsilon$, $S$ is the modulus of the mean rate of strain tensor $S = \sqrt{2S_{ij}S_{ij}}$ and $S_{ij} = 0.5(\partial U_j/\partial x_i + \partial U_i/\partial x_j)$. The coefficients are derived explicitly in the RNG procedure. For completeness, these coefficients are listed here as: $C_\mu = 0.0845$, $C_1 = 1.42$, $C_2 = 1.68$, $\delta_k = 0.7194$, $\delta_\varepsilon = 0.7194$. Figure 3.1 provides a illustration of the viscosity and velocity streamline computed with RNG $k - \varepsilon$ model. An further improvement over the RNG $k$-$\varepsilon$ model is the Realizable turbulence model [127], which contains a new formulation for the turbulent viscosity and new transport equation for the dissipation rate:

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon U - (\nu + \frac{\nu_T}{\delta_\varepsilon})\nabla \varepsilon) = C_1 S \varepsilon - C_2 \frac{\varepsilon^2}{k + \sqrt{\nu \varepsilon}} \tag{3.6}$$

where $C_1 = \max[0.43, \eta/(\eta + 5)]$ and the coefficient $C_\mu$ in Eq. (3.1) is replaced by $C_\mu = 1/(A_0 + A_s k U^*/\varepsilon)$, $U^* = \sqrt{S_{ij}S_{ij} + \Omega_{ij}\Omega_{ij}}$, and $\Omega_{ij}$ is the mean rate-of-rotation tensor. The constants $A_0 = 4.04$ and $A_s = \sqrt{6}\cos\phi$, where $\phi = \cos^{-1}(\sqrt{6}W)/3$, $W = S_{ij}S_{jk}S_{ki}/(S_{ij}S_{ij})^{3/2}$.

Modeling of fabric surface also poses many challenges. A fabric surface may fold and wrinkle, thus making it difficult to model than an elastic plate [29]. We proposed a mesoscale model using the spring-mass system and

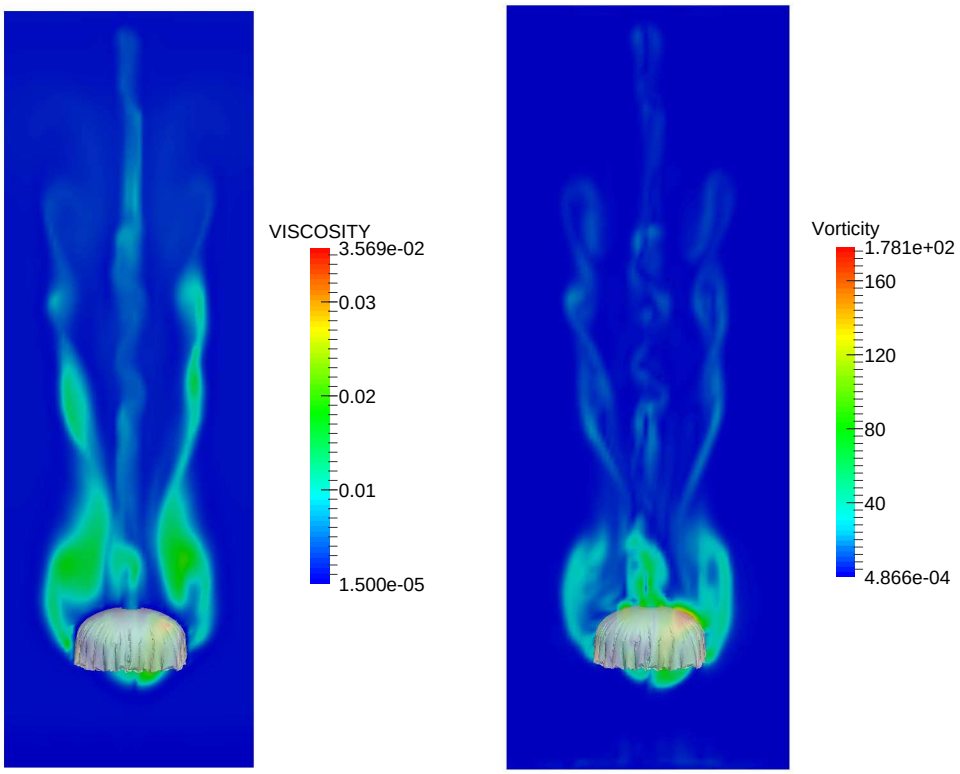


Figure 3.1: Viscosity and vorticity field around parachute computed by RNG $k-\varepsilon$ model

strengthened it by the inclusion of both tensile and angular stiffness, a model originally proposed by Delingette [38]. The mathematics model for $n$ particles system is formulated as 1.17 and 1.18. In our new spring mass model, the formula for computing the force is:

$$\mathbf{f} = -\frac{\partial W}{\partial \mathbf{x}} + \mathbf{f}_d + \mathbf{f}_e \tag{3.7}$$

This model is well defined except the energy function $W$ for each mass

point. In fact, the elastic membrane can be discretized into triangles and then the potential energy in a deformed triangle consists of two parts: the energy of three tensile springs that prevent edges from stretching; the energy of three angular springs that prevent any change of vertex angles.

In the following section, $W(T_p)$ is used to present the energy necessary to deform a triangle $T_{P_0}$ consisting of vertices $\{\mathbf{P}_{1,0}, \mathbf{P}_{2,0}, \mathbf{P}_{3,0}\}$ into its deformed position $T_P$ with vertices $\{\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3\}$; $l_{ij}$ and $l_{ij}^0$ denote the current and original length between vertex $\mathbf{P}_i$ and $\mathbf{P}_j$ respectively; $d_{ij}$ represents the edge elongation $dl_{ij} = l_{ij} - l_{ij}^0$. The potential energy for a single triangle is then given as [38]

$$W(T_P) = \sum_{i=1, j\neq i}^{3} \frac{1}{2} \kappa_{ij}^{T_{P_0}} (dl_{ij})^2 + \sum_{\substack{i=1, j\neq i \\ k\neq i, k\neq j}}^{3} \gamma_i^{T_{P_0}} dl_{ij} dl_{ik}$$

where

$$\kappa_{ij}^{T_{P_0}} = \frac{(l_{ij}^0)^2 (2\cot^2 \alpha_k (\lambda + \mu) + \mu)}{8 A_{P_0}}$$

is the tensile stiffness and

$$\gamma_i^{T_{P_0}} = \frac{l_{ik}^0 l_{ij}^0 (2\cot \alpha_j \cot \alpha_k (\lambda + \mu) - \mu)}{8 A_{P_0}} \tag{3.8}$$

is the angular stiffness; $\alpha_j$ is the vertex angle in the undeformed triangle; $A_{P_0}$ is the area of triangle $T_{P_0}$. $\gamma$ and $\mu$ are the Lamé coefficients [49] of the material. These coefficients are simply related to the physically meaningful parameters for elastic membrane, that is, Young's modulus $E$ [49] and the Poisson ratio

$\nu$ [49]:

$$\lambda = \frac{E\nu}{1-\nu^2} \quad \text{and} \quad \mu = \frac{E(1-\nu)}{1-\nu^2}.$$

Young's modulus quantifies the stiffness of the material [49], whereas the Poisson ratio characterizes the material compressibility [49]. The elastic force on each vertex of the spring system is therefore derived by computing the derivative of the energy with respect to the node position $\mathbf{P}_i$.

$$\begin{aligned}
\mathbf{F}_i(T_P) &= -\frac{\partial W}{\partial \mathbf{P}_i} && (3.9)\\
&= \sum_{j=1, j\neq i}^{3} k_{ij}^{T_{P_0}}(dl_{ij})\mathbf{e}_{ji} && (3.10)\\
&+ \sum_{\substack{j=1, i\neq j \\ k\neq i, k\neq j}}^{3} (\gamma_j^{T_{P_0}} dl_{jk} + \gamma_i^{T_{P_0}} dl_{ij})\mathbf{e}_{ji} && (3.11)
\end{aligned}$$

where $\mathbf{e}_{ji}$ is the unit vector pointing from $\mathbf{P}_i$ to $\mathbf{P}_j$. The above derivation considers only one triangle, and therefore the actual spring force on vertex $P_i$ is a linear combination of the forces calculated from all the triangles containing $P_i$.

It is interesting to notice that, if the second term in Eq. (3.11) is neglected, Delingette's model is almost the same model used by [77] except variable tensile stiffness. Numerical study suggests that both the variation of tensile stiffness [38] and the additional term due to angular stiffness [38] are significant for realistic fabric simulation. Numerical implementation and testing also show that such model is robust and conforms with both the Young modulus and

Poisson ratio of a given fabric material. In many literatures [11, 30], the internal force is decomposed to several parts, such as stretch forces, shear forces and bend forces. Our model is based on the formulation for elastic membrane, and thus only the in-plane forces are considered. The bend forces are relatively small comparing to the the other forces, so that is neglected in the current model.

The parachute simulation in this paper is uniformly built based on the above model, but a few modifications are needed in order to simulate different parts of the parachute system. The parachute canopy [81] is discretized into a elements using Delaunay triangulation [91] and then we can apply the spring mass model directly. However, the suspension lines are modeled by lists of joint mass points with their ends connecting to the canopy surface. The intensity of a real suspension line are much higher than a fabric surface, so that deserves special treatments. One attempt is to increase the tensile stiffness of the strings, but will reduce the time step size considering the numerical stability of explicit method. An alternative way is to use multiple parallel springs, which is able to increase the equivalent stiffness without affecting the numerical stability. Finally, dome-type parachute canopy are constructed from a series of fabric sectors called gores [81]. The gores help the parachute to resemble a semisphere shape after inflation. Similar as the suspension lines, the gore boundaries are also modeled by curves with higher tensile stiffness. Fig. 3.2 displays the mesh structure of such model. Each vertex point in the mesh represents a mass point with point mass $m$. Each edge of a triangle has a tensile stiffness. With the equilibrium lengths set during the initialization, the

changing length of each side exerts a tensile spring force on the two neighboring vertices in opposite directions. Each angle of a triangle has an angular stiffness which is set during the initialization. An additional tensile force is generated when the the angle is changed.

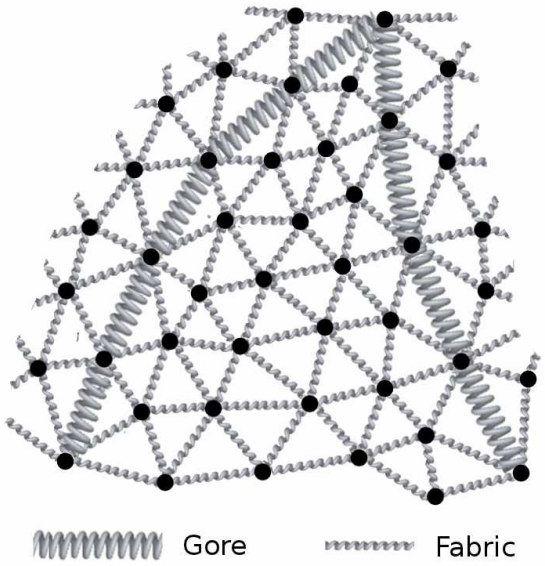


Figure 3.2: The spring model on a triangulated mesh.

In order to interact with the environment, we also add external forces, such as gravity and air-drag, to the force term. The gravity force is a constant force, and thus easy to implement:

$$\mathbf{f}_{g,i} = m_i \mathbf{g} \tag{3.12}$$

An accurate calculation of air-drag should include both the velocity shear (friction drag) and the stress (pressure drag) to the surface [1]. However,

since calculating the friction drag requires substantially more computational resources to resolve the boundary layer and its contribution to the drag force is minor in comparison with the pressure drag for the bluff body [1], we have followed [73] to neglect the friction forces in the current model, then the air-drag is formulated on a per-point basis:

$$\mathbf{f}_{p,i} = \sigma(p_i^- - p_i^+)\mathbf{n} \tag{3.13}$$

where $p_i^-$ and $p_i^+$ are the pressure of point $i$ on lower and upper sides of the canopy, $\sigma$ is the mass density of canopy per unit area, and $\mathbf{n}$ is the unit normal vector pointing from lower to upper side of the canopy. The pressure $p^-$ and $p^+$ are calculated by making use of the left state and right state of the interface point in the front tracking method [53].

Robust dynamic cloth simulation is critically dependent on well-chosen damping forces [11]. A strong damping force must be applied to the spring system with strong stretch force to prevent anomalous in-plane oscillations. However, this strong damping force should confine itself solely to damping in-plane motions without affecting other forces. In previous literatures [124], the in-plane damping force for point $i$ is chosen as:

$$\mathbf{f}_{d,i} = -k_d m_i \sum_{j \neq i} (\mathbf{v}_i - \mathbf{v}_j)^T \mathbf{e}_{ij} \mathbf{e}_{ij} \tag{3.14}$$

It is tempting to formulate a damping function as above. However, this damping function only works for the case with fabric surface solely and gives nonsensical results for the case involving fabric connecting with strings. An al-

ternative method is [142, 141, 26, 113]'s treatment of cloth that used a simple viscous damping function which dissipates kinetic energy. We improve this method by subtracting the damping function with the external impulses, so that its influence on the external forces can be excluded:

$$\mathbf{f}_{d,i} = -k_d m_i(\mathbf{v}_i - I_{ext,i}) \tag{3.15}$$

where $I_{ext,i}(t, \mathbf{x}_i) = \int_0^t (\mathbf{f}_{g,i} + \mathbf{f}_{p,i})dt$. This method works well for all our cases including the joint fabric and strings, and produces visually appealing results. One problem of this method is that a linear function of velocity does not match the quartic energy functions of the continuum formulation [11], but will put this as future studies in the new paper.

As discussed above, the fluid affects the motion of parachute canopy through the pressure drag and friction drag on the surface. In the opposite side, the parachute canopy is treated as an internal moving boundary for the fluid field [77]. In this dissertation, the fluid field is modeled with incompressible Navier-Stokes equation, whose dynamics is controlled by the boundary conditions. The external boundary, such as inflow, outflow, periodic and wall boundaries can be handled trivially by defining an appropriate boundary condition following the underlying numerical scheme. As for the internal boundaries, the parachute canopy, we use the normal velocity computed from the ODEs of the spring system as the boundary value. Therefore, at each time step, we solve a Dirichlet boundary problem for the Naiver-Stokes equation.

## 3.2 Porosity modeling

Although the canopy surface can be modeled using a pure particle system, a canopy surface still has many unique properties differing with a group of particles. Due to the porous structure of a fabric surface, a fraction of the free-stream flow can go through the canopy surface and stabilize the descent of the parachute [70]. Meanwhile, this secondary flow reduces the drag force on the canopy. An intuitive way to study the porous medium via numerical method is to model the microstructure and solve the fluid equation at the pore level. However, to resolve the details of the porous layer would require an extremely refined computational grid. Due to limited computational resources, it is more practical to describe the average aerodynamic motion of the canopy surface using a mesoscale model. In the mesoscale model, the fabric surface can be treated as an interface immersed in the fluid. The intractable complexities of the porous structures are then released by approximating it with an equivalent homogenized porosity. This simplification is reasonable since the parachute canopy thickness is extremely small compared to its radius. In the past few years, several authors have attempted to study fabric permeability under this simplification. Tazduyar et al. [144] proposed a homogenized porosity model based on the finite element method and applied it to the simulation of T-10 parachute. This method is then modified in [136] to simulate the geometric porosity (gaps and slits in ring-sail parachute) by assigning a locally-varying homogenized porosity to the canopy surface. Kim and Peskin studied the influence of porosity on parachute stability in two dimensions [79]. They generalized the immersed boundary method [112] to handle the

porous immersed boundary by coupling it with Darcy's law [109]. Wang [152] simulated three-dimensional fluid-structure interaction with porous parachute canopy using the LS-DYNA code in which the interface is approximated by a rigid and porous internal boundary coupling with the force computed using the Ergun equation [109]. Although specific details vary, there is a deep commonality amongst all the approaches: the porosity effects are seen by the fluid mechanics computation through establishing a relationship between the pressure drop and the permeability velocity through the interface.

Through this section, we present an alternative approach which combines good accuracy for holding the Ergun's equation with a very simple and efficient computational procedure. The idea of our method is outlined as the following: The fluid dynamics is described by the incompressible Navier-Stokes equations and numerically solved by the projection method [76], which is then coupled to the Ergun equation [109] by imposing a pressure drop boundary condition at the interface [24]. Since the pressure is obtained through the Poisson equation by using the projection method at each time step, it is convenient to use the GFM technique similar to the work in [48, 74, 94] to handle the discontinuity of the pressure at the fabric surface with the Poisson solver. Liu et al. [94] used level set function to interpolate the interface position, whereas with the front tracking method the interface is explicitly tracked as a set of topologically linked marker points.

The fabric porosity is simulated with the pressure drop condition at the

interface:

$$[p]_\Gamma = \alpha \mathbf{u}_\Gamma \cdot \mathbf{n} + \beta |\mathbf{u}_\Gamma \cdot \mathbf{n}| \mathbf{u}_\Gamma \cdot \mathbf{n} \tag{3.16}$$

$$[u]_\Gamma = 0 \tag{3.17}$$

where $[p]_\Gamma = p^+ - p^-$ is the pressure drop across the interface $\Gamma$; $p^+$ and $p^-$ are the pressure in $\Omega^+$ and $\Omega^-$ respectively; $\mathbf{n}$ is the local unit normal vector at the interface; $\mathbf{u}_\Gamma$ is the relative velocity between interface and fluid field at the interface location. For the two parameters in Eq. (3.16), $\alpha$ is the viscous porosity coefficient and $\beta$ presents the inertial porosity coefficient [152]. $\beta \neq 0$ is used in the case of turbulent flow with high Reynolds number. Note that the sign of the subdomains is decided by the normal vector of surface triangles which points from $\Omega^+$ to $\Omega^-$. In our application of parachute simulation, the interface is an open surface, which means that $\Omega^+$ and $\Omega^-$ are connected. Therefore $\Omega^+$ and $\Omega^-$ is only a valid concept locally for the immediate vicinity of the fabric surface.

The interaction between fluid and fabric surface structure is handled by the impulse method [78] on the *FronTier++* platform [15]. The jump condition Eq. (3.16) is considered by coupling the GFM with the finite difference scheme at the projection step. We use the pressure-Poisson version of projection method because the jump condition can be applied directly to the Poisson equation.

Consider the computational domain $[L_x, U_x] \times [L_y, U_y] \times [L_z, U_z]$ which is discretized into $N_x \times N_y \times N_z$ cells of size $\Delta x \times \Delta y \times \Delta z$. Let $\mathbf{u}_{ijk}^n$ represent the

numerical solution of velocity field at grid node $\mathbf{x}_{i,j,k} = [L_x + (i+0.5)\Delta x, L_y + (j+0.5)\Delta y, L_z + (k+0.5)\Delta z]$ at time $t^n = n\Delta t$ and an analogous definition holds for the pressure $p^n_{ijk}$ ($i = 0, 1, 2, ..., N_x - 1$, $j = 0, 1, 2, ..., N_y - 1$, $k = 0, 1, 2, ..., N_z - 1$). The standard pressure-Poisson version of projection method consists of the following steps:

Step 1: Compute the tentative velocity

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + [(\mathbf{u} \cdot \nabla_h)\mathbf{u}]^{n+1/2} = \frac{\mu}{2\rho}\Delta_h(\mathbf{u}^* + \mathbf{u}^n) \tag{3.18}$$

$$B(\mathbf{u}^*) = 0, \tag{3.19}$$

here $B(\cdot) = 0$ is the boundary condition for $\mathbf{u}$, $[(\mathbf{u} \cdot \nabla_h)\mathbf{u}]^{n+1/2}$ is computed by the WENO scheme [67] for the advection equation from $t_{n-1}$ to $t_n$ and then extrapolated to $t_{n+1/2}$ [76].

Step 2: Projection step

$$\frac{1}{\rho}\Delta_h p^{n+1/2} = \nabla_h \cdot \mathbf{u}^*/\Delta t \tag{3.20}$$

Step 3: Update new velocity

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla_h p^{n+1/2} \tag{3.21}$$

where

$$\nabla_h \cdot u = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x} + \frac{v_{i,j+1,k} - v_{i,j-1,k}}{2\Delta y} + \frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta z} \tag{3.22}$$

$$\nabla_h p = [\frac{p_{i+1,j,k} - p_{i-1,j,k}}{2\Delta x}, \frac{p_{i,j+1,k} - p_{i,j-1,k}}{2\Delta y}, \frac{p_{i,j,k+1} - p_{i,j,k-1}}{2\Delta z}] \tag{3.23}$$

$$\begin{aligned}\Delta_h p = \frac{p_{i+1,j,k} + p_{i-1,j,k} - 2p_{i,j,k}}{\Delta x^2} + \frac{p_{i,j+1,k} + p_{i,j-1,k} - 2p_{i,j,k}}{\Delta y^2} \\ + \frac{p_{i,j,k+1} + p_{i,j,k-1} - 2p_{i,j,k}}{\Delta z^2}\end{aligned} \tag{3.24}$$

The pressure jump condition is implemented while discretizing the pressure function. Since the derivative is not defined through the interface, therefore, the method by Liu in [94] for the variable coefficient Poisson equation is used to discretize the Laplacian operator in Eq. (3.20) and the gradient operator in Eq. (3.21). Liu used level set function [111] to determine the sign of the domain. In the front tracking framework, one can use the local normal vector to decide the sign of a grid point in the computational domain since the interface is explicitly represented by topologically connected marker points [54]. For example in Fig. 3.3, when discretizing at grid point $p_a$ along the $x$ direction, the sign of this point is defined by the sign of $\Phi_a = \mathbf{v}_a \cdot \mathbf{n}_\Gamma$. Therefore, the sign of a grid point is allowed to vary when discretizing along different directions.

Applying Liu's method to Eq. (3.20) and Eq. (3.21), we have

$$\Delta_h p^{n+1/2} = \nabla_h \cdot \mathbf{u}^*/\Delta t + F^x + F^y + F^z \tag{3.25}$$

where $F^x = F^W + F^E$, $F^y = F^N + F^S$ and $F^z = F^T + F^B$.

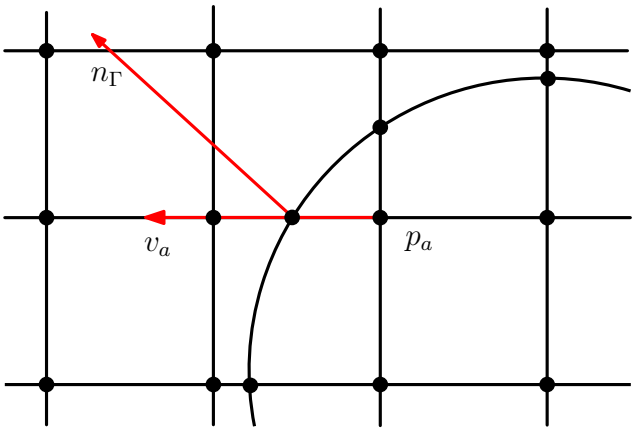


Figure 3.3: Numerical discretization near the interface. The plot shows the algorithm for calculating $\Phi_a = \mathbf{n}_\Gamma \cdot \mathbf{v}_a$, indicating the sign of the domain at grid point $p_a$ when discretizing along $x$ direction. The normal vector $\mathbf{n}_\Gamma$ at the interface-grid crossing is calculated by *FronTier++* with second order accuracy.

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t(\nabla_h p^{n+1/2} + G^x + G^y + G^z) \tag{3.26}$$

where $G^x = G^W + G^E$, $G^y = G^N + G^S$ and $G^z = G^T + G^B$.

The underlying idea to derive the additional terms in Eq. (3.25) and Eq. (3.26) is to define a ghost value for each grid node by imposing the jump condition

$$p^+_{i,j,k} - p^-_{i,j,k} = a(\mathbf{x}_\Gamma) \tag{3.27}$$

where the indices "$\pm$" stand for the domain as $\Omega^\pm$ and $a(\mathbf{x}_\Gamma)$ is the pressure drop at the interface location $\mathbf{x}_\Gamma$. In consequence, every grid node has two values for the solution, the real value and the ghost value.

According to the GFM, all the variables in the difference equation discretized at point $\mathbf{x}_{i,j,k}$ should use the same domain indices as $\mathbf{x}_{i,j,k}$. For example, if $\mathbf{x}_{i,j,k}$ is in domain of $\Omega^+$, then Eq. (3.23) becomes

$$\nabla_h p = [\frac{p^+_{i+1,j,k} - p^+_{i-1,j,k}}{2\Delta x}, \frac{p^+_{i,j+1,k} - p^+_{i,j-1,k}}{2\Delta y}, \frac{p^+_{i,j,k+1} - p^+_{i,j,k-1}}{2\Delta z}] \tag{3.28}$$

and Eq. (3.24) becomes

$$\begin{aligned}\Delta_h p = \frac{p^+_{i+1,j,k} + p^+_{i-1,j,k} - 2p^+_{i,j,k}}{\Delta x^2} + \frac{p^+_{i,j+1,k} + p^+_{i,j-1,k} - 2p^+_{i,j,k}}{\Delta y^2} \\ + \frac{p^+_{i,j,k+1} + p^+_{i,j,k-1} - 2p^+_{i,j,k}}{\Delta z^2}\end{aligned} \tag{3.29}$$

In the case of $\Phi_{i,j,k} \cdot \Phi_{i+1,j,k} < 0$, that is $\mathbf{x}_{i+1,j,k}$ is in the different domain from $\mathbf{x}_{i,j,k}$, the ghost value $p^+_{i+1,j,k}$ should be replaced by its real value using

the jump condition Eq. (3.27). It is easy to obtain $F^E = -a(x_\Gamma)/(\Delta x^2)$ and $G^E = -a(\mathbf{x}_\Gamma)/(2\Delta x)$.

In the case of $\Phi_{i,j,k} \cdot \Phi_{i+1,j,k} > 0$, that is no interface exists between point $\mathbf{x}_{i,j,k}$ and $\mathbf{x}_{i+1,j,k}$, then $F^E = G^E = 0$. The derivation of other terms is ignored here, since they are a straightforward extension of the above procedure.

In our situation, $a(\mathbf{x}_\Gamma) = [p]^n_\Gamma = \alpha \mathbf{u}^n_\Gamma \cdot \mathbf{n} + \beta |\mathbf{u}^n_\Gamma \cdot \mathbf{n}| \mathbf{u}^n_\Gamma \cdot \mathbf{n}$, where $\mathbf{u}_\Gamma = \mathbf{u}_{interface} - \mathbf{u}_{fluid}$, $\mathbf{u}_{interface}$ is the velocity of the interface point and $\mathbf{u}_{fluid}$ is obtained by using the bilinear interpolation of the velocity field at the interface location.

## 3.3 Collision handling

Resolution of collision between different parts of fabric surface or between fabric surface and environment is a very delicate problem in mathematical algorithm and computational geometry. Since all points are on the surface and all points may collide with each other, collision has been a major bottleneck in cloth simulation [17]. In order to resolve fabric collision during parachute inflation process or its reverse procedure (parachute folding), we have followed the algorithm proposed by [17] and implemented a collision handling function to detect and unwrap the fabric surface in each time step. Comparing with other previous methods, such as [115, 151], which allow self-interference and recover from it later, this approach is based on repulsion forces and non-stiff penalty forces. It can guarantee no dynamic self-interference of cloth, and thus do not need complicated and unreliable algorithms for detecting and fixing self-intersection. Furthermore, this method is able to produce physically plausible

details, such as bulk and small scale crumpling, by considering the friction forces.

Our approach for collision detection/response is similar to [17], but with some key differences in the implementation. Since in the parachute simulation, various types of materials are involved in the collision, such as parachute canopy, suspension lines and rigid cargoes, a unified collision framework is required to handle their interactions.

The basic procedure of the algorithm is as follows:

- Select a collision time step size $\Delta t$ and set $t^{n+1} = t^n + \Delta t$
- Record positions $\mathbf{x}^n$ of mass points at $t^n$
- Compute spring mesh interior dynamics to get candidate positions $\bar{\mathbf{x}}^{n+1}$
- Calculate average velocity with $\bar{\mathbf{v}}^{n+1/2} = (\bar{\mathbf{x}}^{n+1} - \mathbf{x}^n)/\Delta t$
- Call collision solver to modify $\bar{\mathbf{v}}^{n+1/2}$ to avoid any interferences and obtain the final mid-step velocity $\mathbf{v}^{n+1/2}$
- Compute the final position $\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2}\Delta t$
- Advance the mid-step velocity $\mathbf{v}^{n+1/2}$ to $\mathbf{v}^{n+1}$

We will focus on the step of collision resolution since other steps are trivial. As in [17], the collision detection and response consist of a few subroutines: proximity detection, collision detection and fail-safe method. For efficient proximity detection, an axis aligned bounding boxes hierarchy (AABB

tree) is built bottom-up at the beginning of the simulation and will be re-calculated for each iteration of the collision algorithm. The bounding boxes are created by taking a box around each geometric element enlarged by the thickness of the cloth. In the parachute simulation, two types of elements need to be considered: triangle for parachute canopy and bond for suspension lines. After creating the AABB tree, we can efficiently traverse the potential collision by only considering the geometric elements with intersected bounding boxes. It is obvious that the potential collisions drop into three categories: triangle to triangle, triangle to bond , and bond to bond. These cases can be further simplified into the cases of "edge to edge" and "point to triangle". Note that we don't check a point against a triangle that contains it or two edges shares an endpoint. If the distance between a point and a triangle or two edges is smaller than the fabric thickness, a repulsion force is applied to the two elements to stop the proximity and a non-stiff penalty force is also used to prevent the collisions in the future.

The second subroutine, collision detection, is similar to the proximity detection, except that the bounding boxes are enlarged to contain both the old state and the current state. This helps to detect the intersection between the moving trajectories of two elements. When a real collision occurs, the corresponding two elements are rewound back to the exact moment of collision and applied with the repulsion forces. Fig. 3.4 shows the difference between treatment of proximity and collision for different cases.

Finally, the fail-safe method, impact zone method, is necessary since the collision handling procedure follows the iteration methodology and it is not

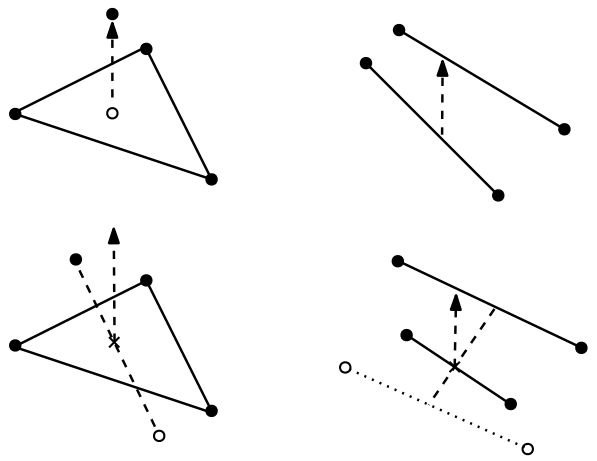

Figure 3.4: Comparison between proximity (upper level) and collision handling (lower level). The solid disks show the position before proximity/collision happens; the circles illustrate the candidate position by applying the average velocity; the cross gives the exact position where collision happens. The solid lines illustrate the basic geometries, such as mass point, bond and triangle. The dashed lines show the path the candidate motion and the dashed lines with arrows give the direction of the impulse.

guaranteed to terminate in finite number of steps. This is mainly because resolving old collision may bring new collision to the surface. As stated in [115], the principle of the impact zone method is to combine all the points in collision to form a large rigid surface. The points in the rigid surface have no relative motion, and hence all collisions can be quickly eliminated. However, this impact zone method has a tendency to freeze the cloth into nonphysical clumps. Therefore, we only switch to the impact zone method after a few iterations to quickly eliminate all collisions.

It is known that the parachute inflations process includes the interactions of rigid bodies (cargo, parachutist), elastic surfaces (parachute canopy) and elastic curves (suspension line). The combination gives six unique situations and make the algorithm not easy to implement. In the programming design point of view, our solution to this problem is to use an abstract C++ class to represent the collision elements: both the "bond" and "triangle". This abstract class provides interface to give their minimum and maximum coordinates in each direction, as well as iterators for all the points in the element. By doing this, the collision algorithm works without knowing the element is a triangle or a line segment. As for the rigid body, we merge all the rigid points to form a large impact zone before the collision detection and handling. Since the impact zone method essentially duplicates the motion of a rigid body, the remaining collision procedure does not need to be changed at all.

There are many ways to construct the impact zone. Our implementation uses the famous "union and find" algorithm. Before the collision algorithm starts, each point is in its own list. Then, when a point-triangle or edge-edge

collision occurs, the lists containing the four points are merged together into one larger impact zone. The impact zones are grown until the cloth is collision free. The velocity of the points in the impact zone is derived from a rigid body motion that preserves linear and angular momentum.

In summary, this method can guarantee no dynamic self-interference of cloth after a successful call of the function. We coupled our collision handling function with the *FronTier++* library, and utilized OpenMP to parallelize the traverse of the collision elements. Our algorithm can universally handle various collision situations between rigid bodies, fabrics and strings.

## 3.4 Folding algorithm

In order to save valuable space, portable fabric equipments, such as parachute or airbag, can be folded to a package before complete inflation. These equipments may have various geometric shapes and folding patterns to serve different functions. One folding pattern may contain several folding operations, such as flat fold, long fold and accordion fold (see Fig. 3.5 for illustration). The preparation of a folded parachute state is also essential for the simulation of the parachute inflation process. However, it is a challenge to design a folding algorithm that can universally deal with all kinds of folding patterns. Based on our best knowledge, three different methods have been proposed to deal with the folding problem in the past years. A representative method is the initial metric method (IMM) [137, 164, 163]. Although with high modeling efficiency, it is very complex, fallible, and can fold only simple patterns. The second method is the simulation-based approach [61, 128, 138],

which requires preparation and calculation time but theoretically can be applied to almost any kind of folding pattern. The principle of this algorithm is to simulate a real operation on the fabric, such as roll, move, tuck and fold, and therefore it can produce physically correct folding procedure. The folding process has to be split into several simple folding operation, which is intuitive but difficult to design. In addition, the intermediate state of the simulation-based folding process is difficult to control since only the boundary points has preset motion while the interior mass-points are completely determined by the ordinary differential equation. The third approach is the origami-based folding algorithm [12, 135, 156, 39], since there is some similarities between folding a paper and folding a fabric: the paper should not stretch and the fabric should also not induce excessive stress and strain that could artificially alter the canopy geometry prior to the inflation; the paper/fabric should not be ripped or have holes; the paper/ fabric does not intersect itself. Therefore, the algorithm for folding a paper can be utilized to fold the parachute canopy. The only difference is that the faces of the folded paper are flat during the folding procedure, while the faces of the folded canopy can be curved. The main advantage of the origami-based method is that after inputting the creases and final folding angles, the intermediate folding process can proceed automatically. This method can deal with all kinds of folding patterns in an universal way, and therefore adding new design pattern does not require any modification on the computer program. However, the original origami-based folding algorithm does not suitable for the parachute folding because the parachute canopy can be bended and stretched due to the physical properties of the

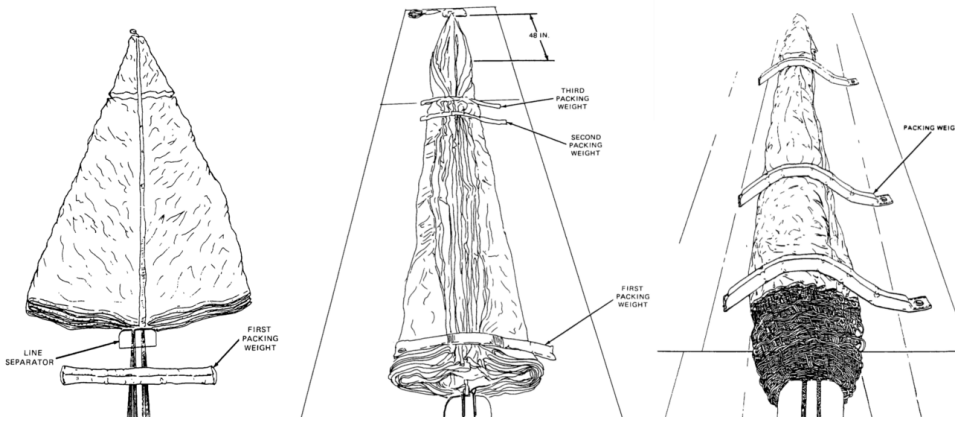


Figure 3.5: Snapshots of folding processes in [40], from left to right are flat fold, half-long fold and long fold.

cloth. Due to these reasons, we improve the original origami-based folding algorithm [156] by combing it with the simulation-based folding approach. This combination makes the folding algorithm suitable for the simulation of folding a fabric.

The mathematical model of the origami-based folding algorithm is based on the theory proposed in [12]. Firstly, each crease (folding lines) is assigned with a folding angle. The folding angle is negative when the final shape is "mountain" and the folding angle is positive when the final shape is "valley". The idea of mountain/valley and folding angles are illustrated in Fig. 3.6. Secondly, the folding angles should satisfy a mathematical constraint during the folding procedure, that is the multiplication of the rotation matrices around the crease should equal to an identity matrix. This constraint is a necessary condition that should hold if an intermediate state is foldable. Based on this necessary condition and the algorithm in [156], the $n$ folding angles

can be treated as a point moving through $n$-dimensional constraint space. The optimization algorithm connects the original point and the final point by randomly searching an optimized path, such that the multiplication of the rotation matrices at each step approximates an identity matrix. The details and implementation of this algorithm can be found in [156].

In order to apply this algorithm to the parachute folding, we have made several important modifications. Firstly, the final point-mass system will be sent to the cloth simulator, in which the bending force, stretch and shear are applied to the canopy. This helps the parachute canopy smooth out the sharp angle at the folding lines and hence can reproduce some cloth properties, such as folds and wrinkles. We need to keep monitoring the total potential energy during this process, and make sure that the final potential energy will be eventually below a threshold. Moreover, for some complicated folding patterns, we split the folding process into several sub-steps by combining origami folding and simulation-based folding. This will simplify the design of the folding pattern. For example, the flat fold of the parachute canopy can be achieved by origami folding using a simple folding pattern in Fig. 3.6 and the accordion fold and long fold can be easily achieved by appending a simulation-based folding operations.

## 3.5 Parallelism

Most of the experiments are carried on the "Intruder" Linux Cluster. The "Intruder" HPC (High Performance Computing) cluster named after a type of parachute was assembled by Advanced Cluster Technologies, inc. It consists

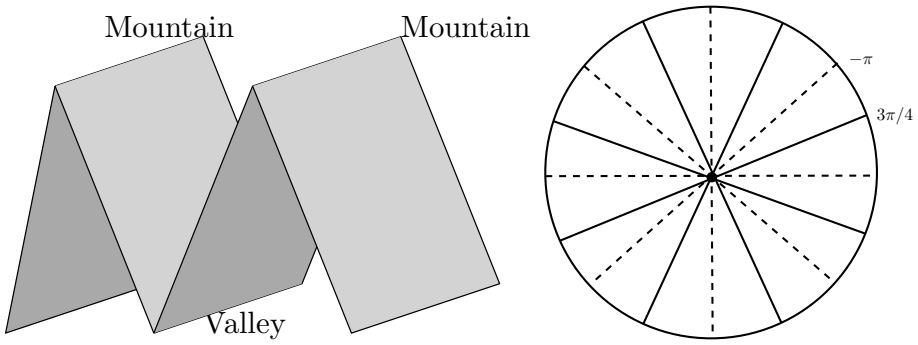


Figure 3.6: Illustration of creases and folding angles. The left panel illustrates the idea of mountain/valley creases and the right panel provides the creases and folding angles of a parachute canopy.

of one head node, 21 computing nodes (20 CPU (Central Processing Unit) nodes and 1 GPU (Graphic Processing Unit) node), connected with $56Gb/s$ InfiniBand and $1000MB/s$ Ethernet. Each node was populated with dual Eight-Core Intel E5-2630v3 "Haswell" $2.4GHz$ processors with different size of RAM (Random-access Memory) and Storage. The head node and compute node have $32GB$ of RAM for each, and the GPU node has $128GB$ of RAM. A $32TB$ of network file system using RAID6 (Redundant Array of Independent Disks) is installed in the head node, and is shared with other nodes. Each node also has a clone of the operation system on its local disk: $2TB$ SSD on head node, $1TB$ SATA drive on compute nodes, and $240GB$ SSD on GPU node. The parallel computing can be further accelerated by including the GPU node, which contains 7 NVIDIA Tesla K40 GPUs with $12GB$ of RAM for each device. A hybrid computation utilizing both CPU and GPU has been performed on this platform. A detailed description of the hardware is summarized in Table 3.1.

| Node Type | CPU | RAM | Storage for data | Storage for OS | GPU |
|---|---|---|---|---|---|
| Head node | 2x8-core Intel E5-2630v3 | 32GB | 32TB (RAID 6) | 2TB (SSD) | none |
| CPU node | 2x8-core Intel E5-2630v3 | 32GB | none | 1TB (SATA) | none |
| GPU node | 2x8-core Intel E5-2630v3 | 128GB | none | 240GB (SSD) | 7 Tesla K40 |

Table 3.1: Summary of the hardware

The software environment consists of the CentOS 6.7 operating system; the Intel OpenMPI compiler 1.8 for MPI-based applications; Sun Grid Engine 2011.11pl as the job scheduler, PETSc3.6 with benchmark tests; *FronTier++* with computational fluid dynamics applications; visualization and post-processing tools such as Paraview and VisIt. Benchmark tests have been performed within this environment, and the performance are compared with the workstation and Cray supercomputer.

The *FronTier++* library offers functions for parallelized operations of initialization and front propagation. On parallel platform, the computational domain is divided into a partition of dimensions. The typical 16 partitions of parachute simulation are displayed in 3.7. For the vector/scalar field, a buffer zone is attached at the boundary of each sub-domain to preserve the information of the ghost cells and this information needs to be updated after one time step. The parallelization of particle system is more challenging. For the connected particles in fabric modeling, a buffer interface is attached at the boundary of each subdomain. After propagation of the interface in each time step, the buffer surface is updated through exchange of interface geometry with the neighboring subdomains. The ODE solver is parallelized through the OpenMP or the GPU platform within one node.

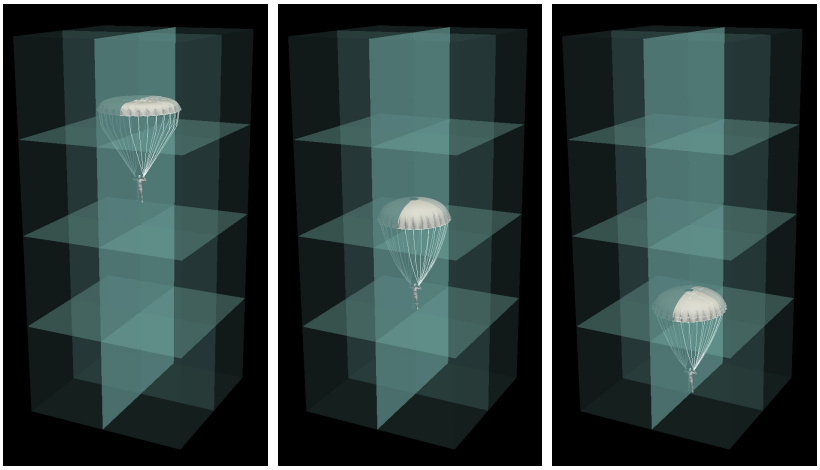

Figure 3.7: Parallel simulation of parachute descending on Intruder Linux Cluster with 16 subdomains. The neighboring subdomains are communicated via MPI while the spring model is solved on OpenMP or GPU system. The numerical algorithm is capable of running on parallel computer with many processors but requires tuning for efficiency and load balance.

Graphics Processing Unit (GPU) computing [80] is to use the GPUs together with CPUs to accelerate a general-purpose scientific and engineering application. GPU computing can offer dramatically enhanced application performance by offloading computation-intensive portions of the programming code to the GPU units, while the remainder of the code still runs on the CPU. Joint CPU/GPU applications constitute a powerful combination because CPUs consist of a few cores optimized for serial processing, while GPUs consist of thousands of smaller, more efficient cores designed for massive parallel calculations. Serial portions of the code with logical comparison run on the CPU while floating point operation intensive parallel portions of the code run on the GPU.

Global indexing is a new feature added to the *FronTier++* computational library for fluid structure interaction. Since the original *FronTier++* [42] deals with frequent surface mesh optimization and topological reconstruction, its parallelization is based on the floating point matching. The floating point matching is not completely reliable therefore more complicated algorithms were implemented as the reinforcement. For fabric-like surface, especially when a spring-mass model is used, the inter-connectivity and proximity of the interface marker points are static. Therefore, global indexing is ideal for the parallel communication of the surface information and is now employed in the work. This new feature enables the parallel communication of interface topology and geometry on a much more reliable and robust basis. It has greatly reduced the run-time interruption due to bugs and efficiency in geometrical and topological matching.

# Chapter 4

# Cloud entrainment and mixing

Reliable knowledge of cloud droplet size distributions and related microphysical properties (e.g., droplet concentration, liquid water content, and relative dispersion) is crucial for many cloud-related areas such as precipitation, weather and climate modeling, and remote sensing. A long-standing problem in cloud physics is that observed droplet size distributions are generally much broader than those predicted by the classical uniform model (e.g., [64, 66, 160, 161]. Understanding the issue of so-called spectral broadening has been a fundamental focus of cloud physics over the last decades, and a number of ideas have been proposed, including stochastic condensation theory that considers the growth of droplet populations as a stochastic process and relates the spectral broadening to turbulence-related fluctuations [65, 123, 107, 75]; systems theory that applies statistical physics ideas to cloud physics [100, 98, 99, 96, 158]; turbulence-induced preferential concentration of droplets [125]; and turbulent entrainment-mixing processes [153, 7, 63, 139, 131, 103]. Another outstanding problem is related to the for-

mation of warm rain [106, 97]. It is observed that precipitation in warm clouds can be initiated within 30 minutes after cloud formation [159]. However, according to adiabatic condensational growth theory, too much time is required for cloud droplets to grow large enough to initiate the collision-coalescence process and moreover cloud droplet size distribution becomes narrowed as cloud droplets grow, hampering realistically fast growth of cloud droplets to raindrops.

Despite the progress (see [41, 57] for recent reviews), details of the processes involved remain poorly understood and elusive. Furthermore, it is commonly accepted that the accuracy and reliability of climate models in projecting the climate change caused by climate forcing depend heavily on cloud feedbacks and thus on representations (parameterizations) of still poorly understood cloud processes, which have the potential to dampen or enhance changes in essential climatic variables such as temperature and precipitation. The situation worsens when interactions with natural and anthropogenic aerosols are included. Indeed, the latest Inter-governmental Panel on Climate Change continues to assigns "very low confidence" to aerosol-cloud-precipitation interactions, with even the sign of the resulting climate forcing remaining uncertain. Understanding such complex processes and upscaling them to adequate representation in climate models present additional challenges to the scientific community, which become more acute for extreme precipitation and weather events and as climate models progress to ever-increasing resolutions. In particular, many key processes, including microphysics, turbulent entrainment-mixing between clouds and environmental air, turbulence and their mutual

interactions, occur at scales smaller than typical grid sizes of even large eddy simulation (LES) models (e.g., 100 m) or cloud-resolving models (CRM) (e.g., 1 km) are either not represented at all or represented in very rudimentary ways, seriously hindering progress of climate and weather modeling and prediction.

Despite their differences, virtually all the ideas tie the outstanding problems to turbulence-related processes that occur on sub-LES scales (e.g. $< 100$ m) such as turbulence-microphysics interactions and turbulent entrainment-mixing processes. There are significant knowledge gaps on such sub-LES scale processes because of the limitations in conventional modeling and observations. Fully addressing these vital knowledge gaps at the fundamental level calls for a particle-resolved direct numerical simulation (DNS) model that not only resolves the smallest turbulent eddies in clouds, but also tracks motion and growth of individual particles as a supplement to measurements. Since the pioneering works in early 2000s [146, 147], a few studies have contributed to developing and applying DNS to cloud microphysics. In a series of publications, [2, 4, 5], Andrejczuk and his coauthors developed the EULAG model as a DNS to study the cloud-clear air interfacial mixing and effects of mixing processes on cloud microphysics in decaying moist turbulence. They examined the effects of initial turbulence kinetic energy (TKE), cloud fraction, droplet sizes, and the relationship between the mixing mechanisms and the Damköhler number. The initial cloud filaments and velocity field were preset to focus on the details of the decaying turbulence. Both bulk microphysics and detailed bin microphysics were used in the model. [105] compared laboratory measurements with the results of [2, 4]. [37] added more features such as sedimentation

and particle inertial in the bulk formulation. Other researchers explored DNS by treating the droplet field as discrete particles and explicitly tracking these particles. In [90] and [27], a model combining Eulerian description of the turbulent velocity and supersaturation fields with a Lagrangian population of cloud droplets was used to study the condensation and evaporation of cloud droplets in turbulent flows. A more complicated model was considered in [146, 147] by including temperature and vapor mixing ratio field. The authors investigated the influence of nonuniformity in the spatial distribution of sizes and positions of cloud droplets on the droplet size distribution. The relationships among preferential concentration, sedimentation and Stokes number were also discussed. The DNS solves the forced incompressible NavierStokes equations in 3D by use of the method of [133]. The effect of entrainment-mixing processes on cloud microphysics was not addressed in these studies. Similar to [147, 90, 84, 83] developed a particle resolved DNS to study turbulent entrainment mixing processes. In their work, a slab-like vapor field was adopted to mimic the supersaturated cloudy area and subsaturated environment. The effects of temperature and buoyancy were ignored while an artificial isotropic volume forcing is introduced to maintain the turbulence. In [85], the authors extended their previous work to both forced and decaying turbulence, and claimed that the buoyancy due to droplet evaporation played minor role in the mixing process.

Despite the progress, many questions regarding turbulence-microphysics interactions and turbulent entrainment-mixing processes remain elusive, still posing challenges for fundamentally understanding and representing clouds in

coarse-resolution climate model. This work expands on these previous studies with three primary objectives. First, as for the entrainment and mixing study, it is interesting to note that the settings in [83, 2] are similar, except the initial configuration of the cloudy area. The cloudy area consists of worm-like area determined by the velocity field of the turbulent flow in [2], while a slab-like cloud filament is adopted in [83]. It is expected that differences in the configuration of cloudy area may cause differences in the results. However, no study has been concentrated on such configuration impact. Thus, one of the objectives of this paper is to explore the effects of initial configuration of cloudy area. Second, it has been recently demonstrated that different mixing scenarios can occur and change during one single cloud evolution [5, 22, 92] and therefore it becomes more and more important to have a reasonable and accurate estimation of the mixing scenario for a sub-grid model. [103] proposed the transition scale number to measure the occurrence probability of homogeneous or inhomogeneous entrainment-mixing process. Both of the cloud observations and numerical simulations imply a positive relationship between the transition scale number and the homogeneous mixing degree. Thus, the second objective is to systematically investigate the potential of unifying the parameterization of mixing types for larger scale models. Last but not least, to the best of our knowledge, all DNS models have been based on pseudo-spectral methods due to its superior accuracy [120, 110, 27, 83]. However, the standard pseudo-spectral method has some limitations. As claimed in [83], the spectral method requires smooth initial conditions to avoid the Gibbs phenomenon, and thus unable to address sharp or zeroth-order discontinuous interfaces that

likely exist in real clouds [e.g., [16]. Moreover, the spectral method requires a periodic boundary condition in each direction and thus cannot be applied to flows that require a non-periodic, physical boundary condition. To be flexible enough to deal with various initial profiles with sharp cloud-air interfaces as well as applying different boundary conditions in the future, we develop a new particle-resolved DNS using finite difference method coupled with WENO (Weighted Essentially Non-Oscillatory) scheme [68], which has the capability of dealing with discontinuity without causing numerical overshoots at sharp interfaces. Thus, the third objective is to develop a new particle-resolved model based on the finite different method for more general application.

The rest of the paper is organized as follows. Section 2 introduces the system of equations and the numerical schemes used to solve these equations. Section 3 describes the design of the numerical experiments, including configurations of different cases studied and initial and boundary conditions for numerical simulations. The results and discussion are provided in Section 4. Concluding remarks are presented in Section 5.

## 4.1 Description of New Particle-Resolved DNS

Similar to most previous DNS, our new DNS is based on the incompressible Boussinesq fluid system [2]. Briefly, the dynamical field is give by

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho_0}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}_b + \mathbf{f}_e \tag{4.1a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4.1b}$$

where $\mathbf{u}$ is the velocity field, $p$ is the pressure field, $\nu = 1.5 \times 10^{-5} m^2 s^{-1}$ is the kinetic viscosity, $\rho_0$ is the density of dry air. Here $\mathbf{f}_b$ is the buoyancy force given by

$$\mathbf{f}_b = -\mathbf{g}[\frac{T - T_0}{T_0} + 0.608(q_v - q_{v0}) - q_c] \tag{4.2}$$

where $\mathbf{g}$ is the gravity, $T$ and $q_v$ are temperature and vapor mixing ratio field respectively with the subscript "0" denoting the reference value. The force $\mathbf{f}_e$ is introduced as an external, "large" scale forcing to maintain the turbulence, and is determined by the low-wavenumber forcing in the Fourier space:

$$\mathbf{f}_e(\mathbf{k}, t) = \epsilon_{in} \frac{\mathbf{u}(\mathbf{k}, t)}{\sum_{\mathbf{k}_f \in \kappa} |\mathbf{u}(\mathbf{k}_f, t)|^2} \delta_{\mathbf{k}, \mathbf{k}_f} \tag{4.3}$$

where $\mathbf{u}(\mathbf{k}, t)$ is the velocity function in Fourier space, $\mathbf{k}_f$ is chosen from a subset of the wavenumber space $\kappa$ containing a few wavevectors, for example $(2\pi/L_x, 2\pi/L_y, 4\pi/L_z)$ plus all permutations with respect to components and sign, $\epsilon_{in}$ is the input energy rate [52]. $\delta_{k,k_f}$ is a delta function. Therefore, statistically stationary homogeneous turbulence can be obtained in DNS by forcing the low-wavenumber modes. For decaying turbulence simulations, $\mathbf{f}_e$ is set to zero.

The temperature $T$ and vapor mixing ratio $q_v$ are described by the following equations ([84]):

$$\partial_t T + (\mathbf{u} \cdot \nabla)T = \frac{L_h}{c_p} C_d + \mu_T \nabla^2 T \tag{4.4}$$

$$\partial_t q_v + (\mathbf{u} \cdot \nabla)q_v = -C_d + \mu_v \nabla^2 q_v \tag{4.5}$$

where $L_h$ is the latent heat of water vapor condensation, $c_p$ is the specific heat at constant pressure, $\mu_T = \mu_v$ are the molecular diffusivity for temperature and water vapor, respectively and assumes to be equal to $2.16 \times 10^{-5} m^2 s^{-1}$. The condensation rate $C_d$ denotes the rate of exchange between liquid and vapor, and is described by:

$$C_d(\mathbf{X}, t) = \frac{d(m_l(\mathbf{X}, t))}{m_a dt} = \frac{4\pi \rho_l A}{\rho_0 a^3} \Sigma_{i=1}^{n} S(\mathbf{X}_i, t) R_i(t) \tag{4.6a}$$

where $A$ is a function of temperature and pressure given by:

$$A = 1/[(\frac{L_h}{G_v T} - 1)\frac{L_h \rho_l}{\mu_T T} + \frac{\rho_l G_v T}{\mu_v e_s(T)}] \tag{4.6b}$$

where $G_v$ is the individual gas constant, $e_s(T)$ is the saturation vapor pressure. The supersaturation $S(X, t)$ is calculated directly directly from the water vapor mixing ratio and temperature based on the definition

$$S(\mathbf{X}, t) = \frac{q_v(\mathbf{X}, t)}{q_{v,s}(\mathbf{X}, t)} - 1 \tag{4.6c}$$

where $q_{v,s}$ is the corresponding saturation water vapor mixing ratio. The droplets grow or shrink depending on the sign of supersaturation $S$. A positive and negative $C_d$ mean condensation and evaporation, respectively.

The liquid water mixing ratio is given by

$$q_c(\mathbf{X}, t) = \frac{4\pi\rho_l}{3\rho_0 a^3} \sum_{i=1}^{n} R_i^3(t) \tag{4.7}$$

where $a$ is the size of a grid cell, $n$ is the number of droplets in the grid cell; $\rho_l$ and $\rho_0$ are the densities of water and air. $R_i(t)$ is the radius of the $i$-th droplet.

To describe the motion and condensation(or evaporation) of cloud droplets, we use

$$R_i(t)\frac{R_i(t)}{dt} = K \cdot S(\mathbf{X}_i, t) \tag{4.8}$$

$$\frac{d\mathbf{X}_i(t)}{dt} = \mathbf{V}_i(t) \tag{4.9}$$

$$\frac{d\mathbf{V}_i(t)}{dt} = \frac{1}{\tau_p}[\mathbf{u}(\mathbf{X}_i, t) - \mathbf{V}_i(t)] + \mathbf{g} \tag{4.10}$$

where $R_i(t)$ is the radius, $\mathbf{X}_i(t)$ is the position coordinate and $\mathbf{V}_i(t)$ is the droplet velocity of the $i$-th particle; $\mathbf{g}$ is the gravitational constant. The particle response time $\tau_p$ measures the droplet inertial effect and is given by

$$\tau_p = \frac{2\rho_l R_i^2}{9\rho_0 \nu} \tag{4.11}$$

Eq. (4.11) is appropriate for the Stokes particles whereby the Reynolds number based on the relative velocity between the particle and fluid is significantly

less than one and the drag follows the Stokes law [43]. For Stokes particles, The particle diameter is also smaller than the Kolmogorov microscale $\eta$, the smallest length scales of the turbulent flow field. The last term in Eq. (4.10) is the sedimentation term that accounts for the effect of gravity on droplets motion. When $\tau_p$ is set to be zero, Eq. (4.10) becomes $\mathbf{V}_i(t) = \mathbf{u}(\mathbf{X}_i, t)$, which implies that the droplets exactly follow the turbulent flows. It is assumed that direct interactions between droplets are negligible during condensation/evaporation considering that the droplet sizes are too small comparing with the mean distance between two droplets. The fluid velocity $\mathbf{u}(\mathbf{X}_i, t)$ is obtained through bilinear interpolation of the Eulerian field at position $\mathbf{X}_i$.

## 4.2 Numerical Implementation

The numerical code consists of three packages (Eulerian fluid dynamics, Lagrangian droplet, and Coupling). The dynamic equations Eq. (4.1a) and Eq. (4.1b) are solved following the fraction-step algorithm [19]. The thermodynamic fields Eq. (4.4), Eq. (4.5) are solved with semi-implicit method coupling with fifth-order WENO scheme for the discretization of the hyperbolic term. The use of WENO scheme here is critical since it can well handle the numerical overshoots as well as keep the high order of the overall accuracy. To simplify the implementation, we adopt the external package Portable Extensible Toolkit for Scientific Computation (PETSc) [9] as the parallel linear solver and Parallel High Performance Preconditioners (HYPRE) [47] as the preconditioner. These two packages are widely used in the community of computational fluid dynamics and has a good parallel scaling in both Linux

clusters and supercomputers. The droplets position Eq. (4.9) and motion Eq. (4.10) are solved by implicit Euler method in consideration of efficiency and stability. The dynamical and thermodynamical fields are represented on an Eulerian rectangular grid, while the particles are explicitly tracked during the simulation and can utilize the information of the Eulerian field through bilinear interpolation. The Lagrangian particles impact on the Eulerian field through Eq. (4.6a), which acts as a source or sink term in Eq. (4.4) and Eq. (4.5). The fluid field is not directly affected by the particle ensemble, but is influenced by the thermodynamical field through Eq. (4.2). The time step size is adaptive to satisfy the Courant-Friedrichs-Lewy (CFL) condition. Parallel computing techniques are implemented with MPI library to increase the computational speed.

The numerical domain is set to be $0.512m^3$ with triply periodic boundary conditions. The computational grid is $256^3$, corresponding to grid spacing of $2mm$, close to the typical Kolmogorov length.

## 4.3 Turbulence initialization and external forcing

Since the DNS is performed in a small-scale turbulence environment, turbulence field is generated and maintained before injecting the particles into it. Many literatures [45] have demonstrated that the small-scale behavior in turbulent flows tends to be characterized by statistical homogeneity, isotropy and universality. Because of the universality we can hope to understand small-scale behavior by studying the simplest turbulent flows: homogeneous, isotropic turbulence. The two most frequently studied types of isotropic turbulence are

freely decaying, and forced statistically stationary turbulence, which are both studied in this dissertations. The decaying turbulence can be easily obtained by providing a solenoidal isotropic initial velocity field with random phases and prescribed energy spectrum, and then directly solve the Navier-Stokes equation to evolve the turbulence. In addition to the initial condition, the forced turbulence further requires a method to artificially force the low-wavenumber modes, so as to supply the energy dissipated by viscous effects. The initialization and external forcing method for homogeneous, isotropic turbulence are introduced below.

Similar to Rogallo's procedure, the initial velocity field is constructed in Fourier space satisfying continuity, isotropy, and having a given energy spectrum. Given the coordinates in three-dimensional Fourier space $\mathbf{k} = \{k_1, k_2, k_3\}$ and energy spectrum $E(k)$, the Fourier transformation $\hat{\mathbf{u}}$ of the velocity field is determined by:

$$\hat{\mathbf{u}} = \left\{ \frac{\alpha k k_2 + \beta k_1 k_3}{k(k_1^2 + k_2^2)^{1/2}}, \frac{\beta k_2 k_3 - \alpha k k_1}{k(k_1^2 + k_2^2)^{1/2}}, -\frac{\beta(k_1^2 + k_2^2)^{1/2}}{k} \right\} \tag{4.12}$$

where $k = (k_1^2 + k_2^2 + k_3^2)^{1/2}$. The coefficients $\alpha$ and $\beta$ are

$$\alpha = (\frac{E(k)}{4\pi k^2})^{1/2} e^{i\theta_1} \cos\phi, \quad \beta = (\frac{E(k)}{4\pi k^2})^{1/2} e^{i\theta_2} \sin\phi \tag{4.13}$$

where $\theta_1$, $\theta_2$, and $\phi$ are uniformly distributed random numbers on the interval $(0, 2\pi)$. It can be verified that $\hat{\mathbf{u}}$ satisfies the continuity condition $\hat{\mathbf{u}} \cdot \mathbf{k} = 0$

The energy function is defined as:

$$E(k) = \frac{16}{\sqrt{\pi/2}} \frac{u_0^2 k^4}{k_0^5} \exp(-\frac{2k^2}{k_0^2}) \tag{4.14}$$

where $u_0$ is the initial root-mean-square (r.m.s) velocity, and $k_0$ is the wavenumber at which the maximum of $E(k)$ occurs. The parameters $u_0$ and $k_0$ determine the exact power spectral shape. Different from the commonly used Kolmogorov spectrum, this function enforces the kinetic energy be concentrated in a relatively narrow band at the initial time, so as to not affect the turbulence behavior in larger wave number space. As turbulence evolves, the spectrum will quickly spread to the inertial range and dissipation range according to the Navier-Stokes equation. Fig. 4.1 illustrates the energy spectrum with different parameters. The parameters for most cases in this paper are $u_0 = 0.35m/s$ and $k_0 = |(1, 1, 2)| \approx 2.4$, which allows one to generate an initial turbulence field with reasonable Reynolds number and narrow energy band in large wave length.

As for the external volume force, several solutions have been proposed in the literature, representing two main approaches. The first is to construct the force in Fourier space to keep a constant energy injection rate, and thus this method requires knowledge of Fourier-transformed velocities. In [52, 25], the authors applied a force to the wavenumbers in the chosen shell, and guarantee a constant energy injection rate. The Ghosal's approach [52] can be simply

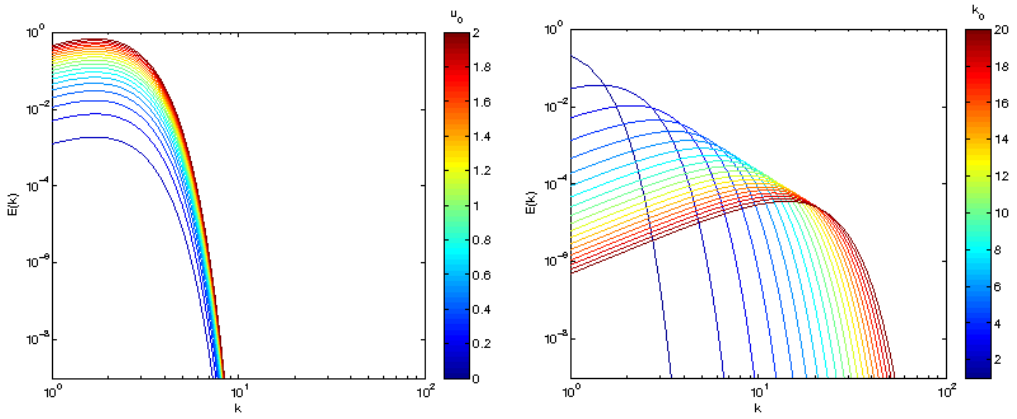


Figure 4.1: Initial energy spectrum with different parameters: left figure shows the energy spectrum with fixed $k_0 = 2.4$ while varing $u_0$ from $0m/s$ to $2m/s$, right figure shows the one with fixed $u_0 = 0.35m/s$ and varing $k$ from 1 to 20.

formulated in the Fourier space:

$$\mathbf{f}_e(\mathbf{k}, t) = \epsilon_{in} \frac{\mathbf{u}(\mathbf{k}, t)}{\sum_{\mathbf{k}_f \in \kappa} |\mathbf{u}(\mathbf{k}_f, t)|^2} \sigma_{\mathbf{k}, \mathbf{k}_f} \tag{4.15}$$

where $\mathbf{u}(\mathbf{k}, t)$ is the Fourier-transformed velocity function, $\mathbf{k}_f$ is chosen from a subset of the wavenumber space $\kappa$ containing a few wavevectors, for example $(1, 1, 2)$ plus all permutations with respect to components and sign, $\epsilon_{in}$ is a constant input energy rate. $\delta_{k,k_f}$ is a delta function. Statistics stationary homogeneous turbulence can be obtained in DNS by forcing the low-wavenumber modes. There still exist many other approaches based on the Fourier transformation. For example, Sullivan and Chasnov attempted to maintaining constant kinetic energy in the lowest wavenumbers [132, 28]. Eswaran and Pope in [45] utilized a stochastic process to determine external forcing scheme.

The second group is to evaluate the external force in physical space. This approach is attractive for applications since it does not require transformation to Fourier space and is easily integrated into physical-space numerical codes. Lundgren proposed a forcing function which is directly proportional to the velocity [104]. Rosales studied the properties of this linear forcing scheme for isotropic turbulence, and showed the linearly forced system converges to a stationary state that does not depend on the spectral shape of the initial condition [121]. The Lundgren's linear forcing scheme is determined with the following equation:

$$\mathbf{f}(\mathbf{x}, t) = \frac{\epsilon_{in}}{3u_{rms}^2}\mathbf{u} \tag{4.16}$$

During the simulation, the root-mean-square velocity $u_{rms}$ is recalculated every time step while the energy injection rate $\epsilon_{in}$ is kept equal to a constant. The cross section of the initial and stationary vorticity field is displayed in Fig. 4.2.

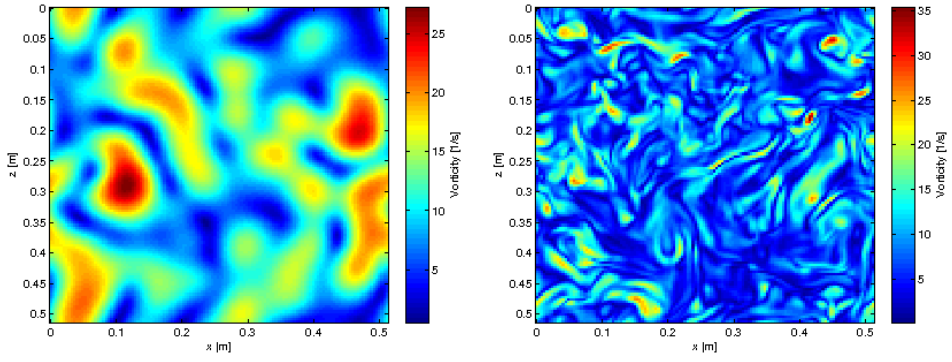


Figure 4.2: initial and final vorticity field $(1/s)$ in x-z cross-sectional plane for forced cases

## 4.4 Parallelism

The numerical experiments for DNS of cloud entrainment mixing are carried out on the Linux cluster "FASTER", named after the FASTER (Fast-Physics-System-Testbed and Research) project. The parallelization of this application involves three components: computation of fluid dynamics, computation of particles and runtime data analysis. The parallelization for the computation of fluid dynamics is done through domain decomposition and buffer extension. The computational domain is first uniformly divided into several partitions, and each partition has a buffer in each direction storing the information of its neighbor cells. The buffer is then updated every time step to provide the boundary information for the finite difference method. A typical $4 \times 4 \times 4$ partition of the computational domain is displayed in Fig. 4.3.

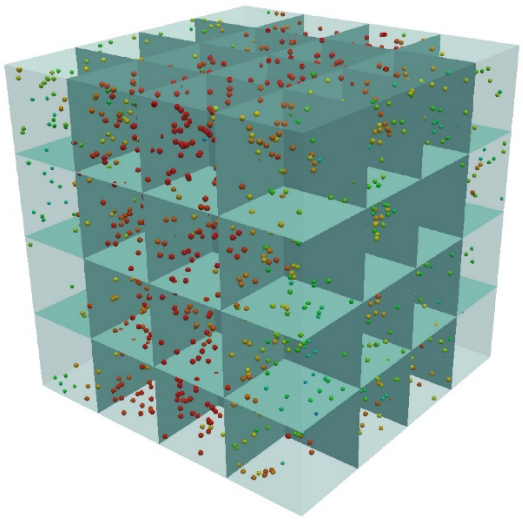

Figure 4.3: $4 \times 4 \times 4$ partitions of computational domain for the simulation of cloud droplets

The computation of the particles is parallelized using a hybrid computing technique. After each time step, the particles with zero radius or exceeding the limits of the local domain are removed from the current partition or sent to its neighbors. The computation can be further accelerated using OpenMP or GPU technique within one time step. The combination of using MPI between computing nodes and shared memory computing inside a node results in a hybrid computing technique, which has become the mainstream of high performance computing. In addition, the runtime data analysis also needs parallelization. For example, a parallel algorithm is required to calculate the mean and standard deviation of the droplets radius, temperature field, vapor mixing ratio field and velocity field. The calculation is first carried out in each processor by itself, and then aggregated to the answer using MPI reduce subroutine. However, since the size of the numbers $n$ is too large ($n = O(10^7)$ for cloud droplets and $n = O(10^6)$ for fields), simply summing a sequence of finite precision floating point numbers may accumulate the numerical error. To overcome this, we have applied the Kahan summation algorithm [72], in which the worst-case error bound is independent of $n$. Therefore, a large number of values can be summed with an error that only depends on the floating-point precision.

Next chapter gives a detailed description of the numerical experiments and results. The data processing and analysis will also be presented.

# Chapter 5

# Results and Analysis

The numerical results are displayed in this chapter for both parachute simulation and DNS of cloud entrainment process. In each part, the experimental setups are firstly introduced, and then the numerical results and quantitative analysis are presented.

## 5.1 Results for parachute simulation

This section aims to demonstrate the improvements and features of the new parachute model. Firstly, the experimental setup are introduced including the domain size, boundary conditions, and the initialization of parachute canopy and flow field. Then the turbulent-viscosity models are incorporated to the current code and compared with the laminar-version simulation. In the meanwhile, we try to investigate the new porosity model, and compare its effects with the previous non-porous parachute. Finally, the collision handling modules are applied to simulate the fabric collision and folding procedure.

### 5.1.1 Experimental setup

The experiments are carried out in a wind tunnel, which is set to be $14m \times 14m \times 40m$ as default with constant velocity at the inlet, outflow boundary condition with pressure $p = 0\ Pa$ at the outlet, and periodic boundary condition for the rest of faces. In some situations, the periodic boundary condition can be replaced by the non-slip boundary condition or wall boundary condition. The domain size may also change for some specific reasons. As for the parachute types, we consider several descending round canopies: C-9, G-11, G-12, G-14 and T-10.

The C-9 type parachutes are used in the Advanced Concept Ejection Seat (ACES) in all current U.S. jet fighters and personal parachutes in cargo airplanes. It has $8.5m$ nominal diameter, 28 number of gores, $7m$ suspension lines. The G-11 100-feet parachute is heavy capacity cargo parachute designed primarily for used in the aerial delivery of vehicular and bulk-type platform loads. It has $30.3m$ diameter, 120 suspension lines with length of $10.6m$. Each 10 of the consecutive suspension lines are connected to a suspension riser, and thus giving 12 suspension risers with length of $18.28m$. These suspension risers are further evenly divided into three suspension riser assemblies terminating in three riser attaching loops. The G-12 64-feet cargo parachute is designed for medium capacity use with the A-22 air delivery cargo bag. It has the similar structure as G-11 while with 64 suspension lines of $15.5m$ and 8 risers of $18.2m$. The G-14 34-feet cargo parachute provides the capability to deliver non-fragile supplies and equipment using low-velocity air delivery method. It can also be assembled to a cluster of three to support payloads up to 1500 $lbs$.

| Type | Diameter | Shape | Suspensions | Risers |
|---|---|---|---|---|
| C-9 | $8.5m$ | flat circular | $28 \times 7m$ | none |
| G-11 | $30.3m$ | flat circular | $120 \times 10.6m$ | $12 \times 18.2m$ |
| G-12 | $19.5m$ | flat circular | $64 \times 15.5m$ | $8 \times 18.2m$ |
| G-14 | $10.3m$ | flat circular | $32 \times 8.3m$ | $2 \times 0.76m$ |
| T-10 | $10.7m$ | parabolic | $30 \times 8.5m$ | $2 \times 0.76m$ |

Table 5.1: The technical specifications of parachutes

It has 32 suspension lines of length $8.3m$ with two risers. The T-10 35-feet personnel parachute is designed for combat mass-assault airborne operations and training. It is a parabolic-shape and has a nominal diameter of $10.7m$ with 30 suspension lines. The technical specifications for these canopies are listed in Table 5.1, and Figure. 5.1 illustrates the scales and initial shapes of different types of parachutes. The strings attached to the canopies of G-11 and G-12 reflect the result of combined effects of suspension lines and riser lines.

The computational mesh consist of two parts: the Eulerian mesh of fluid field and Lagrangian mesh of the parachute canopy. The fluid field is computed in a three-dimensional rectangular grid with uniform grid spacing. Various resolutions are tested depending on the domain size and the required accuracy. The canopies' mesh is generated by CGAL 2D triangulations package [62] based on the constrained Delaunay triangulation. The computational mesh of T-10 parachute is displayed in Fig. 5.2 for illustration.

Two major settings have been applied in the wind tunnel: the drop test

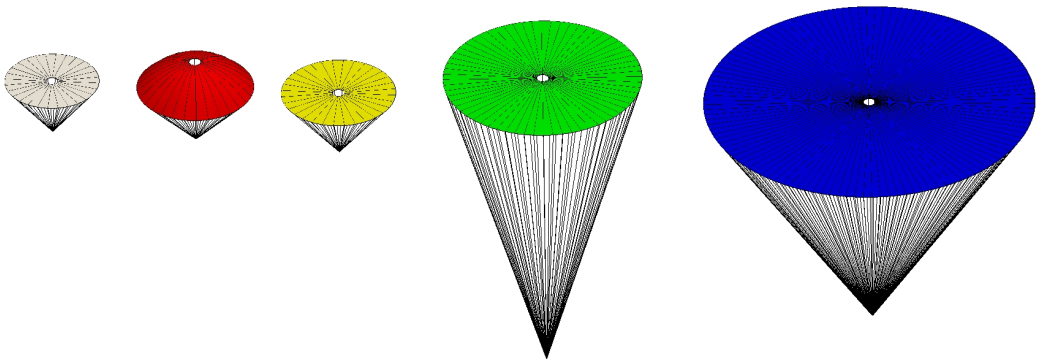

Figure 5.1: Initial shapes of parachutes, from left to right are C-9, T-10, G-14, G-12, G-11.

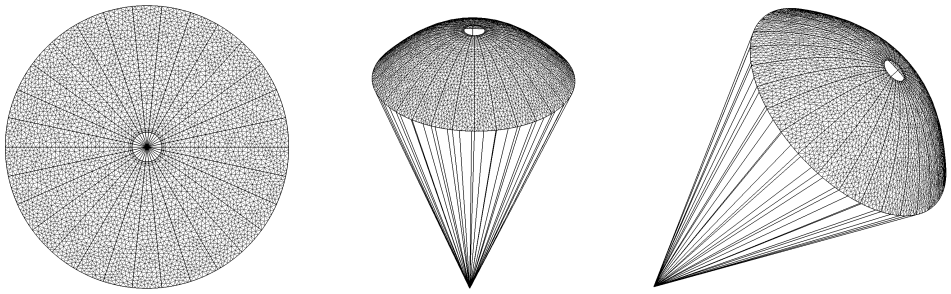

Figure 5.2: Initial mesh of T-10 canopy.

and the fixed-end test. In the drop test, the parachute is released at the top of the wind tunnel from either folded or unfolded shapes, and then the parachute freely falls in the tunnel until it hits the ground. This experiment aims to simulate the inflation stage of the parachute falling and study the factors affecting this procedure. The settings of drop test is close to a real situation, it, however, may terminate before the parachute reaching the stable state. Instead of free falling, the fixed-end test fixes one end of the parachute, and let the wind blow into the tunnel to inflate the canopy. This test can be used to measure the drag force on the canopy and is able to run arbitrarily long until the drag becomes stable.

### 5.1.2 Turbulent-viscosity models

The Reynolds number around a full-size parachute usually exceeds several millions, and hence the fluid flow has to be treated as turbulence [70]. The turbulent-viscosity models is one of many approaches to simulate the turbulent effects. In the current parachute simulation, a two-equation model has been developed attempting to duplicate several features during the turbulence-parachute interactions. The principle of the two-equation model is to modify the viscosity coefficient in the Reynolds-averaged Navier-Stokes equation, so as to approximate the effects of Reynolds stresses. The turbulent viscosity is computed from turbulence quantities such as $k$ and $\varepsilon$, which are solved from a couple of transport equations. In this section, three types of $k-\varepsilon$ model, standard, Re-Normalisation Group (RNG), and realizable turbulence model are investigated in the parachute drop test, and meanwhile compared with the

simulation results without using turbulence model, in which the viscosity is a constant. The velocity magnitude and turbulence viscosity for each model are recorded.

Fig. 5.3 shows the velocity magnitude and turbulent viscosity for different models at the same time frame. When the inlet velocity is $3m/s$, it shows that all the $k-\varepsilon$ models produce similar flow patterns, and they also resemble the pattern in the simulation without using turbulence model. However, the standard model tends to predict higher velocity magnitude than the RNG model and Realizable model and the flow patterns near the parachute vent are significantly different. As for the turbulent viscosity, the standard model resembles the pattern of RNG model while its numeric value is one order of magnitude larger. It is interesting to see that, in the realizable model, the viscosity after the parachute canopy is significantly smaller than the other two models. This is because the realizable model predicts higher turbulence dissipation rate $\varepsilon$ in the parachute wake, and thus the viscosity quickly decreases according to the formula $\mu_t = C_\mu \rho k^2/\varepsilon$.

Fig. 5.4 shows the same test, yet with inlet velocity $10m/s$. It is reasonable that the parachutes are subjected to stronger drag force than the case of $3m/s$, and hence falls much slower. The Realizable model still predicts the smallest value of the viscosity in the wake of the parachute. However, since the inflow velocity increases, the kinetic energy and dissipation rate produced at the inlet are able to propagate further and dominate the viscosity field. As the parachute falls and interacts with the inlet flow, it pushes the turbulent viscosity field away to each side and generates vortex-like pattern [70].

Fig. 5.5 displays the velocity and viscosity field with inlet velocity $15m/s$. This test gives a real turbulence field, and therefore the predictions of the three models are expected to be reasonable. The viscosity field predicted by the RNG model resembles the standard one but with more details in the wake, while the realizable model produces a completely different viscosity field. This field results in a more chaotic velocity field than the other two cases. We also notice that using turbulence model makes the parachute drop significantly faster than the case without turbulence model. This can be explained by the fact that the turbulence viscosity plays as a friction between fluid elements, and hence reduce the velocity magnitude and the drag force.

In summary, the standard $k - \varepsilon$ model assumes that the flow is fully turbulent, and the effects of molecular viscosity are negligible. This model was demonstrated to work fairly well for a wide range of wall-bounded and free shear flows. However, since the parachute canopy prevents the flow from passing through and the flow behind the parachute is less-turbulent when the inlet velocity is small, the standard model tends to overestimate the turbulent viscosity in the parachute wake. The RNG model is integrated to obtain an accurate description of how the effective turbulent transport varies with the effective Reynolds number, allowing the model to better handle low-Reynolds-number and near-wall flows. The RNG model is expected to be more responsive to the effects of rapid strain and streamline curvature than the standard model. Hence in the rapidly strained flows, it yields a lower turbulent viscosity than the standard model. The Realizable model is proposed by satisfying certain mathematical constrains on the normal stresses, consistent with the

physics of turbulent flows. It is believed that the modified form of the equation for $\varepsilon$ better represents the spectral energy transfer. The performance of the model has been found to be substantially better than that of the standard model.

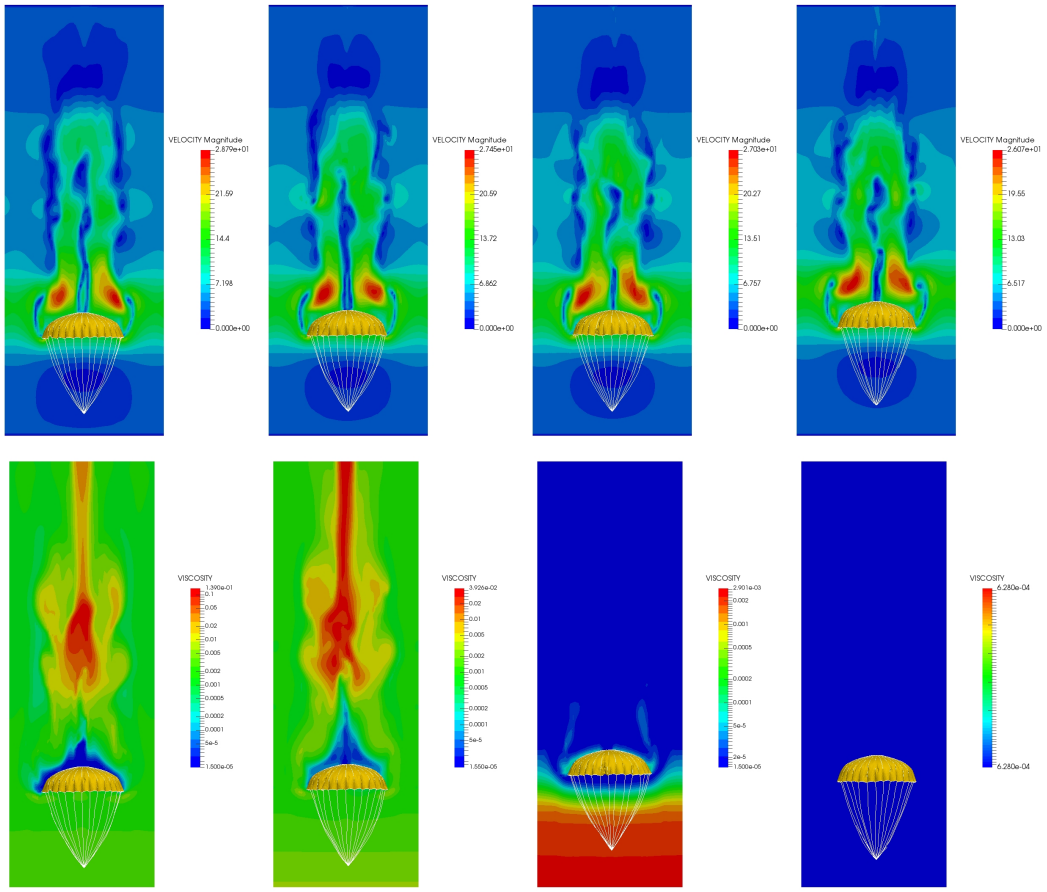


Figure 5.3: Velocity magnitude and turbulent viscosity for C-9 parachute using different turbulent models at the same time frame. The inlet velocity is $3m/s$. The figures in the upper row displays the velocity magnitude and the figures in the lower row shows the turbulent viscosity.

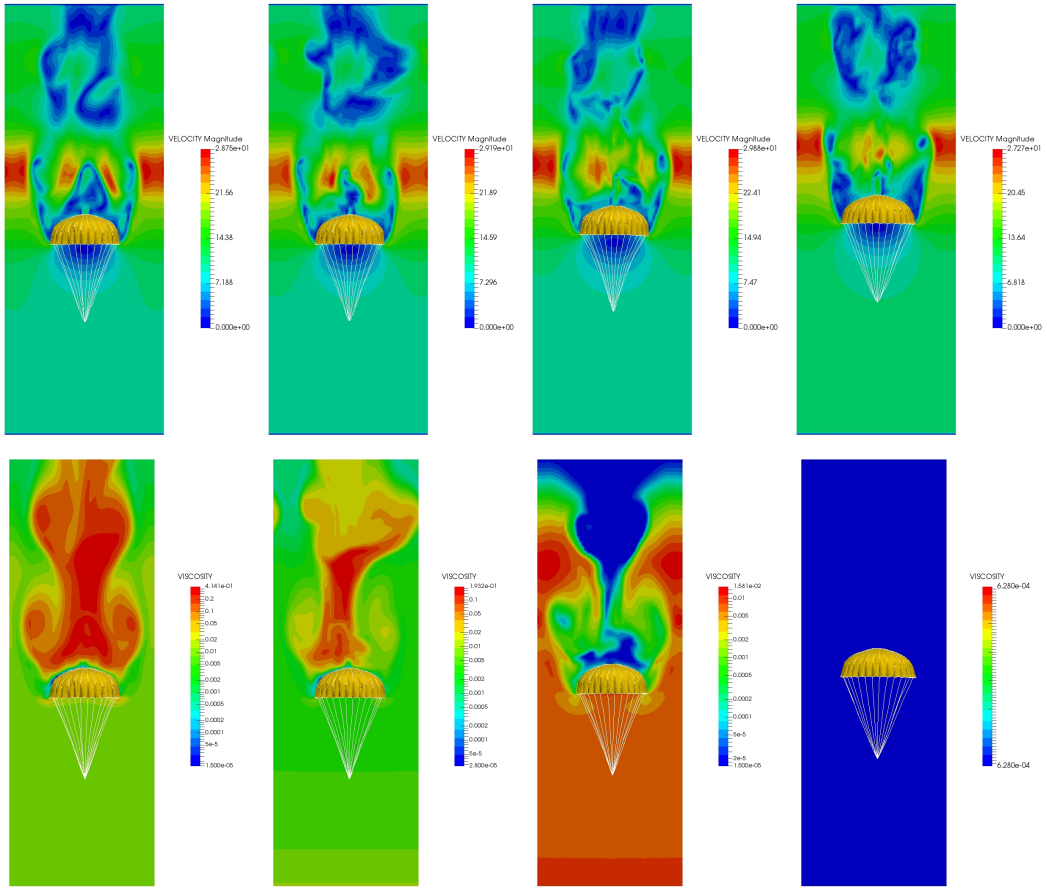


Figure 5.4: Velocity magnitude and turbulent viscosity for C-9 parachute using different turbulent models at the same time frame. The inlet velocity is $10m/s$. The figures in the upper row displays the velocity magnitude and the figures in the lower row shows the turbulent viscosity.

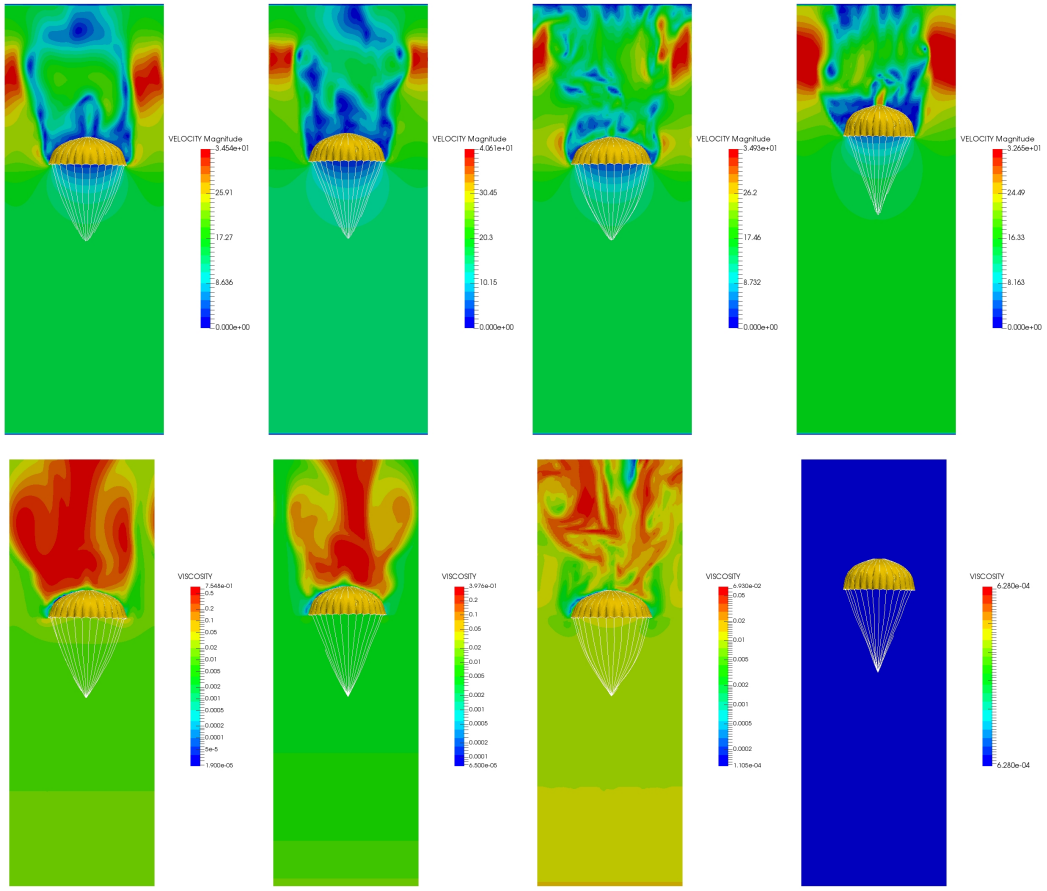


Figure 5.5: Velocity magnitude and turbulent viscosity for C-9 parachute using different turbulent models at the same time frame. The inlet velocity is $15m/s$. The figures in the upper row displays the velocity magnitude and the figures in the lower row shows the turbulent viscosity.

### 5.1.3 Porosity model

Two sets of numerical tests are carried out to validate the porosity model and its underlying numerical method. The first set of tests is designed to study a steady incompressible flow through a porous interface. The purpose of this numerical test is to verify the implementation of our method. The pressure drop is expected to appear exactly at the interface position as the model described in Eq. (3.16). The computational domain is set to be $4m \times 0.4m \times 0.4m$ in the $x$, $y$, $z$ directions, respectively. The velocity is driven by an inflow with a parabolic profile:

$$\mathbf{u}(x = 0, y, z) = [16U_{max}yz(L_y - y)(L_z - z)/(L_y^2 L_z^2), 0, 0] \tag{5.1}$$

where $U_{max}$ is the maximum velocity at the center of the inlet, $L_y$ and $L_z$ are the width of the channel in $y$ and $z$ directions. An outflow boundary condition together with the pressure $p = 0Pa$ is applied to the outlet at $x = 4m$ and a non-slip boundary condition of $\mathbf{u} = \mathbf{0}$ is imposed for the remaining faces. The porous interface is orthogonal to the streamline of flow and is placed at $x = 2m$. The coefficients in equation Eq. (3.16) are set as $\alpha = 10kgm^{-1}s^{-1}$ and $\beta = 0kgm^{-2}$. The profiles of velocity and pressure at different positions are shown in Fig. 5.6.

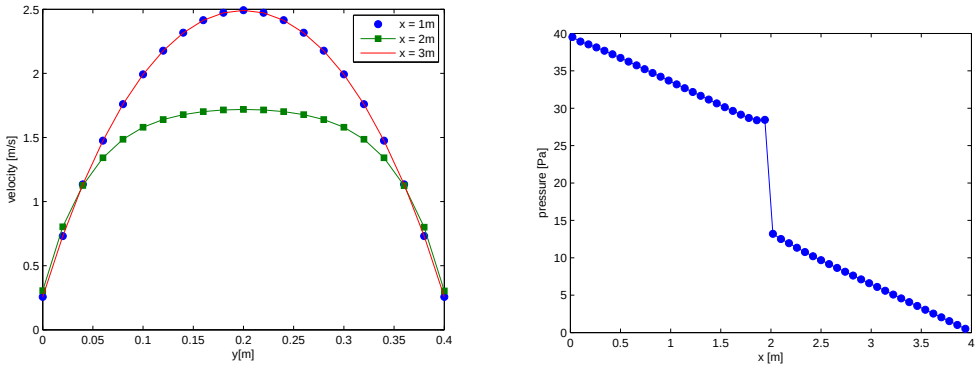


Figure 5.6: (left) Streamwise velocity profile at different sliced cross section and (right) pressure along a axial direction with $y = 0.2m, z = 0.2m$. The plot shows the velocity changes its profile as the fluid passes through the interface. The pressure drop at the interface is well captured.

In the second set of tests, a full-scale model is used to simulate the response of a realistic fabric surface in the channel flow. The boundary conditions of the computational domain are the same as the ones in the first test except that the domain size is $30m \times 10m \times 10m$. The elastic fabric surface located at $x = 5m$ is from the spring-mass model described in [126]. Thus the surface can be stretched or compressed due to the pressure drop at the interface. The fabric tested in the simulations resembles the properties of MIL-C-7020 type III fabric [46] with density $533.77kgm^{-3}$, Young's modulus $0.4309Gpa$. The values of viscous and inertial parameters are $\alpha = 162kgm^{-1}s^{-1}$ and $\beta = 48.82kgm^{-2}$, which are calculated by fitting the experimental data using a quadratic function. The permeability velocity and the pressure drop are measured by taking the average of the velocity and pressure drop over the entire surface. By imposing different values of inflow velocity, the functional relationship between

these two variables can be obtained. We approximate the experimental data in [46] with formula $[p]_\Gamma = \alpha U_n + \beta U_n^2$ and use it as the reference solution. The results presented in Fig. 5.7 show that our model can successfully reproduce the quadratic relationship observed in the experiments.

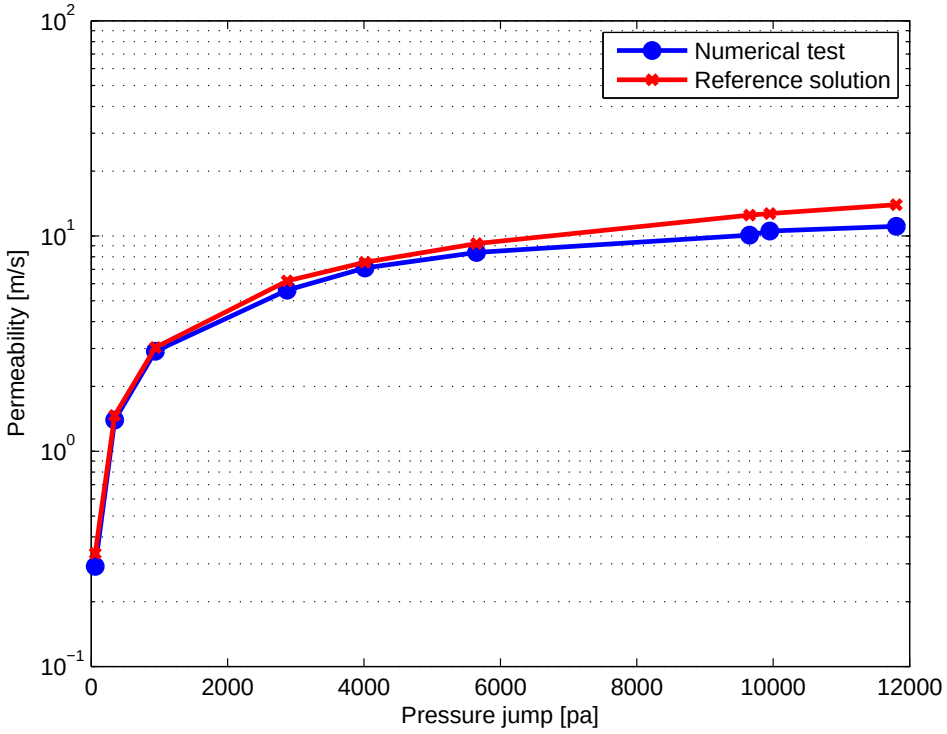


Figure 5.7: Plot of permeability velocity vs. pressure drop for the test case. The numerical results show that there is a quadratic relationship between the permeability velocity and pressure drop. Such relationship is observed in the experiment (red line)

In this section, we report our application of the porosity model to the parachute simulation and compare the drag force with the same force as in the

impermeable cases. The drag forces and drag coefficients are calculated with varying freestream velocity at the inlet. The drag measurement is carried out in a wind tunnel setting on G11 cargo parachute with its nominal diameter $10m$ and point of load fixed. The computational domain is set to be $14m \times 14m \times 40m$ with constant velocity at the inlet, outflow boundary condition with pressure $p = 0Pa$ at the outlet and periodic boundary condition for the rest of faces. The shapes of the parachute at different times are displayed in Fig. 5.8. First, The parachute canopy is inflated by the inflow air. It then oscillates for a few seconds due to the elasticity of the string and the canopy, a process that is called parachute breathing [119]. Eventually, it relaxes to a steady state shape due to damping friction force.

The drag force on the parachute is calculated by firstly integrating the pressure difference over all the surface elements (triangles) of the canopy and then projecting to the direction of the freestream velocity. After recording the drag force, the drag coefficient is calculated by the following formula:

$$C_d = \frac{F_d}{0.5\rho v_0^2 A_0} \tag{5.2}$$

where $\rho$ is the air density $= 1.2kg/m^3$, $v_0$ is the freestream velocity at the channel inlet and $A_0 = 78.5m^2$ is the area of parachute canopy at initial state. $F_d$ is the mean value of the drag force during the last $1s$ of the simulation. We observed that the drag coefficient increases at low descent velocities as displayed in Table 5.2. This can be explained by the fact that the drag force (or the pressure drop) increases linearly at a very small velocity according to

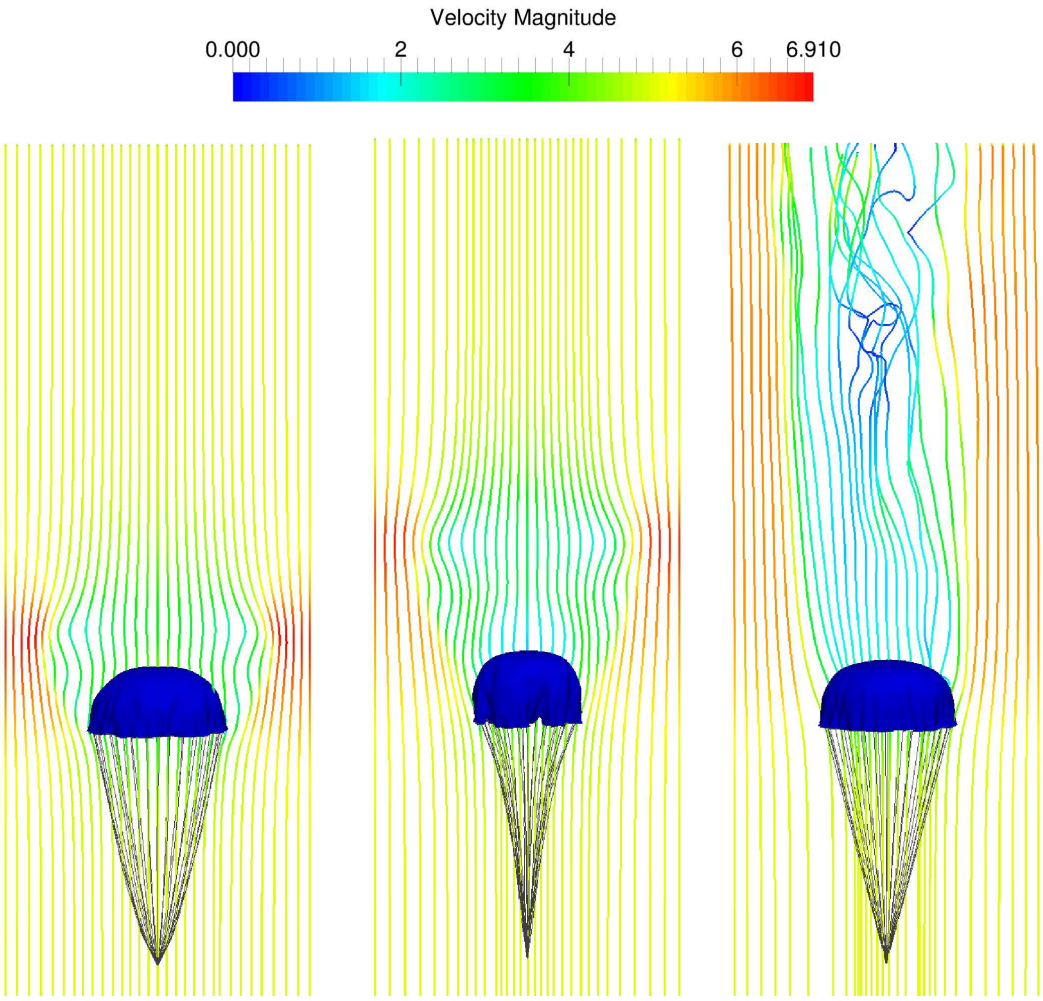


Figure 5.8: G11 cargo parachute in a numerical wind tunnel test with the inlet velocity $5m/s$. The three plots from left to right are the parachute shapes and velocity streamlines at the times $1.5s$, $2.5s$, and $15s$, respectively. The porosity coefficients in this case are set to be $\alpha = 6.7kgm^{-1}s^{-1}$, and $\beta = 3.1kgm^{-2}$. The oscillation of parachute canopy (parachute breathing) due to the elasticity of the parachute string and the canopy is observed during the initial a few seconds. The streamlines are plotted with color to show the velocity magnitude.

Ergun's equation, while the denominator has a quadratic growth.

| Inlet velocity $(m/s)$ | Drag coefficients |
|---|---|
| 1.0 | 1.89 |
| 2.5 | 1.31 |
| 5.0 | 0.92 |
| 7.5 | 0.63 |
| 10.0 | 0.49 |

Table 5.2: Drag measurements of the G11 parachute in wind tunnel tests with varying inlet velocity and fixed the porosity coefficients $\alpha = 6.7kgm^{-1}s^{-1}$, $\beta = 3.1kgm^{-2}$. The drag coefficient decreases as the inlet velocity increases.

To study the effects of the porosity on the parachute system, we fix the inlet velocity at $3m/s$ while gradually increasing the permeability of the parachute canopy. Although the porosity is not explicitly defined in our model, its relationship with the parameters $\alpha$ and $\beta$ can be obtained from the Ergun theory [145]:

$$\alpha = \frac{150\mu(1-\gamma)^2}{D^2\gamma^3}e, \qquad \beta = \frac{1.75\rho(1-\gamma)}{D\gamma^3}e, \tag{5.3}$$

where $\mu$ is the dynamic viscosity, $\rho$ is the density of the air, $\gamma$ is the porosity of the parachute canopy, $D$ is the characteristic length and $e$ is the thickness of the porous surface. The porosity is defined as the fraction of the volume of voids over the total volume [145]. Since the porosity is proportional to $(\alpha/\beta^2)^{1/3}$, we can use the quantity $(\alpha/\beta^2)^{1/3}$ instead of $\gamma$ to characterize the permeability of the parachute canopy for our model. In fact, the parachute system is af-

fected by the porosity through in two ways. On one hand, the porosity model will reduce the drag force on the parachute surface by lowering the pressure difference. Since the viscous drag is ignored here, the pressure drag becomes the driving force affecting the shape and behaviors of the parachute. On the other hand, the permeability of parachute canopy could significantly affect the aerodynamic field variables of the surrounding fluid such as pressure, flow velocity and vorticity. These, in turn, will impact the stability of the parachute system.

Another observation is that the parachute system would be more stable with finite porosity than that of solely impermeable fabric. This is shown by comparing the drag force on the parachutes with and without fabric permeability (see Fig. 5.9).

### 5.1.4 Collision handling and folding algorithm

In this section, the results of collision detection, collision handling, and folding algorithm are presented. Firstly, we have tested the performance of the collision handling algorithm. Several benchmark tests have been carried out: a round fabric falling on a rigid box; a round fabric falling on strings. These tests demonstrate the capability of our algorithm that can universally handle the interactions between fabric surface, rigid bodies and strings. Fig. 5.10 shows that the collision algorithm can produce visually plausible results and successfully eliminate all the intersections in the fabric. In order to test the performance of the collision algorithm on a more complicated geometry, a simulation of a fabric dropping from above of a rigid human model has been

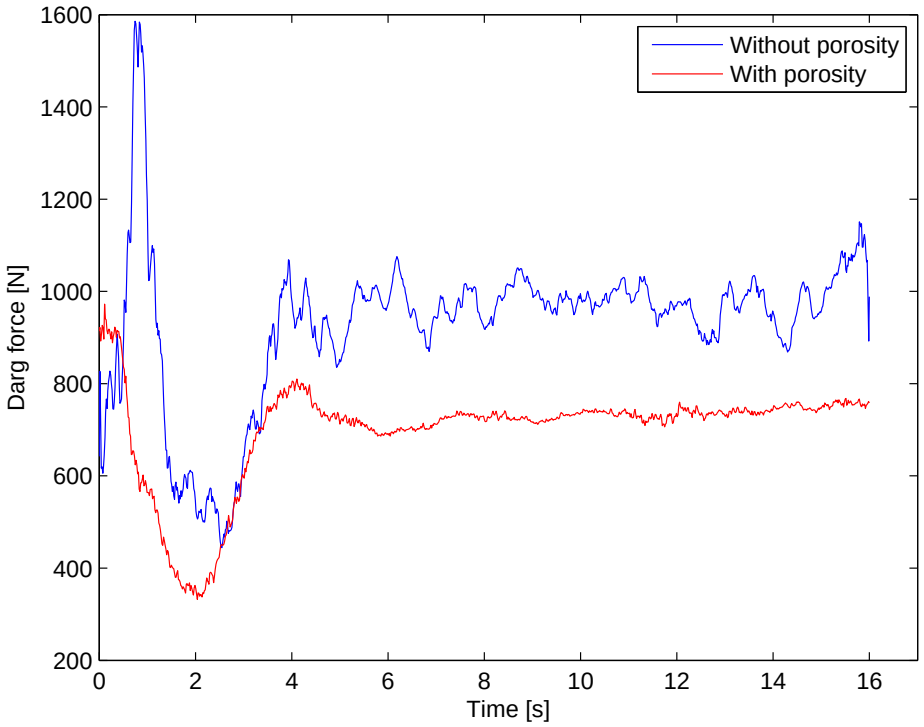


Figure 5.9: Comparison of drag force for G11 parachute between the permeable (red) and impermeable (blue) canopies in the same simulation. The porosity coefficients are set to be $\alpha = 6.7 kgm^{-1}s^{-1}$, $\beta = 3.1 kgm^{-2}$. The increase of porosity leads to the reduction of drag force and the vorticity in the wake of the canopy, thus make the drag force less oscillatory.

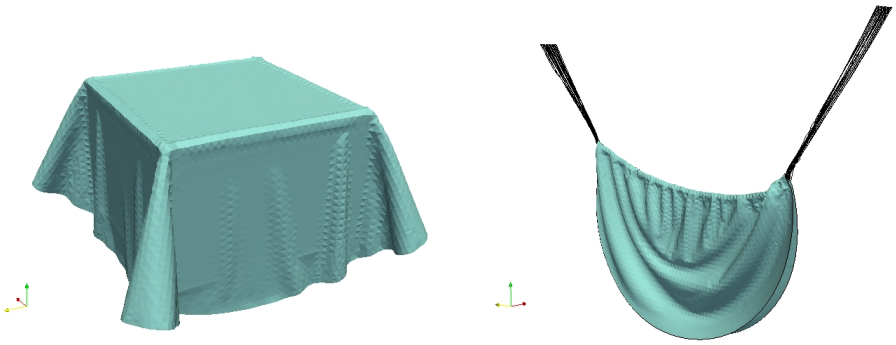

Figure 5.10: Benchmark test for collision detection and handling. Two cases are considered: interactions of fabric with rigid body (left) and elastic (right). Note that the fabric's self-interactions are also handled well during the collision procedure.

carried out. Fig. 5.11 displays three frames of the simulation and shows that the algorithm can well handle the collision between fabric and a non-trivial geometry.

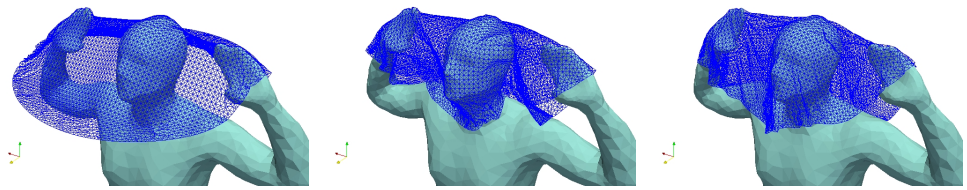

Figure 5.11: Three movie frames in the simulation of collision between elastic fabric and rigid human model.

The folding algorithm is verified by close a round parachute canopy with 16 creases. Here we name this folding pattern as "close folding". Two groups of the folding angles are assigned corresponding to the mountain and valley

crease alternatively. The final folding angle of the mountain crease is $-\pi$ and the folding angle of the valley crease is $3\pi/4$. The folding algorithm can automatically proceed and reach the desired state without any artificial controls. Three frames of the folding procedure are displayed in Fig. 5.12. It can be seen that the canopy remains unstretched and the final state is perfectly symmetric. To explain the folding algorithm more clearly, the 16 folding angles of the close folding are recorded and plotted in Fig. 5.13. The left panel is the behavior of the folding angles in the entire simulation and the right panel is a plot of the first 25 steps. It is clear that the angles are moved randomly while gradually approach to the final state. This is because the optimization algorithm is based on a random search, and hence the folding angle can not be guaranteed to move uniformly. However, this behavior would not affect the eventually folded state, and in the meanwhile, the folding angles still strictly satisfy the necessary condition of the origami design. In conclusion, this folding algorithm provides a highly flexible and universal way to design various folding patterns. Only the folding angles and creases pattern are required as inputs while the intermediate folding process is completely automatic.

## 5.2 Results for cloud entrainment and mixing

### 5.2.1 Initial conditions

Three different initial configurations of cloudy area are used to investigate the impact of cloudy area configuration. Case 1 follows that used in [2] whereby water mixing ratio is defined according to the sign of the velocity function in

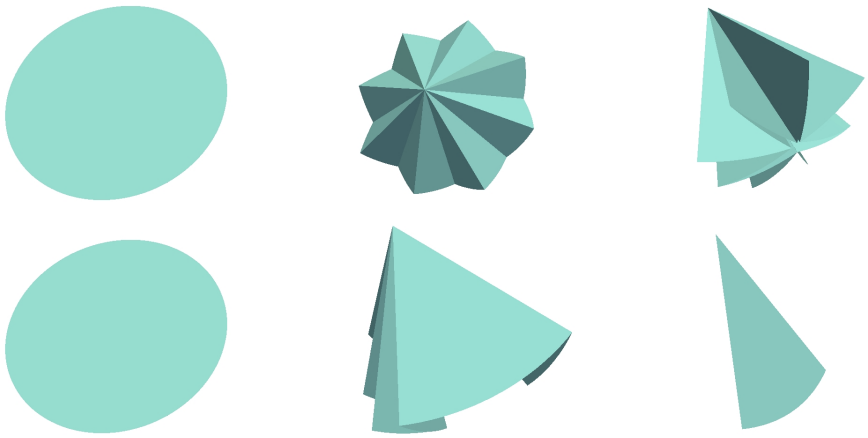

Figure 5.12: Three movie frames of the parachute canopy shape in the folding process. The upper panel displays the close of a parachute canopy and the lower panel displays the flat fold.

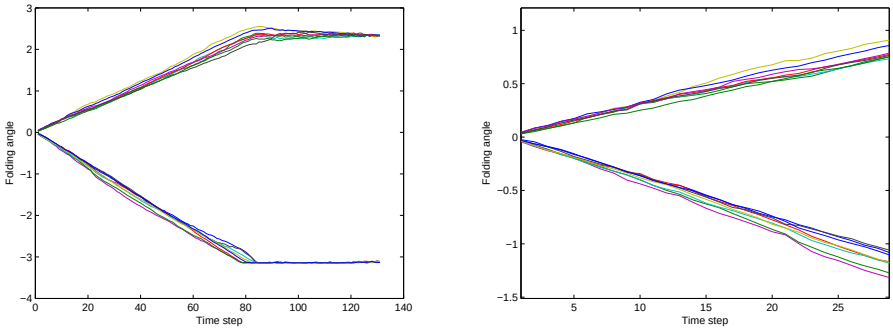


Figure 5.13: Folding angles of the 16 creases. Right panel shows the magnification of the irst 25 steps.

physical space such that

$$\text{case 1: } q_v(\mathbf{x}, t=0) = \begin{cases} q_v^{max}, & u(\mathbf{x}) > 0 \\ q_{v,e}, & u(\mathbf{x}) \leq 0 \end{cases} \tag{5.4}$$

where $q_v^{max} = 3.95g/kg$ is the maximum amplitude of $q_v$, which exceeds $q_{v,s}$ by 2%, and $q_{v,e} = 0.03g/kg$ is the vapor mixing ratio of the clear air. $u(\mathbf{x})$ is the first component of the fluid velocity.

In [84], the author investigated a slab-like cloud configuration approximated with a smooth function to avoid the Gibbs phenomenon (numerical overshoots at sharp interfaces). Similarly, our Case 2 is designed to study the slab-like configuration but approximiated with a simple discontinuous function given by

$$\text{case 2: } q_v(x, t=0) = \begin{cases} q_v^{max}, & (L-d)/2 \leq x < (L+d)/2 \\ q_{v,e}, & \text{elsewhere} \end{cases} \tag{5.5}$$

where the $q_v^{max}$ and $q_{v,e}$ are the same as in Case 1. $L$ is the length of computational domain, and $d = L/2$ is the width of the cloud slab.

It is well known that entrainment-mixing processes can also occur near cloud tops, esp., for stratiform clouds [101, 162]. To mimic the cloud-top entrainment-mixing process, herein we add a new cloud configuration by rotating Case 2 by 90 degree, and name it Case 3.

$$\text{case 3: } q_v(z, t=0) = \begin{cases} q_v^{max}, & (L-d)/2 \leq z < (L+d)/2 \\ q_{v,e}, & \text{elsewhere} \end{cases} \tag{5.6}$$

The temperature field is initialized by imposing the neutral buoyancy condition [85] such that:

$$T(x, t = 0) = T_0 - 0.608 T_0 [q_v(x, t = 0) - q_{v0}] \tag{5.7}$$

where the reference values are defined by the domain averages $T_0 = \langle T(t = 0) \rangle_V$ and $q_{v0} = \langle q_v(t = 0) \rangle_V$. This neutral buoyancy condition ensures the initial cloudy area having higher water vapor mixing ratio but lower temperature compared to the environment. Note that this procedure is only performed for the initial temperature field; later temperature field completely follow Eq. (4.4) afterwards. Fig. 5.14 compares the initial fields of water vapor mixing ratio and temperature for the three cases. The discrepancies between the initial water vapor and temperature fields are self-evident, allowing for examination of the impacts of the initial configuration of cloudy area on entrainment-mixing processes. Note that all the three initial configurations have the same initial cloud fraction of 0.5, and the same dynamical field.

At beginning, a total of $10^7$ droplets with the same radius of $15 \mu m$ are randomly placed in the cloudy area according to the Poisson point process, giving a droplet number concentration of $153 cm^{-3}$. Note that for the forced turbulence scenario, the velocity field needs a few steps (5 seconds here) to relax to a steady state. Therefore, the droplets are released to move and change their sizes according to the physics law after this spin-up period. For the decaying turbulence, the droplets are released at time $t = 0s$ since there is no needs to seek for a steady state. The simulation is terminated when droplets

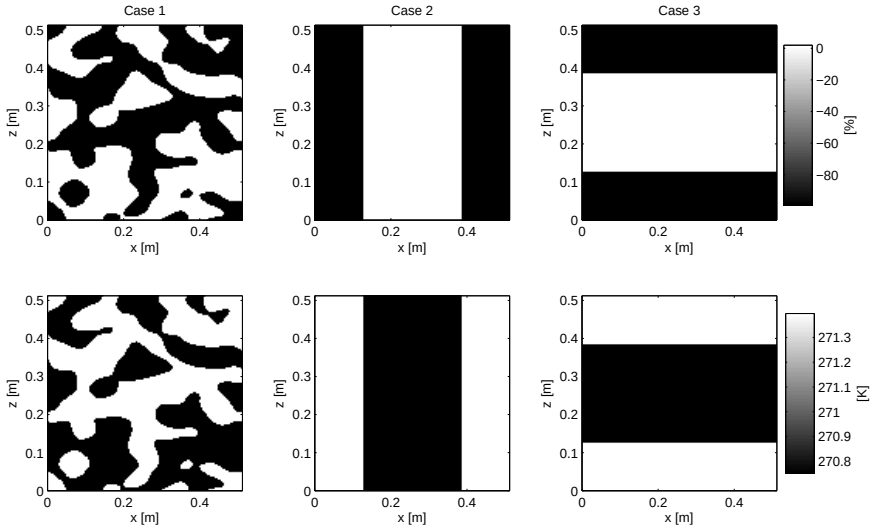


Figure 5.14: Cross sectional view of the initial supersaturation and temperature (K) field for different cases: case 1, 2, 3 from left to right. The cloudy part occupies about half of the computational domain.

| Quantity | Symbol | Value | Quantity | Symbol | Value |
|---|---|---|---|---|---|
| Grid points | $N$ | 256 | Droplet radius | $R_0$ | $15\mu m$ |
| Box length | $L$ | $0.512m$ | Environ supersat | $S_e$ | $-99\%$ |
| Grid size | $a$ | $0.002m$ | Cloud supersat | $S_c$ | $2\%$ |
| Viscosity | $\nu$ | $1.5 \times 10^{-5} m^2 s^{-1}$ | Number concentration | $N_c$ | $153 cm^{-3}$ |
| Dissip rate | $\epsilon$ | $2.0 \times 10^{-3} m^2 s^{-3}$ | Eddy turnover time | $\tau_L$ | $4.27s$ |
| Dissip length | $\eta$ | $10^{-3}m$ | Evaporation time | $\tau_{evap}$ | $2.09s$ |
| Dissip time | $\tau_\eta$ | $0.087s$ | Reaction time | $\tau_{react}$ | $4.52s$ |

Table 5.3: Summary of key model parameters and initial conditions

completely evaporate or the field becomes nearly uniform ($std < 0.0002$).

For convenience, Table 5.3 summarizes the key quantities and initial conditions.

### 5.2.2 Dynamical fields and microphysics

Fig. 5.15 compares the temporal evolutions of the domain mean, standard deviation and relative dispersion of turbulent kinetic energy (TKE, a, b, c), of temperature (d,e,f) and water vapor mixing ratio (g, h, i), and supersaturation (j, k, l) between the six scenarios. As expected, the mean TKE and its standard deviation for the three forced simulations (F1, F2 and F3) remain approximately constant determined by the large scale forcing after a short relaxation at the initial time. However, it is interesting to observe a transient turbulence enhancement before gradually decaying to zero in the decaying cases, especially for D2 and D3. This transient enhancement likely results from the buoyancy effect, which is caused by the deviation of tempera-

ture and vapor mixing ratio to the reference value according to Eq. (4.2). The D3 simulation exhibits the strongest enhancement, followed by D2. But for D2 the enhancement lasts longer. The mixing in D3 is accelerated by the sedimentation effect, making it a slightly stronger and faster than D2. Note that D1 can be regarded as the intermediate stage of mixing process in D2 or D3, and therefore the buoyancy effect quickly disappear and show little enhancement in the figure. Most of the droplets have a chance to enter into the clear air and evaporate at an early stage. Evaporation process absorbs latent heat from the environment, resulting in deviation of the temperature field from the mean value. The transient enhancement can be seen more clearly from the standard deviation of temperature. The transient enhancement is weaker for the three forced simulations F1, F2 and F3 (but still stronger than that of TKE). It is noteworthy that the behavior of transient turbulence enhancement does not appear in the field of water vapor mixing ratio, which is consistent with [85]. Notice that the vapor mixing ratio in the clear air is much lower than in the cloudy air. The droplets entering into the clear area will quickly turn into vapor while the droplets staying in the cloudy area continue to grow by condensation. This phase transition process reduces the difference of vapor mixing ratio between clear air and cloudy air, thus the transient growth of the deviation can hardly be observed. The behavior of supersaturation reflects the combination of temperature and water vapor mixing ratio, as expected. The variations manifest themselves in the plots of relative dispersion defined as standard deviation to the mean of the corresponding variables.

Fig. 5.16 shows temporal variation of the cloud droplet size distribution

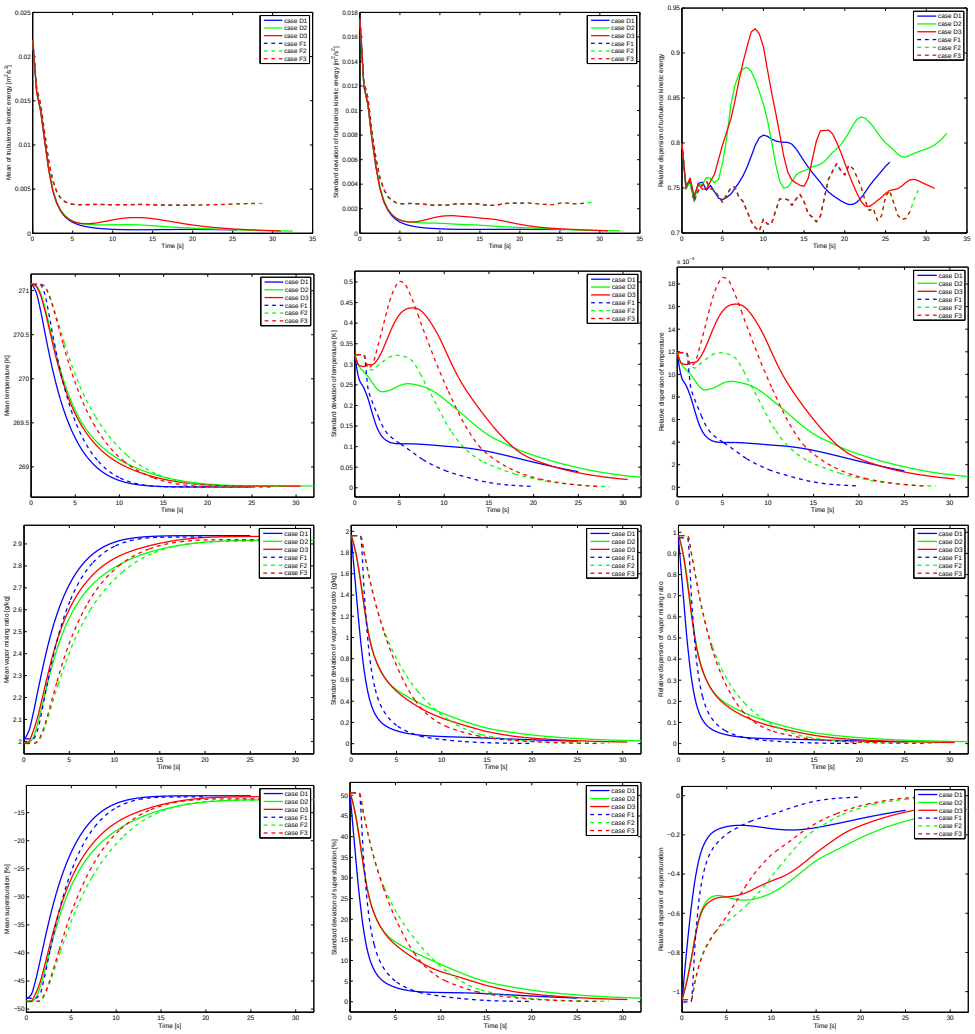

Figure 5.15: Thermodynamics of three cases. The left column is the mean value, middle column is the standard deviation and left column is the relative dispersion defined as the ratio of standard deviation and mean. The rows from top to bottom are turbulent kinetic energy, temperature, vapor mixing ratio and supersaturation.

for all the six simulations. A few points are evident. First, the droplet size distributions start with a monodisperse distribution with a uniform droplet radius of $15\mu m$. As the turbulent mixing between the subsaturated environment and the supersaturated cloudy air proceeds, some droplets evaporate, the size distributions gradually shift to small sizes and broadens until all droplets completely evaporate or the background environment become saturated. Due to the initial configuration, the final states of all the cases contain no droplets and result in unsaturated environments. Second, the three cases with decaying turbulence (D1, D2 and D3) are quite different in their evolutions of size distributions. However, the difference between the three forced turbulence cases (F1, F2 and F3) almost disappear, demonstrating that the buoyancy effect is overwhelmed by the external forcing and the differences in the decaying cases are caused by the buoyancy term in Eq. (4.2). The role of buoyancy in broadening is also evident from the comparison of the corresponding simulations with decaying and forced turbulence.

To better illustrate the impacts of different simulation scenarios, Fig. 5.17 shows the temporal evolution of the domain-mean liquid water content (a), droplets concentration (b), mean volume radius (c), mean radius (d), standard deviation (e) and relative dispersion (f). Several points are evident. First, in all the simulations, LWC and droplet concentration decrease as turbulent mixing and droplet evaporation proceed. The mean volume radius and mean radius also decreases with time because the decrease of liquid water content is stronger/faster than that of droplet concentration. Second, in all the simulations, standard deviations first increase, peak at some time, and

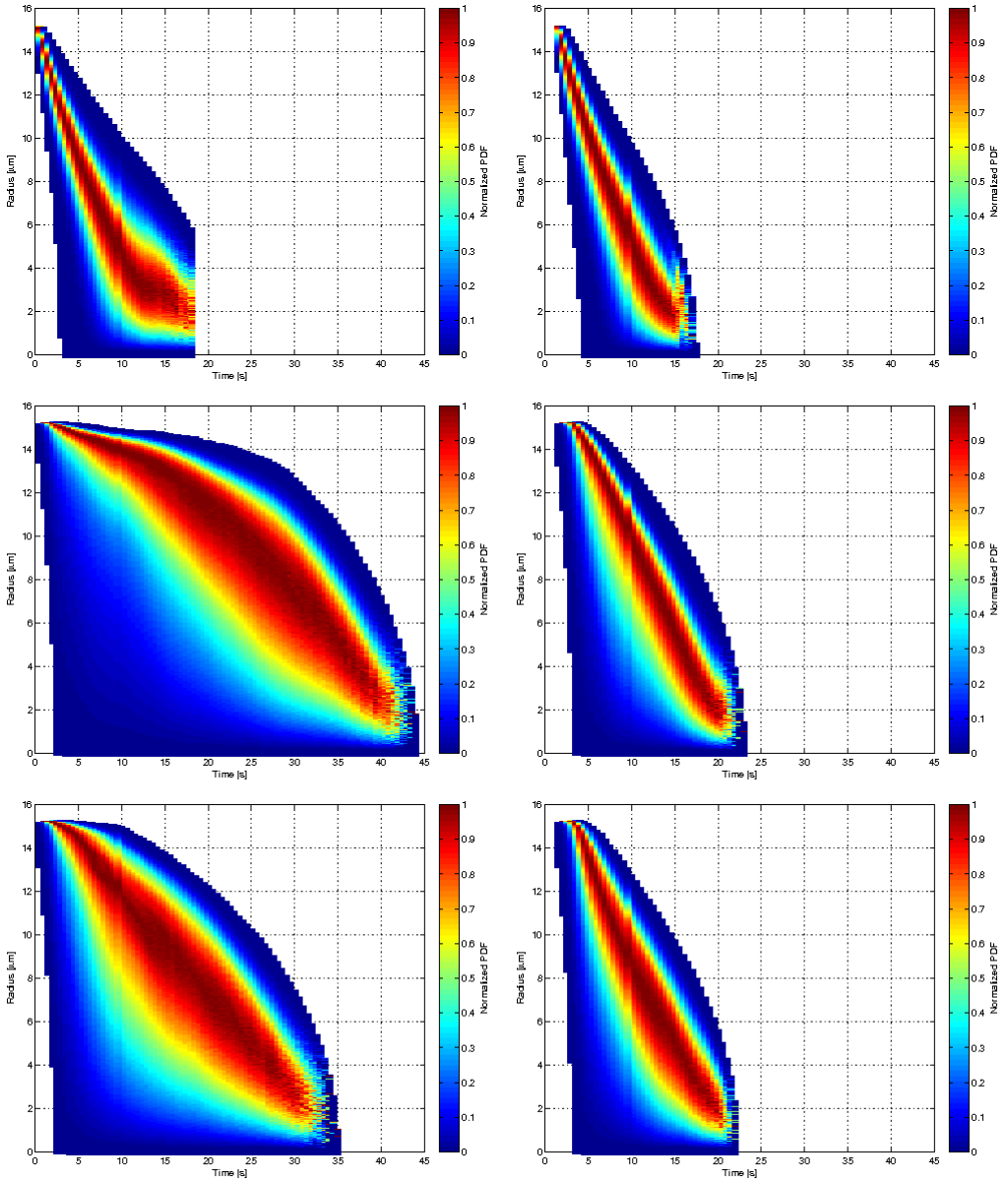


Figure 5.16: Evolution of radius distribution for decaying turbulence (left column) and forced turbulence (right column). From up to bottom are case 1, case 2 and case 3 respectively.

then decrease beyond the peak time. The occurrence of maximum standard deviation steps from the combined spectral broadening related to entrainment-mixing processes and the shrinking of droplet populations due to evaporation. Also noteworthy is that the peak standard deviations occur between $x$ and $y$ for all the six simulation scenarios. The coupled variations of mean radius and standard deviation result in relative dispersion peaks at a much later time compared to standard deviation. Finally, despite the commonalities, the differences among the different scenarios are evident. Since the configuration of Case 1 is close to an already-mixed case, its number concentration and mean radius decay at a faster rate, and the standard deviation of the droplet size is lower than other cases. Case 2 and Case 3 have no big difference except the number concentration. Since the mixing process of Case 3 is accelerated by the buoyancy effects in the vertical direction, the number concentration of Case 3 has a stronger decrease than Case 2. Comparing the forced cases and decaying cases, at least two phases can be observed. In the first stage, the forced cases contain more liquid water content, larger number concentration and mean radius than the corresponding decaying cases, while exhibit an opposite result in the later stage. This implies that the decaying cases initially have faster mixing and evaporation while are overtaken by the forced turbulence later.

Note that a droplet with radius smaller than $1\mu m$ will be immediately removed from the computational domain, and therefore will not contribute to any statistical results.

In summary, the shape of the cloud filament has no influence on the final state after the mixing but will affect the intermediate process, but cannot be

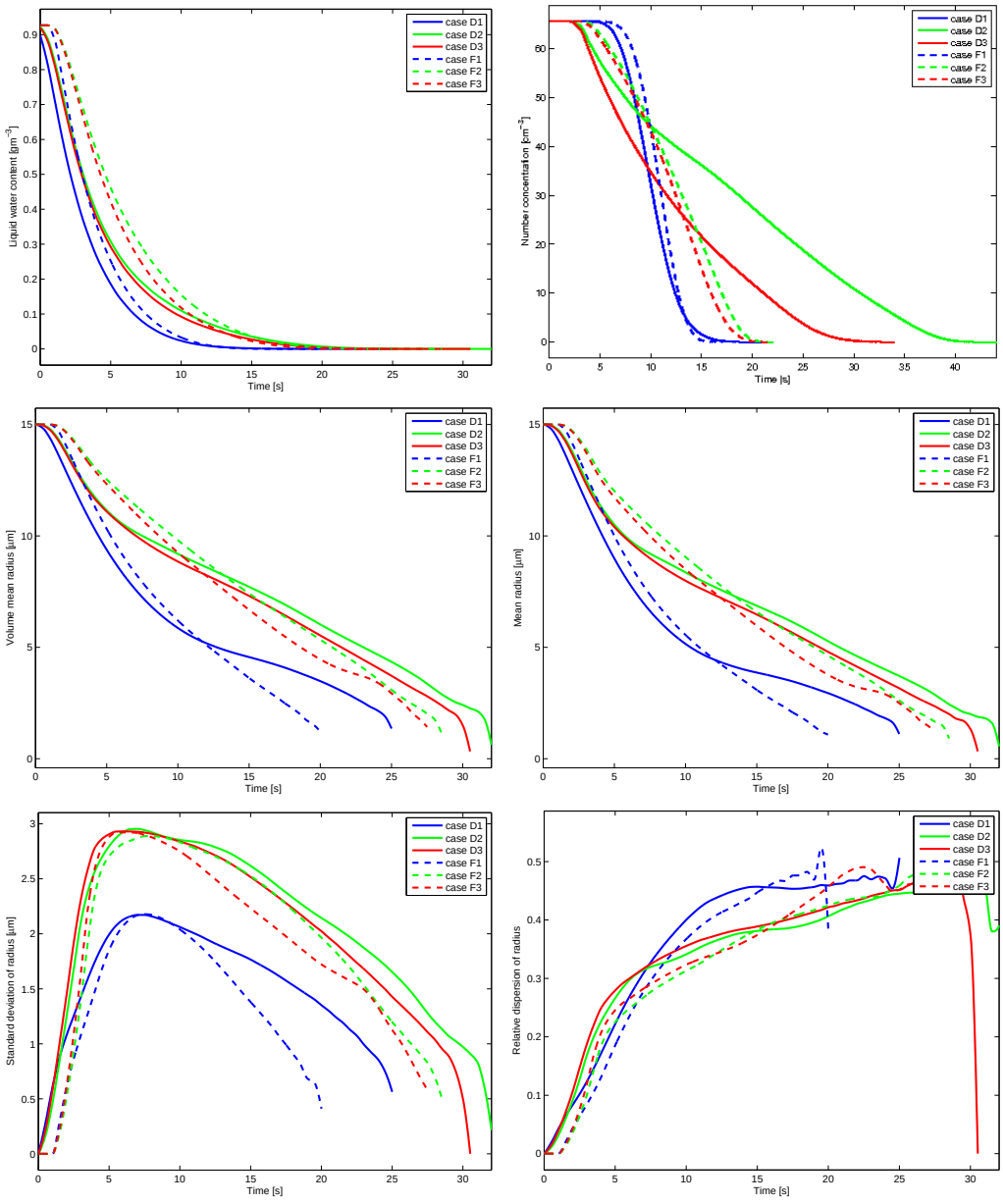


Figure 5.17: From top to bottom and left to right are temporal evolutions of (a) droplets concentration, (b) liquid water content, (c) mean volume radius, (d) mean radius, (e) standard deviation and (f) relative dispersion.

completely ignored for their intermediate states. The results suggest that the initial shape of cloud filaments should be considered as an important factor when studying mixing scenario with or without external forcing. All cases have the same equilibrium state with a zero liquid water content, i.e. all droplets eventually evaporate. The rate at which droplets evaporate is higher for the forced turbulence than the decaying turbulence except D1, in which all the droplets are quickly exposed to the same environment and begin to evaporate, leading to its number concentration curve to be similar as the forced case.

### 5.2.3 Turbulent entrainment-mixing processes

Turbulent entrainment of dry environmental air and subsequent turbulent mixing between cloudy air and environmental air and associated droplet evaporation are likely primary factors that affect the droplet size distributions and corresponding microphysical properties. There are two limiting entrainment-mixing mechanisms proposed in the literature. One is that the entrained air and the cloudy air are mixed evenly and all cloud droplets evaporate with the same proportion [153]. This type of mixing is referred to as homogeneous mixing. The other type of mixing is inhomogeneous mixing, where the entrained air mixes with only some portion of cloud parcel and evaporate all droplets in this portion completely while the droplets in the rest of the cloud parcel remain intact [7]. Ambient clouds often fall between the two limiting mechanisms. To characterize the effect of turbulent entrainment-mixing processes on microphysical properties, the $R_v^3 - N_c$ diagram was introduced in [21], and has been widely applied to study the homogeneous/inhomogeneous entrainment-

mixing process in observational studies and DNS simulations. [2, 4, 5] were probably the first studies that applied the mixing diagram analysis to DNS simulations with bin microphysics. [85] further applied to the mixing diagram analysis to their particle-resolved DNS simulations. In addition to their model differences, [2, 4, 5] used Case 1 initial configuration of cloudy area whereas Kumar et al used a configuration similar to our Case 2. This section extends these pioneering studies to examine the results of all the six scenarios by use of the mixing diagram analysis.

In addition to the domain mean as examined by Andrejczuk et al, we also examine smaller averaging boxes to obtain better ideas of statistics by following Kumar et al. [85] to divide the computational domain into 64 equal-sized sample boxes. We keep tracking the volume mean radius and number concentration at each time step in each sample box. Fig. 5.19 shows the mixing diagrams for the six scenarios. The solid green dot represents the value sampled in each sample box at each time step; the red curve denotes the DNS domain average, with arrows indicating the direction of temporal evolution. The corresponding homogeneous mixing line (black dot) and extreme inhomogeneous mixing line (black solid) are plotted in the diagram as references. Note that in the top panel, the mixing diagrams for D1 and F1 do not start from the $(1, 1)$ point since the initial droplets in a sample box have already been diluted and their number concentration are thus less than the adiabatic value. The droplet number concentration remains nearly unchanged as the droplet size decreases until some time has elapsed, suggesting an extreme homogeneous mixing. The difference between forced turbulence and decaying

turbulence is not obvious, except a wider range of variability in the shape of the mixing trajectories for the decaying turbulence D1, since the forced turbulence will foster the mixing procedure, resulting in similar states in different sample boxes. As claimed in [2], this configuration excludes the initial dilution process and can only be used to simulate the mixing process after dilution.

The middle panel shows the mixing diagrams for case D2 and F2. These cases have the same configurations with [85]. However, a sharp initial profile of vapor mixing ratio was used in our simulation. This results in an unsaturated vapor mixing ratio at final state, leading to completely evaporation of the droplets. The phenomenon of inhomogeneous offset described by [85] can also be observed in the figures: the mixing trajectories tend to shift to smaller values of $N_c/N_{c,a}$. This inhomogeneous offset is due to the initial dilution process, in which the droplets number concentration in the sample boxes is diluted while the droplets mean radius in the sample box doesn't change too much. As mixing proceeds, the turbulent time scale in the decaying case continues to increase while the time scale for the forced turbulence remains unchanged. Therefore, the inhomogeneous mixing is more likely to occur in D2, leading to a slightly stronger deviation from the homogeneous mixing line. A similar conclusion can be obtained in the bottom panel for case D3 and F3.

It is noteworthy to observe that the points in Case 3 are more scattered than other cases in the mixing diagrams. During the initial several seconds, a part of the points move along the inhomogeneous line and others are below the red curve and closer to the homogeneous line. As mixing proceeds, this two groups move towards $(0, 0)$ point and finally merge together. We interpret

this divergence by considering the following facts. According to our way of selecting sample boxes, the cloud slab will be divided into two groups: the upper layer and the lower layer. On one hand, due to the sedimentation effect, a part of the droplets will escape from the upper layer and enter into the lower layer, thus making the number concentration of upper level sample boxes decreased and their volume mean radius unchanged. On the other hand, the evaporating droplets below the lower layer may reentering into the lower sample boxes by the turbulent mixing, leading to reduced volume mean radius and slowly decreasing number concentration.

Also noteworthy is that the difference between the mixing diagrams lies primarily in the cloudy area configurations, esp., Case 1 vs. Case 2 or Case 3, instead of lying in whether or not the turbulence is forced or freely decaying as shown Fig. 5.16 and Fig. 5.18 for the temporal evolutions of droplet size distribution and supersaturation, respectively. This result suggests the potential for a unifying parameterization of different mixing mechanisms detailed next.

Since the real entrainment-mixing mechanism can fall anywhere between the limiting mixing mechanisms, it is desirable to define some measure that covers all the possible mixing processes. We generically called such a measure as homogeneous mixing degree since a larger homogeneous mixing degree indicates that the mixing process is closer to the limiting homogeneous mixing process. Based on the fact that the horizontal line in the $R^3 N_c$ diagram corresponds to the extremely inhomogeneous mixing whereas the vertical line implies extremely homogeneous mixing [5], the homogeneous mixing degree can be quantified by the instantaneous slope of the trajectories in the mixing

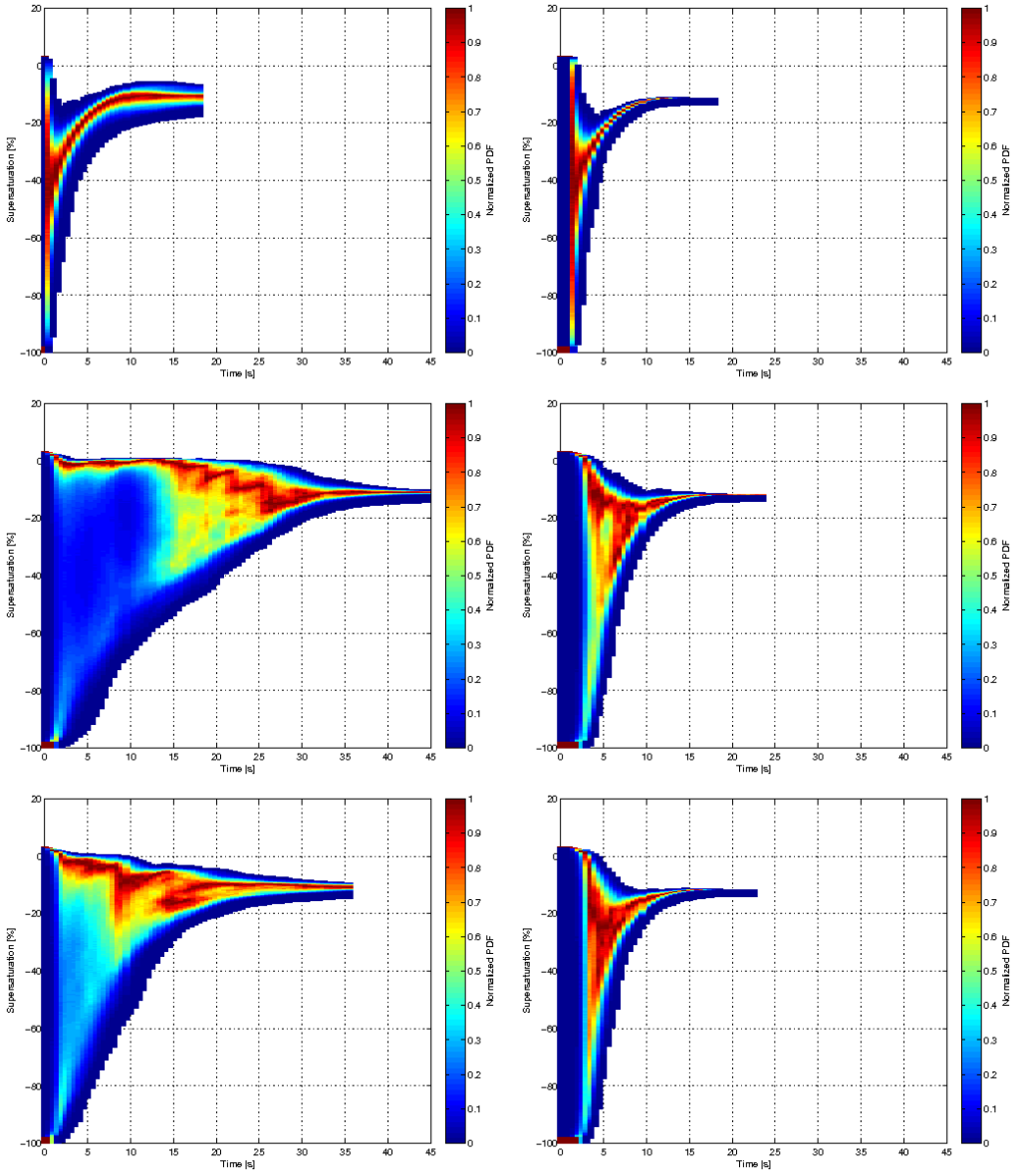


Figure 5.18: Evolution of supersaturation distribution for decaying turbulence (left column) and forced turbulence (right column). From up to bottom are case 1, case 2 and case 3 respectively.

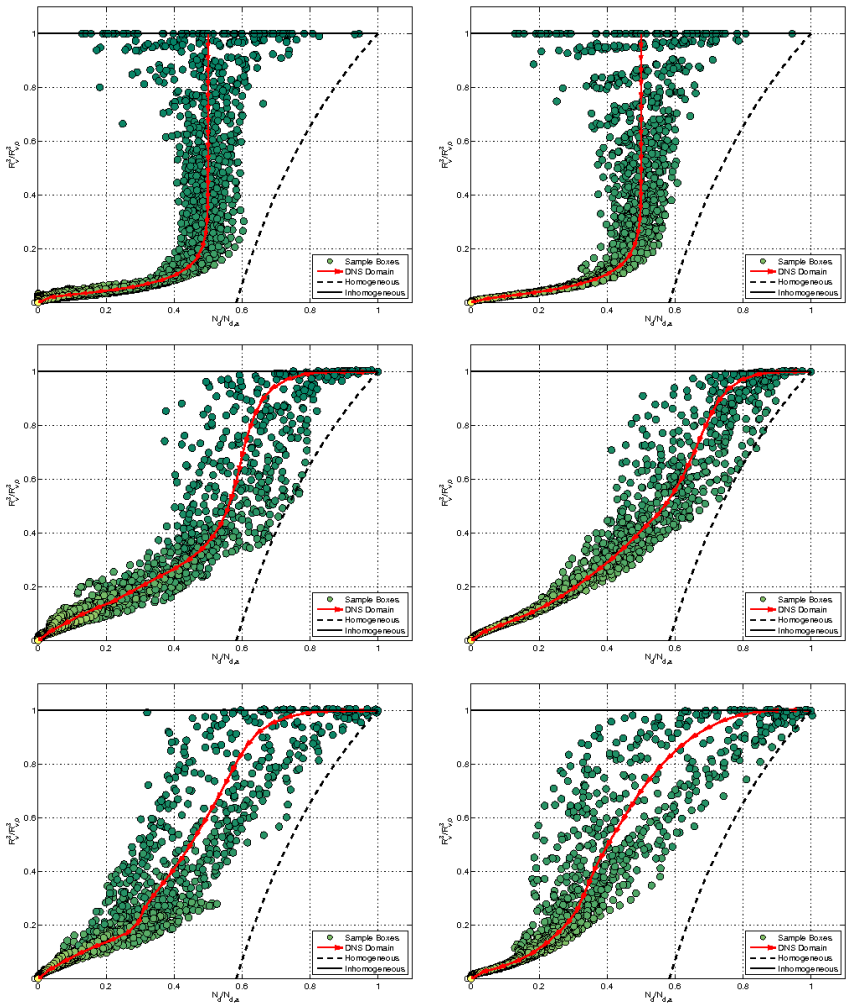


Figure 5.19: Mixing diagram for case D1, D2, D3, F1, F2 and F3. Mean cubic radius and mixing fraction have been calculated in 64 equal-sized samples boxes. The circle represent the time trajectories of $R_v^3/R_{v,0}^3$ and $N_d/N_{d,a}$ in different sample boxes and the triangles represent the same in the entire domain. Color indicates the time for each record. Only the boxes with non-zero particles at the initial time are considered.

diagram, and is calculated using central differencing in time:

$$\psi_1 = \frac{R_{j+1}^3/R_a^3 - R_{j-1}^3/R_a^3}{N_{j+1}/N_a - N_{j-1}/N_a} \tag{5.8}$$

where the subscript "$a$" denotes the adiabatic value of the droplet population in the initial cloudy region. Note that $\psi_1$ is in fact the inverse of the slope defined by [5] such that a larger value of $\psi_1$ indicates a higher degree of homogeneous mixing, in line better with intuition and the other microphysical measures discussed below.

More measures of homogeneous mixing degree have been introduced in [101, 102]. These measures are based on the mixing of adiabatic cloudy air and clear air, and slightly modified here to consider the instantaneous homogeneous mixing degree between two adjacent temporal states in time $t_j$ and $t_{j-1}$, such that

$$\psi_2 = \frac{\tan^{-1}(\frac{R_j^3/R_{j-1}^3 - 1}{N_j/N_{j-1} - N_H/N_{j-1}})}{\pi/2} \tag{5.9}$$

$$\psi_3 = 0.5(\frac{N_j - N_I}{N_H - N_I} + \frac{R_j^3 - R_{j-1}^3}{R_H^3 - R_{j-1}^3}) \tag{5.10}$$

$$\psi_4 = \frac{\ln R_j^3 - \ln R_{j-1}^3}{\ln R_H^3 - \ln R_{j-1}^3} \tag{5.11}$$

$$\psi_5 = \frac{1 - R_j^3/R_{j-1}^3}{1 - LWC_j N_{j-1}/(N_H LWC_{j-1})} \tag{5.12}$$

where all the variables are calculated from a sample box; $R$ is the mean volume radius; $N$ is the number concentration; $LWC$ is the liquid water content; the subscript "$j$" means the value is calculated from the $j$-th dataset at time $t_j$. The subscripts $I$ and $H$ indicate that the values are calculated based on the assumption of inhomogeneous and homogeneous mixing, respectively. Briefly,

$$N_H = \chi N_j + (1-\chi) N e_j$$

$$R_H^3 = \frac{N_j R_j^3}{N_H},$$

$$N_I = \frac{R_j^3}{R_{j-1}^3} N_j.$$

The mixing fraction $\chi$ is computed according to the mass conservation of total water between state $j$ and $j-1$:

$$\chi(q_{vc}^{j-1} + q_{lc}^{j-1}) + (1-\chi)(q_{ve}^{j-1} + q_{le}^{j-1}) = q_{lc}^{j} + q_{vc}^{j} \tag{5.13}$$

where the subscripts $c$ and $e$ stand for the mean value of a sample box and its environmental air; $l$ and $v$ stand for the liquid water and water vapor. The environmental air is defined as the air in 4 grids extended from the original sample box; the subscript $j$ indicates the state of the $j$-th dataset collected at time $t_j$. Note that Eq. (5.13) considers the fact that the cloudy air may have been diluted and the environmental air may contain droplets.

It can be readily shown that $\psi_1$ equals to 0 for the extremely inhomoge-

neous mixing, but approach $\infty$ as the mixing process approaches homogeneous mixing. On the other hand, the other four definitions of homogeneous mixing degree all range between 0 and 1, with 0 for extremely inhomogeneous mixing and 1 for homogeneous mixing. Note that the theoretic range of $\psi_1$ is $[0, \infty]$ while $\psi_2$, $\psi_3$, $\psi_4$ and $\psi_5$ are $[0, 1]$, and therefore it is difficult to compare $\psi_1$ with other measures. Fig. 5.20 compares the four measures of homogeneous mixing degree whereby each dot represents an instantaneous domain-mean and the different color denotes the six different scenarios. As expected, all the measures of homogeneous mixing degree are positively related to one another, and can be used as a microphysical measure to quantify the entrainment-mixing process.

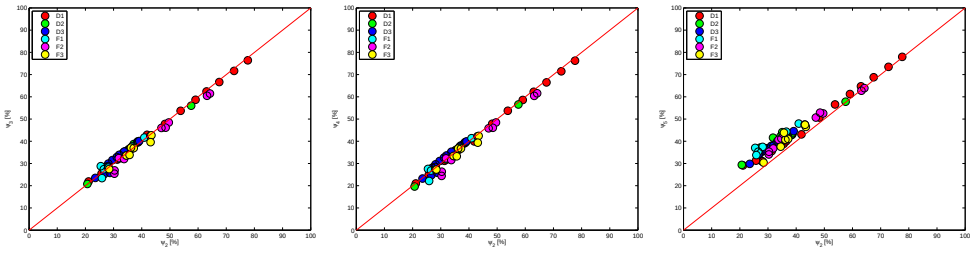


Figure 5.20: Comparison between different mixing degree, from left to right are $\psi_2$ with $\psi_3$, $\psi_4$ and $\psi_5$.

Following the previous work contributed by [82, 55, 21], the entrainment-mixing process can be characterized by the Damköhler number, the ratio of turbulent mixing timescale to a microphysical timescale:

$$Da = \frac{\tau_{mix}}{\tau_{react}} \tag{5.14}$$

where the turbulence mixing time scale can be estimated as $\tau_{mix} = (\lambda^2/\epsilon)^{1/3}$; the length scale $\lambda$ is represented by the mean Taylor microscale for the cloud water, $\lambda = (\lambda_1 + \lambda_2 + \lambda_3)/3$, $\lambda_i = \langle q_c^2 \rangle^{1/2} / \langle (\partial q_c / \partial x_i)^2 \rangle^{1/2}$, and the dissipation rate is estimated with $\epsilon = 2\nu \langle (\nabla \times \mathbf{u})^2 \rangle$. The definition of reaction time scale $\tau_{react}$ will be introduced later. In general, $Da \ll 1$ corresponds to the homogeneous mixing while $Da \gg 1$ is the inhomogeneous one. Ambient clouds often have $Da$ between these two limits.

Recognizing that the turbulent mixing time scale depends on the entrained eddy sizes, [93] introduced the transition length scale defined as the length scale at which $Da = 1$. A larger transition scale length suggests a higher degree of homogeneous mixing. It can be shown

$$l^* = \epsilon^{1/2} \tau_{react}^{3/2}$$

[101] further introduced the transition scale number defined as the ratio of transition length to Kolmogorov length scale as a dynamic measure of homogeneous mixing degree:

$$N_L = \frac{l^*}{\eta} \tag{5.15}$$

where the Kolmogorov length scale is given by

$$\eta = (\frac{\nu^3}{\epsilon})^{1/3},$$

In [93, 103], $\tau_{react}$ is defined as the time when droplets have completely evaporated or relatively humidity has reached 99.5% whichever is first satisfied.

It is calculated by solving the following ordinary differential equation for the mean volume radius and mean supersaturation:

$$\frac{dR_v}{dt} = K\frac{S}{R_v} \tag{5.16}$$

$$\frac{dS}{dt} = -BR_vS \tag{5.17}$$

where $B$ is a function of pressure and temperature:

$$B = \frac{4\pi N\rho_L[\frac{G_dT}{\varepsilon e_s(T)} + \frac{\varepsilon L_h^2}{pTc_p}]}{(\frac{L_h}{G_vT} - 1)\frac{L_h\rho_L}{\mu_TT} + \frac{\rho_LG_vT}{\mu_ve_s(T)}} \tag{5.18}$$

where $L_h$ is latent heat, $G_v$ is individual gas constant for water vapor, $T$ is air temperature, $\rho_L$ is density of liquid water, $\mu_T$ is coefficient of thermal conductivity of air, $\mu_v$ is coefficient of diffusion of water vapor in air, $e_s(T)$ is saturation vapor pressure over a plane water surface at temperature $T$, $N$ is number concentration of droplets, $G_d$ is individual gas constant for dry air, $\varepsilon = G_d/G_v$, $p$ is air pressure, and $c_p$ is specific heat with pressure held constant ([101]). This definition considers the interactions between liquid water and vapor water, and hence its value is expected to be smaller than the previous definition.

Another microphysical time scale is the so-called evaporation time defined as the time that a droplet needs to complete the evaporation [6, 21]:

$$\tau_{evap} = R_v(\frac{dR_v}{dt})^{-1} = \frac{R_v^2}{-KS_e} \tag{5.19}$$

where $R_v$ is the mean volume radius of a group of droplets; $K$ is the constant in the droplet diffusional growth equation, and $S_e$ is the supersaturation of the dry air. Evidently, the evaporation time scale assumes that the environmental dry air is always unsaturated with a constant negative $S_e$.

The impact of entrainment and cloudy-clear air mixing on the spectra of cloud droplets remains an important yet still unresolved issue in cloud physics. Because warm (ice free) clouds are close to water saturation, conservation of the total water and moist static energy is sufficient to determine the temperature, water vapor, and cloud water mixing ratios of the homogenized mixture of cloudy and cloud-free unsaturated air. Predicting the evolution of a cloud droplet spectrum, on the other hand, requires additional constraints because, as far as bulk conservation principles are concerned, cloud water after homogenization can be distributed over either a large number of small droplets or a small number of large droplets. The concentration and size of cloud droplets critically depend on whether the mixing is homogeneous (i.e., all droplets are exposed to the same subsaturation during mixing) or inhomogeneous (i.e., the degree of droplet evaporation varies; [8, 7, 21]. In the homogeneous mixing scenario, the number of droplets does not change and the mean droplet size decreases. In the extreme inhomogeneous mixing scenario, droplets from a fraction of the cloudy volume evaporate completely to bring the mixture to saturation, and the droplets from the rest of the cloudy volume are dispersed over the combined volumes without changing their size. It follows that the extremely inhomogeneous mixing is associated with the change of droplet concentration, but not the droplet size. The homogeneous mixing and the ex-

tremely inhomogeneous mixing set the limits for all possible mixing scenarios. Whether cloud dilution is associated with homogeneous or inhomogeneousmixing has been shown to significantly affect radiative properties of stratocumulus [32] and shallow convective clouds [58, 129].

[21] showed that the Damköhler number, defined as the ratio between the mixing time scale ($\tau_m$) of dry air and cloud air and the evaporation time scale ($\tau_e$) could be used as a parameter that indicates which mixing mechanism is dominant. [93] argued that the mixing mechanism could be better determined with the transition length scale instead of the Damköhler number because of the uncertainty in knowing the turbulent mixing length scale. The results varied by cloud region; homogeneous mixing (HM) appeared more frequently in the vicinity of the cloud core, while inhomogeneous mixing (IM) appeared more frequently in more diluted cloud regions [93]. Similarly, [101] proposed that the transition scale number defined as the ratio of transition length to the Kolmogorov length scale could be used as a parameter to estimate mixing mechanisms; a higher transition scale number corresponds to a greater tendency of homogeneous mixing. The transition scale number Eq. (5.15), Damköhler number Eq. (5.14) and the various microphysical measures of homogeneous mixing degree Eq. (5.8), Eq. (5.9), Eq. (5.10), Eq. (5.11) and Eq. (5.12) are expected to be correlated since they are two measures quantifying the probability of the homogeneous mixing process from different perspective. These relationships are examined using the numerical data produced from the six simulations. First, the slope Eq. (5.8) is plotted against the Damköhler number and transition scale number in Fig. 5.21. The left figure shows that the

Damköhler number has a positive relationship with the reciprocal of the slope. This duplicates the results in [5] and is consistent with the heuristic argument relating homogeneity of mixing to the time scale ratio. We also compare the results using transition scale number and Damköhler number as the dynamical measures in the right figure. It yields that the transition scale number has a wider range of values but gives consistent conclusion with Damköler number.

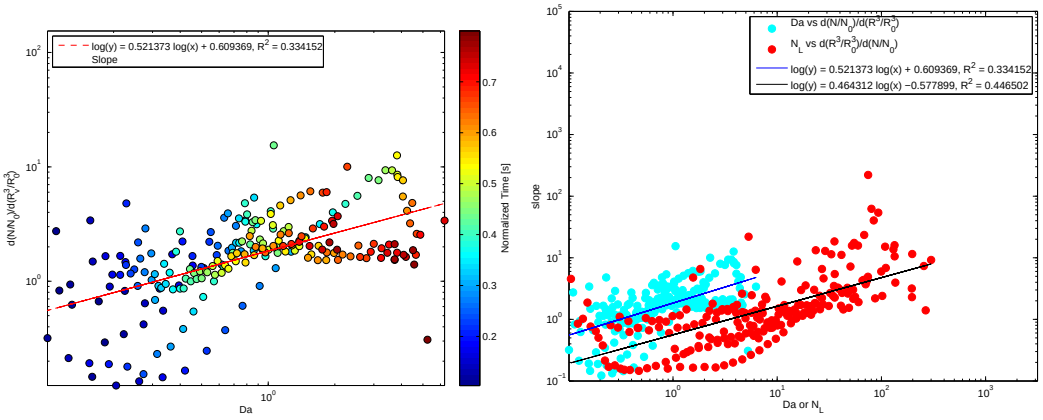


Figure 5.21: The left figure displays the scatter plot of the slope in the $R-N$ diagram as a function of Damköhler number and transition scale number. The comparison between using Damköhler number and transition scale number is shown in the right figure.

The scatterplot of the homogeneous mixing degree as a function of the transition scale number is shown in Fig. 5.22 with color indicating the normalized simulation time. The fitting curves have a close slope for different cases, and a tight relationship can be observed in the critical range of the slopes, and therefore one can suggest a simple parameterization.

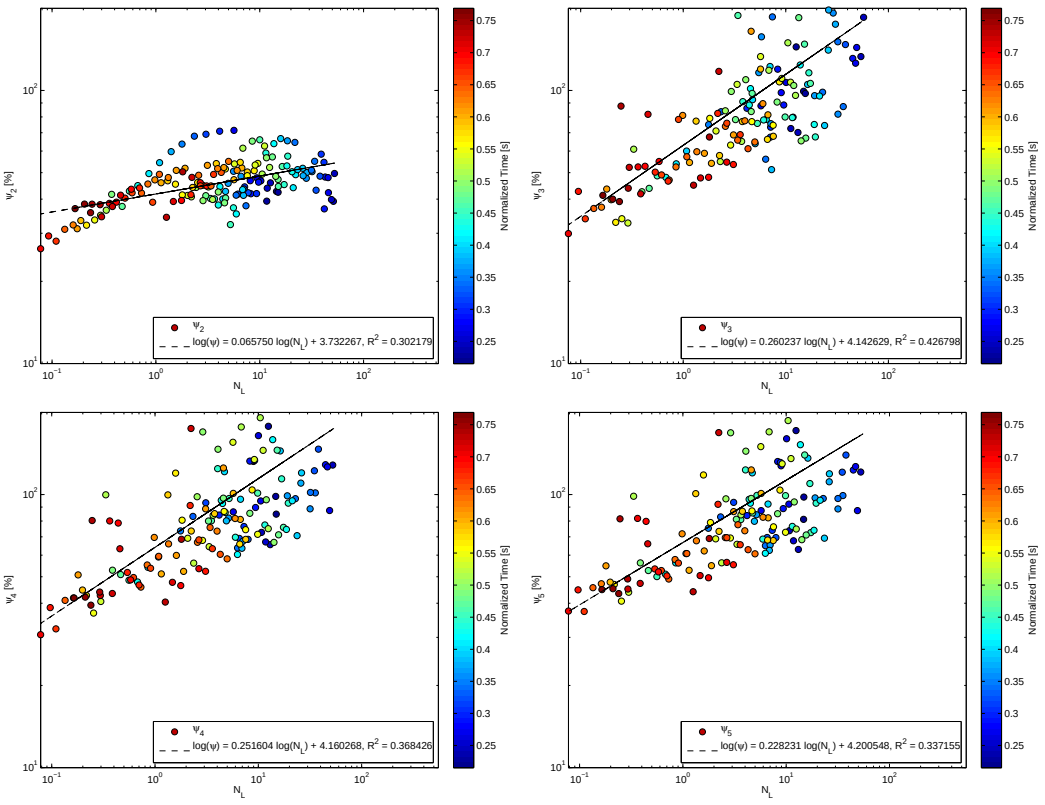


Figure 5.22: Scatter plot of the homogeneous mixing degree vs the transition scale number. All the cases are shown in one figure. From left to right, up to bottom are $\psi_1$, $\psi_2$, $\psi_3$ and $\psi_4$.

### 5.2.4 Effects of sedimentation on preferential concentration

Clustering of inertial particles has been extensively studied via both experiments and numerical simulations [134, 116], but the sedimentation effects on the clustering are poorly understood. In this section, a series of numerical test are performed by gradually increasing the gravity force. Since we are only interested on the functional relationship between clustering index and gravity, the particles are not allowed to evaporate or condensate during the simulation, thus keeping their sizes unchanged. The clustering index calculated with [148]

$$C_L = \hat{V}_L(n)/V_L(n) - 1 \tag{5.20}$$

where $\hat{V}_L(n)$ is the measured variance of the droplets number concentration and $V_L(n)$ is the Poisson variance equal to the mean droplet number concentration. The droplets are uniformly placed in the domain at initial time, thus their number concentration will follow the Poisson point distribution.

From the history of the clustering index in Fig. 5.23, one can tell that the clustering indexes increase at the beginning stage due to the strong turbulence fluctuation and then decrease as the turbulence decaying. The forced turbulence differs from the decaying case by remaining on average of clustering index in the latter stage of the simulation. The result suggests that gravity weakens the preferential concentration. This phenomenon can also be visualized in Fig. 5.24 by plotting the clustering index as a function of the gravity at the final step.

Preferential concentration can also be measured with Pearson correlation coefficient between droplet number concentration and vorticity magnitude. Fig. 5.25 shows the correlation coefficients of four cases, decaying or force turbulence with or without considering sedimentation as a function of time. All the cases show negative correlations, which agrees with that particles tend to accumulate within the low vorticity area of turbulence field. Two nondimensional numbers are useful for understanding the mechanism of preferential concentration. One is the Stokes number, which measures the relative time scale of particle and turbulence flow:

$$S_t = \tau_p/\tau_\eta \tag{5.21}$$

where $\tau_p = 2\rho_w R^2/9\mu$ is the particle response time and $\tau_\eta$ is the Kolmogorov time scale.

Previous studies [56, 149] have shown that maximum preferential concentration occurs at $S_t \sim 1$. For sedimenting particles, another useful nondimensional number is based on the terminal velocity of the particles:

$$S_v = v_T/v_\eta \tag{5.22}$$

where $v_T = \tau_p g$ is the terminal velocity and $v_\eta$ is the Kolmogorov velocity scale. The droplets have no time to interact with the eddies when $S_v \gg 1$ and sedimentation can be neglected when $S_v \ll 1$, thus $S_v \sim 1$ represents the case that sedimentation effects should not be ignored.

In our simulation, the initial condition gives $\tau_p = 0.0028s$, $\tau_\eta = 0.087s$ and

$S_t = 0.032$; $v_T = 0.027m/s$, $v_\eta = 0.011m/s$ and $S_v = 2.45$. The fact of $S_t \ll 1$ tells that the particle motion will almost follow the turbulence flow, thus the negative correlation between number concentration and vorticity magnitude could be too weak to observe. To be specific, the decaying turbulence has a decreasing dissipation rate and increasing $\tau_\eta$, thus the correlation will be reduced further as the turbulence dissipating. The fact of $S_v \sim 1$ implies that the sedimentation will break the correlation in some instance as shown in Fig. 5.25. For the forced turbulence, the dissipation rate is maintained by the volume force, so that $S_t$ and $S_v$ will remain on the average. Therefore, stable correlation coefficients can be observed in Fig. 5.25. However, even if $S_v \sim 1$, the sedimentation has little influence on the correlation for the forced turbulence. This could be explained by the fact that the particle motion is almost determined by the forced turbulence flow and $S_v$ becomes insignificant in this case.

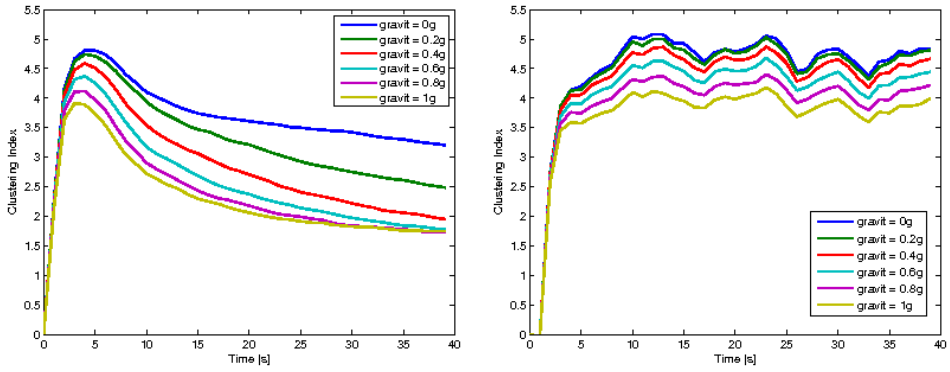


Figure 5.23: Time evolution of clustering index with different sedimentation term in decaying turbulence (see left) and forced turbulence (see right figure).

In this section, a series of numerical simulation is performed to study the

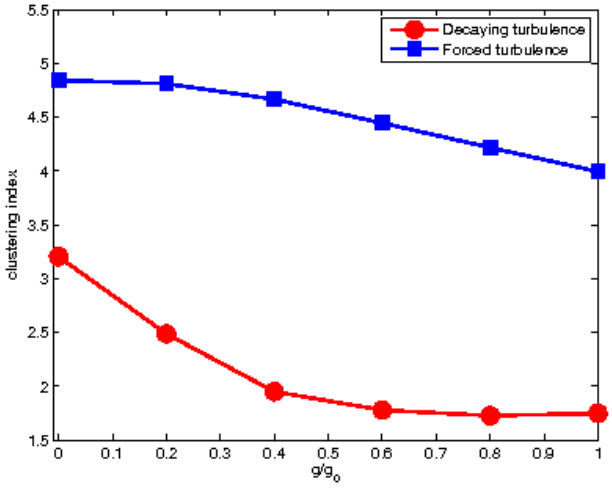


Figure 5.24: Clustering index as a function of gravitational accleration. The parameters are normalized by the original gravitational accerlation $g_0 = 9.8m/s^2$.

cloud entrainment and mixing phenomena. Three different configurations are compared with each other to inspect the influence of the initial cloudy shape on the cloud microphysics in the mixing process. The simulation duplicates the configurations in [3] and [87] and agrees with their main results. Case 1 corresponds to [3], which is aiming to study the final stage of the mixing process. Case 2 tries to mimic the idealized cloud slab in [87], presenting a complete view of entrainment and mixing. Case 3 is created by rotating case 2 with 90 degree clockwise to show the sedimentation effects on the droplet spectrum.

The work described in this paper almost follows the configurations in [87]. However, due to the Gibb's phenomena of the pseudo-spectral method, there is an inconsistency between the initial profile of cloud droplets and vapor mixing

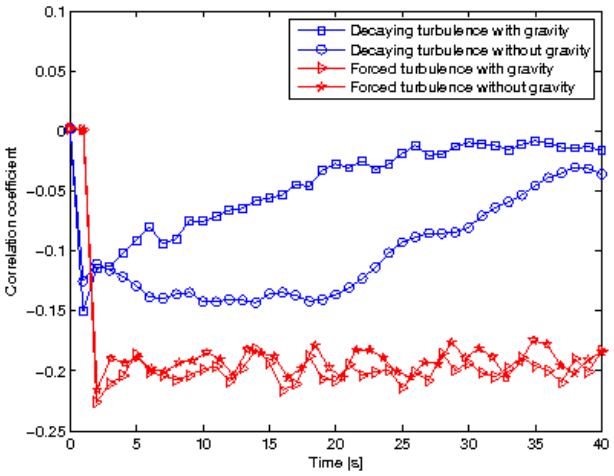


Figure 5.25: Correlation coefficient between vorticity magnitude and droplet number density as a function of time. The correlation coefficients are computed following the Pearson product-moment correlation coefficients, which measures the linear correlation between two variables with positive and negative correlations inclusive.

ratio in [87], in which an artificial continuous function was used to connect the area of cloudy air and clear air, while the cloud droplets were treated as a simple slab. This inconsistency is not desirable and can be overcome by taking advantage of the high resolution finite difference WENO scheme, which is designed for problems with piecewise smooth solutions containing discontinuities. Therefore, we are able to perform the simulation with the same sharp initial interface for both cloud droplets and the vapor mixing ratio.

All the simulation have been tested in both decaying turbulence and forced turbulence. The thermodynamics, cloud microphysics and mixing diagram are studied to make a comparison between different cases. The transient growth of turbulence kinetic energy due to buoyancy effects in the decaying cases agrees with the observation in [88]. The spectrum of droplets size and supersaturation implies that the cloudy shape effectively influence the mixing process in decaying turbulence by affecting the reaction time and the size distribution. However, this effect seems to be much smaller in the forced turbulence.

The mixing diagram is then plotted to compare the $R$-$N$ relationship in different cases, which have the same final state with zero liquid water content. The number density in case 1 does not change for a long time due to its already diluted configuration. This implies that the initial reductions of number density in case 2 and 3 are due to dilution process. Case 2 is performed to duplicate the results in [88] for radius $15\mu m$ (note that their results were amended in [86]). Our results disagree the conclusion in [88] but support their amendment in [86], that is the mixing trajectories are not scattered around

the homogeneous mixing curve. The configuration of case 3 is the same as case 2 except the direction of the cloud slab. The results infer that the sedimentation will accelerate the mixing to a certain degree when comparing with case 2. Two groups of mixing trajectories are observed in the results of case 3. We interpret the separation of the curves as an indication that the sedimentation will push the particles moving downwards, so as to accelerate the mixing process. The experiment designed to test the relationship between gravity and clustering index gives a better understanding of the sedimentation effect.

## 5.3 Scaling performance

We have carried out several experiments to investigate the impacts of the computing platform on our main application, which highly relies on a few APIs (such as MPI, CUDA) and external packages (such as PETSc, HYPRE). In order to distinguish their impacts on the application, we have created a few independent programs by including PETSc, MPI and CUDA separately. In the first test, a two dimensional Poisson equation was solved with PETSc as the KSP solver and HYPRE as the preconditioner. To test the strong scaling, we fix the domain size to be $4096 \times 4096$ while gradually double the number of processors from 1 to 1024. The speedup of all the cases are illustrated in Fig. 5.26, which yields that the Cray supercomputer has a wider range of linear scaling than "Intruder" cluster and workstation. It also suggests that the speedup will become slower when the machine is nearly fully occupied. We interpret this fact as a indicator of the limit bandwidth of the main memory which is highly slower than the speed of the modern CPU. The speedup of

DNS with grid size $128^3$ for various number of CPUs are also displayed in Fig. 5.26 for comparison. It yields that the optimal speedup is achieved at 128 number of processors.

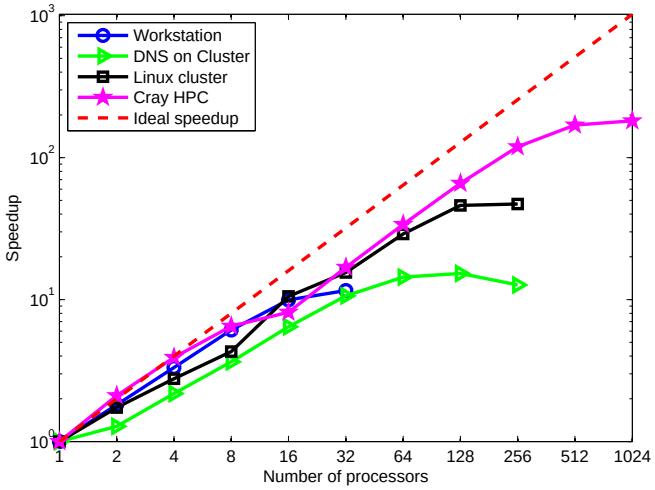


Figure 5.26: The figure displays the speedup of linear solver solving 2-D Poisson equation with domain size ($4096 \times 4096$) on different machines: workstation, linux cluster and Cray supercomputer. The speedups of DNS with different number of processors are also provided in the figure for comparison

Another experiment tests the GPU acceleration on the cloth simulator. The GPU code is implemented with CUDA library, which is a parallel computing platform created by NVIDIA. We compare the computational time of solving spring model for different parachute types using or without using GPU device and calculate the speedup. As shown in Table 5.4, using GPU device can achieve at least 16 times and up to 21 times speedup for cloth simulation.

| Parachute type | CPU/GPU | Time(s) | Avg time per step(s) | Speedup |
|---|---|---|---|---|
| C9 | CPU | 2805.85 | 3.39 | 1.00 |
| | GPU | 131.90 | 0.16 | 21.2 |
| G11 | CPU | 5101.47 | 5.41 | 1.00 |
| | GPU | 243.18 | 0.26 | 20.81 |
| Intruder | CPU | 1252.65 | 2.00 | 1.00 |
| | GPU | 69.67 | 0.11 | 18.18 |
| T10 | CPU | 5540.02 | 5.99 | 1.00 |
| | GPU | 282.74 | 0.36 | 16.64 |
| T11 | CPU | 6791.9 | 5.12 | 1.00 |
| | GPU | 352.07 | 0.29 | 17.66 |

Table 5.4: A comparison of computational time between different parachute type on CPU or GPU. The speedup is calculated based on the computing time by CPU.

# Chapter 6

# Conclusion

In this dissertation, we have proposed a computational framework that combines the Eulerian description of the turbulence field with a Lagrangian particles ensemble. This framework has been applied to the numerical study of parachute deceleration and cloud microphysics. Although these two fields have different backgrounds, they can be uniformly modeled with PDEs and ODEs, and numerically solved in the same framework.

The turbulence field is described by the incompressible Navier-Stokes equation, which uniquely determines the velocity and pressure field in the domain. However, if the computational grid is not fine enough to cover the entire length scale, directly solving the Navier-Stokes may fail to fully resolve the effects of turbulence kinetic-energy cascade and backscatter. In a statistically-averaged sense, the Naiver-Stokes equation can be modified using the Reynolds decomposition. The resulting velocity and pressure field are solved from the Reynolds-Averaged Navier-Stokes equation in the perspective of ensemble average, and the additional term of Reynolds stress can be approximated by the

turbulence model.

The Naiver-Stokes equation is essentially a special case of the nonlinear PDE, and its solving procedure contains the numerical schemes for hyperbolic, parabolic and elliptic equations. Therefore, these methods can be directly reused to solve the associated scalar field, which is described by a parabolic equation with a source or sink term. However, the appearance of the sink terms adds more challenges, since many physics problem requires the underlying field to be positive, such as kinetic energy, dissipation rate, temperature and concentration of chemical species. One way to overcome this difficulty is to linearize the sink term and set a lower bound for the coefficients without touching the final solution. This method keeps the positivity of the solution and not introducing any direct artifacts on it.

A particle system has been introduced in the framework due to its simple structure and wide range of application. There exist two kinds of particle system: independent or connected system. In the independent system, such as cloud droplets, the motion of one particle has no direct impacts on the rests, and hence the system can be decoupled to a set of independent equations. In the opposite case, such as the spring-mass model, the connected particle system consists of many coupled ODEs, which should be solved simultaneously. Moreover, the connected-particle system usually forms an interface, and hence an efficient collision treatment has been developed to avoid the surface self-intersections. In addition, the ODEs for the particle system can be stiff, thus a numerical method with large stability region is desirable. We have examined the implicit Euler method, explicit Runge-Kutta method and BDF method.

We choose the appropriate method by considering its efficiency, accuracy, and stability.

This framework has been used to study two applications: the parachute deceleration and cloud microphysics. The parachute simulation is carried in a turbulence environment with a relatively large Reynolds number, and thus the turbulence model is required. We have examined three different turbulence model: standard, RNG and Realizable $k$-$\epsilon$ model. The numerical experiment shows that the standard and RNG model tends to overestimate the turbulence effects when the flow is laminar while the Realizable model gives a reasonable prediction. In the meanwhile, we have proposed a new porosity model to simulate the porous effects of the parachute canopy. The new model is formulated by combining the Navier-Stokes equation with the Darcy's law through the Ghost Fluid scheme. This model is validated by the numerical experiments and concludes that the porosity effects is able to reduce the oscillation of the drag force. In addition, an efficient handling library has been developed to eliminate the intersections among the parachute canopy, suspension lines and cargoes. The collision treatment follows an iterative methodology, but is guaranteed to finish in finite number of steps thanks to the fail-safe method. The collision handling method is also applied to the folding of the parachute. We have achieved various folding patterns by combining some atomic operations.

The cloud entrainment and mixing process is studied in the same framework. Since the domain size is relatively small, we use direct numerical simulation to solve the turbulence field. The temperature field and vapor mixing ratio field are considered at the same time. The cloud droplets are described

by the particle model, and can grow or shrink according to the local humidity. We have proposed three ways of configuration to model the cloud entrainment and mixing process at different location of the cloud. All the cases have been performed in both the decaying turbulence and forced turbulence. The main purpose of this study is to quantify the mixing degree, which is usually between the two extreme cases: the homogeneous mixing and inhomogeneous mixing. The numerical result suggests a new way to parameterize the mixing degree using the dynamical measures. Finally, the preferential concentration is also studied in this dissertation. The clustering index is used to quantify the degree of clustering, and we have examined the effects of the gravity on the clustering. The results imply that the gravity reduce the clustering effects in both the decaying and forced turbulence. Furthermore, a negative relationship between the number concentration and vorticity field has been verified through the experiment. It also shows that the gravity has less impacts on the forced turbulence than the decaying turbulence.

The two applications demonstrate that this computational framework is flexible and has a broad application. The PDE can be used to describe a wide variety of phenomena, such as sound, heat and fluid dynamics. These distinct physical problems can be formalized similarly in terms of PDE. In the meanwhile, the particle model can be used to simulate various objects, from independent mass points to elastic membranes and rigid bodies, and hence adds more flexibility to this physically based modeling. However, the current framework only considers a diluted system (cloud droplets) or a solid system (fabric). A combination of them can give us a more sophisticated dynamic

system, which enables us to simulate some complicated physical processes. For example, the current study of the cloud microphysics in this paper ignores the collision between two droplets, which is a reasonable assumption in the early stage of entrainment. As the mixing proceeds, however, some droplets may grow to large cloud drops due to condensation, and therefore their effects on other droplets and fluid field cannot be neglected. In this situation, we may set a threshold such that the droplets larger than the threshold are modeled as moving interfaces while the small droplets can still be treated as mass points. In consequence, the cloud drops and cloud droplets coexist in the same environment, and their collision and coalescence can be treated similarly as the method for fabric collision detection whereas adding additional bounding boxes for the mass points is needed. This problem can be a possible future application of this computational framework.

# Bibliography


[1] John David Anderson Jr. *Fundamentals of Aerodynamics*, chapter 1, pages 75–81. McGraw-Hill Education, New York, 5th edition, 1985.

[2] M. Andrejczuk, W.-W. Grabowski, S.-P. Malinowski, and P.-K. Smolarkiewicz. Numerical simulation of cloud-clear air interfacial mixing. *Journal of the Atmospheric Sciences*, 61:1726–1739, 2004.

[3] M. Andrejczuk, W.-W. Grabowski, S.-P. Malinowski, and P.-K. Smolarkiewicz. Numerical simulation of cloud-clear air interfacial mixing. *Journal of the Atmospheric Sciences*, 61:1726–1739, 2004.

[4] M. Andrejczuk, W.-W. Grabowski, S.-P. Malinowski, and P.-K. Smolarkiewicz. Numerical simulation of cloud-clear air interfacial mixing: Effects on cloud microphysics. *Journal of the Atmospheric Sciences*, 63:3204–3225, 2006.

[5] M. Andrejczuk, W.-W. Grabowski, S.-P. Malinowski, and P.-K. Smolarkiewicz. Numerical simulation of cloud-clear air interfaical mixing: Homogeneous versus inhomogeneous mixing. *Journal of the Atmospheric Sciences*, 66:2493–2500, 2009.

[6] Miroslaw Andrejczuk, Wojciech W. Grabowski, Szymon P. Malinowski, and Piotr K. Smolarkiewicz. Numerical simulation of cloud–clear air interfacial mixing: Homogeneous versus inhomogeneous mixing. *Journal of the Atmospheric Sciences*, 66(8):2493–2500, aug 2009.

[7] MB Baker, RG Corbin, and J Latham. The influence of entrainment on the evolution of cloud droplet spectra: I. a model of inhomogeneous mixing. *Quarterly Journal of the Royal Meteorological Society*, 106(449):581–598, 1980.

[8] MB Baker and J Latham. The evolution of droplet spectra and the rate of production of embryonic raindrops in small cumulus clouds. *Journal of the Atmospheric Sciences*, 36(8):1612–1615, 1979.

[9] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Groppand Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.

[10] Barrett Baldwin and Harvard Lomax. Thin-layer approximation and algebraic model for separated turbulentflows. In *16th Aerospace Sciences Meeting*, page 257, 1978.

[11] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery (ACM), 1998.

[12] Sarah-Marie Belcastro and Thomas C Hull. A mathematical model for non-flat origami.

[13] Jiri Blazek. *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.

[14] Reinhard Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989.

[15] W. Bo, B. Fix, J. Glimm, X. Li, X. Liu, R. Samulyak, and L. Wu. Frontier and applications to scientific and engineering problems. *Proceedings in Applied Mathematics and Mechanics*, 2007.

[16] Jean-Louis Brenguier. Observations of cloud microstructure at the centimeter scale. *Journal of Applied Meteorology*, 32(4):783–793, 1993.

[17] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics*, 21(3), jul 2002.

[18] David L. Brown, Ricardo Cortez, and Michael L. Minion. Accurate projection methods for the incompressible navier–stokes equations. *Journal of Computational Physics*, 168(2):464–499, apr 2001.

[19] David L. Brown, Ricardo Cortez, and Michael L. Minion. Accurate projection methods for the incompressible navier–stokes equations. *Journal of Computational Physics*, 168(2):464–499, apr 2001.

[20] L. Brugnano and D. Trigiante. Boundary value methods: The third way between linear multistep and runge-kutta methods. *Computers & Mathematics with Applications*, 36(10-12):269–284, nov 1998.

[21] Frédéric Burnet and Jean-Louis Brenguier. Observational study of the entrainment-mixing process in warm convective clouds. *Journal of the Atmospheric Sciences*, 64(6):1995–2011, jun 2007.

[22] NG Burnet, R Wurm, JR Yarnold, JH Peacock, J Nyman, and I Turesson. Prediction of normal-tissue tolerance to radiotherapy from in-vitro cellular radiation sensitivity. *The Lancet*, 339(8809):1570–1571, 1992.

[23] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley-Blackwell, jun 2003.

[24] Alfonso Caiazzo, Miguel A Fernández, Jean-Frédéric Gerbeau, and Vincent Martin. Projection schemes for fluid flows through a porous interface. *Society for Industrial & Applied Mathematics Journal on Scientific Computing*, 33(2):541–564, Jan 2011.

[25] Daniele Carati, Sandip Ghosal, and Parviz Moin. On the representation of backscatter in dynamic localization models. *Physics of Fluids*, 7(3):606–616, mar 1995.

[26] Michel Carignan, Ying Yang, Nadia Magnenat Thalmann, and Daniel Thalmann. Dressing animated synthetic actors with complex deformable clothes. *ACM SIGGRAPH Computer Graphics*, 26(2):99–104, jul 1992.

[27] A. Celani, G. Falkovich, A. Mazzino, and Seminara. Droplet condensation in turbulence flows. *Europhysics Letters*, 70(6):775–781, 2005.

[28] Jeffrey R. Chasnov. Simulation of the kolmogorov inertial subrange using an improved subgrid model. *Physics of Fluids A: Fluid Dynamics*, 3(1):188–200, jan 1991.

[29] Kwang-Jin Choi and Hyeong-Seok Ko. Research problems in clothing simulation. *Computer-aided design*, 37(6):585–592, 2005.

[30] Kwang-Jin Choi and Hyeong-Seok Ko. Stable but responsive cloth. In *ACM SIGGRAPH 2005 Courses*. Association for Computing Machinery (ACM), 2005.

[31] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104):745, oct 1968.

[32] Frédérick Chosson, Jean-Louis Brenguier, and Lothar Schüller. Entrainment-mixing and radiative transfer simulation in boundary layer clouds. *Journal of the atmospheric sciences*, 64(7):2670–2682, 2007.

[33] R Cortez and DA Varela. The dynamics of an elastic membrane using the impulse method. *Journal of Computational Physics*, 138(1):224–247, 1997.

[34] John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press, 1947.

[35] Germund Dahlquist. Convergence and stability in the numerical integration of ordinary differential equations. *MATHEMATICA SCANDINAVICA*, 4:33, dec 1956.

[36] Germund G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3(1):27–43, mar 1963.

[37] Alberto de Lozar and Juan Pedro Mellado. Direct numerical simulations of a smoke cloud–top mixing layer as a model for stratocumuli. *Journal of the Atmospheric Sciences*, 70(8):2356–2375, 2013.

[38] H. Delingette. Triangular springs for modeling nonlinear membranes. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):329–341, mar 2008.

[39] Erik D Demaine, Martin L Demaine, and Joseph SB Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. In *Proceedings of the fifteenth annual symposium on Computational geometry*, pages 105–114. ACM, 1999.

[40] Departments of the army and air force. *Unit and intermediate direct support maintenance manual for parachute, personnnel type: 35-foot diameter, T-10B troop back parachute assembly*, Sep 1988.

[41] BJ Devenish, P Bartello, J-L Brenguier, LR Collins, WW Grabowski, RHA IJzermans, SP Malinowski, MW Reeks, JC Vassilicos, L-P Wang, et al. Droplet growth in warm turbulent clouds. *Quarterly Journal of the Royal Meteorological Society*, 138(667):1401–1429, 2012.

[42] Jian Du, Brian Fix, James Glimm, Xicheng Jia, Xiaolin Li, Yunhua Li, and Lingling Wu. A simple package for front tracking. *Journal of Computational Physics*, 213(2):613–628, Apr 2006.

[43] J. K. Eaton and J. R. Fessler. Preferential concentration of particles by turbulence. *International Journal of Multiphase Flow*, 20(Supplement 1):169–209, 1994.

[44] J.K. Eaton and J.R. Fessler. Preferential concentration of particles by turbulence. *International Journal of Multiphase Flow*, 20:169–209, aug 1994.

[45] V. Eswaran and S.B. Pope. An examination of forcing in direct numerical simulations of turbulence. *Computers & Fluids*, 16(3):257–278, jan 1988.

[46] EG Ewing, HW Bixby, and TW Knacke. Recovery systems design guide. Technical report, Defense Technical Information Center Document, Dec 1978.

[47] Robert D. Falgout and Ulrike Meier Yang. hypre: A library of high performance preconditioners. In *Lecture Notes in Computer Science*, pages 632–641. Springer Science and Business Media, 2002.

[48] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2):457–492, Jul 1999.

[49] Kang Feng and Zhong-Ci Shi. *Mathematical theory of elastic structures*. Springer Science & Business Media, 2013.

[50] Allen Van Gelder. Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2):21–41, jan 1998.

[51] Wanner. Gerhard and Hairer. Ernst. *Solving ordinary differential equations II: stiff and differential-algebraic problems*. Springer Verlag, 1991.

[52] Sandip Ghosal, Thomas S. Lund, Parviz Moin, and Knut Akselvoll. A dynamic localization model for large-eddy simulation of turbulent flows. *Journal of Fluid Mechanics*, 286(-1):229, mar 1995.

[53] James Glimm, John W Grove, Xiao Lin Li, Keh-ming Shyue, Yanni Zeng, and Qiang Zhang. Three-dimensional front tracking. *SIAM Journal on Scientific Computing*, 19(3):703–727, 1998.

[54] James Glimm, John W Grove, Xiao Lin Li, Kehming Shyue, Yanni Zeng, and Qiang Zhang. Three-dimensional front tracking. *SIAM Journal on Scientific Computing*, 19(3):703–727, May 1998.

[55] Wojciech W. Grabowski. Cumulus entrainment, fine-scale mixing, and buoyancy reversal. *Quarterly Journal of the Royal Meteorological Society*, 119(513):935–956, jul 1993.

[56] Wojciech W Grabowski and Paul Vaillancourt. Comments on preferential concentration of cloud droplets by turbulence: effects on the early evolution of cumulus cloud droplet spectra. *Journal of the atmospheric sciences*, 56(10):1433–1436, 1999.

[57] Wojciech W Grabowski and Lian-Ping Wang. Growth of cloud droplets in a turbulent environment. *Annual Review of Fluid Mechanics*, 45:293–324, 2013.

[58] WW Grabowski, P Bechtold, A Cheng, R Forbes, C Halliwell, M Khairoutdinov, S Lang, T Nasuno, J Petch, W-K Tao, et al. Daytime convective development over land: A model intercomparison based on lba observations. *Quarterly Journal of the Royal Meteorological Society*, 132(615):317–344, 2006.

[59] J.L. Guermond, P. Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6011–6045, sep 2006.

[60] Ami Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 49(3):357–393, 1983.

[61] Shinya Hayashi. Jfold-introducing a new simulation-based airbag folding system for ls-dyna. 2013.

[62] Susan Hert and Michael Seel. dD convex hulls and delaunay triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.9 edition, 2016.

[63] E Hicks, C Pontikis, and A Rigaud. Entrainment and mixing processes as related to droplet growth in warm midlatitude and tropical clouds. *Journal of the atmospheric sciences*, 47(13):1589–1618, 1990.

[64] Wallace E Howell. The growth of cloud drops in uniformly cooled air. *Journal of Meteorology*, 6(2):134–149, 1949.

[65] Chou Hsiu-chi. A study of the microphysical mechanism of warm-cloud precipitation. Technical report, DTIC Document, 1964.

[66] James G Hudson and Seong Soo Yum. Droplet spectral broadening in marine stratus. *Journal of the atmospheric sciences*, 54(22):2642–2654, 1997.

[67] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, jun 1996.

[68] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996.

[69] Guang-Shan Jiang and Cheng-chin Wu. A high-order weno finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 150(2):561–594, 1999.

[70] Hamid Johari and Kenneth J. Desabrais. Vortex shedding in the near wake of a parachute canopy. *Journal of Fluid Mechanics*, 536:185–207, jul 2005.

[71] W.P Jones and B.E Launder. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, 15(2):301–314, feb 1972.

[72] W. Kahan. Pracniques: Further remarks on reducing truncation errors. *Commun. ACM*, 8(1):40–48, January 1965.

[73] Vinay Kalro and Tayfun E. Tezduyar. A parallel 3D computational method for fluid-structure interactions in parachute systems. *Computer Methods in Applied Mechanics and Engineering*, 190:321–332, 2000.

[74] Myungjoo Kang, Ronald P Fedkiw, and Xu-Dong Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

[75] Vitaly I Khvorostyanov and Judith A Curry. Toward the theory of stochastic condensation in clouds. part i: A general kinetic equation. *Journal of the atmospheric sciences*, 56(23):3985–3996, 1999.

[76] J Kim and P Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of Computational Physics*, 59(2):308–323, jun 1985.

[77] Joung-Dong Kim, Yan Li, and Xiaolin Li. Simulation of parachute FSI using the front tracking method. *Journal of Fluids and Structures*, 37:100–119, feb 2013.

[78] Joung-Dong Kim, Yan Li, and Xiaolin Li. Simulation of parachute fsi using the front tracking method. *Journal of Fluids and Structures*, 37:100–119, Feb 2013.

[79] Yongsam Kim and Charles S. Peskin. 2–d parachute simulation by the immersed boundary method. *SIAM Journal on Scientific Computing*, 28(6):2294–2312, jan 2006.

[80] David B Kirk and W Hwu Wen-mei. *Programming massively parallel processors: a hands-on approach*. Morgan Kaufmann, 2010.

[81] Theo W Knacke. Parachute recovery systems design manual. Technical report, DTIC Document, 1991.

[82] Steven K Krueger, Chwen-Wei Su, and Patrick A McMurtry. Modeling entrainment and finescale mixing in cumulus clouds. *Journal of the atmospheric sciences*, 54(23):2697–2712, 1997.

[83] B. Kumar, F. Janetzko, J. Schumacher, and R.-A.Shaw. Extreme response of a coupled scalar-particle system during turbulent mixing. *New Journal of Physics*, 14(115020), 2012.

[84] B. Kumar, J. Schumacher, and R.-A.Shaw. Cloud microphysical effects of turbulent mixing and entrainment. *Theoretical Computational Fluid Dynamics*, 27(361), 2012.

[85] B. Kumar, J. Schumacher, and R.-A.Shaw. Lagrangian mixing dynamics at the cloudy-clear air interface. *Journal of the Atmospheric Sciences*, 71(7):2564–2580, 2014.

[86] B. Kumar, J. Schumacher, and R.-A.Shaw. Corrigedum to lagrangian mixing dynamics at the cloudy-clear air interface. personal communication, 2016.

[87] Bipin Kumar, Jrg Schumacher, and Raymond A. Shaw. Cloud microphysical effects of turbulent mixing and entrainment. *Theoretical and Computational Fluid Dynamics*, 27(3-4):361–376, jun 2012.

[88] Bipin Kumar, Jrg Schumacher, and Raymond A. Shaw. Lagrangian mixing dynamics at the cloudy–clear air interface. *Journal of the Atmospheric Sciences*, 71(7):2564–2580, jul 2014.

[89] Olga A Ladyzhenskaya and Richard A Silverman. *The mathematical theory of viscous incompressible flow*, volume 12. Gordon & Breach New York, 1969.

[90] Alessandra S. Lanotte, Agnese Seminara, and Federico Toschi. Cloud droplet growth by condensation in homogeneous isotropic turbulence. *J. Atmos. Sci.*, 66(6):1685–1697, jun 2009.

[91] Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242, 1980.

[92] K. Lehmann, H. Siebert, and R.-A.Shaw. Homogeneous and inhomogeneous mixing in cumulus clouds: Dependence on local turbulence structure. *Journal of the Atmospheric Sciences*, 66:3641–3659, 2009.

[93] Katrin Lehmann, Holger Siebert, and Raymond A. Shaw. Homogeneous and inhomogeneous mixing in cumulus clouds: Dependence on local turbulence structure. *Journal of the Atmospheric Sciences*, 66(12):3641–3659, dec 2009.

[94] Xu-Dong Liu, Ronald P Fedkiw, and Myungjoo Kang. A boundary condition capturing method for Poisson's equation on irregular domains. *Journal of Computational Physics*, 160(1):151–178, 2000.

[95] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, nov 1994.

[96] Yangang Liu and Peter H Daum. Anthropogenic aerosols: Indirect warming effect from dispersion forcing. *Nature*, 419(6907):580, 2002.

[97] Yangang Liu, Peter H Daum, and Robert McGraw. An analytical expression for predicting the critical radius in the autoconversion parameterization. *Geophysical research letters*, 31(6), 2004.

[98] Yangang Liu and John Hallett. The 1/3power law between effective radius and liquid-water content. *Quarterly Journal of the Royal Meteorological Society*, 123(542):1789–1795, 1997.

[99] Yangang Liu and John Hallett. On size distributions of cloud droplets growing by condensation: A new conceptual model. *Journal of the atmospheric sciences*, 55(4):527–536, 1998.

[100] Yangang Liu, You Laiguang, Yang Weinong, and Liu Feng. On the size distribution of cloud droplets. *Atmospheric Research*, 35(2):201–216, 1995.

[101] Chunsong Lu, Yangang Liu, and Shengjie Niu. Examination of turbulent entrainment-mixing mechanisms using a combined approach. *J. Geophys. Res.*, 116(D20), oct 2011.

[102] Chunsong Lu, Yangang Liu, Shengjie Niu, and Satoshi Endo. Scale dependence of entrainment-mixing mechanisms in cumulus clouds. *Journal of Geophysical Research: Atmospheres*, 119(24), 2014.

[103] Chunsong Lu, Yangang Liu, Shengjie Niu, Steven Krueger, and Timothy Wagner. Exploring parameterization for turbulent entrainment-mixing processes in clouds. *Journal of Geophysical Research: Atmospheres*, 118(1):185–194, jan 2013.

[104] TS Lundgren. Linearly forces isotropic turbulence. Technical report, DTIC Document, 2003.

[105] Szymon P Malinowski, Miroslaw Andrejczuk, Wojciech W Grabowski, Piotr Korczyk, Tomasz A Kowalewski, and Piotr K Smolarkiewicz. Laboratory and modeling studies of cloud–clear air interfacial mixing: anisotropy of small-scale turbulence due to evaporative cooling. *New Journal of Physics*, 10(7):075020, jul 2008.

[106] Robert McGraw and Yangang Liu. Kinetic potential and barrier crossing: A model for warm cloud drizzle formation. *Physical review letters*, 90(1):018501, 2003.

[107] Robert McGraw and Yangang Liu. Brownian drift-diffusion model for evolution of droplet size distributions in turbulent clouds. *Geophysical research letters*, 33(3), 2006.

[108] L.P. Nedel and D. Thalmann. Real time muscle deformations using mass-spring systems. In *Proceedings. Computer Graphics International (Cat. No.98EX149)*. Institute of Electrical and Electronics Engineers (IEEE).

[109] Donald A Nield and Adrian Bejan. *Mechanics of Fluid Flow Through a Porous Medium*, chapter 1, pages 1–29. Springer Science Business Media, New York, Oct 2012.

[110] Steven A. Orszag. Comparison of pseudospectral and spectral approximation. *Studies in Applied Mathematics*, 51(3):253–259, sep 1972.

[111] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, Nov 1988.

[112] Charles S Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[113] John C Platt and Alan H Barr. Constraints methods for flexible models. *ACM SIGGRAPH Computer Graphics*, 22(4):279–288, 1988.

[114] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press (CUP), 2000.

[115] Xavier Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Eurographics*, pages 177–189. Springer Nature, 1997.

[116] Walter C. Reade and Lance R. Collins. Effect of preferential concentration on turbulent collision rates. *Physics of Fluids*, 12(10):2530, 2000.

[117] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Proceedings of the Royal Society of London*, 56(336-339):40–45, jan 1894.

[118] Osborne Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Proceedings of the Royal Society of London*, 56(336-339):40–45, 1894.

[119] Bryan W Roberts. Axisymmetric, self-excited oscillations in parachutes. *Journal of Aircraft*, 11(12):736–744, Dec 1974.

[120] R. S. Rogallo. *Numerical Experiments in Homogeneous Turbulence*, volume 81315. National Aeronautics and Space Administration, 1981.

[121] Carlos Rosales and Charles Meneveau. Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties. *Physics of Fluids*, 17(9):095106, sep 2005.

[122] François G Schmitt. About boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity. *Comptes Rendus Mécanique*, 335(9-10):617–627, 2007.

[123] Yu. S. Sedunov. *Physics of Drop Formation in the Atmosphere*. Wiley-Blackwell, 1974.

[124] A. Selle, J. Su, G. Irving, and R. Fedkiw. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):339–350, mar 2009.

[125] Raymond A Shaw, Walter C Reade, Lance R Collins, and Johannes Verlinde. Preferential concentration of cloud droplets by turbulence: Effects on the early evolution of cumulus cloud droplet spectra. *Journal of the atmospheric sciences*, 55(11):1965–1976, 1998.

[126] Qiangqiang Shi, Daniel Reasor, Zheng Gao, Xiaolin Li, and Richard D. Charles. On the verification and validation of a spring fabric for modeling parachute inflation. *Journal of Fluids and Structures*, 58:20–39, oct 2015.

[127] Tsan-Hsing Shih, William W Liou, Aamir Shabbir, Zhigang Yang, and Jiang Zhu. A new k- eddy viscosity model for high reynolds number turbulent flows. *Computers & Fluids*, 24(3):227–238, 1995.

[128] Richard Taylor Shinya Hayashi. Simulation-based airbag folding system jfold version 2: New capabilities and folding examples. 2014.

[129] Joanna Slawinska, Wojciech W Grabowski, Hanna Pawlowska, and Andrzej A Wyszogrodzki. Optical properties of shallow convective clouds diagnosed from a bulk-microphysics large-eddy simulation. *Journal of Climate*, 21(7):1639–1647, 2008.

[130] PRaA Spalart and S1 Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.

[131] Chwen-Wei Su, Steven K Krueger, Patrick A McMurtry, and Philip H Austin. Linear eddy modeling of droplet spectral evolution during entrainment and mixing in cumulus clouds. *Atmospheric research*, 47:41–58, 1998.

[132] Neal P. Sullivan, Shankar Mahalingam, and Robert M. Kerr. Deterministic forcing of homogeneous, isotropic turbulence. *Physics of Fluids*, 6(4):1612–1614, apr 1994.

[133] Peter P Sullivan, James C McWilliams, and Chin-Hoh Moeng. A subgrid-scale model for large-eddy simulation of planetary boundary-layer flows. *Boundary-Layer Meteorology*, 71(3):247–276, 1994.

[134] Shivshankar Sundaram and Lance R. Collins. Collision statistics in an isotropic particle-laden turbulent suspension. part 1. direct numerical simulations. *Journal of Fluid Mechanics*, 335:75–109, mar 1997.

[135] Tomohiro Tachi. Simulation of rigid origami. *Origami*, 4:175–187, 2009.

[136] Kenji Takizawa and Tayfun E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19(1):125–169, feb 2012.

[137] Anand S. Tanavde, Himanshu Khandelwal, David Lasry, Xiomin Ni, Eberhard Haug, Jutta Schlosser, and Pradeep Balakrishnan. Airbag modeling using initial metric methodology. In *SAE Technical Paper Series*. SAE International, feb 1995.

[138] Richard Taylor and Shinya Hayashi. Using jfold & ls-dyna to study the effects of folding on airbag deployment. 2015.

[139] James W Telford and Steven K Chai. A new aspect of condensation theory. *pure and applied geophysics*, 118(2):720–742, 1980.

[140] Roger Temam. Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5):377–385, 1969.

[141] Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery (ACM), 1988.

[142] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery (ACM), 1987.

[143] Matthias Teschner, Stefan Kimmerle, Bruno Heidelberger, Gabriel Zachmann, Laks Raghupathi, Arnulph Fuhrmann, M-P Cani, François Faure, Nadia Magnenat-Thalmann, Wolfgang Strasser, et al. Collision detection for deformable objects. In *Computer graphics forum*, volume 24, pages 61–81. Wiley Online Library, 2005.

[144] Tayfun E. Tezduyar, Sunil Sathe, Matthew Schwaab, Jason Pausewang, Jason Christopher, and Jason Crabtree. Fluid–structure interaction modeling of ringsail parachutes. *Computational Mechanics*, 43(1):133–142, mar 2008.

[145] Benjamin Tutt, C Richard, S Roland, and Greg Noetscher. Development of parachute simulation techniques in ls-dyna. In *11th International LS-DYNA Users Conference*, pages 25–36, Detroit, 2010.

[146] P. A. Vaillancourt and M. K. Yau. Review of particle-turbulence interactions and consequences for cloud physics. *American Meteorological Society*, 81(2):285–298, 2000.

[147] P. A. Vaillancourt, M. K. Yau, P. Bartello, and W. W. Grabowski. Microscopic approach to cloud droplets growth by condensation. part ii: Turbulence, clustering, and condensational growth. *American Meteorological Society*, 59:3421–3432, 2002.

[148] P. A. Vaillancourt, M. K. Yau, P. Bartello, and W. W. Grabowski. Microscopic approach to cloud droplets growth by condensation. part ii: Turbulence, clustering, and condensational growth. *American Meteorological Society*, 59:3421–3432, 2002.

[149] Paul A Vaillancourt and MK Yau. Review of particle-turbulence interactions and consequences for cloud physics. *Bulletin of the American Meteorological Society*, 81(2):285–298, 2000.

[150] Pascal Volino and Nadia Magnenat Thalmann. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. *Computer Graphics Forum*, 13(3):155–166, aug 1994.

[151] Pascal Volino and Nadia Magnenat Thalmann. Collision and self-collision detection: Efficient and robust solutions for highly deformable surfaces. In *Eurographics*, pages 55–65. Springer Nature, 1995.

[152] Jason Wang, Nicolas Aquelet, Benjamin Tutt, Ian Do, Hao Chen, and Mhamed Souli. Porous euler-lagrange coupling: Application to parachute dynamics. In *9th International LS-DYNA Users Conference*, pages 11–21, Manchester, Jul 2006.

[153] J Warner. The microstructure of cumulus cloud: Part iv. the effect on the droplet spectrum of mixing between cloud and environment. *Journal of the Atmospheric Sciences*, 30(2):256–261, 1973.

[154] David C Wilcox et al. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.

[155] Andrew Witkin. Particle system dynamics. Citeseer.

[156] Zhonghua Xi and Jyh-Ming Lien. Folding rigid origami with closure constraints. In *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V05BT08A052–V05BT08A052. American Society of Mechanical Engineers, 2014.

[157] Victor Yakhot and Steven A. Orszag. Renormalization group analysis of turbulence. i. basic theory. *Journal of Scientific Computing*, 1(1):3–51, 1986.

[158] Jun-Ichi Yano and Mitchell W Moncrieff. Numerical archetypal parameterization for mesoscale convective systems. *Journal of the Atmospheric Sciences*, 73(7):2585–2602, 2016.

[159] Man Kong Yau and RR Rogers. *A short course in cloud physics*. Elsevier, 1996.

[160] Seong Soo Yum and James G Hudson. Vertical distributions of cloud condensation nuclei spectra over the springtime arctic ocean. *Journal of Geophysical Research: Atmospheres*, 106(D14):15045–15052, 2001.

[161] Seong Soo Yum and James G Hudson. Adiabatic predictions and observations of cloud droplet spectral broadness. *Atmospheric research*, 73(3):203–223, 2005.

[162] Seong Soo Yum, Jian Wang, Yangang Liu, Gunnar Senum, Stephen Springston, Robert McGraw, and Jae Min Yeom. Cloud microphysical relationships and their implication on entrainment and mixing mechanism for the stratocumulus clouds measured during the vocals project. *Journal of Geophysical Research: Atmospheres*, 120(10):5047–5069, 2015.

[163] Yanan Zhan, Li Yu, Xue Yang, and Han Cheng. Initial stress correction method for the modeling of folded space inflatable structures. *Aviation*, 18(4):166–173, oct 2014.

[164] Jinhuan Zhang, Chunsheng Ma, Yuanli Bai, and Shilin Huang. Airbag mapped mesh auto-flattening method. *Tinshhua Sci. Technol.*, 10(3):387–390, jun 2005.