3-D Parachute Simulation by the Penalty Immersed Boundary Method

Yongsam Kim* and Charles S. Peskin†

Abstract

We apply the Immersed Boundary (IB) method to the 3-D parachute problem which involves an interaction between the flexible, elastic, porous parachute canopy and the high speed airflow (relative to the parachute) through which the parachute falls. In order to take account of the effect of mass of the parachute including the payload, the panelty Immersed Boundary (pIB) method, which was devised to provide efficiently and effectively an elastic boundary with mass, is used. We simulate the parachute motion in various realistic situations to show that pIB method is well applicable to the 3-D parachute problem.

1 Introduction

The purpose of this paper is to show that the Immersed Boundary (IB) method can be properly applied to the 3-D parachute problem. We have showed that IB method is well applicable to the 2-D parachute problem [12], and so our work here can be considered as the generalization of 2-D to 3-D parachute problem. The problem of parachute aerodynamics experiences several of the most complex phenomena in classical fluid dynamics, such as porous bluff-body aerodynamics and highly deformable structures. The cloth part of the parachute, which is called the canopy, changes its shape rapidly in response to the surrounding flow field, but the airflow that generates the aerodynamic forces depends on the shape of the parachute canopy. Thus,

*Department of Mathematics, University of Texas at Austin, Austin, TX 78712, USA. kimy@math.utexas.edu

†Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012 USA. peskin@cims.nyu.edu

parachute aerodynamics is inherently a fluid-structure interaction phenomenon, and to express this requires time-dependent position of the parachute as well as the usual variables that appear in the Navier-Stokes equations [4, 14, 17, 18].

The IB method was developed to study flow patterns around heart valves, and is a generally useful method for problems in which elastic materials interact with a viscous incompressible fluid. In the IB formulation, the action of the elastic canopy immersed in the air flow appears as a localized body force acting on the fluid. This body force arises from the elastic stresses in the parachute canopy. Moreover, the parachute canopy is required to move at the local fluid velocity as a consequence of the no-slip condition. This condition is modified, however, in the case of canopy porosity, as described below. The central idea of the IB method is that the Navier-Stokes solver does not need to know anything about the complicated time-dependent geometry of the elastic boundary, and that therefore we can escape from the difficulties caused by the interaction between the elastic boundary and the fluid flow. This whole approach has been applied successfully to problems of blood flow in the heart [19, 20, 21, 22, 23, 24], wave propagation in the cochlea [2, 6], platelet aggregation during blood clotting [7], and several other problems [1, 8, 9, 11, 12, 13, 16].

In the context of the IB method, 2-D problems can be easily generalized to 3-D problems. The 2-D space dimension makes 3-D only by adding one more direciton, and the elastic boundary can be represented by a web of many strings, each of which is usually an elastic boundary itself in 2-D case. A continuously approximate Dirac-delta function and fast Fourier transformation, which are widely used in the IB method, have their conterparts respectively in 3-D case. The easy generalization is such a huge advantage of the IB method that, in doing 3-D simulations, it enjoys its power more apparently in 3-D case than in 2-D one. Even though the problem contains a very complicated structure of an elastic boundary which interacts with 3-D fluid, as long as the initial configuration of the elastic boundary is defined, we can solve the system of equations governing the problem by the IB method with an endurable addition of amount of work and time. In this paper on 3-D parachute, all we have given a great effort in is how we construct the initial configuration of the parachute the deformation of which accounts totally for the external force (except the gravity) in the fluid equation. After a careful construction of the initial shape of parachute, mostly of the canopy part, the simulation is quite straight forward as the generalization of the 2-D parachute case.

To explore the falling parachute, we need the mass of the parachute, especially of the payload. But most previous applications of the IB method have assumed that the elastic materials have no mass. If the elastic material is light enough compared to the density of fluid to ignore the mass of material, the massless assumption is quite reasonable simplification. The applications mentioned above shows this fact well if we consider the results of each simulation. Some problem, however, must have a mass of the material which affects the overall dynamics [13, 28]. To give the mass to the canopy as well as the payload, we use the penalty Immersed Boundary (pIB) method which was designed by the authors and which was applied to the simulatation of several problems to show its applicability and effectiveness [13]. The basic idea of this method is, first, to make the massive boundary imitating massless boundary. Then the two boundary move following their own dynamics but a force that tries to keep them equal comes up to affect two dynamics. The merit of this method, with a constant density in the Navier-Stokes equations, enables us to use the Fast Fourier methodology.

The rest of the paper is organized as follows. We introduce the mathematical derivation and numerical implementation of the pIB in section 2 and 3, respectively. In Section 4, we construct and specify our 3D parachute model and introduce the initial configuration, physical and numerical parameters. Section 5 is for the simulation results which include the inflation of parachute, freely dropping parachute, comparison of drag coefficients of three different situations, and the interaction of two parachutes. The conclusions follow in section 6.

2 Equations of Motion for Penalty IB Method

We begin by stating the mathematical formulation of the equations of the motion for a system comprised of a three-dimensional viscous incompressible fluid containing an immersed, elastic boundary.

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\mathbf{F} = -\frac{\partial E}{\partial \mathbf{X}} + \mathbf{F}_M, \tag{3}$$

$$\mathbf{F}_M = -M \frac{\partial^2 \mathbf{X}}{\partial t^2} - M g \mathbf{e}_3, \tag{4}$$

$$\mathbf{f}(\mathbf{x},t) = \int \mathbf{F}(r,s,t)\delta(\mathbf{x} - \mathbf{X}(r,s,t))drds, \tag{5}$$

$$\begin{aligned} \frac{\partial \mathbf{X}}{\partial t}(r,s,t) &= \mathbf{u}(\mathbf{X}(r,s,t),t) \\ &= \int \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(r,s,t))d\mathbf{x}. \end{aligned} \tag{6}$$

Eqs (1) and (2) are the familiar Navier-Stokes equations for a viscous incompressible fluid. The constant parameters ρ and μ are the fluid density and viscosity, respectively. The unknown functions in the fluid equations are the fluid velocity, $\mathbf{u}(\mathbf{x},t)$; the fluid pressure, $p(\mathbf{x},t)$; and the force per unit volume applied by the immersed boundary to the fluid, $\mathbf{f}(\mathbf{x},t)$; where $\mathbf{x} = (x,y,z)$ are fixed Cartesian coordinates, and t is the time.

Eqs (3) and (4) are the immersed boundary equations which are written in Lagrangian form. The unknown $\mathbf{X}(r,s,t)$ completely describes the motion of the immersed boundary, and also its spatial configuration at any given time t. Note that $\mathbf{X}(r,s,t)$ represents a 2-D surface in 3-D space, which is applicable to both the canopy and payload here. In Eq (3), $\mathbf{F} = \mathbf{F}(r,s,t)$ is the force density applied by the immersed boundary to the fluid, in the sense that $\mathbf{F}(r,s,t)drds$ is the force applied to the fluid by the patch of immersed boundary $drds$. The elastic contribution to this force density is given by the variational derivative $-\frac{\partial E}{\partial \mathbf{X}}$ of the elastic energy functional $E[\mathbf{X}(\cdot,\cdot,t)]$. This variational derivative is implicitly defined by

$$dE(t) = \int \frac{\partial E}{\partial \mathbf{X}}(r,s,t) \cdot d\mathbf{X}(r,s,t)drds, \tag{7}$$

where $d\mathbf{X}$ is a perturbation of the boundary configuration and dE is the resulting perturbation in the elastic energy of the boundary (to first order).

In this paper, the energy functional $E[\mathbf{X}(\cdot,\cdot,t)]$ is chosen a simple way. The 2-D immersed boundaries are modeled as collections of 1-D strings each of which resists stretching and compression:

$$E[\mathbf{X}(\cdot)] = \frac{1}{2}c_s \int (|\frac{\partial \mathbf{X}}{\partial s}| - 1)^2 ds, \tag{8}$$

where c_s is constant. Since, for a fixed r or s, $\mathbf{X}(r,s)$ is a string which is parameterized as a function of s or r, respectively, each such curve can be assigned the same strain-stress relation as in Eq (8). Finally all the forces generated from all these strings are summed up to make an elastic force $\mathbf{F}(r,s)$.

The contribution of the boundary mass to the force density applied by boundary to the fluid in which it is immersed is denoted $\mathbf{F}_M$ in Eq (3) and is written out in Eq (4), where $M = M(r,s)$ is the mass density of the immersed boundary in the sense that $M(r,s)drds$ is the mass of the patch of boundary $drds$. The two terms on the right hand side of Eq (4) are, of course, the inertial force and the gravitational force. In the latter, $\mathbf{e}_3$ is the unit vector pointing "up", i.e. against gravity.

Note that Eqs (3) and (4) can be combined to yield Newton's law of motion for the immersed boundary:

$$M\frac{\partial^2 \mathbf{X}}{\partial t^2} = -\frac{\partial E}{\partial \mathbf{X}} - \mathbf{F} - Mg\mathbf{e}_3. \tag{9}$$

Since $\mathbf{F}drds$ is (by definition) the force applied by the immersed boundary to the fluid, $-\mathbf{F}drds$ is (according to Newton) the force applied by the fluid to the immersed boundary. Although $-\mathbf{F}$ could be expressed in terms of the jump in the fluid stress tensor across the boundary, we shall have no need to do so.

Eqs (5) and (6) which are called interaction equations involve the three-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)\delta(z)$, which expresses the local character of the interaction. Eq (5) expresses the relation between the two corresponding force densities $\mathbf{f}(\mathbf{x},t)d\mathbf{x}$ and $\mathbf{F}(r,s,t)drds$. We can see this fact by integrating each side of the Eq (5) over an arbitrary region Ω. Note, however, that $\mathbf{f}(\mathbf{x},t)$ is a delta function layer with support on the immersed elastic boundary. This is because the delta function that we use is the three-dimensional but the integral in Eq (5) is only two-dimensional. Thus $\mathbf{f}(\mathbf{x},t)$ is infinite on the immersed boundary and zero elsewhere, but in such a manner that its integral over finite volumes is finite.

Eq (6) is the equation of motion of the immersed elastic boundary. It is just the no-slip condition which says that the boundary moves at the local fluid velocity. This is rewritten in terms of the Dirac delta function in the second form of Eq (6). We do so in order to expose certain symmetry with Eq (5), in which the force generated by the immersed boundary is re-expressed as body force acting on the fluid. This symmetry is important in the construction of our numerical scheme. Note, however, that the integral in Eq (6) is a triple integral ($dx_1dx_2dx_3$), unlike the integral in Eq (5). Thus $\partial\mathbf{X}/\partial t$ is finite, unlike $\mathbf{f}$.

Now we modify Eqs (1)-(6) to make the penalty IB Method with which we can handle the mass of elastic material efficiently. For this, we first introduce a configuration $\mathbf{Y}(r,s,t)$ of massive boundary which is exact copy of $\mathbf{X}(r,s,t)$, and then assume that $\mathbf{X}(r,s,t)$ has no mass. The massive boundary $\mathbf{Y}(r,s,t)$ has mass density

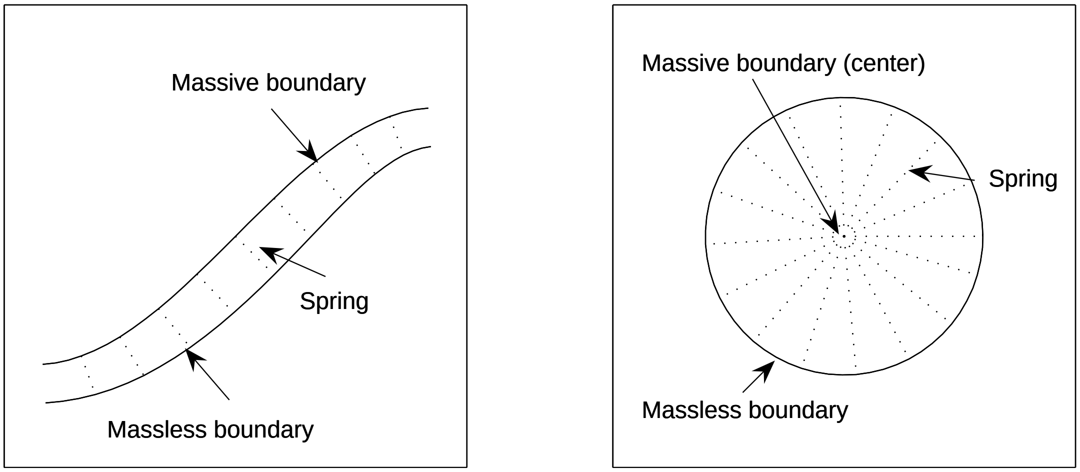

Figure 1: Massive and massless boundaries are linked together with a very stiff spring of which the rest length is zero (left) or the radius of a disc (right). The disc in 2-D is simply generalized into a 3-D ball. The circle (boundary) represents the massless boundary and the center is the massive boundary (point mass).

$M(r,s,t)$ but do not interact with fluid directly. On the other hand, the massless boundary $\mathbf{X}(r,s,t)$ interacts with fluid: it moves with the local fluid velocity and exerts force locally on the fluid. Both massless and massive boundary are supposed to represent the exactly same material surface. But if a pair of corresponding points moves apart, a restoring force appears in order to make them move close together, see the left of Figure 1. The restoring force $\mathbf{F}_K$ acting on the massless boundary is

$$\mathbf{F}_K(r,s,t) = K(\mathbf{Y}(r,s,t) - \mathbf{X}(r,s,t)), \tag{10}$$

where K is a large constant.

Since the massless elastic boundary $\mathbf{X}(r,s,t)$ now has no mass, Eq (4) has no meaning. Instead, the massive boundary $\mathbf{Y}(r,s,t)$ moves by the following equation:

$$M(r,s)\frac{\partial^2 \mathbf{Y}(r,s,t)}{\partial t^2} = -\mathbf{F}_K(r,s,t) - M(r,s)g\mathbf{e}_3. \tag{11}$$

Note that the only forces acting on the massive boundary are the reaction force $-\mathbf{F}_K$ and the force of gravity. The massive boundary does not interact with the fluid directly.

Consider the case in which K in Eq (10) goes to infinity. Then the massless boundary $\mathbf{X}(r,s,t)$ and the massive boundary $\mathbf{Y}(r,s,t)$ should be the same and, F_K

in Eq (11) approaches exactly to F_M in Eq (4). In practice K cannot be infinite but we can keep $\mathbf{X}(r,s,t)$ and $\mathbf{Y}(r,s,t)$ as close as we like by choosing K sufficiently large (depending on the time duration Δt).

To complete pIB Method, F_M and Eq (4) are replaced by F_K and Eqs (11), respectively, to express the motion of the massive boundary, and Eq (10) is added to take the force generated from the massive boundary into account.

In order to introduce a massive payload which is a ball in our simulation, we modify pIB method explained so far. We assume that all the mass of the ball exists only at its center (point mass). Then the massless boundary $\mathbf{X}(r,s,t)$ is a sphere (boundary of the ball) and the massive boundary $\mathbf{Y}(t)$ is a point (center of the ball). When we previously assumed that the two boundaries were supposed to be in the same position, we used $\mathbf{F}_K(r,s,t) = K(\mathbf{Y}(r,s,t) - \mathbf{X}(r,s,t))$ (Eq (10)) which implies that any distance between two boundaries generates a restoring force to make them as close as possible. Imitating this idea, since the sphere and the center of the ball has the distance R, the radius of the ball (payload), we change Eq (10) to generate the restoring force into the following:

$$\mathbf{F}_K(r,s,t) = K(|\mathbf{Y}(t) - \mathbf{X}(r,s,t)| - R)(\mathbf{Y}(t) - \mathbf{X}(r,s,t)), \tag{12}$$

where KR is the spring constant, see the right of Figure 1 which is a 2-D simplification.

For the motion of the center, we need to sum all the reaction forces from the boundaries on the center, i.e.,

$$M\frac{\partial^2 \mathbf{Y}(t)}{\partial t^2} = -\int \mathbf{F}_K(r,s,t)\,drds - Mg\mathbf{e}_3, \tag{13}$$

where M is the point mass and the integration is done over the sphere. When we deal with the force and motion related to the payload, we use Eqs (12) and (13) instead of Eqs (10) and (11) which are mainly used for the force and motion of the massive canopy.

3 Numerical Method

We now describe a formally second-order IB method to solve the equations of motion [15, 25]. The word 'formally' is used as a reminder that this scheme is only second-order accurate for problems with smooth solutions. Even though our solutions are not smooth (the velocity has jumps in derivative across the immersed boundary), the use of the formally second-order method results in improved accuracy, see [15].

The specific formally second-order method that we use is the one described in [25], generalized here to take into account the massive boundary that is linked to the immersed elastic boundary by stiff springs. In this method, each time step proceeds in two substeps, which are called the preliminary and final substeps. In the preliminary substep, we get data at time level $n + \frac{1}{2}$ from data at time level n by a first-order accurate method. Then the final substep starts again at time level n and proceeds to time level $n + 1$ by a second-order accurate method. This Runge-Kutta framework allows the second-order accuracy of the final substep to be the overall accuracy of the scheme.

We use a superscript to denote the time level. Thus $\mathbf{X}^n(r, s)$ is shorthand for $\mathbf{X}(r, s, n\Delta t)$, where Δt is the duration of the time step, and similarly for all other variables. Our goal is to compute updated $\mathbf{u}^{n+1}$, $\mathbf{V}^{n+1}$, where $\mathbf{V}$=$d\mathbf{Y}/dt$, $\mathbf{X}^{n+1}$, and $\mathbf{Y}^{n+1}$ from given data $\mathbf{u}^n$, $\mathbf{V}^n$, $\mathbf{X}^n$ and $\mathbf{Y}^n$.

The grid on which the fluid variables are defined is a fixed uniform cubic lattice of meshwidth $h = \Delta x_1 = \Delta x_2 = \Delta x_3$. Now we introduce the central difference operator D_i, defined for $i = 1, 2, 3$, as follows:

$$(D_i\phi)(\mathbf{x}) = \frac{\phi(\mathbf{x} + h\mathbf{e}^i) - \phi(\mathbf{x} - h\mathbf{e}^i)}{2h}, \tag{14}$$

where $\mathbf{e}^i$ is the unit vector in the i-th coordinate direction. As the notation suggests, the difference operator in i-th direction D_i corresponds to the i-th component of the differential operator ∇. Thus $\mathbf{D}p$ will be the discrete gradient of p, and $\mathbf{D} \cdot \mathbf{u}$ will be the discrete divergence of $\mathbf{u}$.

We shall also make use the central difference Laplacian L:

$$(L\phi)(\mathbf{x}) = \sum_{i=1}^{3} \frac{\phi(\mathbf{x} + h\mathbf{e}^i) + \phi(\mathbf{x} - h\mathbf{e}^i) - 2\phi(\mathbf{x})}{h^2} \tag{15}$$

The fluid mesh and the elastic boundary mesh defined below are connected by a smoothed approximation to the Dirac delta function. It is denoted δ_h and is of the following form:

$$\delta_h(\mathbf{x}) = h^{-3}\phi(\frac{x_1}{h})\phi(\frac{x_2}{h})\phi(\frac{x_3}{h}), \tag{16}$$

where $\mathbf{x} = (x_1, x_2, x_3)$, and the function ϕ is given by

$$\phi(r) = \begin{cases} \frac{3-2|r|+\sqrt{1+4|r|-4r^2}}{8} & , \quad \text{if} \quad |r|<1 \\ \frac{5-2|r|-\sqrt{-7+12|r|-4r^2}}{8} & , \quad \text{if} \quad 1\leq|r|<2 \\ 0 & , \quad \text{if} \quad 2\leq|r| \end{cases} \tag{17}$$

The motivation and derivation for this particular choice is discussed in [21, 23].

We are now ready to describe a typical timestep of the numerical scheme. The preliminary substep which goes from time level n to $n + \frac{1}{2}$ proceeds as follows:

First, update the position of the massless boundary $\mathbf{X}^{n+\frac{1}{2}}(r, s)$.

$$\frac{\mathbf{X}^{n+\frac{1}{2}} - \mathbf{X}^n}{\Delta t/2} = \sum_{\mathbf{x}} \mathbf{u}^n(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^n(r, s))h^3 \tag{18}$$

Here and throughout the paper $\sum_{\mathbf{x}}$ denotes the sum over the cubic lattice in physical space on which the fluid variables are defined. Similarly, $\sum_{r,s}$ will denote the sum over rectangular lattice in (r, s) space on which the two boundary positions $\mathbf{X}$, $\mathbf{Y}$, the force density $\mathbf{F}$ and the mass density M are defined.

The key to generalizing the formally second-order method to the massive case is to handle the massive boundary in a manner that closely parallels the numerical treatment of the immersed boundary itself. Thus, we update $\mathbf{Y}^{n+\frac{1}{2}}$ in a similar manner to $\mathbf{X}^{n+\frac{1}{2}}$:

$$\frac{\mathbf{Y}^{n+\frac{1}{2}} - \mathbf{Y}^n}{\Delta t/2} = \mathbf{V}^n, \tag{19}$$

where $\mathbf{V}^n$ is the velocity vector of the massive boundary (the known value at time $n\Delta t$, like $\mathbf{u}^n$).

Next, calculate the force density $\mathbf{F}^{n+\frac{1}{2}}$ which is the sum of two parts: one is elastic force and the other is from the spring linked between massless and massive boundaries.

$$\mathbf{F}_K^{n+\frac{1}{2}} = K(\mathbf{Y}^{n+\frac{1}{2}} - \mathbf{X}^{n+\frac{1}{2}}), \tag{20}$$

$$\mathbf{F}^{n+\frac{1}{2}} = -\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}^{n+\frac{1}{2}}) + \mathbf{F}_K^{n+\frac{1}{2}}. \tag{21}$$

These are discretization of Eqs (10) and (3) (with change of the notation from F_M to F_K), respectively.

Now we have to change this elastic force density defined on Lagrangian grid points into the force at Eulerian spatial grid points to be applied in the Navier-Stokes equations. This is done by a discretization of Eq. (5).

$$\mathbf{f}^{n+\frac{1}{2}}(\mathbf{x}) = \sum_{r,s} \mathbf{F}^{n+\frac{1}{2}}(r,s)\delta_h(\mathbf{x} - \mathbf{X}^{n+\frac{1}{2}}(r,s))\Delta s \Delta r. \tag{22}$$

With $\mathbf{f}^{n+\frac{1}{2}}$ in hand, we can turn to solving the Navier-Stokes equations:

$$\rho(\frac{u_i^{n+\frac{1}{2}} - u_i^n}{\Delta t/2} + \frac{1}{2}(\mathbf{u}\cdot\mathbf{D}u_i + \mathbf{D}\cdot(\mathbf{u}u_i))^n) + D_i\tilde{p}^{n+\frac{1}{2}} = \mu L u_i^{n+\frac{1}{2}} + f_i^{n+\frac{1}{2}}, \tag{23}$$

for $i = 1, 2, 3$, and

$$\mathbf{D}\cdot\mathbf{u}^{n+\frac{1}{2}} = 0. \tag{24}$$

The notation $\tilde{p}^{n+\frac{1}{2}}$ is used to distinguish this pressure from the one that is computed when solving Eqs (28)-(29), below. Note that the unknowns in Eqs (23)-(24) are $u_i^{n+\frac{1}{2}}$ and $\tilde{p}^{n+\frac{1}{2}}$ and that they enter into these equations linearly. Since all the coefficients of these equations are constants, the system of Eqs (23)-(24) can be solved by Fast Fourier Transform with the periodic boundary condition [23, 25].

Again we have to calculate the velocity $\mathbf{V}^{n+\frac{1}{2}}$ of the massive boundary in the same fashion.

$$M(\frac{\mathbf{V}^{n+\frac{1}{2}} - \mathbf{V}^n}{\Delta t/2}) = -\mathbf{F}_K^{n+\frac{1}{2}} - Mg\mathbf{e}_3. \tag{25}$$

This is nothing but Euler's method for Eq (11) and completes the preliminary substep.

The final substep is the use of $\mathbf{u}^{n+\frac{1}{2}}$, $\mathbf{V}^{n+\frac{1}{2}}$, $\mathbf{X}^{n+\frac{1}{2}}$ and $\mathbf{Y}^{n+\frac{1}{2}}$ obtained in the preliminary substep to compute the corresponding quantities at time level n+1 by the second-order accurate midpoint rule.

First, using the fluid velocity $\mathbf{u}^{n+\frac{1}{2}}$ and massive boundary velocity $\mathbf{V}^{n+\frac{1}{2}}$, we can find the massless boundary configuration $\mathbf{X}^{n+1}$ and massive boundary position $\mathbf{Y}^{n+1}$.

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t} = \sum_{\mathbf{x}} \mathbf{u}^{n+\frac{1}{2}}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^{n+\frac{1}{2}}(r,s))h^3, \tag{26}$$

$$\frac{\mathbf{Y}^{n+1} - \mathbf{Y}^n}{\Delta t} = \mathbf{V}^{n+\frac{1}{2}}. \tag{27}$$

The last thing that we have to do is to update the fluid velocity data:

$$\rho(\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{2}(\mathbf{u}\cdot\mathbf{D}u_i + \mathbf{D}\cdot(\mathbf{u}u_i))^{n+\frac{1}{2}}) + D_i p^{n+\frac{1}{2}} = \frac{1}{2}\mu L(u_i^{n+1} + u_i^n) + f_i^{n+\frac{1}{2}}, \tag{28}$$

for $i = 1, 2, 3$, and

$$\mathbf{D}\cdot\mathbf{u}^{n+1} = 0. \tag{29}$$

Just as in the case of Eqs (23)-(24), Eqs (28)-(29) are a constant-coefficient linear system in the unknowns u_i^{n+1}, $p^{n+\frac{1}{2}}$. This linear system is solved by fast Fourier transform. For the velocity of the massive boundary, we have

$$M(\frac{\mathbf{V}^{n+1}-\mathbf{V}^{n}}{\Delta t}) = -\mathbf{F}_K^{n+\frac{1}{2}} - Mg\mathbf{e}_3. \tag{30}$$

Since we have now computed $\mathbf{u}^{n+1}$, $\mathbf{V}^{n+1}$, $\mathbf{X}^{n+1}$ and $\mathbf{Y}^{n+1}$, the timestep is complete.

To complete the description of the numerical pIB method, we need to explain the boundary conditions imposed along the edges of our computational domain. We use periodic boundary conditions which are very convenient for the solution of the linear systems of Eqs (23)-(24) and Eqs. (28)-(29) by fast Fourier transform, but we also break the periodicity in various ways depending on the situation of simulation.

In order to prevent the parachute from falling down out of domain, we need to impose an "inflow velocity" which is an upward wind. This is done on two (or four) adjacent parallel grid planes. It is very important to use two adjacent planes rather just one, in order to avoid the spatial oscillations that would otherwise propagate throughout the domain via the central-difference structure of our numerical scheme. Although we typically choose either the first or last two grid planes in some coordinate direction, this is of no fundamental significance, since the underlying domain is periodic and all grid points are created equal.

The way of driving the "inflow velocity" in this paper is to apply an external force per unit volume equal to

$$\mathbf{f}_0(\mathbf{x},t) = \begin{cases} \alpha_0(\mathbf{u}_0(t)-\mathbf{u}(\mathbf{x},t)) & , \quad \mathbf{x}\in\Omega_0 \\ 0 & , \quad \text{otherwise,} \end{cases} \tag{31}$$

where Ω_0 is the set of grid points comprised of the four grid planes on which we want to control the velocity, $\mathbf{u}_0(t)$ is the desired velocity on those planes, and α_0 is a constant. When α_0 is large, the grid velocity is driven rapidly towards $\mathbf{u}_0(t)$ within Ω_0.

Another way to break the periodicity is to put no-slip walls on faces of the domain. The method that we shall describe for this purpose can indeed be used to put no-slip walls of any shape anywhere within the domain; planar walls along the faces are just a special case.

We create fixed no-slip boundaries by laying out an array of "target points" to mark their desired positions. To avoid leaks, the target points should be spaced about half a meshwidth apart (or closer). These target points are fixed in position and do not interact with fluid. Each of them is connected, however, by a stiff spring to a massless immersed boundary point that moves at the local fluid velocity and applies the force generated by the stiff spring locally to the fluid. This provides a feedback mechanism for computing the boundary force needed to enforce the no-slip condition.

Note the close analogy between the above construction and the pIB Method, the massive boundary of which is essentially a target position that moves according to Newton's law of motion instead of being fixed in space.

4 Three-dimensional Parachute Model

The model of a parachute which we use here is a conical ribbon parachute which is most widely manufactured and used in practice [4, 14, 17, 18].

The basic component of a conical parachute is an isosceles triangle which is called a gore. The several gores are put together and seamed with each side to make one canopy (cloth part of a parachute), see the top in Figure 2. One gore is made of three kinds of fabrics or ribbons: one is a vertical line, another horizontal line, and the other radial line. These lines are different in their dimensions and strengths. Because a falling parachute in air experiences different drag forces in different parts, the part exerted by a bigger force needs a stronger line or ribbon. Our computational model can take this into account by changing the tension coefficients of these lines.

Some real parachutes have many gaps between ribbons to make its canopy to be a porous body. In computation, however, it is impractical to resolve these gaps individually. To do so would require that each gap be at least a few fluid meshwidths in diameter. Otherwise the hole would be effectively closed. Thus a very fine mesh would be required. Because of this difficulty, we do not make gaps between the ribbons except one big hole called a vent on the top of the canopy, see Figure 3. For a different approach, however, see [12].

The vent on the top of a real parachute is known to be important both to stabilize the parachute motion and to help its secure inflation at the first stage [18]. Because it is difficult to see these facts due to the limitation of the size of the computational domain and computer capacities, we do not explore its effect here. Instead, the vent is made in our model to generate reasonable amount of force from the parachute top. We use the each line (string or ribbon) to make a gore and thus a canopy. In the procedure of the IB method, these lines are acting as springs generating the Lagrangian force

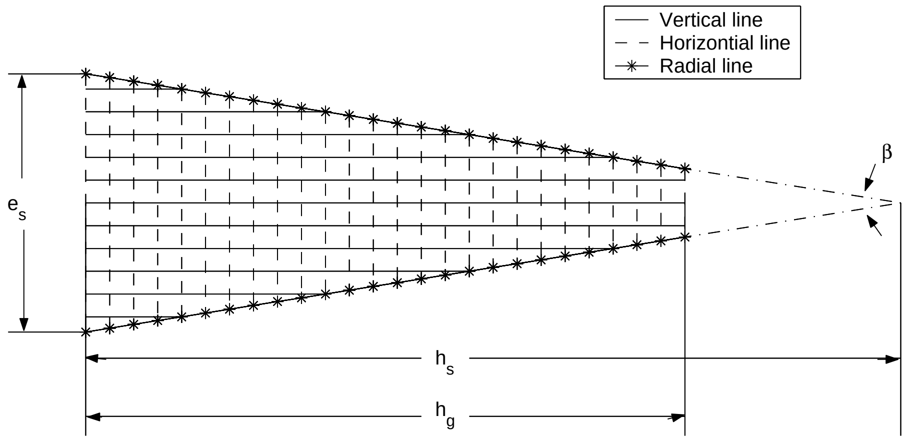

Figure 2: Gore layout. Gore which is made up of three types of lines is an element of a canopy.

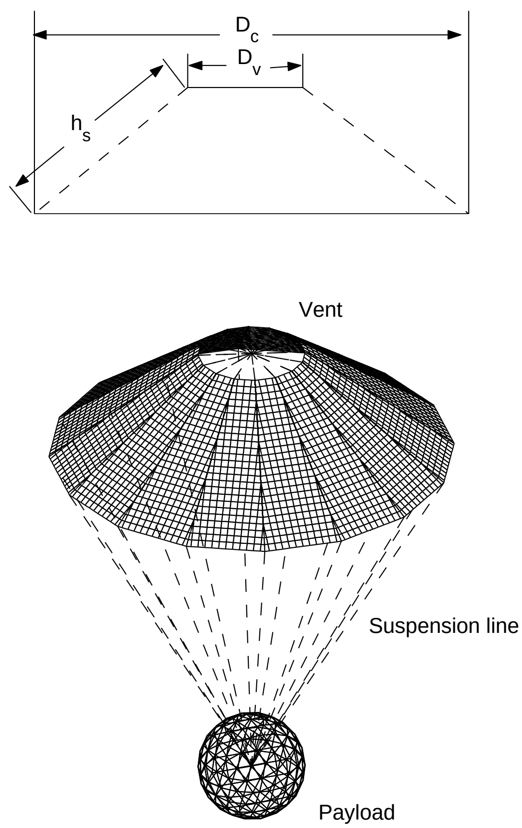

Figure 3: Constructed profile of a parachute with canopy, payload (ball) and suspension lines. The constructed profile of a parachute is shown on the top. The canopy and payload (ball) are connected by suspension lines (bottom).

Table 1: Construction parameters of parachute (unit:m)

Parameter	symbol	magnitude
Height of Triangle (Gore)	h_s	0.42
Height of Gore	h_g	0.3432
Number of Gores	N_g	16
Gore Base Length	e_s	0.1106
Vertex Angle of Gore	β	$\pi/12$
Base Cone Angle	μ	0.27π
Length of Cone Base	D_c	0.2780
Length of Vent	D_v	0.1063
Suspension Line Length	l_s	0.45
Radius of Payload	R	0.07

which is transformed into a body force in the Navier-Stokes equations (see Section 2). If a canopy has no vent at its apex, it would let so many strings gather at the apex that the force around it is too large compared to the other parts of the canopy. By making a hold at the top of canopy, we can avoid this unpredictably large force which would induce an unstable condition of the computation. For the same reason, the real parachute has a vent at its apex without which such a large drag force would be generated around the apex to plunge the parachute in any direction.

With N_g gores of the same size, we attach these gores with each sides to make a cone shape canopy (Figure 3). For a canopy to have a cone shape, only criterion needed is $\beta < 2\pi/N_g$ where β is the vertex angle of the gore (Figure 2 and Table 1). The top in Figure 3 shows the constructed profile of a conical parachute, and the bottom one shows the completed canopy with other elements of a parachute: suspension lines and payload. In order to make the suspension lines, in each time step, a fixed point (determined initially) of the moving payload is just connected to one low vertex of each gore of the canopy. These suspension lines are pure force generators in that they do not interact with the fluid along their length, but simply act as linear springs which apply force to the two low vertices of each gore and the center of the payload.

Now consider an incompressible viscous fluid in a cube (0m,2m)$\times$(0m,2m)$\times$(0m,2m)

Table 2: Physical parameters of parachute simulations.

Physical parameters	symbol	magnitude	unit
Density	ρ	1.2	kg/m^3
Viscosity	μ	2×10^{-3}	kg/(m·s)
Gravitational acceleration	g	9.8	m/s^2
Mass of Payload	M_p	$0.003 - 0.012$	kg
Density of Canopy	M_c	0.06	kg/m^2
Inflow velocity	W_0	$1.0 - 1.2$	m/s
Reynolds number		up to 250	
Computational Domain		$2 \times 2 \times 2$	m×m×m

with periodic boundary conditions (but see above) which contains a parachute and a payload as immersed boundaries. Using the physical parameters in Table 1, we construct our model parachute which is shown in Figure 3. In the table, we use conventional terminologies widely known in parachute manufacturing industries such as gore size and constructed parachute size. The parameters in Table 1 determine one size of a conical parachute: once the number of gores N_g, the gore height h_g, and the vertex angle of the gore β are chosen, the unique canopy is determined from these parameters. Note that the overall size of the parachute is small and the vent size rate to that of canopy is big compared to a real parachute.

Table 2 shows the physical and computational parameters used in the simulation. To get a modified Reynolds number, we choose a larger viscosity than that of air. In real world, parachute dynamics has around $300,000$ Reynolds number at its steady descent. Based on our modified viscosity, the canopy size measured as the diameter of its fully inflated shape, and the inflow speed made to keep a parachute inside the domain (see below), our simulation is done with Reynolds number up to 250.

The most natural way to model a parachute would be to let it fall, under the influence of gravity acting an its canopy and payload, through air which would be at rest at a large distance from the parachute. For the simulation which has a fixed domain, however, if we allow the parachute to move freely with its own dynamic law, it can run away from the domain. In order to avoid it, i.e., to keep the parachute within the domain, we use a control mechanism to adjust the inflow velocity in such a manner that the y-coordinate of the payload settles to a predetermined value. The equation governing this control mechanism is as follows:

$$\frac{dW_0(t)}{dt} = \gamma(z_{target} - Z_p(t)) - \sigma W_p(t) \tag{32}$$

Here, $Z_p(t)$ and $W_p(t)$ are obtained by taking the z-components of the position and the velocity of the center of payload, respectively. These values are calculated by using Eq (30) in the procedure of each time iteration. The velocity $(0, 0, W_0(t))$ is the inflow velocity at time t, z_{target} is the fixed value at which the z-coordinate of the payload center is supposed to be in its equilibrium state, and γ and σ are constant coefficients. The equation says that if, at some time, the height of the payload $Z_p(t)$ is lower than the target position of the payload z_{target}, the inflow velocity increases, and if $Z_p(t)$ is greater than z_{target}, the inflow velocity decreases. The change of the inflow velocity, however, is damped according to $W_p(t)$ in order to avoid large oscillations of the inflow velocity.

5 Results and Discussions

5.1 Inflation and Landing of Parachute

The first computational experiment that we consider involves the process of parachute inflation, starting from a nearly closed configuration and studying the changes in shape of the parachute at early times, see Figure 4. Of all the results that we consider, it is this one that shows most clearly the need for a method that can handle the unknown changes in shape of the parachute canopy. In the IB method, this is done without any re-gridding, since the canopy is represented in the fluid dynamics computation by a force field defined on a uniform grid. For alternative approaches, see [5, 26, 27].

Figure 4 shows the inflation of a parachute as time goes on. The initial configuration is modified from Figure 3 to be of a shape of a nearly closed cone, see the left-top picture of the figure. The upper two rows show the inflation procedure of a parachute in time which begins at left-top and increases with columns and then with columns in the next row. At the beginning of the parachute inflation, the canopy inflates from the bottom through to the top, and the upper part of canopy is over-inflated, see the figure in second column and second row. But air captured by the canopy finally applies enough pressure to the other part so that the parachute inflates completely.

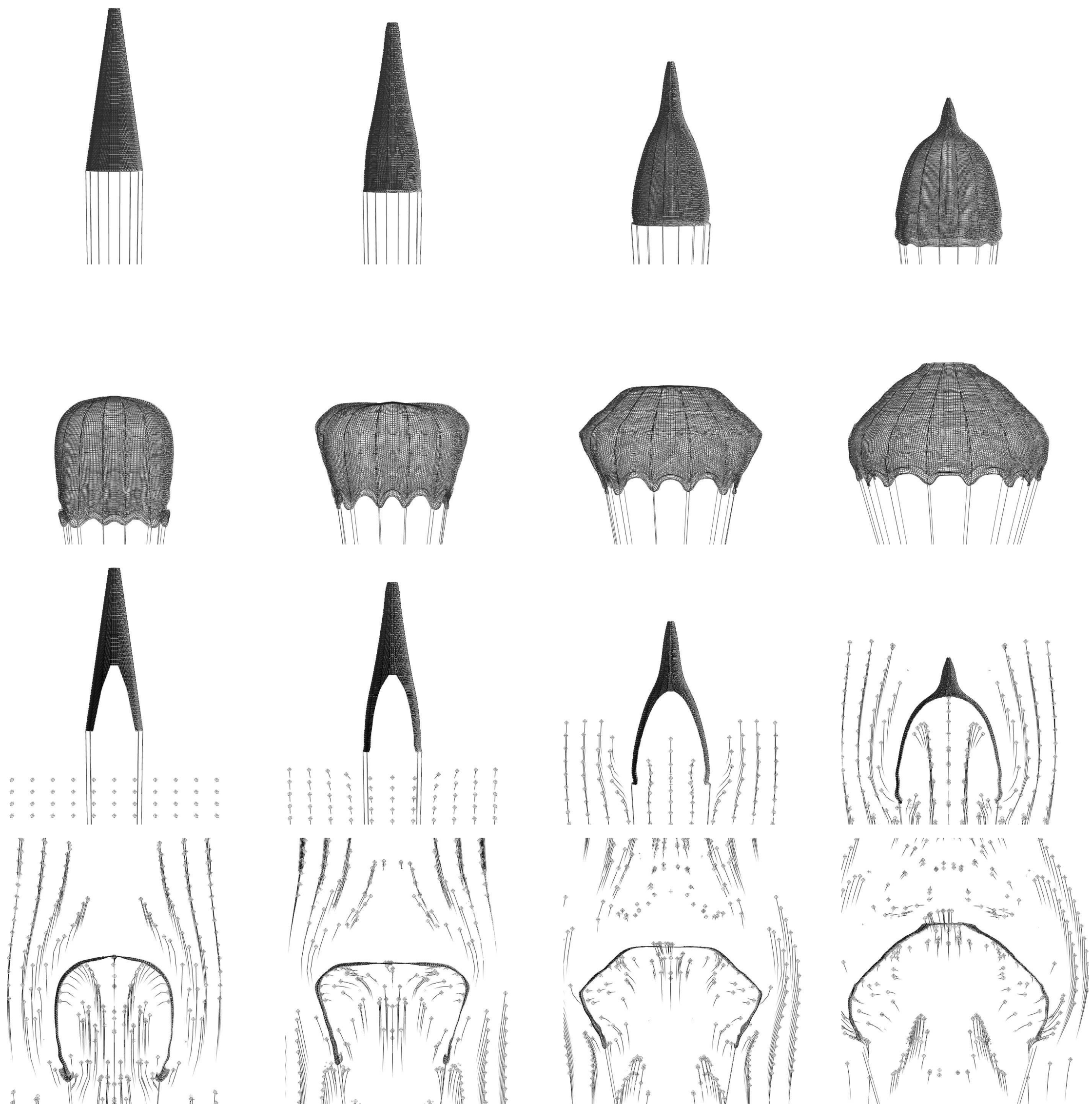

Figure 4: The inflation of a parachute is shown in early times: time=0s, 0.2s, 0.4s, and 0.6s in the fist row, and 0.8s, 1.0s, 1.2s, and 1.4s in the second row. The bottom two rows show the same parachute at the same time as the ones at the corresponding positions in the upper two rows.

This inflation procedure is very similar to the results in [5, 12, 27], though the latters were 2-D simulations.

The bottom two rows of Figure 4 show the same parachute at the same time as the ones at the corresponding positions in the upper two rows. While the latters

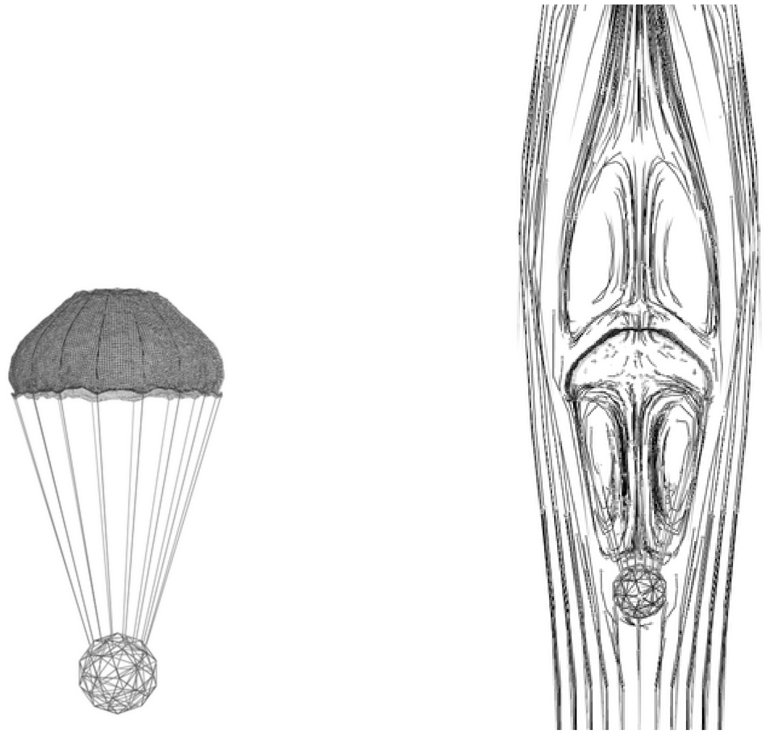

Figure 5: Fully inflated parachute at time=6.0s.

are seen from a far distance, the formers from cross sections containing the middles of the parachute and the domain. From the trails which are left from some fluid markers and which show their recent trajectories, we can observe that there are vortex sheddings outside the canopy, and that the fluid motion inside the canopy push the canopy outward. After the time proceeds beyond the last time shown in Figure 4, the parachute moves stably without further change in its configuration in some time. We can see the motion of the parachute at a later time $t = 6.0$ s at Figure 5.

It is interesting to see a parachute landing on a fixed ground. For this simulation, the inflow is removed and, instead, we add some shear flow driven left to right to create a more realistic situation. The ground on which the parachute lands is made by the way of generating a fixed no-slip boundary which we have introduced in Section 3. Figure 6 shows the motion of a landing parachute and the fixed ground as time goes on. The initial configuration of the parachute at $t = 0s$ (left-top of the figure) is a modified one with a smaller initial opening than that of Figure 3. Note that, even though there is an impact force when the payload collides into the ground, we have found that the ground experiences almost no dent, which is attributed either to a too

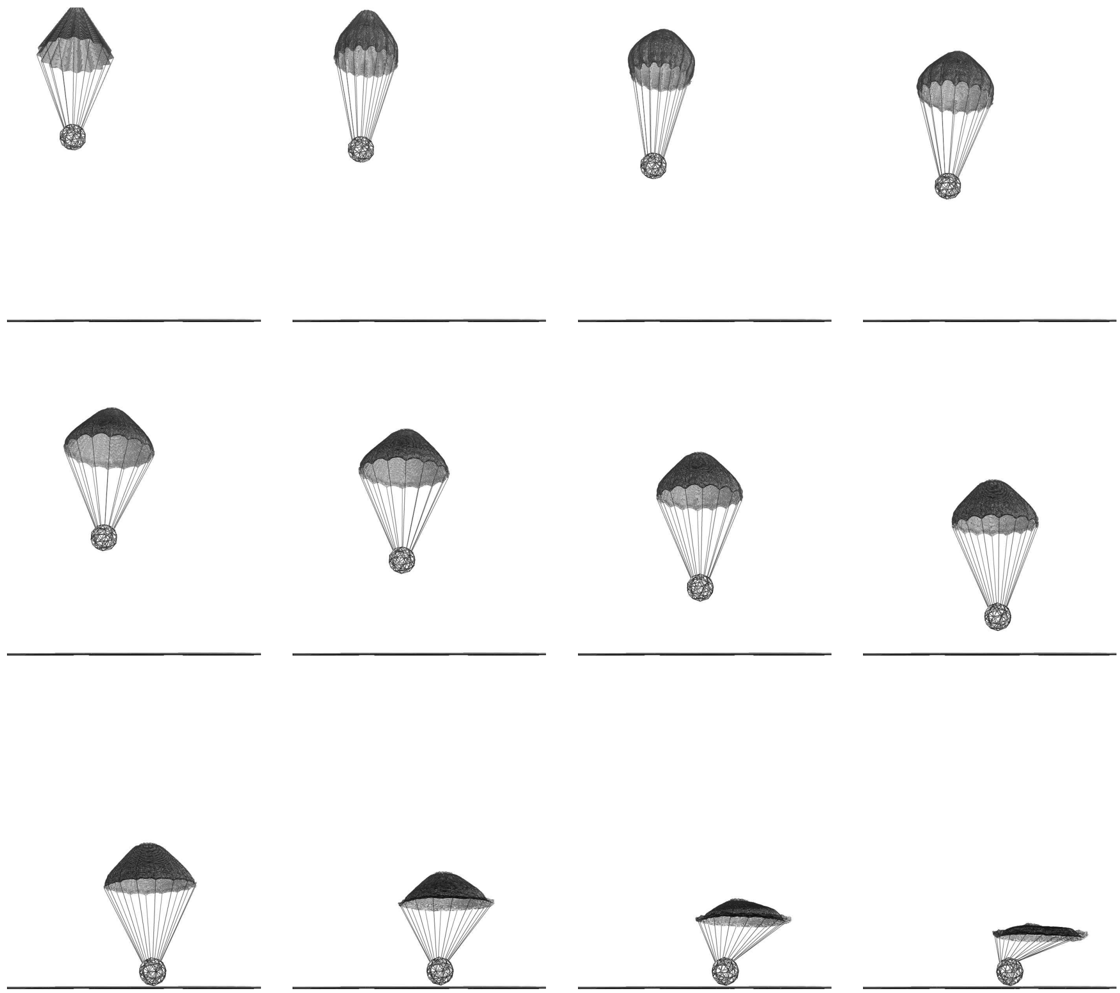

Figure 6: A parachute is falling with gravity to the ground which is a fixed no-slip boundary. While there is no upward wind, a small shear flow from left to right exists: time=0, 0.2, 0.4, and 0.6 s in the fist low, and 0.8, 1.0, 1.2, and 1.4 s in the second low, 1.6, 1.8, 2.0, and 2.2 s in the third low.

small impact force due to a low initial height of the parachute (1 m high), or to the strong attachment of the immersed ground to the fixed boundary which is dependent on the spring coefficient connecting them.

5.2 Parachute with One and Multiple Canopies

In this section, we compare the drag coefficients of a payload in three different situations: (1) payload dropping alone, (2) supported by one-canopy parachute, and (3) supported by three-canopy parachute.

When a body is placed in and partially blocks a flow, the body feels a drag force. A dimensionless measure of this effect is the drag coefficient C_D which is defined as follows:

$$C_D = \frac{F_D}{\frac{1}{2}\rho V_0^2 A}, \tag{33}$$

where ρ is the density of fluid, V_0 is the far-field uniform velocity, and A is the projected area of the body which is π times square of the radius of the ball in our case. F_D is the force per unit length of the payload exerted parallel to the direction of the flow. See [3] for more details. Note that the drag coefficient C_D is dimensionless.

The overall performance of a parachute can be summarized by the relationship between the drag force it generates and the speed at which it is falling (relative to the air at a large distance from the parachute). In physical experiments and also in computational simulations, the usual way to determine the drag force and coefficient of a body is to have the body fixed in a given flow. If the payload is fixed in place and the inflow velocity is arbitrary, then the speed of the parachute (relative to the distant air) is the independent variable, and the drag force could be computed. Indeed, the drag force can be determined simply by examining the tensions and angles of the suspension lines.

In our simulation, however, we drop the payload (in our case, a ball) attached to a parachute and let it fall freely, and the inflow velocity $W_0(t)$ is adjusted to keep the parachute from falling or rising. In this case, the drag force is the independent variable, since it has to be equal to the specified weight of the payload, and the speed corresponding to the given drag force is just the equilibrium value of the inflow velocity, as set by the control mechanism (Eq (32)). When the position of the payload and the inflow velocity $W_0(t)$ have both settled down, the total force acting on the ball must be zero, and this implies that the drag force per unit length of the payload F_D must be equal to the weight of the payload $M_p g$, where M_p is the mass of the payload and g is the gravitational acceleration. The value to which $W_0(t)$ settles down may be thought of as the free stream velocity that generates the drag F_D.

In order to get a better performance, the parachute makers often produce parachutes which have several canopies supporting one payload. Adapting the same model of the one-canopy parachute used in the last sections, we can simply create a three-canopy parachute by connecting the center of the payload to three canopies with suspension lines, see Figure 7. The figure shows the configurations from top and side in the initial

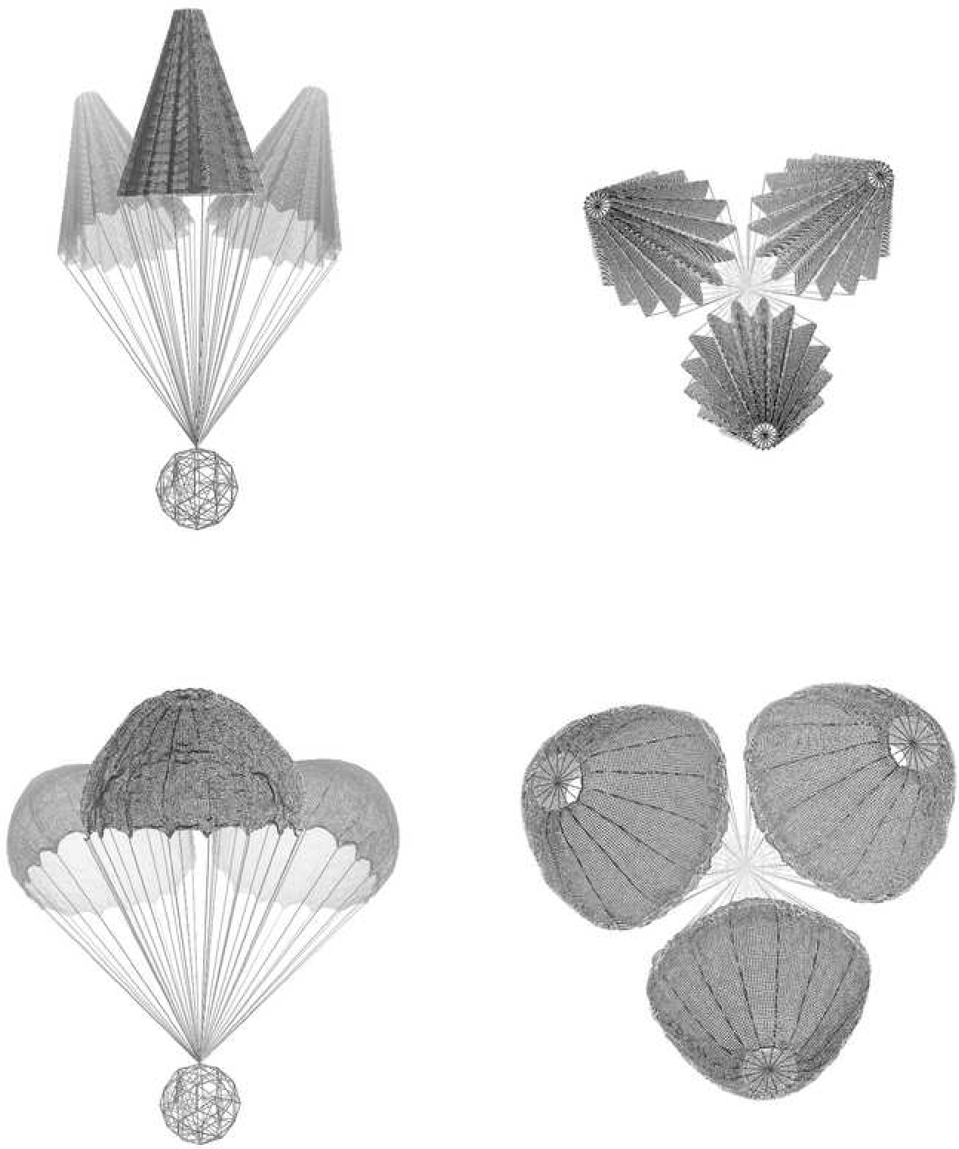

Figure 7: Three-canopy parachute in time;top=0.0s and bottom=1.2s.

(0.0 s) and later times (1.2 s).

Before comparing the performance of the parachutes, we first have to check whether our control mechanism (Eq (32)) results in the desired effect that the controlled inflow makes the ball center stay around one position with a uniform inflow. To see that, we plot the height of the ball center $Z_p(t)$ and the applied inflow velocity $W_0(t)$ in Figure 8. The dotted, dashed, and solid lines represent the results of (1) the payload

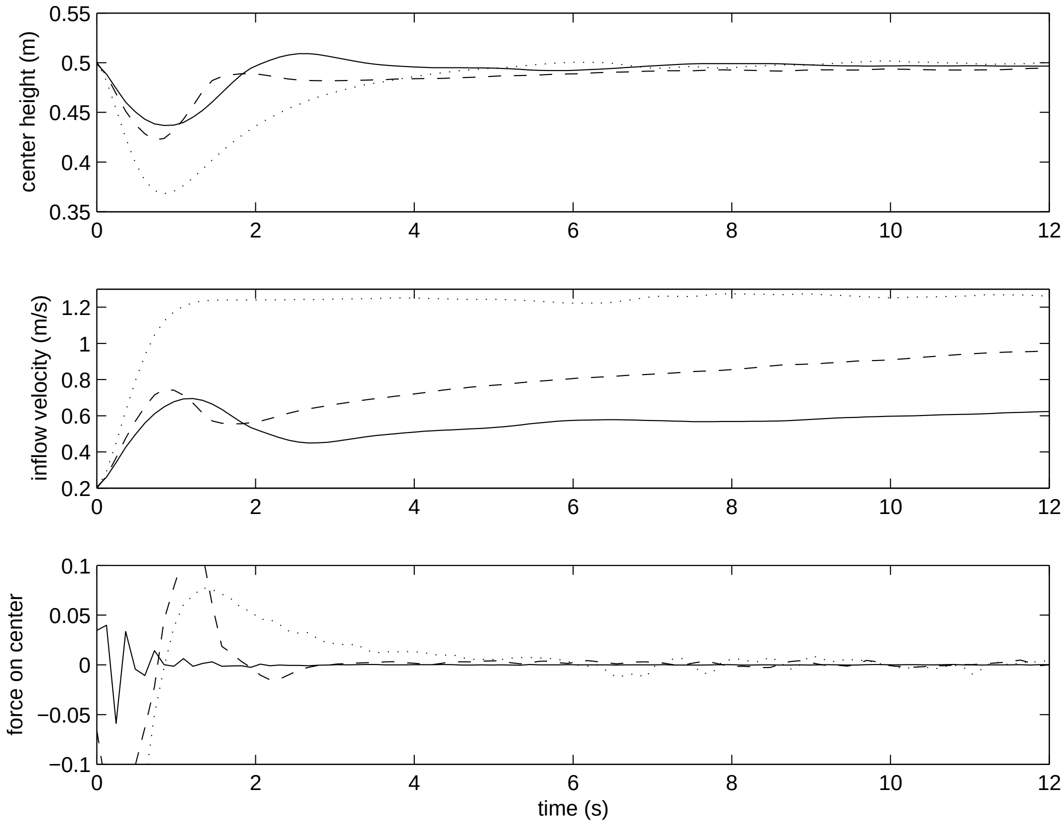

Figure 8: The height of the ball center $Z_p(t)$ (top), the desired inflow velocity $W_0(t)$ (middle), and the force acting on the center $-\int \mathbf{F}_M(s,t)ds - M_p g\mathbf{e_3}$ (bottom) are plotted: dotted line for the payload only, dashed line for the payload with one canopy, and solid line for the payload with three canopies.

only, (2) payload with one canopy, and (3) payload with three canopies, respectively. We can see that the height of the ball centers in the three cases approach one value $z_{target} = 0.5$ m (top), and that the inflow velocity stays around values 1.28 m/s in case (1), 1.0 m/s in case (2), and 0.6 m/s in case (3). This is exactly what the control mechanism is designed to achieve. The parameters $\sigma = 4.0/\text{s}^2$ and $\gamma = 5.0/\text{s}$ are used for the constants in (32). The bottom panel plots the force acting on the center of the ball in terms of time and shows that this force settles down to zero (equilibrium).

With the almost uniform velocity $W_0(t)$, we are ready to calculate the Reynolds number and the drag coefficient. To do that, we also need some parameters used in the simulation which are displayed in Table 2. To get a reasonable inflow velocity, we use M_p=0.003 kg for the case (1), M_p=0.006 kg for the case (2), and M_p=0.012 kg

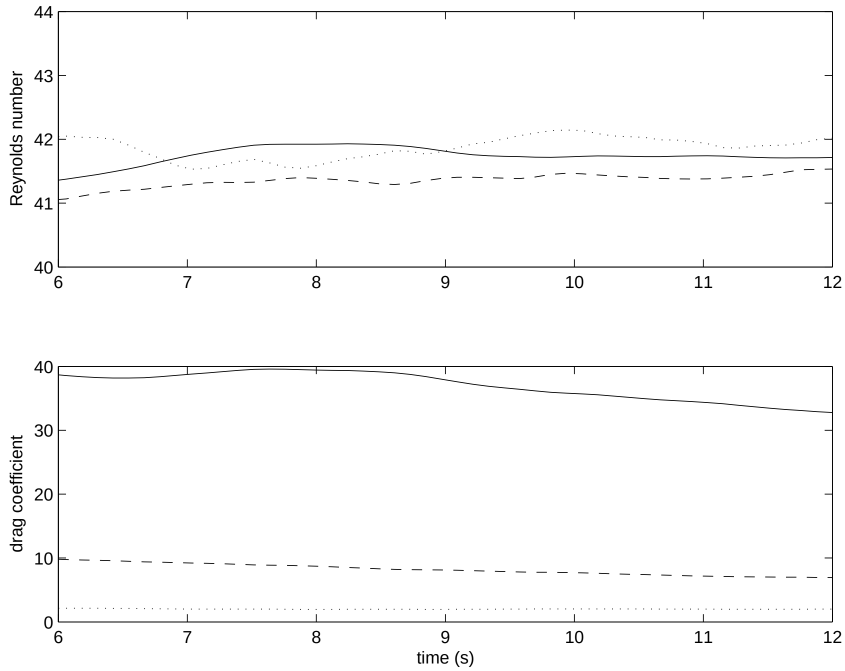

Figure 9: The Reynold number (top) and the drag coefficient (bottom) as functions of time.

for the case (3). After the height of the parachutes approache almost steady states, we plot the Reynolds numbers (top) and the drag coefficient (bottom) as functions of time in Figure 9. Note that, while the Reynolds numbers here are based on the diameter of the payload, the actual Reynolds numbers in the simulations are bigger than these values when based on the size of the dimension of parachute canopy.

Our result shows that the Reynolds numbers are about 42 in all the three cases. The drag coefficient, however, are different: the first case approaches about 2.0, the second case does 7.1, and the third one does 32.0. The drag coefficient obtained by the experiment with a ball is also around 2.0 with Reynolds number 40 which is comparable to our result in case (1), see [3]. Note that the performance of three-canopy parachute is more than 4 times better than that of one-canopy parachute.

5.3 Interactions between Two Parachutes

The last simulation is done with two parachutes to explore their interaction. When two parachutes get close, the motion of one parahcute can be strongly influenced by its interaction with the other parachute. These interactions sometimes can lead to the

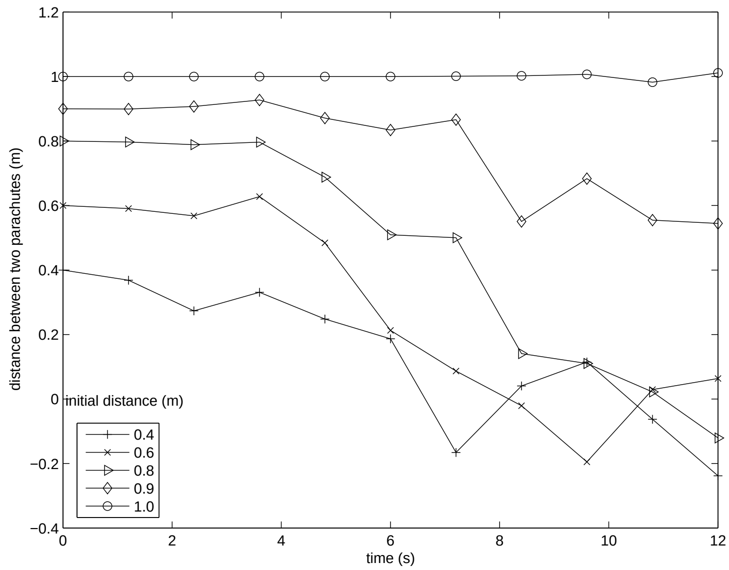

Figure 10: Distance of two parachutes as a function of time. Each curve represents a different initial distance. As time goes on, all the distances deviate from their initial distance except the one with the initial distance 1.0 m which is an extreme case.

failure of the parachutes such as a unstable motion and an incomplete inflation. One reason for that is because there is the vortex shedding of the parachutes, especially, if its position is lower than the other, which prevents enough and regular pressure difference for the other parachute.

In order to see how the distance between two parachutes affect the overall dynamics, two parachutes are placed in the computational domain with various distances in both x-direction and z-direction. The lower parachute is fixed at its payload, the higher one can move freely as in the previous sections, and their z-directional distance is 0.9 m. The payload of the lower parachute does not participate as a force generator, and only its canopy affects the whole dynamics, see the left-top in Figure 11. (We can do with both parachutes moving freely too. However, it would need a much bigger computational space.)

Now we change the x-directional distance of the two parachutes to see the influence of their initial distance on the overall dynamics of the interaction. The distances are chosen 0.4, 0.6, 0.8, 0.9 and 1.0 m. The Figrue 10 shows the x-directional distance of the two parachutes as a function of time. Each curve represents a different initial distance. As time goes on, all the distances deviate from their initial distance except the one with the initial distance 1.0 m. The times of the deviation, however, are clearly different between the distances below 0.8 and above 0.8. While, in the former cases, the deviation occurs suddenly and largely at time around 4 s, that of the latter cases slowly begins at 8 s and is not big compared to the formers. Note that the parachutes with the distance 1.0 m is an extreme case in that 1.0 m is perfectly half the size of the domain and, because of the periodicity, the repitition of parachutes with the same distance prevent the deviation.

Figure 11 and Figure 12 show the motions of the two parachutes with different initial distances: 0.6 m (Figure 11) and 1.0 m (Figure 12). In both figures, the upper two rows show the parachutes motion seen from the far distance and the lower two rows show the parachutes and fluid markers from cross sections containing the middles of the parachute and the domain. From the trails which are left from some fluid markers and which show their recent trajectories, we can observe that there are vortex sheddings around the parachutes. Notice that, in the case of the 0.6 m distance, the upper parachute moves over the lower parachute.

6 Conclusions

We have presented numerical experiments concerning the parachute problem in the three-dimensional case. Two basic configurations have been studied: one with a fixed payload in a prescribed updraft, and the other with a free payload in a controlled updraft, the controller being designed to adjust the updraft so that the parachute stays within the computational domain. The coupled equations of motion of the air and the flexible parachute canopy have been solved by the Immersed Boundary(IB) method. We have used this methodology to simulate the details of parachute inflation, and to study the influence of canopy porosity on the lateral stability of the parachute.

Acknowledgement

This work was supported by the National Science Foundation under KDI research grant DMS-9980069.

References

[1] K.M.Arthurs, L.C.Moore, C.S.Peskin, E.B.Pitman, and H.E.Layton. *Modeling arteriolar flow and mass transport using the immersed boundary method.* J. Com-

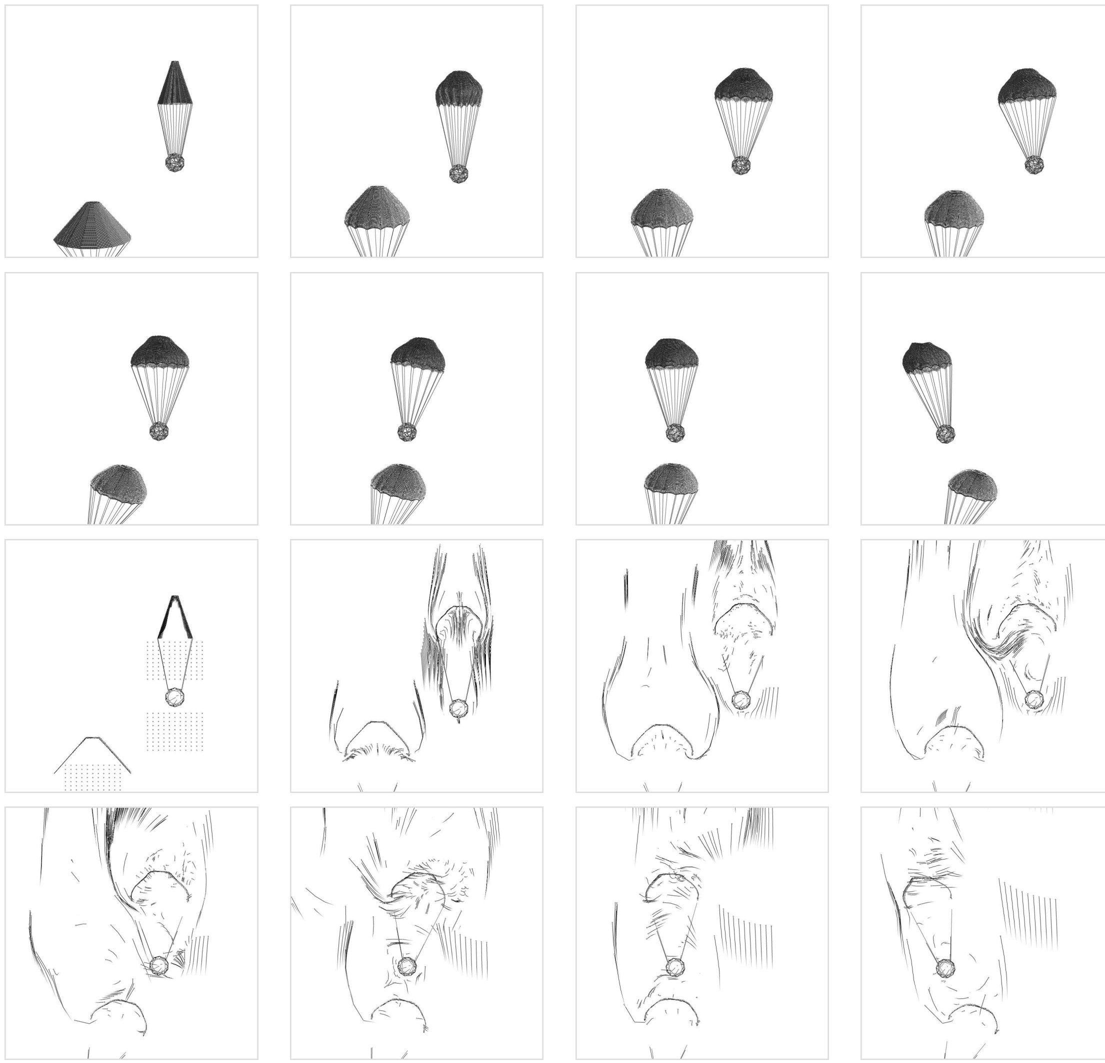

Figure 11: Interaction of two parachutes with a close initial distance: time=0s, 1.2s, 2.4s, and 3.6s in the fist row, and 4.8s, 6.0s, 7.2s, and 8.4s in the second row. The lower two rows have have the same time in the corresponding position.

put. Phys. 147:402-440, 1998

[2] R.P.Beyer. *A computational model of the cochlea using the immersed boundary method.* J. Comput. Phys. 98:145-162, 1992

[3] R.D.Blevins. *Applied Fluid Dynamics Handbooks.* Van Nostrand Reinhold Company, page=279–381, 1984.

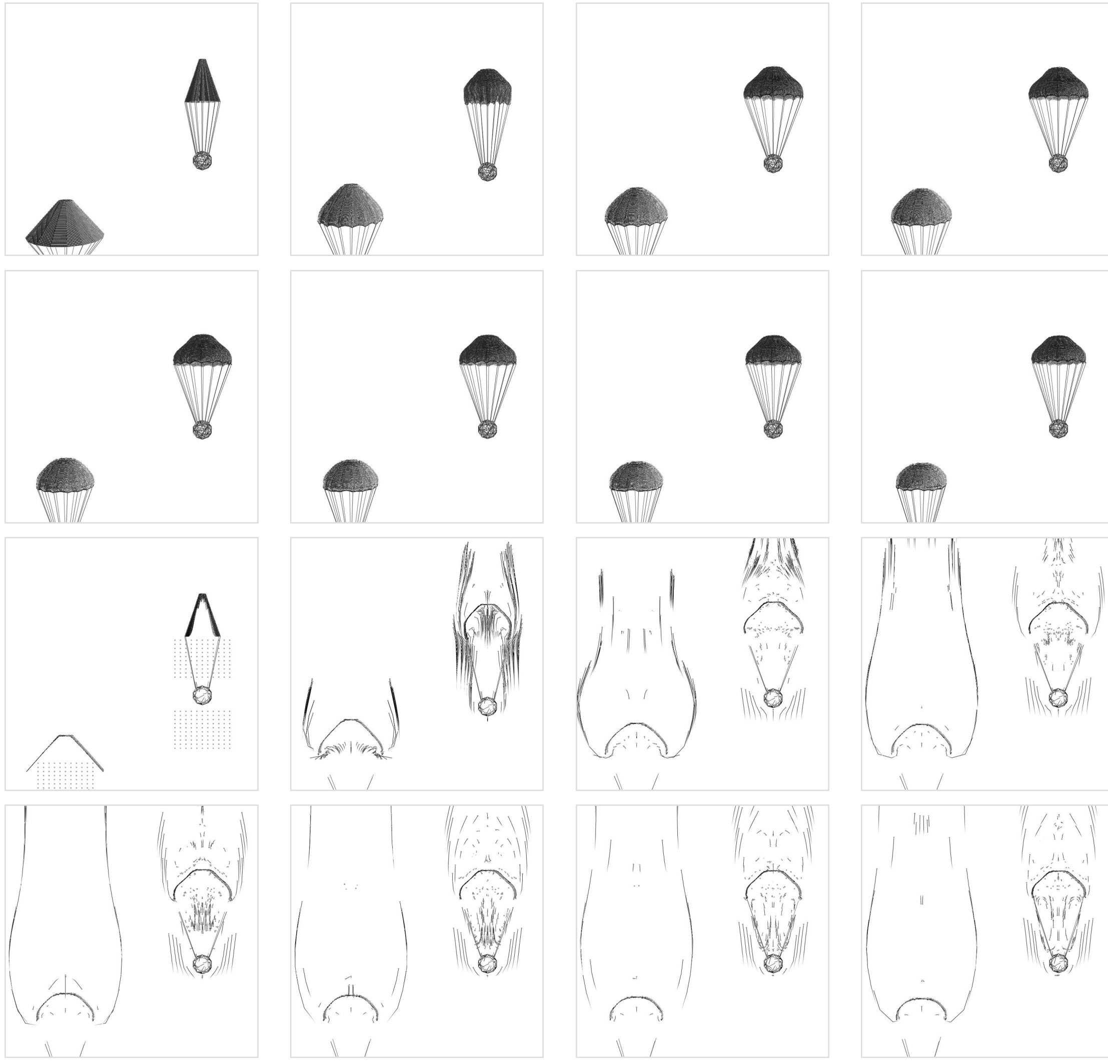

Figure 12: Interaction of two parachutes with a wide initial distance: time=0s, 1.2s, 2.4s, and 3.6s in the fist row, and 4.8s, 6.0s, 7.2s, and 8.4s in the second row. The lower two rows have have the same time in the corresponding position.

[4] D.J.Cockrell. *The Aerodynamics of Parachutes.* AGARDograph No. 295, July 1987.

[5] I.V.Dneprov. *Computation of aero-elastic characteristics and stress-strained state of parachutes.* AIAA Pap. 93-1237, 1993

[6] E.Givelberg. *Modeling elastic shells immersed in fluid.* Ph.D.Thesis, Mathematics, New York University, 1997.

[7] A.L.Folgelson. *A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting.* J. Comput. Phys. 56:111-134, 1984

[8] A.L.Folgelson and C.S.Peskin. *A fast numerical method for solving the three-dimensional Stoke's equations in the presence of suspended particles.* J. Comput. Phys. 79:50-69, 1988

[9] L.J.Fauci and C.S.Peskin. *A computational model of aquatic animal locomotion.* J. Comput. Phys. 77:85-108, 1988

[10] H.Higuchi and F.Takahashi. *Flow past two-dimensional ribbon parachute models.* J. Aircr. 26(7):641-649, 1989

[11] E.Jung and C.S.Peskin *Two-dimensional simulations of valveless pumping using the immersed boundary method.* SIAM J.Sci.Comput. 23:19-45, 2001

[12] Y.Kim and C.S.Peskin. *2-D parachute simulation by the Immersed Boundary Method.* preprint. submitted to SIAM journal on Scientific Computing.

[13] Y.Kim and C.S.Peskin. *Penalty Immersed Boundary Method for an Elastic Boundary with Mass.* preprint. submitted to J. Comput. Phys.

[14] T.W.Knacke. *Parachute Recovery Systems Design Manual.* NWC TP 6575, Naval Weapons Center, China Lake, California, June 1987

[15] M.C.Lai and C.S.Peskin. *An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity.* J. Comput. Phys. 160:705-719, 2000

[16] S.Lim and C.S.Peskin *Simulations of Whirling Instability by the immersed boundary method.* SIAM J.Sci.Comput. 25:2066-2083, 2004

[17] R.C.Maydew and C.W.Peterson. *Design and Testing of High-Performance Parachutes.* AGARD-AG-319. 1991

[18] C.W.Peterson, J.H.Strickland, and H.Higuchi. *The fluid dynamics of parachute inflation.* Annu.Rev.Fluid.Mech. 28:361-387. 1996

[19] C.S.Peskin. *Flow patterns around heart valves:A numerical method.* J. Comput. Phys. 10:252-271,1972

[20] C.S.Peskin. *Numerical analysis of blood flow in the heart.* J. Comput. Phys. 25:220-252, 1977

[21] C.S.Peskin and D.M.McQueen. *Three dimensional computational method for flow in the heart :Immersed elastic fibers in a viscous incompressible fluid.* J. Comput. Phys. 81:372-405, 1989

[22] C.S.Peskin and D.M.McQueen. *A general method for the computer simulation of biological systems interacting with fluids.* Symposia of the society for Experimental Biology. 49:265-276, 1995

[23] C.S.Peskin and D.M.McQueen. *Fluid dynamics of the heart and its valves. In: Case studies in Mathematical Modeling: Ecology, Physiology, and Cell Biology.* Prentice Hall, Englewood Cliffs NJ, 1996, pp. 309-337

[24] C.S.Peskin and B.F.Printz. *Improved volume conservation in the computation of flows with immersed elastic boundaries.* J.Comput. Phys. 105:33-46,1993

[25] C.S.Peskin and D.M.McQueen. *Heart Simulation by an Immersed Boundary Method with Formal Second-order Accuracy and Reduced Numerical Viscosity.* In: Mechanics for a New Millennium, Proceedings of the International Conference on Theoretical and Applied Mechanics(ICTAM) 2000, (H.Aref and J.W.Phillips,eds.)Kluwer Academic Publishers,2001

[26] J.Sahu, G.Cooper, and R.Benney. *3-D parachute descent analysis using coupled CFD and structural codes.* AIAA Pap. 95-1580, 1995

[27] K.R.Stein and R.J.Benney. *Parachute inflation: a problem in aero-elasticity.* US Army Tech.Rep. NATICK/TR-94/015, Natick, MA. 1994

[28] L.Zhu and C.S.Peskin. *Simulation of a flapping flexible filament in a flowing soap film by the Immersed Boundary method.* preprint, submitted to J.Comput. Phys.