Simulation of a Flapping Flexible Filament in a Flowing Soap Film by the Immersed Boundary Method

Luoding Zhu

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics

New York University

September 2001

Advisor: Charles S. Peskin

Dedicated to my wife Ying and my son Eric

Acknowledgements

I would like to take this opportunity to express my deep gratitude to my advisor, Charles Peskin, for his excellent guidance, frequent help, great patience, and constant support throughout my thesis research.

I want to thank Jun Zhang very much for many discussions of his laboratory experiment and for his generosity of giving me the file to plot the experimental setup. I want to thank Michael Shelley very much for many helpful discussions of the problem and for giving me the copy of his note on the instability analysis of the problem. My thanks go to Peter Schmid for helpful discussion of multigrid method, go to Stephen Childress, Donna Calhoun and Daniel Forger for helpful discussions of the problem, go to David McQueen, Estarose Wolfson, and Boyce Griffith for help in using computers and softwares.

Finally, I want to thank my wife and my father-in-law for their support and help in life during my thesis research.

Abstract

This thesis reports the computer simulation of a flapping flexible filament in a flowing soap film using the Immersed Boundary Method. Our mathematical formulation includes filament mass and elasticity, gravity, air resistance, and the two wires that bound the flowing soap film. The incompressible viscous Navier-Stokes equations , which are used to describe the motion of the soap film and filament in our formulation, are discretized on a fixed uniform Eulerian lattice while the filament equations are discretized on a moving Lagrangian array of points which do not necessarily coincide with the fixed Eulerian mesh points of the fluid computation. The interaction between the filament and the soap film is handled by a smoothed approximation to the Dirac delta function. This delta function approximation is used not only to interpolate the fluid velocity and to apply force to the fluid (as is commonly done in immersed boundary computations), but also to handle the mass of the filament, which is represented in our calculation as delta function layer of fluid mass density supported along the immersed filament. Because of this nonuniform density, we use a multigrid (7-grid V-cycle) method for solving the discretized fluid equations. This replaces the FFT based method that is commonly used in the uniform-density case.

Our main numerical results are: 1)the sustained flapping of the filament only occurs when filament mass is included in the formulation of the model; within

a certain range of mass, the more mass of the filament the larger amplitude of the flapping. 2) When the length of filament is short enough (below some critical length), the filament always approaches its straight (rest) state in which the filament points downstream; but when the length is larger, the system is bi-stable, which means that it can settle into either state (rest state or sustained flapping) depending on the initial conditions. 3) The bi-stability of the film-filament system depends on the filament bending rigidity; the filament motion can be switched in either direction, from static to flapping, or from flapping to static, depending on whether the bending rigidity is decreased or increased sufficiently.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many problems in biofluid dynamics involve interactions between deformable elastic bodies and incompressible viscous fluids, for instance, blood flow in human heart, insect flight, swimming motions of eel, sperm and flagella. Unlike flow past a fixed rigid body or flow through a fixed pipe, where the flow does not change the shape and location of the boundary, flow involving a flexible elastic body will affect the position and shape of the immersed boundary by applying pressure and viscous shear stress to the interface separating the elastic body and the fluid. At the same time, the immersed body will exert action on the surrounding fluid through its inertia and elastic response and its surface immersed in the fluid will move with the fluid local velocity due to the viscosity of the fluid and the non-penetrable property of the body. Understanding of such interactions could lead to important applications in bionics. However this kind of coupling action between an elastic body and ambient fluid is very complicated and not yet well understood even in the passive case of a flag-in-wind. Readers who are interested in fluid-structure interaction can refer to [1] [2].

As a model of hydrodynamic interaction of deformable bodies with sur-

rounding fluid flows, Zhang [3] recently studied experimentally the dynamics of flexible filaments in a flowing soap film (See Fig. 2.1 in the next chapter.): separating at a nozzle attached to the bottom of a soapy water reservoir two thin nylon wires extend at a angle, then run parallel downwards and finally converge to a receiving container below. With the stopcock being turned on, which controls the rate of flow through the nozzle, under the actions of gravity and air resistance, a thin flowing soap film is formed on the two wires and reaches its terminal velocity soon. A flexible filament (thread) is introduced at the middle line of the two wires with the top end anchored by using a thin tube perpendicular to the soap film below the position where the film reaches its terminal velocity. Such a system (a filament in a thin film) is a two dimensional version of the flag-in-wind problem. In the past several decades, people commonly believed that flapping of a flag in the wind arises by a linear instability mechanism. However, recent experiments performed at the Courant Institute Wetlab by Zhang [3] have shown that the flexible filament in the flowing soap film is, under certain condition, bi-stable. In addition to that, the system itself (a free boundary problem) is very interesting: we have a one-dimensional immersed flexible moving boundary with one end tethered in a two-dimensional laminar flow, the boundary has mass and applies elastic forces (stretching, compression and bending) to the film and moves at the local film velocity. Vortices are shed from the free end of the filament and get carried away by the flow and are diffused by the film viscosity and air resistance. A vortex street is formed downstream.

As a result of the experiment, many people are interested in this problem. Currently Shelley [4] is working on the instability analysis of this system; Fast [5] is working on the simulation by the overset grid method. Here we report our

numerical simulation of such a system by the Immersed Boundary Method.

The Immersed Boundary Method has turned out to be a very practical and efficient way to simulate numerically the interaction of an elastic structure and surrounding fluid in the incompressible flow setting. The immersed boundary (or body) can be active like human heart and can be passive like a flexible filament; the boundary can be neutrally buoyant in the ambient fluid or not neutrally buoyant. The philosophy underlying the Immersed Boundary Method is that the whole system (fluid + elastic body) is treated as a incompressible composite continuum and an Eulerian description is used to describe its dynamics; in addition, a Lagrangian description is used to depict the motion of the immersed elastic body; the connections between Eulerian and Lagrangian variables are realized by the Dirac δ function. In general, the incompressible viscous Navier-Stokes equations with additional forcing term from the immersed boundary (and variable density resulting from the mass of the immersed body if it is not neutrally buoyant.) are used to govern the whole system and are discretized on an fixed Eulerian uniform grid, while the equations of the elastic structure are discretized on a collection of Lagrangian moving points which do not necessarily coincide with the fixed Eulerian mesh points. (Note that the shape and position of the immersed boundary is not known in advance and has to be determined as a part of the whole problem.) Thus, the Navier-Stokes solver does not need to know the shape and location of the moving elastic body; the influence of the latter has been taken into account by spreading its mass, if any, and force to the nearby fluid. The motion of immersed boundary is updated by the surrounding fluid velocity via interpolation because of the no-penetration and no-slip conditions at the boundary. The interaction between the fluid and the elastic body is mediated by a smoothed approximation to the Dirac δ func-

tion, δ_h, which plays an important role in the Immersed Boundary Method. This δ_h is used to spread the mass and force of the boundary to the fluid and to compute the new position of the boundary by interpolating its velocity from the surrounding fluid velocity field. The function δ_h is chosen such that the conservation of total mass and force, momentum and angular momentum are guaranteed when the mass and force of the boundary are transferred to fluid mesh points. The same δ_h is used to interpolate the velocity of the structure from the fluid, and this guarantees that the rate at which the immersed structure does work on the fluid is the same whether it is expressed in Lagrangian (structure) variables or in Eulerian (fluid) variables.

Since its birth in 1970's, the immersed Boundary Method has been applied successfully to a wide range of problems which involve an incompressible viscous fluid and an elastic deformable body, particularly in computational biofluid mechanics : blood flow in the human heart [6] [7] [8] [9] [10] [11] [12] [13] , the design of prosthetic cardiac valves [14], aquatic animal locomotion [15] [16] [17], wave propagation in the cochlea [18] [19], platelet aggregation during blood clotting [16] [20] , flow of suspensions [21] [22], valveless pumping [23], flow in a collapsible tube[24], flow and transport in a renal arteriole [25], cell and tissue deformation under shear flow [26] [27] [28]. At present there exist several versions of the Immersed Boundary Method. The version we use here is different from most existing versions [6] [8] [10] [12] [29] [31] in two aspects: 1) the discretization of the Navier-Stokers equations is different; the fractional step projection scheme is applied and the skew symmetrical scheme is used for the non-linear term instead of upwind differencing. 2) The numerical method to solve the resultant system of linear algebraic equations is different: a multigrid method (seven-grid V-cycle) is applied to solve the system of linear algebraic

equations with non-constant coefficients (therefore the FFT method is no longer applicable) which result from discretizing Navier-Stokers equations with variable density as a consequence of the mass of the filament. The first example (unpublished) of such a computation can be found in [32], here we report on another such example: simulation of a flapping flexible filament in a flowing soap film.

Our mathematical formulation includes filament mass and elasticity, gravity, air resistance, and the two wires that bound the flowing soap film. The incompressible viscous Navier-Stokes equations, which are used in our formulation to depict the motion of the whole system (soap film + filament), are discretized on a fixed uniform Eulerian lattice while the the filament equations are discretized on a moving Lagrangian array of points which do not necessarily coincide with the fixed Eulerian mesh points of the fluid computation. The interaction between the filament and the fluid (the soap film) is handled by a smoothed approximation to the Dirac delta function. This delta function approximation is used not only to interpolate the fluid velocity and to apply force to the fluid (as is commonly done in immersed boundary computations), but also to handle the mass of the filament, which is represented in our calculation as delta function layer of fluid mass density supported along the immersed filament. Because of this nonuniform density, we use a multigrid (seven-grid V-cycle) method for solving the discretized fluid equations. This replaces the FFT based method that is commonly used in the uniform-density case.

Our main numerical results are: 1)the sustained flapping of the filament only occurs when filament mass is included in the formulation of the model; within a certain range of mass, the more mass of the filament the bigger amplitude of the flapping. 2) When the length of filament is short enough (below some critical

length), the filament always approaches its straight state, but when the length is between some critical values, the system is bi-stable, which means that it can settle into either state (rest state or sustained flapping) depending on the initial conditions. 3) The bi-stability of the film-filament system depends on the filament bending rigidity; the filament motion can be switched in either direction, from static to flapping, or from flapping to static, depending on whether the bending rigidity is decreased or increased sufficiently.

This numerical result 2) agrees qualitatively very well with that of the experiment even though the Reynolds number of the computations (around 200) is lower than that of laboratory experiment (around 20,000) by two orders of magnitude.

Our numerical method used here can be generalized to the three dimensional case to study numerically problems involving interactions of fluids and immersed boundaries which are not neutrally buoyant, as is usually the case in aerodynamic problems such as flag-in-wind, insect flight, parachute simulation, etc.

The rest of the thesis is structured as following: Chapter 2 describes the physical problem, including the experiment setup, the experiment parameters and the main findings of the experiment. Chapter 3 presents our model problem abstracted from the real problem and the mathematical formulation of our model problem. Chapter 4 addresses the numerical methods used to discretize the system of partial differential-integral equations given in previous chapter. The simulation results are given in Chapter 5 and summary and conclusion are made in Chapter 6. At the end are the references.

Chapter 2

Physical Problem

In this chapter we talk about the experiment setup, the experiment parameters and main findings of the experiment.

Fig. 2.1 shows the setup of the laboratory experiment. The gravitational acceleration points from top to bottom. At the top of the setup, we have a reservoir containing soapy water. A stopcock is attached to the bottom of the container controlling the rate of the flow. Below, two thin nylon wires separate at the nozzle, extend downward under tension and then run parallel , and finally converge to a receiving container 240 cm below the nozzle. If the stopcock is opened, the soapy water will flow down under the action of gravity and will soon reach its terminal speed set by the balance of gravity and air resistance. As it falls, the soapy water forms a layer of thin film spanning the two wires. Halfway between the two wires a flexible filament (thread) is introduced with its upper end fixed at a place where the film has already reached its terminal speed (a distance of 80 cm from the nozzle is used in experiment). It is held there by a thin tube which is perpendicular to the film plane. The filament remains within the film plane because of surface tension.

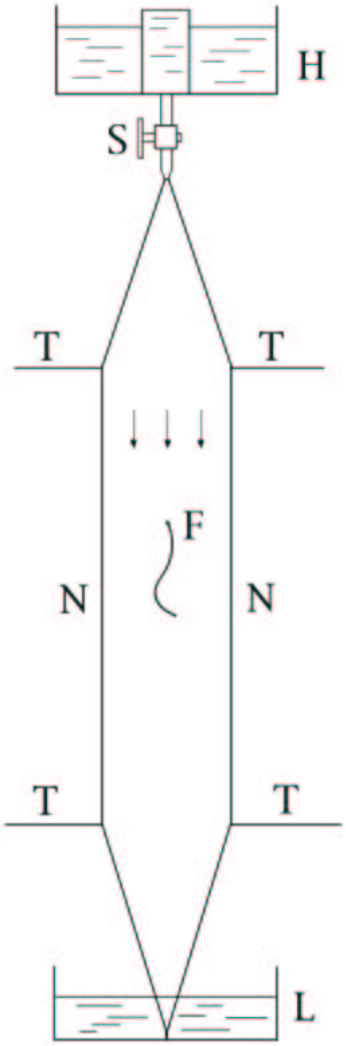

Figure 2.1: The laboratory experimental setup

Without the filament, the flowing soap film can be approximately treated as a two-dimensional channel flow which has a simple analytical solution. (Readers who are interested in the hydrodynamics of soap films can refer to [33] [34] [35] [36].) After the introduction of the filament, however, because of the complex interaction between the flexible filament and the flowing soap film, the motion of the system becomes very complicated and interesting: the flowing film exerts pressure and viscous force on the filament, and the filament acts back on the film through its inertia and elastic response and moves passively at the local film velocity. Vorticity is generated along the filament by the complex interacting force between the film and the filament and by film viscosity, and some of the vorticity is washed down along the filament by the flowing film. At the free end of the filament, vortices are shed and get advected by the flow, diffused by the film viscosity, and dissipated by air resistance. The sinuous flapping motion of the filament adds more complexity to the problem.

The experimental parameters are listed in 2.1.

The main experimental findings are: 1) there exists a critical length L_c, below which the filament always returns to its stretched straight state after a few oscillations no matter how large the external perturbation is. 2) When the filament length is greater than L_c, the filament-film system becomes bi-stable: two distinct stable dynamical states are found–a stretched straight state (the filament is motionless and aligned with the flow direction) and a sustained flapping state (the filament executes a sinuous motion). If the filament is in either of these two states and a small disturbance is applied, the filament will return to the state it was in before, but a large enough disturbance can switch it from one state to the other.

film inflow velocity	$200 - 300\, cm/sec$
film dynamic viscosity	$1.2 \times 10^{-7}\, g/(cm \cdot sec)$
film density	$3 \times 10^{-4}\, g/cm^2$
film thickness	$(3 - 4) \times 10^{-4}\, cm$
filament diameter	$0.015\, cm$
filament length	$2 - 6\, cm$
filament density	$2 \times 10^{-4}\, g/cm$
filament rigidity	$0.1\, erg \cdot cm$
width of the film	$8.5\, cm$
length of of the film	$120\, cm$

Table 2.1: Parameters of the laboratory experiment

Chapter 3

Mathematical Formulation

At the beginning of this chapter a model problem is abstracted from the real problem in the experiment, based on which our simulation is carried out. The mathematical formulation of this model problem is presented in the second part.

3.1 Model Problem

As we can see from Table 2.1, the filament diameter ($150\,\mu m$) is much greater than the film thickness ($3 - 4\,\mu m$), so the actual physical situation involves a three-dimensional filament moving in an almost two-dimensional film. See Fig. 3.1.

As we always do in going from the real world problem to a mathematical problem, we make assumptions. The main hypotheses we made for the filament-in-film problem are:

1) the filament is a one-dimensional curve without any volume, with mass uniformly distributed on this one-dimensional curve, which is totally immersed in the soap film. The curve behaves like a linear elastic material line which

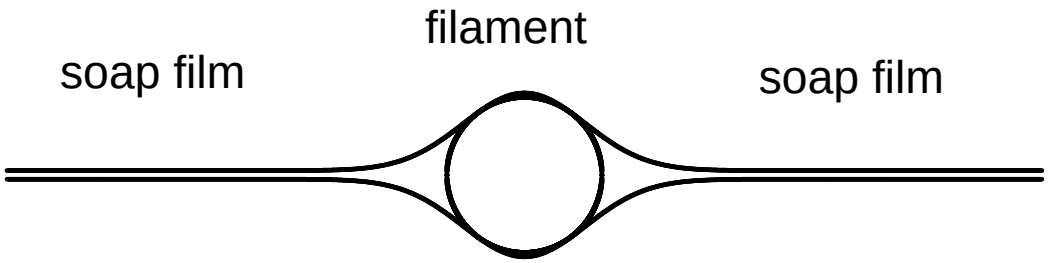

Figure 3.1: Overview of the film-filament system

can be stretched, compressed, and bent, and which resists these deformation by elastic forces.

2) the flowing soap film is an incompressible, viscous, Newtonian fluid membrane with no thickness which experiences no stretching or compression. The flow within the soap film is a two-dimensional plane laminar flow and the surface tension between the film and the filament is negligible.

3) the magnitude of air resistance is proportional to the film speed and the proportional factor is a constant in space and in time.

4)The actual mass of the filament is twice as much as that of a filament wet by soapy water. The extra mass comes from bulges in the film that form around the filament as a result of surface tension, which increase the effective filament mass. Note that in the laboratory experiment the film thickness is about $3\,\mu m$ while the diameter of the filament is about $150\,\mu m$. See Fig. 3.1 .

5)The filament-film system is not sensitive to the film length provided it is long enough compared with the filament length and the system is not sensitive to the outflow condition (the actual flow situation at the bottom of the film) provided the film bottom is far enough away from the free end of the filament not to interfere with the filament motion.

With these assumptions, our model problem is shown in Fig. 3.2, and our

mathematical formulation of the model problem is in the next section.

3.2 Mathematical Formulation of the Model Problem

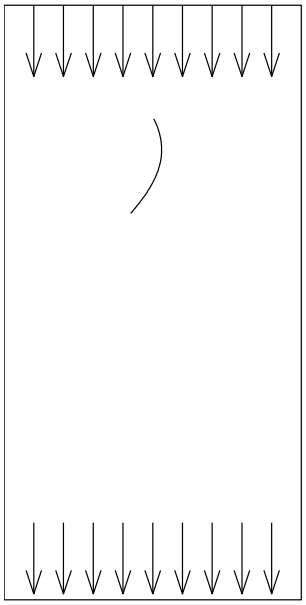

Figure 3.2: The model problem

We use an Eulerian description of the system (soap film and filament) as a whole supplemented by a Lagrangian description of the filament. The independent Eulerian variables are the Cartesian coordinates $\mathbf{x}$=(x,y) (x is the coordinate in horizontal direction and y is the coordinate in vertical direction.) and the time t, and the independent Lagrangian variables are the curvilinear material coordinate s and again the time t. The dependent Eulerian variables are the velocity $\mathbf{u}(\mathbf{x}, t)$, the pressure $p(\mathbf{x}, t)$, the density $\rho(\mathbf{x}, t)$, and the Eulerian

force density $\mathbf{f}(\mathbf{x},t)$. The reason $\rho(\mathbf{x},t)$ is not constant is that we use $\rho(\mathbf{x},t)$ to describe not only the mass density of the film but also the mass density contributed by the filament, see equation (3.5). The dependent Lagrangian variables are the position of the filament $\mathbf{X}(s,t)$, the Lagrangian force density $\mathbf{F}(s,t)$, and the filament velocity $\mathbf{U}(s,t)$. With this notation, the equations of motion of the film and filament system read as follows:

$$\rho(\mathbf{x},t)(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p+\mu\Delta\mathbf{u}+\mathbf{f}(\mathbf{x},t)-\lambda\mathbf{u}-\rho(\mathbf{x},t)g\hat{\mathbf{e}}_2 \tag{3.1}$$

$$\nabla\cdot\mathbf{u}=0 \tag{3.2}$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t)=\mathbf{U}(s,t) \tag{3.3}$$

$$\mathbf{f}(\mathbf{x},t)=\int \mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))ds \tag{3.4}$$

$$\rho(\mathbf{x},t)=\rho_0+\int M\delta(\mathbf{x}-\mathbf{X}(s,t))ds \tag{3.5}$$

$$\mathbf{U}(s,t)=\int \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x} \tag{3.6}$$

$$\mathbf{F}(s,t)=-\frac{\partial E}{\partial \mathbf{X}}=-\frac{\partial(E_{stretch}+E_{bend})}{\partial \mathbf{X}} \tag{3.7}$$

$$E_{stretch}=\frac{1}{2}K_{stretch}\int(|\frac{\partial \mathbf{X}}{\partial s}|-1)^2ds \tag{3.8}$$

$$E_{bend} = \frac{1}{2} K_{bend} \int |\frac{\partial^2 \mathbf{X}(s,t)}{\partial s^2}|^2 ds \tag{3.9}$$

These equations (without the viscous and air-resistance terms) can be derived formally from the principle of least action, see [10] for details. Equations (3.1) and (3.2) are the Navier-Stokes equations for incompressible, viscous, Newtonian fluids with variable density and multiple forcing terms: force applied by the immersed filament and the two nylon wires $\mathbf{f}$, air resistance $-\lambda \mathbf{u}$, and gravity $-\rho g \hat{\mathbf{e}}_2$. The constant ρ_0 is the mass per unit area of the soap film. (Note unusual units: this is a 2-D problem!) The constant g is the gravitational acceleration, the constant μ is the two-dimensional soap film viscosity, and λ is the air resistance coefficient, which is assumed to be constant and can be found by identifying $\lambda |\bar{V}_0| = \rho_0 g$, where $\bar{V}_0$ is the measured terminal velocity of the film as it falls under gravity. The reason we can estimate the air resistance coefficient in this simple way is that the film terminal velocity profile $V_0(x)$ is found to be almost flat (rather than a parabola) because of air resistance. The vector $\hat{\mathbf{e}}_2$ in equation (3.1) denotes the unit vector in the vertical direction (y direction). The function $\delta(\mathbf{x})$ is the Dirac δ function.

Equation (3.3) describes motion of the filament. The filament velocity is found from that of the film by equation (3.6). This is equivalent to saying that the filament moves at the local film velocity because of the no-slip and no-penetration boundary conditions.

The Eulerian force density $\mathbf{f}(\mathbf{x}, t)$ resulting from the immersed filament and the two wires bounding the film is connected to the Lagrangian force density $\mathbf{F}(s, t)$, which is defined on the filament, through the integral relation (3.4). The δ function in (3.4) is two-dimensional, but integration is done only once

with respect to s, so the resultant $\mathbf{f}(\mathbf{x},t)$ is singular along the filament like a one-dimensional δ function. However, the integration of $\mathbf{f}(\mathbf{x},t)$ over any finite region produces a finite value, which is equal to the total force over that part of the filament and/or the two wires happening to lie within the integration region. The mathematical explanation is the following. Let L represent the filament, let D be any bounded open subset of $\Re^2$, and note that $L \cap D$ denotes the part of the filament that lies in the integration region D. (See Fig. 3.3.)

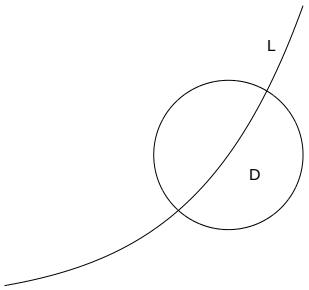

Figure 3.3: The integration over part of the filament

Then the total force exerted by the filament in the region D is

$$\begin{aligned}
\int_D \mathbf{f}(\mathbf{x},t)d\mathbf{x} &= \int_D \int_L \mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))dsd\mathbf{x} \\
&= \int_L \mathbf{F}(s,t)ds \int_D \delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x} \\
&= \int_{L\cap D} \mathbf{F}(s,t)ds
\end{aligned}$$

In the first step we make use of equation (3.4). Then we change the order of integration and use the fact that the integration of $\delta(\mathbf{x}-\mathbf{X}(s,t))$ is 1 or 0 depending on whether $\mathbf{X}(s,t)$ lies in D or not. Note that the last expression is just the force generated by the part of the filament that lies in the integration region. The result is the same for any integration over any part of the two nylon

wires.

The variable mass density in our problem comes from equation (3.5), which accounts for both the mass of the film and that of the immersed filament. Here M is the Lagrangian mass density of the filament (mass per unit length). [1] Like the Eulerian force density $\mathbf{f}(\mathbf{x},t)$, the Eulerian mass density $\rho(\mathbf{x},t)$ is also singular along the filament akin to a one-dimensional δ function supported only along the filament, for the same reason given above in the case of the Eulerian force. Also integration of $\rho(\mathbf{x},t)$ performed over any finite region gives a finite result: the total mass that lies within that region. Note that this includes a contribution from the filament and a contribution from the film. The mathematical explanation reads as follows. Here we use the same notation for the filament and the integration region as in the case of Eulerian force density $\mathbf{f}(\mathbf{x},t)$. See Fig. 3.3.

$$\begin{aligned}
\int_D \rho(\mathbf{x},t)d\mathbf{x} &= \int_D \rho_0 d\mathbf{x} + \int_D \int_L M\delta(\mathbf{x}-\mathbf{X}(s,t))dsd\mathbf{x} \\
&= \int_D \rho_0 d\mathbf{x} + \int_L Mds \int_D \delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x} \\
&= \int_D \rho_0 d\mathbf{x} + \int_{L\cap D} Mds
\end{aligned}$$

In the first step, we use Eq. (5), and the rest of the reasoning is the same as in the case of the Eulerian force. Note that the last expression is just the total mass of the part of the film and the part of the filament that lie in region D.

[1] In the general case of immersed boundary formulation, M is the density difference of the fluid and immersed structure. In our case, however, M is the mass density of the immersed filament. This is because in our problem the filament does not displace the film: it simply adds mass to the soap film.

Although the mass density $\rho(\mathbf{x},t)$ is variable, the fluid is still incompressible ($\nabla \cdot \mathbf{u} = 0$). This situation, although somewhat unusual, also arises in other contexts, such as stratified fluids with variable densities as a result of variable salinity in geophysical problems. In such problems, the mass density $\rho(\mathbf{x},t)$ is transported as a fluid marker, that is

$$\frac{D\rho(\mathbf{x},t)}{Dt} = \frac{\partial \rho(\mathbf{x},t)}{\partial t} + \mathbf{u}(\mathbf{x},t) \cdot \nabla \rho(\mathbf{x},t) = 0$$

We shall not explicitly use this equation because it is implied by equations (3.2) and (3.5). To show this, we proceed as follows: Let Ω be any bounded closed subset of $\Re^2$, let $\phi(\mathbf{x},t)$ denote any smooth function in Ω which vanishes on $\partial\Omega$. Then we have

$$\begin{aligned} \frac{\partial \rho(\mathbf{x},t)}{\partial t} &= - \int M(s) \nabla \delta(\mathbf{x} - \mathbf{X}(s,t)) \cdot \frac{\partial \mathbf{X}(s,t)}{\partial t} ds \\ &= - \int M(s) \nabla \delta(\mathbf{x} - \mathbf{X}(s,t)) \cdot \mathbf{u}(\mathbf{X}(s,t),t) ds \end{aligned}$$

In the first step we make use of equation (3.5) and the chain rule of differentiation, in the second step we use the equation (3.6). (The gradient on δ function is taken in the sense of weak derivative.) Multiplying by $\phi(\mathbf{x},t)$ on both sides and integrating over Ω, we get

$$\begin{aligned} \int_\Omega \frac{\partial \rho(\mathbf{x},t)}{\partial t} \phi(\mathbf{x},t) d\mathbf{x} &= - \int M(s) ds \int_\Omega \nabla \delta(\mathbf{x} - \mathbf{X}(s,t)) \cdot \mathbf{u}(\mathbf{X}(s,t),t) \phi(\mathbf{x},t) d\mathbf{x} \\ &= \int M(s) ds \int_\Omega \delta(\mathbf{x} - \mathbf{X}(s,t)) \mathbf{u}(\mathbf{X}(s,t),t) \cdot \nabla \phi(\mathbf{x},t) d\mathbf{x} \\ &= \int M(s) (\mathbf{u} \cdot \nabla \phi)(\mathbf{X}(s,t),t) ds \end{aligned}$$

In the second step we use integration by parts and the fact that $\phi(\mathbf{x},t)$ vanishes on $\partial\Omega$. In the third step we use the defining property of the δ function.

On the other hand, by using the equation (3.5) we have the following

$$\begin{aligned}\mathbf{u}(\mathbf{x},t)\cdot\nabla\rho(\mathbf{x},t) &= \mathbf{u}(\mathbf{x},t)\cdot\int M(s)\nabla\delta(\mathbf{x}-\mathbf{X}(s,t))ds \\ &= \int M(s)\mathbf{u}(\mathbf{x},t)\cdot\nabla\delta(\mathbf{x}-\mathbf{X}(s,t))ds\end{aligned}$$

Multiplying by $\phi(\mathbf{x},t)$ on both sides and integrating over Ω, we get

$$\begin{aligned}\int_\Omega \phi(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)\cdot\nabla\rho(\mathbf{x},t)d\mathbf{x} &= \int M(s)ds\int_\Omega \phi(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)\cdot\nabla\delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x} \\ &= -\int M(s)ds\int_\Omega \nabla\cdot(\phi(\mathbf{x},t)\mathbf{u}(\mathbf{x},t))\delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x} \\ &= -\int M(s)(\mathbf{u}\cdot\nabla\phi)(\mathbf{X}(s,t),t)ds\end{aligned}$$

In the second step we use the integration by parts, and the fact that $\phi(\mathbf{x},t)$ vanishes on $\partial\Omega$. In the third step, we use $\nabla\cdot\mathbf{u}(\mathbf{x},t)=0$ (equation (3.2)) and again the defining property of the δ function.

From the above two results, we immediately have

$$\int_\Omega \phi(\mathbf{x},t)(\frac{\partial\rho(\mathbf{x},t)}{\partial t}+\mathbf{u}(\mathbf{x},t)\cdot\nabla\rho(\mathbf{x},t))d\mathbf{x}=0$$

and then, since $\phi(\mathbf{x},t)$ is arbitrary, we have

$$\frac{D\rho(\mathbf{x},t)}{Dt}=\frac{\partial\rho(\mathbf{x},t)}{\partial t}+\mathbf{u}(\mathbf{x},t)\cdot\nabla\rho(\mathbf{x},t)=0$$

in the weak sense, as claimed above.

The Lagrangian force density $\mathbf{F}(s,t)$ is obtained by taking the Frechet derivative[2] of the elastic potential energy E, which consists of two terms: the energy

[2]Let E be a mapping, E: $Z\longrightarrow\Re^1$, where Z is an open subset of a Hilbert space $(H,(\cdot,\cdot))$, for any ξ in Z and any η in H such that $\xi+\eta$ is in Z, if $E(\xi+\eta)-E(\xi)$ can be written as $A(\eta)+o(||\eta||)$ and exists an element in H such that $A(\eta)=(\chi,\eta)$, then we say the Frechet derivative of E is χ , and write it as $\frac{\partial E}{\partial\xi}=\chi$.

associated with stretching and compression, $E_{stretch}$, and the energy associated with bending, E_{bend}. Equations (3.8) and (3.9) define the stretching energy and bending energy, respectively. The constant $K_{stretch}$ is filament stretching coefficient which is chosen in computation so that the filament has almost no stretch, and constant K_{bend} is the bending rigidity which was measured in the laboratory experiment. Equation (3.7) is essentially the principle of virtual work. Here we give another derivation as following:

In response to external force applied by the surrounding fluid, strain is resulted in the immersed elastic body, and internal stress is generated to resist the strain. The stress does work on the strain and the work is stored in the elastic body as elastic potential energy. Assuming the interacting process of the fluid and immersed body is adiabatic and noting that the elastic structure carries no mass (the mass of the immersed structure has already been transferred to nearby fluid), by the first law of thermodynamics, the work done by the fluid to the structure is equal to the change of the elastic potential energy, i.e.

$$\triangle E = \int \mathbf{F}_f \cdot d\mathbf{X} ds$$

$\mathbf{F}_f$ is the force density applied by fluid to structure (so the force applied by the structure to the fluid is $-\mathbf{F}_f$), E denotes the total potential energy of the structure. By definition of Frechet derivative, we have

$$-\mathbf{F}_f = -\frac{\partial E}{\partial \mathbf{X}}$$

Since $-\mathbf{F}_f$ is just the force applied by immersed body to fluid, this proves equation (3.7).

The initial velocity field for soap film without the filament is $\mathbf{u}(x, 0) = (0, V_0(x))$, where $V_0(x)$ is the film terminal velocity profile, which solves the

following boundary value problem:

$$\begin{cases} \mu V_{xx} - \lambda V - \rho_0 g & = 0 \\ V(a) = V(b) & = 0 \end{cases} \tag{3.10}$$

Where a and b are the x-coordinates of the position of the two wires. Equation (3.10) is obtained by setting $\mathbf{u} = (0, V_0(x))$, $\frac{\partial}{\partial t} = 0$, $p(\mathbf{x}, 0) = constant$ in the incompressible Navier-Stokers equations. The solution to (3.10) is:

$$V(x) = C_1 e^{r_1 x} + C_2 e^{r_2 x} - \bar{V}_0$$

where

$$r_1 = \sqrt{\frac{\rho g}{\bar{V}_0 \mu}}\ ,\ r_2 = -\sqrt{\frac{\rho g}{\bar{V}_0 \mu}}$$

$$C_1 = \frac{\bar{V}_0(e^{r_2 b} - e^{r_2 a})}{e^{r_1 a + r_2 b} - e^{r_1 b + r_2 a}}\ ,\ C_2 = \frac{\bar{V}_0(e^{r_1 a} - e^{r_1 b})}{e^{r_1 a + r_2 b} - e^{r_1 b + r_2 a}}$$

See Fig. 3.4 for the solution for two different sets of involved parameters.

The value of $\mathbf{X}(s, 0)$ is specified as initial condition for the filament and the boundary condition at the upstream end of the filament is that $\mathbf{X}(0, t)$ is constant, i.e. the target position [3] of $\mathbf{X}(0, t)$ is constant. The fluid velocity profile $(0, V_0(x))$ is specified at inflow and outflow, and the fluid velocity is equal to zero on the two side wires. The initial condition for the soap film is that the velocity field is given as $(0, V_0(x))$.

[3]The known position of the set of selected computational Lagrangian grid points, which are intended to describe the shape and location of the immersed boundary, is called the target position of the immersed boundary.

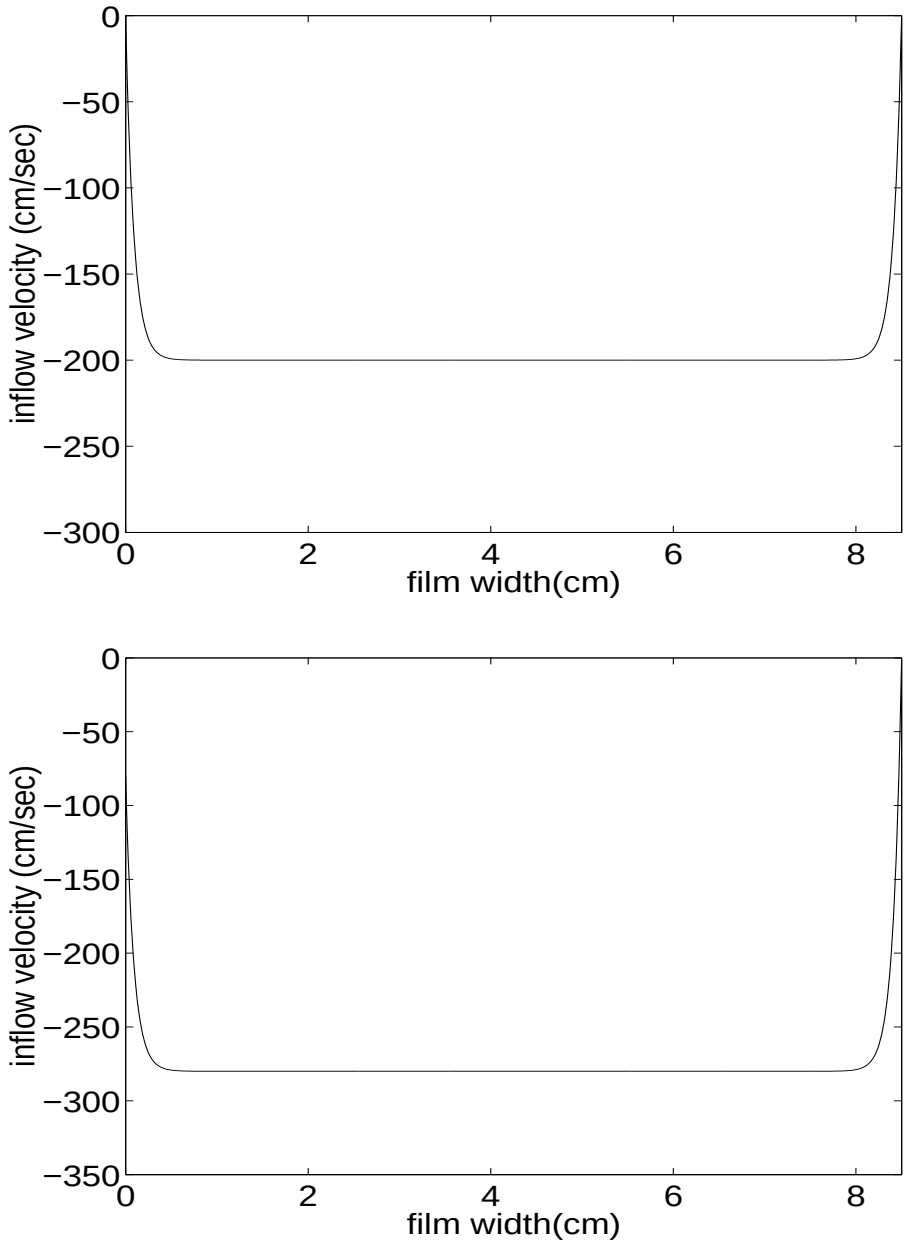

Figure 3.4: The inflow velocity profile; the film terminal velocity is $200 cm/sec$ on top and $280 cm/sec$ on bottom.

Chapter 4

Numerical Methods

In this chapter, we first address the discretizations of the above differential-integral equations in two different ways corresponding to two different situations—-whether the immersed boundary is neutrally buoyant or not; then we talk about the method to solve the resultant linear algebraic equations of variable coefficients—- the multigrid method (seven-grid V-cycle). Finally we compare the simulation results from two different discretizing methods for the nonlinear term in Navier-Stokes equations.

4.1 Discretizations of the differential-integral equations

The above system of differential-integral equations are numerically solved by the Immersed Boundary Method. As we mentioned in the Chapter 1, the incompressible viscous Navier-Stokes equations are discretized on a fixed uniform Eulerian lattice while the filament equations are discretized on a moving La-

grangian array of points which do not necessarily coincide with the fixed Eulerian mesh points of the fluid computation. The interaction between the filament and the fluid (the soap film) is handled by a smoothed approximation to the Dirac delta function. This delta function approximation is used not only to interpolate the fluid velocity and to apply force to the fluid (as is commonly done in immersed boundary computations), but also to handle the mass of the filament, which is represented in our calculation as delta function layer of fluid mass density supported along the immersed filament. The details are as follows:

Let Δt be the duration of the time step, let n be the time step index: $\mathbf{X}^n(s) = \mathbf{X}(s, n\Delta t)$, $\mathbf{u}^n = \mathbf{u}(\mathbf{x}, n\Delta t)$, $p^n = p(\mathbf{x}, n\Delta t)$, $\rho^n = \rho(\mathbf{x}, n\Delta t)$. Let the filament be represented by a discrete collection of points: $s = m\Delta s$, where m is an integer. The "half-integer" points are given by $s = (m + 1/2)\Delta s$. For any function $\phi(s)$, let

$$(D_s\phi)(s) = \frac{\phi(s + \frac{\Delta s}{2}) - \phi(s - \frac{\Delta s}{2})}{\Delta s} \tag{4.1}$$

Then the stretching energy and corresponding stretching force are discretized as the following:

$$E_{stretch} = \frac{1}{2}K_{stretch}\sum_m(|D_s\mathbf{x}| - 1)^2\Delta s = \frac{1}{2}K_{stretch}\sum_{m=1}^{n_f-1}(\frac{|\mathbf{X}_{m+1} - \mathbf{X}_m|}{\Delta s} - 1)^2\Delta s \tag{4.2}$$

$$(\mathbf{F}_{stretch})_l = \frac{K_{stretch}}{(\Delta s)^2}\sum_{m=1}^{n_f-1}(|\mathbf{X}_{m+1} - \mathbf{X}_m| - \Delta s)\frac{\mathbf{X}_{m+1} - \mathbf{X}_m}{|\mathbf{X}_{m+1} - \mathbf{X}_m|}(\delta_{ml} - \delta_{m+1,l}) \tag{4.3}$$

We discretize the bending energy and the corresponding bending force as follows:

$$E_{bend} = \frac{1}{2}K_{bend}\sum_m |D_sD_s\mathbf{X}|^2\Delta s = \frac{1}{2}K_{bend}\sum_{m=2}^{n_f-1}[\frac{|\mathbf{X}_{m+1} + \mathbf{X}_{m-1} - 2\mathbf{X}_m|^2}{(\Delta s)^4}]\Delta s \tag{4.4}$$

$$(\mathbf{F}_{bend})_l = \frac{K_{bend}}{(\Delta s)^4} \sum_{m=2}^{n_f-1} (\mathbf{X}_{m+1} + \mathbf{X}_{m-1} - 2\mathbf{X}_m)(2\delta_{ml} - \delta_{m+1,l} - \delta_{m-1,l}) \quad (4.5)$$

Where n_f is the total number of grid points of the filament, the δ_{kl} is the Kronecker symbol whose definition is:

$$\delta_{ml} = \begin{cases} 1, & \text{if } m = l, \\ 0, & \text{if } m \neq l. \end{cases}$$

Of course the sums in equations (4.3) and (4.5) can be simplified by making use of the Kronecker delta property, but it is actually better *not* to make this simplification but to compute the forces directly from equations (4.3) and (4.5) , since this avoids complications at the ends of the filament.

Note that the total Lagrangian force density $\mathbf{F}(s,t) = \mathbf{F}_{stretch}(s,t) + \mathbf{F}_{bend}(s,t)$. The three integral relations can be discretized as follows:

$$\mathbf{f}^n(\mathbf{x}) = \sum_s \mathbf{F}^n(s)\delta_h(\mathbf{x} - \mathbf{X}^n(s))\Delta s \quad (4.6)$$

$$\rho^n(\mathbf{x}) = \rho_0 + \sum_s M\delta_h(\mathbf{x} - \mathbf{X}^n(s))\Delta s \quad (4.7)$$

$$\mathbf{U}^{n+1}(s) = \sum_{\mathbf{x}} \mathbf{u}^{n+1}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^n(s))h^2 \quad (4.8)$$

Here the notation $\sum_s$ means the sum over all the discrete collection of points of the form $s = m\Delta s$, where m is integer. The notation $\sum_{\mathbf{x}}$ means the sum over all the discrete points of the form $\mathbf{x} = (ih, jh)$, where i and j are integers, h is meshwidth. The δ_h is a smoothed approximation of the two dimensional Dirac δ function. In our computation, we choose the following δ_h:

$$\delta_h(\mathbf{x}) = h^{-2}\phi(\frac{x}{h})\phi(\frac{y}{h}) \tag{4.9}$$

Where h is the mesh spacing, $\mathbf{x} = (x, y)$, and ϕ is chosen as:

$$\phi(r) = \begin{cases} \frac{1}{4}(1 + \cos(\frac{\pi r}{2})), & \text{if } |r| \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

See [8] for details regarding the choice of $\phi(r)$. Note that the support of δ_h is a square with width $4h$ at each point instead of a circle with diameter of $4h$. (See Fig. 4.1).

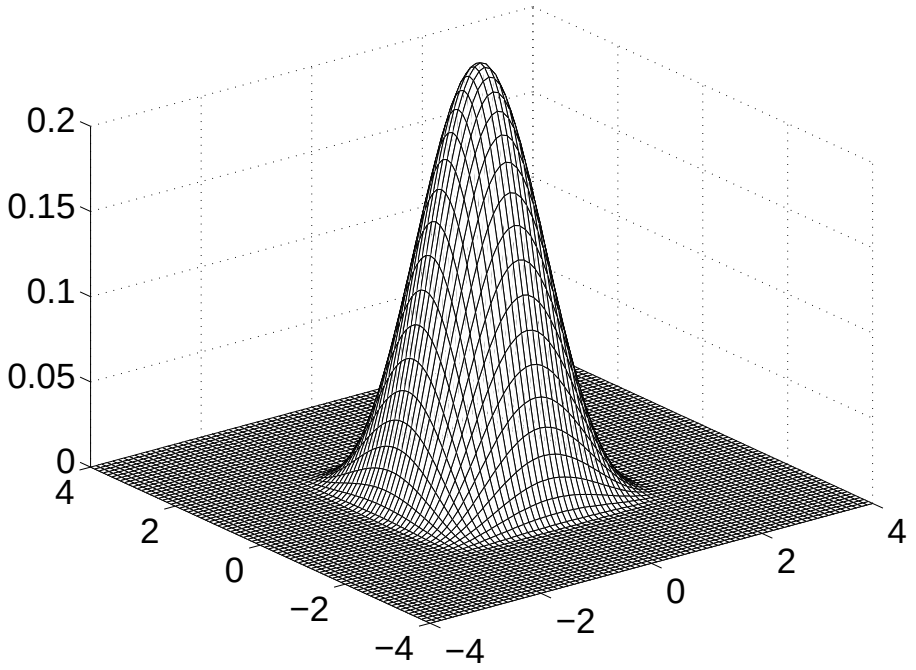

Figure 4.1: The graph of the two-dimensional smoothed delta function.

With $\mathbf{U}^{n+1}(s)$ known, the filament motion equations are discretized as follows:

$$\frac{\mathbf{X}^{n+1}(s) - \mathbf{X}^n(s)}{\Delta t} = \mathbf{U}^{n+1}(s) \tag{4.10}$$

Let the fluid velocity, pressure and density be defined on the square lattice of points $\mathbf{x} = \mathbf{k}h$, where h is the meshwidth and $\mathbf{k} = (i, j)$ is a vector with integer components. With $\mathbf{f^n}(\mathbf{x})$ and $\rho^n(\mathbf{x})$ defined, we can now handle the discertization of Navier-Stokes equations. First we state the following definitions:

$$(D_\alpha^0)(\mathbf{x}) = \frac{\phi(\mathbf{x} + h\hat{\mathbf{e}}_\alpha) - \phi(\mathbf{x} - h\hat{\mathbf{e}}_\alpha)}{2h} \tag{4.11}$$

$$(D_\alpha^+)(\mathbf{x}) = \frac{\phi(\mathbf{x} + h\hat{\mathbf{e}}_\alpha) - \phi(\mathbf{x})}{h} \tag{4.12}$$

$$(D_\alpha^-)(\mathbf{x}) = \frac{\phi(\mathbf{x}) - \phi(\mathbf{x} - h\hat{\mathbf{e}}_\alpha)}{h} \tag{4.13}$$

Where $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2\}$ is the standard basis of $\Re^2$, $\alpha = 1$ or 2. Thus $\mathbf{D}^0 = (D_1^0, D_2^0)$ is the central difference approximation to the gradient operator ∇, and $\sum_{\alpha=1}^{2} D_\alpha^+ D_\alpha^-$ is a 5-point difference approximation to the Laplace operator $\mathbf{\Delta}$.

There are many numerical methods for the incompressible Navier-Stokers equations; here we employ two different types of numerical schemes corresponding to whether the immersed structure is neutrally buoyant or not. If the immersed structure is neutrally buoyant, the discretization in [8] is used except for the nonlinear term, which is discretized here using the central difference scheme in the form of $\frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{u} + \nabla \cdot (\mathbf{u}\mathbf{u}))$, instead of the upwind scheme in the original form of $\mathbf{u} \cdot \nabla \mathbf{u}$. The discretization is as follows:

$$\rho(\frac{\mathbf{u}_k^{n+1} - \mathbf{u}_k^n}{\triangle t} + \frac{1}{2}(\mathbf{u} \cdot \mathbf{D}^0 \mathbf{u}_k + \mathbf{D}^0 \cdot (\mathbf{u}\mathbf{u}_k))^n) = -\mathbf{D}^0 p^{n+1} + \mu \sum_{\beta=1}^{2} D_\beta^+ D_\beta^- \mathbf{u}_k^{n+1} + \mathbf{f}_k^n - \lambda \mathbf{u}_k^n - \rho g \hat{\mathbf{e}}_2 \tag{4.14}$$

$$\mathbf{D}^0 \cdot \mathbf{u}^{n+1} = 0 \tag{4.15}$$

Where $\mathbf{u}_k$ is the k^{th} component of velocity $\mathbf{u}$ and similarly for the components of any vectors. The discrete fast Fourier transform can be used to solve the system of linear algebraic equations with constant coefficients defined by (4.14) and (4.15). Note that ρ is constant in this case since $M = 0$ is the case being considered here. To apply discrete FFT, first writing equation (4.14) in the form:

$$(I - \frac{\mu\Delta t}{\rho}\sum_{\beta=1}^{2} D_\beta^+ D_\beta^-)\mathbf{u}^{n+1} + \frac{\Delta t}{\rho}\mathbf{D}^0 p^{n+1} = \mathbf{v}^n \tag{4.16}$$

where I is an identity operator, and

$$\mathbf{v}^n = \mathbf{u}^n - \frac{1}{2}(\mathbf{u}\cdot\mathbf{D}^0\mathbf{u}_k + \mathbf{D}^0\cdot(\mathbf{u}\mathbf{u}_k))^n + \frac{\Delta t}{\rho}(\mathbf{f}^n - \lambda\mathbf{u}^n - \rho g\hat{\mathbf{e}}_2)$$

Note that the only difference between our case here and the case in [8] is now shown in the expression of $\mathbf{v}^n$, which is known at the beginning of each time step. Then the discrete FFT is applied on both side of (4.15) and (4.16); after some algebraic calculations, expressions of the velocity and pressure in Fourier space, $\hat{\mathbf{u}}^{n+1}$ and $\hat{p}^{n+1}$ are formulated. Finally, the inverse discrete FFT is applied to $\hat{\mathbf{u}}^{n+1}$ and $\hat{p}^{n+1}$, and the velocity and pressure at time $(n+1)\Delta t$ in physical space are recovered. See [8] for details.

If the immersed structure is not neutrally buoyant, the FFT is no longer applicable because $\rho(\mathbf{x}, t)$ is a variable coefficient. One could still consider the scheme given by equations (4.14)-(4.15), with some iterative method used to solve the variable-coefficient Stokes system. Instead of this approach, we apply a projection method [37] [38], which is a fractional step scheme. (Our scheme differs from the original projection method in the treatment of the nonlinear term, which is explicit and skew-symmetric.) First an intermediate velocity

field $\tilde{\mathbf{u}}(\mathbf{x},t)$ is introduced which is not divergence free and is the solution to the following difference equations:

$$\rho^n(\frac{\tilde{\mathbf{u}}_k^{n+1}-\mathbf{u}_k^n}{\triangle t}+\frac{1}{2}(\mathbf{u}\cdot\mathbf{D}^0\mathbf{u}_k+\mathbf{D}^0\cdot(\mathbf{u}\mathbf{u}_k))^n)=\mu\sum_{\beta=1}^{2}D_\beta^+D_\beta^-\tilde{\mathbf{u}}_k^{n+1}+\mathbf{f}_k^n-\lambda\tilde{\mathbf{u}}_k^{n+1}-\rho^n g\hat{\mathbf{e}}_2 \tag{4.17}$$

for $k=1$ and 2; where, again $\mathbf{u}_k$ is the k^{th} component of velocity $\mathbf{u}$ and similarly for the components of any vectors. Note that for incompressible flow, the convection term $\mathbf{u}\cdot\nabla\mathbf{u}$ can be rewritten as $\frac{1}{2}(\mathbf{u}\cdot\nabla\mathbf{u}+\nabla\cdot(\mathbf{u}\mathbf{u}))$ in which form we discretize it. (We also used the upwind scheme for the nonlinear term in its original form $\mathbf{u}\cdot\nabla\mathbf{u}$, and the simulation result is given in contrast with that from skew-symmetric scheme at the end of this chapter.)

Then with $\tilde{\mathbf{u}}^{n+1}$ known, we update (project) the velocity field with the pressure gradient and make the velocity be divergence free.

$$\rho^n(\frac{\mathbf{u}^{n+1}-\tilde{\mathbf{u}}^{n+1}}{\Delta t})=-\mathbf{D}^0p^{n+1} \tag{4.18}$$

$$\mathbf{D}^0\cdot\mathbf{u}^{n+1}=0 \tag{4.19}$$

Note the variable coefficient ρ^n. Equations (4.18)-(4.19) define an orthogonal projection in the norm which uses ρ as a weight function. To solve for pressure, we divide both sides of equation (4.18) by ρ^n and then apply the central difference operator $\mathbf{D}^0$, and finally use the divergence free condition (4.19). Thus we obtain a system of difference equations for the unknown pressure p^{n+1} which is decoupled from the unknown velocity field $\mathbf{u}^{n+1}$.

$$\mathbf{D}^0\cdot(\frac{1}{\rho^n}\mathbf{D}^0p^{n+1})=\frac{\mathbf{D}^0\cdot\tilde{\mathbf{u}}^{n+1}}{\Delta t} \tag{4.20}$$

Because of the use of the central difference operator $\mathbf{D}^0$, equation (4.20) give us four separate systems of linear algebraic equations with variable coefficients, each of which resembles the one generated by a 5-point scheme for Poisson's equation. Now comes the question of how to solve numerically the systems of difference equations (4.17) and (4.20), both of which contain the non-constant coefficient ρ^n (so FFT will not work any longer). Instead we use another efficient technique—the multigrid method [39] [40] [41] to solve these equations. First equations (4.17) are solved for $\tilde{\mathbf{u}}^{n+1}$, and then with $\tilde{\mathbf{u}}^{n+1}$ in hand, equation (4.20) is solved for p^{n+1} . Finally the velocity field $\mathbf{u}^{n+1}$ is calculated from equations (4.18). This completes the computations at each time step.

4.2 Solution of the discretized differential-integral equations

The multigrid method solves a problem on a series of gradually coarsened grids instead of on a single grid. The basic idea of the multigrid method can be explained as follows in a simple setting: the two-grid V-cycle. Suppose we have a linear scalar partial differential equation

$$Lu = f$$

and

$$L_h u_h = f_h$$

is a discrete form of $Lu = f$ on a grid with spacing h by any numerical method (finite difference, finite element, or finite volume). Let u_h and $\tilde{u}_h$ be the exact solution and computed solution to $L_h u_h = f_h$, respectively. Define the residual

as

$$r_h = f_h - L_h \tilde{u}_h$$

Define the error as

$$e_h = u_h - \tilde{u}_h$$

It is easy to see that the error satisfies the following equation (residual equation)

$$L_h e_h = r_h$$

(Since the PDE is linear.) The key idea of multigrid method is that the residual equation is solved on the coarse grid rather than on the original fine grid. To be specific, let the fine grid and coarse grid be Ω^h and Ω^H, respectively. Generally, the spacing of coarse grid H is twice as wide as that of the fine gird. Then the two-grid multigrid method goes as follows:

- Solve $L_h u_h = f_h$ on Ω^h by iterating ν_1 times with initial guess $\tilde{u}_h$; the computed solution is also denoted by $\tilde{u}_h$
- Calculate the residual $r_h = f_h - L_h \tilde{u}_h$ on Ω^h
- Restrict the residual r_h from fine grid Ω^h to coarse grid Ω^H: $r_H = R_h^H r_h$
- Find the exact solution to the residual equation $L_H e_H = r_H$ on the coarse grid Ω^H, denoted by e_H (called coarse grid correction)
- Find the correction on fine grid Ω^h by interpolating e_H from Ω^H to Ω^h: $e_h = I_H^h e_H$
- Let $\tilde{u}_h \leftarrow \tilde{u}_h + e_h$, iterate $L_h u_h = f_h$ on fine grid Ω^h again for ν_2 times, with initial guess $\tilde{u}_h$,the corrected computed solution is still denoted by $\tilde{u}_h$

- Check for convergence, i.e. check whether the following condition is satisfied:

$$\frac{||r_h||_{L_2}}{||f_h||_{L_2}} \leq \epsilon$$

, which says the relative residual in L_2 norm is less than or equal to a prescribed tolerance ϵ. If it is satisfied, we think $L_h u_h = f_h$ is solved by the computed approximation $\tilde{u}_h$; if not, go to the first step to start another V cycle.

The parameters ν_1 and ν_2 are given fixed constants depending on the specific problem. The operator R_h^H is the restriction operator from the fine grid to the coarse grid (transferring data from a fine grid to the next coarse grid). The operator I_H^h is the interpolation operator from the coarse grid to the fine grid (transferring data from a coarse grid to the next fine grid). The operator L_H is the coarsening operator, which defines the coarse grid problem (the version of the original problem on the coarse grid). There are as many coarsening methods as discretization schemes themselves (discretization of the original problem on the finest grid). Usually, the coarse grid problem is the discretization of the original problem on the coarse grid by the same discretizing method used for discretization on the finest grid. See Fig. 4.2 for the illustration of a two-grid V-cycle, where $\bullet$ stands for iteration, $\circ$ stands for finding exact solution. Because of the pattern in the diagram, this algorithm is commonly called V-cycle.

For multigrid with more than two grids, the algorithm is basically the same as the two-grid algorithm except that rather than finding the exact solution of the coarse grid problem $L_H e_H = r_H$ (r_H is also denoted by f_H) on Ω^H we iterate on $L_H e_H = f_H$ for ν_1 times, compute the residual, and move on to an even coarser grid Ω^{4h}, and the process is repeated till the coarsest grid is

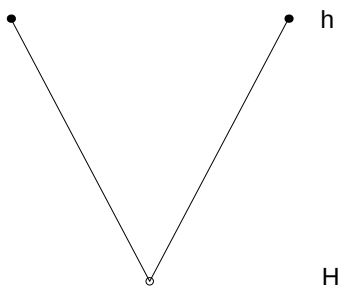

Figure 4.2: The two grid V-cycle

reached, where exact solution is found. The resolution of the finest grid is determined by the need for resolution in the problem, and the resolution of the coarsest grid is such that the problem on that grid can be easily solved exactly. (One possibility is that the coarsest gird contains only a single interior point, but this is not used here.) Suppose we have n grids with the spacing $h, 2h, 4h, \cdots, Nh$, where $N = 2^{n-1}h$, let Ω^{lh} denote the grid whose spacing is lh, with $l = 2, 4, 8, \cdots, 64$, we denote the problem on Ω^{lh} by

$$L_{lh}e_{lh} = f_{lh}$$

, (The problem on Ω^h is $L_h u_h = f_h$.) where f_{lh} is the right hand side of the problem on Ω^{lh}. Then the n-grid multigrid V-cycle reads as follows:

- Iterate on $L_h u_h = f_h$ for ν_1 times with initial guess $\tilde{u}_h$ on Ω^h; after iteration the approximated solution is still denoted by $\tilde{u}_h$
- Compute the residual $r_h = f_h - L_h\tilde{u}_h$ on Ω^h
- Restrict the residual r_h from Ω^h to Ω^{2h}: $f_{2h} = R_h^{2h} r_h$ (note the residual is now denoted by f_{2h})
- Iterate on $L_{2h}e_{2h} = f_{2h}$ on Ω^{2h} with initial guess $\mathbf{0}$ (the zero vector) for ν_1

times; the approximated solution to the residual equation is denoted by $\tilde{e}_{2h}$

- Compute the residual on Ω^{2h}: $r_{2h} = f_{2h} - L_{2h}\tilde{e}_{2h}$
- Restrict the residual r_{2h} from Ω^{2h} to Ω^{4h}: $f_{4h} = R_{2h}^{4h} r_{2h}$
- Iterate on $L_{4h}e_{4h} = f_{4h}$ on Ω^{4h} with initial guess $\mathbf{0}$ for ν_1 times; the approximated solution to the equation is denoted by $\tilde{e}_{4h}$
- Compute the residual on Ω^{4h}: $r_{4h} = f_{4h} - L_{4h}\tilde{e}_{4h}$
- Restrict the residual r_{4h} from Ω^{4h} to Ω^{8h}: $f_{8h} = R_{4h}^{8h} r_{4h}$

$\bullet\bullet\bullet$

- Find the exact solution to $L_{Nh}e_{Nh} = f_{Nh}$ on the coarsest grid Ω^{Nh}, denote it by $\bar{e}_{Nh}$
- Transfer $\bar{e}_{Nh}$ back from Ω^{Nh} to $\Omega^{\frac{N}{2}h}$ by interpolation: $\bar{e}_{\frac{N}{2}h} = I_{Nh}^{\frac{N}{2}h}\bar{e}_{Nh}$
- Correct $\tilde{e}_{\frac{N}{2}h} \leftarrow \tilde{e}_{\frac{N}{2}h} + \bar{e}_{\frac{N}{2}h}$ and iterate on $L_{\frac{N}{2}h}e_{\frac{N}{2}h} = f_{\frac{N}{2}h}$ for ν_2 times with initial guess $\tilde{e}_{\frac{N}{2}h}$, the new computed solution is still denoted by $\tilde{e}_{\frac{N}{2}h}$

$\bullet\bullet\bullet$

- Transfer $\tilde{e}_{8h}$ back from Ω^{8h} to Ω^{4h} : $\bar{e}_{4h} = I_{8h}^{4h}\tilde{e}_{8h}$
- Correct $\tilde{e}_{4h} \leftarrow \tilde{e}_{4h} + \bar{e}_{4h}$; iterate on $L_{4h}e_{4h} = f_{4h}$ on Ω^{4h} with initial guess $\tilde{e}_{4h}$ for ν_2 times; the new computed solution is still denoted by $\tilde{e}_{4h}$
- Transfer $\tilde{e}_{4h}$ back from Ω^{4h} to Ω^{2h}: $\bar{e}_{2h} = I_{4h}^{2h}\tilde{e}_{4h}$
- Correct $\tilde{e}_{2h} \leftarrow \tilde{e}_{2h} + \bar{e}_{2h}$; iterate on $L_{2h}e_{2h} = f_{2h}$ on Ω^{2h} with initial guess $\tilde{e}_{2h}$ for ν_2 times, the new computed solution is still denoted by $\tilde{e}_{2h}$

- Transfer $\tilde{e}_{2h}$ back from Ω^{2h} to Ω^h: $\bar{e}_h = I_{2h}^h \tilde{e}_{2h}$

- Correct $\tilde{u}_h \leftarrow \tilde{u}_h + \bar{u}_h$; iterate on $L_h u_h = f_h$ on Ω^h for ν_2 times with initial guess $\tilde{u}_h$; the new computed solution is still denoted by $\tilde{u}_h$

- Check for convergence i.e. check whether the following condition is satisfied:

$$\frac{||r_h||_{L_2}}{||f_h||_{L_2}} \leq \epsilon$$

, which says the relative residual in L_2 norm is less than or equal to a prescribed tolerance ϵ. If it is satisfied, we think $L_h u_h = f_h$ is solved by the computed approximation $\tilde{u}_h$; if not, go to the first step to start another V cycle.

The above n-grid V-cycle can also be written in a compact recursive form. First we need to make some simplification of our notation. We can treat the residual equation $L_h e_h = r_h$ and the original equation $L_h u_h = f_h$ as a single equation, and use f_h to denote the right hand side of the residual equation, rather than r_h, since it is just another right hand side vector. We call the exact solution of residual eqution u_h instead of e_h, since it is just another exact solution. Rather than using $\tilde{e}_h$ to represent the computed approximation to the residual equation, we call it $\tilde{u}_h$ since it is just another computed approximation. Furthermore, we use the following notation

$$\tilde{u}_h \leftarrow V_h(\tilde{u}_h, f_h)$$

to denote a V-cycle beginning on Ω^h with initial guess $\tilde{u}_h$ and ending on Ω^h with the computed solution, again denoted by $\tilde{u}_h$ (after correction and relaxation on Ω^h). Then the recursive specification of the V-cycle algorithm is as follows:

Recursive definition of $\tilde{u}_h \leftarrow V_h(\tilde{u}_h, f_h)$

1. Iterate ν_1 times on $L_h u_h = f_h$ on Ω^h with given initial guess $\tilde{u}_h$

2. **If** Ω^{2h} is the coarsest grid,

   $f_{2h} \leftarrow R_h^{2h}(f_h - L_h \tilde{u}_h)$

   find the exact solution of $L_{2h} u_{2h} = f_{2h}$

   Else

   $f_{2h} \leftarrow R_h^{2h}(f_h - L_h \tilde{u}_h)$

   $\tilde{u}_{2h} \leftarrow \mathbf{0}$

   $\tilde{u}_{2h} \leftarrow V_{2h}(\tilde{u}_{2h}, f_{2h})$

   End If

3. Correct: $\tilde{u}_h \leftarrow \tilde{u}_h + I_{2h}^h \tilde{u}_{2h}$

4. Iterate ν_2 times on $L_h u_h = f_h$ with initial guess $\tilde{u}_h$

5. Check for convergence on the finest grid, i.e. check whether the following condition is satisfied:

$$\frac{||r_h||_{L_2}}{||f_h||_{L_2}} \leq \epsilon$$

, which says the relative residual in L_2 norm is less than or equal to a prescribed tolerance ϵ. If it is satisfied, we think $L_h u_h = f_h$ is solved by the computed approximation $\tilde{u}_h$; if not, go to the first step to start another V cycle.

In our computation a 7-grid V-cycle is used with the finest grid 256×512 and the coarsest grid 4×8. See Fig. 4.3 for the illustration of a seven-grid V-cycle. Again, the $\bullet$ represents iteration, the $\circ$ represents finding the exact solution. The left branch (descending from Ω^h to Ω^{64h}) is restriction and the right branch (ascending from Ω^{64h} to Ω^h) is interpolation. The labels on the left hand side give the mesh spacing on each grid level and those on the right hand side show the mesh points used on each grid.

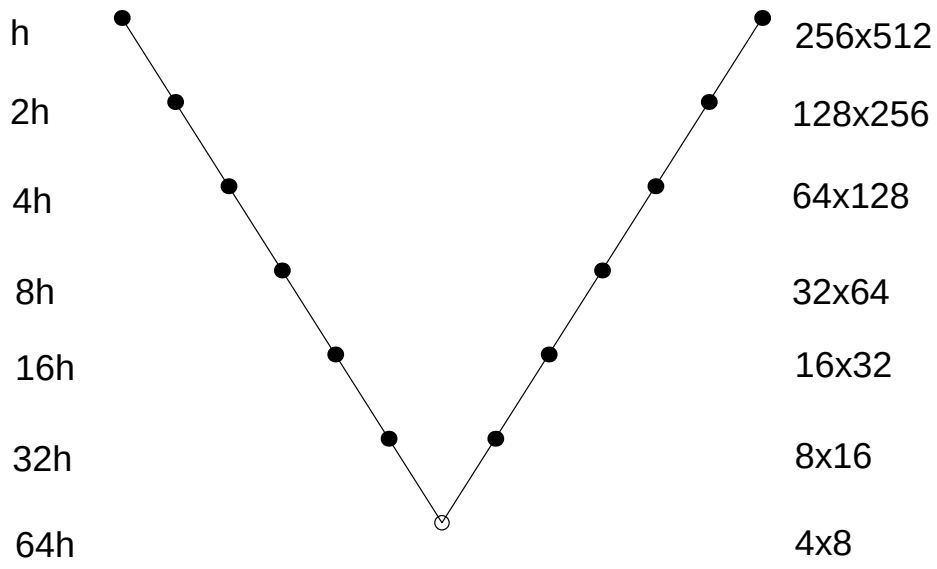

Figure 4.3: The 7-grid V-cycle used in our simulation.

The red-black Gauss-Seidel method is used for iteration.[1] A red point is a point whose index sum is even and a black point is a point whose index sum is odd (assuming that the index corresponding to a boundary in x direction is zero, and the same for y direction). First, all the red points are updated simultaneously by using the old values on black points, then all the black points are updated simultaneously by using the newly-computed values on red points.

The full-weighting scheme is applied for residual restriction from a fine grid to the next coarse grid. For any point on a coarse grid, the full weighting is actually an average of values at the nearest points on the next fine grid centered at the corresponding point on the fine grid. Put mathematically, suppose the fine grid is Ω^{lh} ,the coarse grid is Ω^{2lh},here $l = 1, 2, 4, 8, ..., 64$, let w_{ij}^{2lh} denote the value of a function $w(x, y)$ at grid point $(i\Delta x, j\Delta y)$ on grid Ω^{2lh}, similar for

[1]In the case of the pressure equations the red-black ordering is applied separately to each of the four subgrids on which the system of equations for the pressure is defined.

the notation w_{ij}^{lh}. The full-weighting scheme means, for any point indexed by (i,j) on coarse grid Ω^{2lh},

$$\begin{aligned} w_{ij}^{2lh} &= \frac{1}{4}w_{2i,2j}^{lh} + \frac{1}{8}(w_{2i,2j-1}^{lh} + w_{2i,2j+1}^{lh} + w_{2i-1,2j}^{lh} + w_{2i+1,2j}^{lh}) \\ &+ \frac{1}{16}(w_{2i-1,2j-1}^{lh} + w_{2i-1,2j+1}^{lh} + w_{2i+1,2j-1}^{lh} + w_{2i+1,2j+1}^{lh}) \end{aligned}$$

The linear interpolation is employed to transfer data back from a coarse grid to the next fine grid. Using the same notation as above, the linear interpolation is defined as follows:

$$\begin{aligned} w_{2i,2j}^{lh} &= w_{i,j}^{2lh} \\ w_{2i+1,2j}^{lh} &= \frac{1}{2}(w_{i,j}^{2lh} + w_{i+1,j}^{2lh}) \\ w_{2i,2j+1}^{lh} &= \frac{1}{2}(w_{i,j}^{2lh} + w_{i,j+1}^{2lh}) \\ w_{2i+1,2j+1}^{lh} &= \frac{1}{4}(w_{ij}^{2lh} + w_{i+1,j}^{2lh} + w_{i,j+1}^{2lh} + w_{i+1,j+1}^{2lh}) \end{aligned}$$

QR factorization is utilized to find exact solution on the coarsest grid.

We want to point out that using the simple injection [2] for density in transferring $\rho^n(\mathbf{x})$ from Ω^h (the finest grid) to Ω^{lh} $(l = 2, 4, 8, 16, 32, 64)$ results in a rather slowly convergent multigrid algorithm; instead, it is much better to use the following way to define ρ_{lh}^n, the density on Ω^{lh} at time n:

$$\rho_{lh}^n(\mathbf{x}) = \rho_0 + \sum_s M\delta_{lh}(\mathbf{x} - \mathbf{X}^n(s))\Delta s \tag{4.21}$$

Here lh means the grid whose meshwidth is l times h, where h is the meshwidth of the finest grid. The δ_{lh} means the smoothed approximation to the

[2] Injection means a point on a coarse grid takes the same value as the corresponding point on the next fine grid.

Dirac delta function on a grid with spacing lh. The reason is that in the case of using simple injection, the discretized Navier-Stokers equations on a sufficiently coarse grids does not "feel" the existence of the filament, thus the coarse grid correction does not help very much in accelerating convergence. Note that the smoothed approximation of delta function in equation (4.21) has width $4lh$, that is, it gets wider as the grid is coarsened. This ensures that the mass of filament is well represented on each level grid. Except for this important detail about the width of the delta function being adjusted to the grid level, our multigrid method is standard, see [39] [40] [41] for detail. It is also worthwhile to note that equation (4.20) actually involves four decoupled subsystems of linear algebraic equations, which will be solved separately by the multigrid method.

The last issue we want to address in this chapter is the duration of time step. In our computation the following two constraints are respected in choosing the time step Δt while the spacing h is given.

$$\Delta t \leq \frac{h}{max\{u^n(\mathbf{x}), v^n(\mathbf{x})\}} \tag{4.22}$$

$$\Delta t \leq \sqrt{\frac{8h}{3K_{stretch}}} \tag{4.23}$$

The inequality (4.22) ensures that within each time step, the fluid at a grid point will not flow out of a box with width $2h$ centered at that point. The inequality (4.23) is taken from [30], which was derived from a one-dimensional model problem for studying the Immersed Boundary Method. This ends the description of our numerical method.

4.3 Upwind vs skew-symmetric for nonlinear term

Before we start the next chapter (the simulation results), we want to make a comparison of the simulation results from using upwind scheme for the nonlinear term $\mathbf{u} \cdot \nabla \mathbf{u}$ in the Navier-Stokes equations and using skew-symmetric scheme for the nonlinear term. We found that the skew-symmetric scheme gives much better results. Fig. 4.4 and Fig. 4.5 show the simulation results at two different times; Fig. 4.6 shows the peak-to-peak amplitude of the filament free end against time. The inflow velocity is $280\, cm/sec$, the filament length is $3\, cm$. Note that the vortices are better defined and that the amplitude of the filament motion is larger with the skew-symmetric scheme. The clear contrast in these figures shows that, compared to the skew-symmetric scheme the upwind scheme introduces much more numerical viscosity, which may help stablize the computation, but results in more non-physical energy dissipation and therefore less action of the flow. On the contrary, the skew-symmetric scheme has the desirable property that $\frac{d}{dt}||\mathbf{u}||_{L_2}^2 = 0$, (See [30]) so the kinetic energy of the soap film is conserved, and as a result, more detailed flow structure is captured.

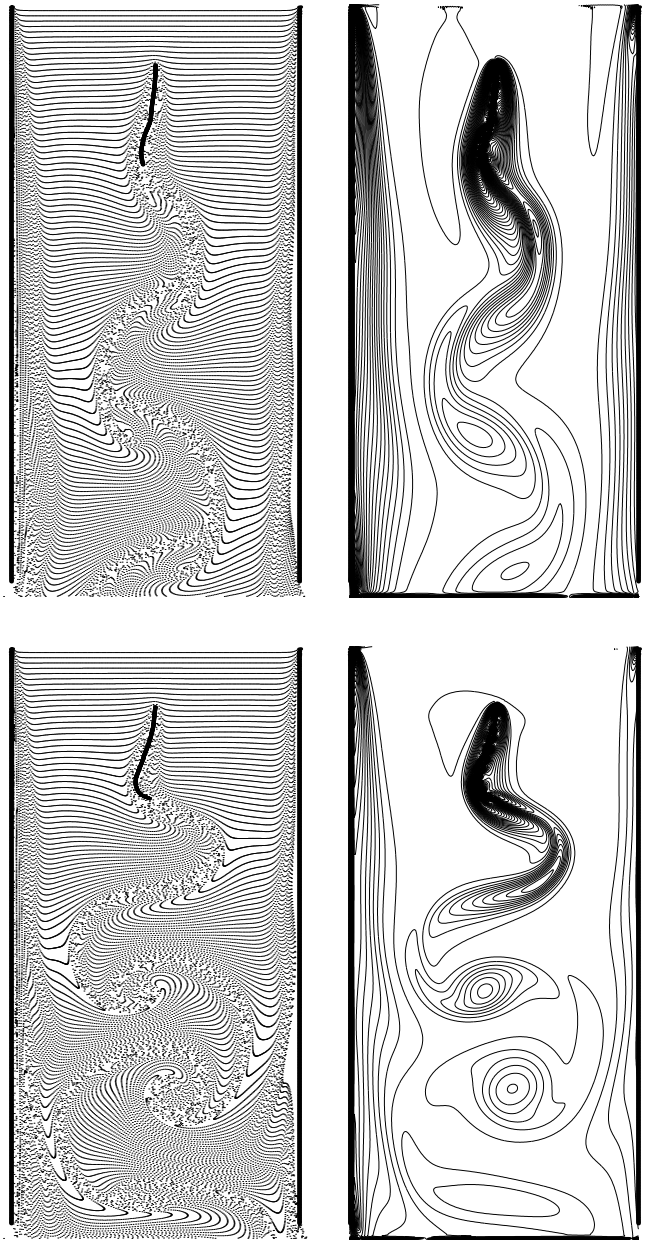

Figure 4.4: Computation results using two different discretizations for the nonlinear term: upwind scheme (top panel) and skew-symmetric scheme (bottom panel). The inflow velocity is $280\,cm/sec$,the filament length is $3\,cm$, $time = 0.1825\ sec$.

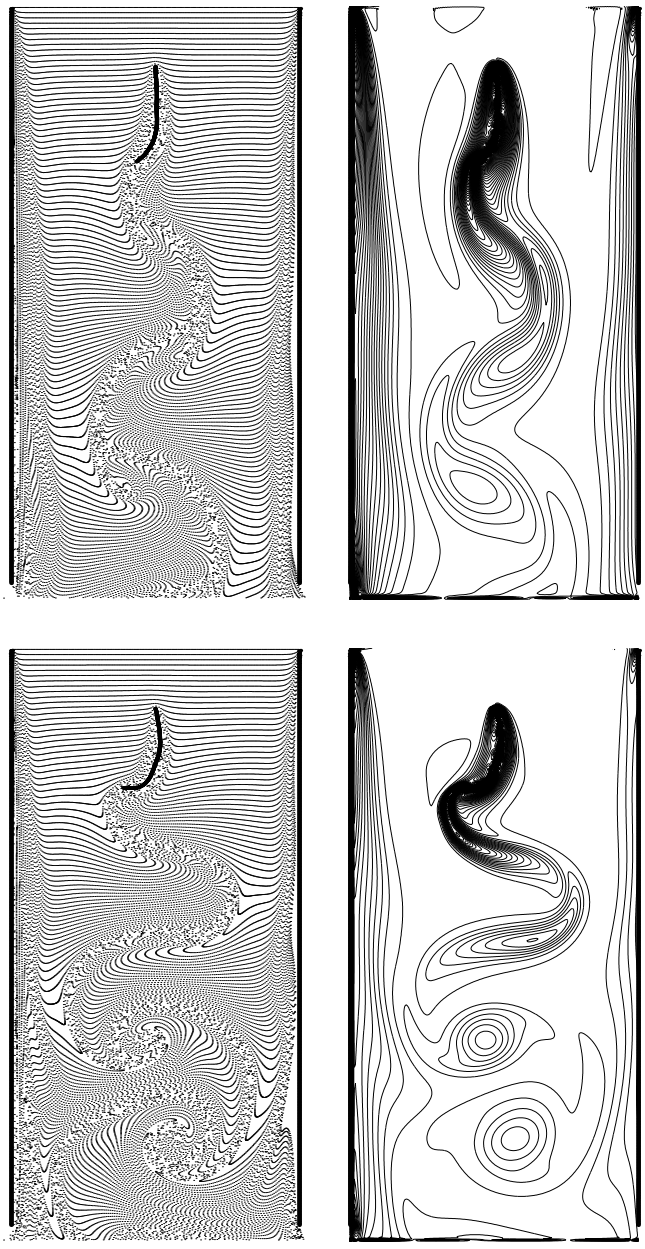

Figure 4.5: Continued from Fig. 4.4, $time = 0.1875$ *sec.*

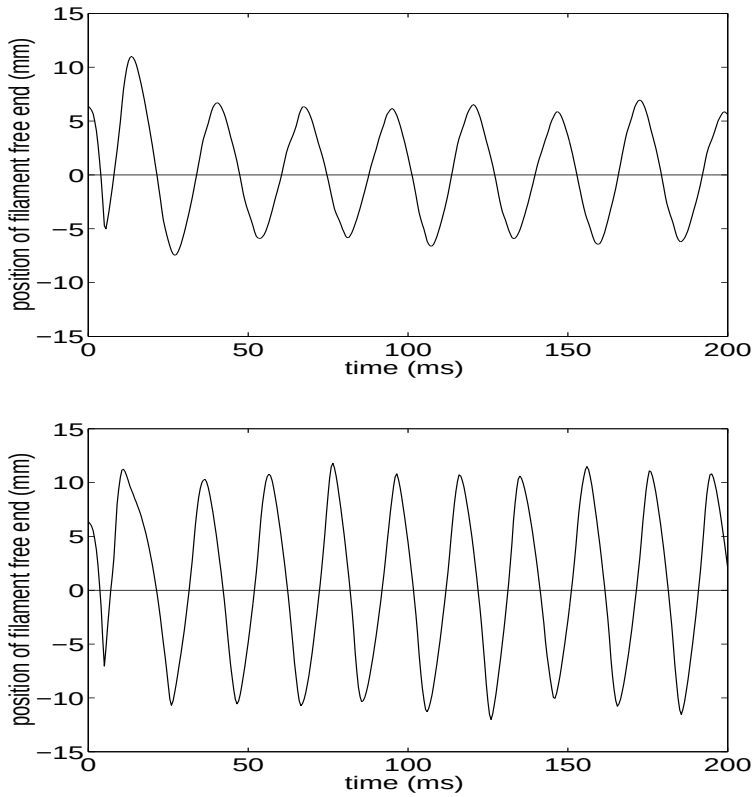

Figure 4.6: The position of the free end of the filament as a function of time: the upwind scheme (top) and the skew-symmetric (bottom).

Chapter 5

Simulation Results

The parameters of most of our simulations are shown in Table 5.1.

film inflow velocity	$200 - 280\, cm/sec$
film dynamic viscosity	$1.2 \times 10^{-5}\, g/(cm \cdot sec)$
film density	$3 \times 10^{-4}\, g/cm^2$
filament length	$1 - 6\, cm$
filament density	$4 \times 10^{-4}\, g/cm$
filament rigidity	$0.1\, erg \cdot cm$
gravitational acceleration	$980\, cm/sec^2$
air resistance coefficient	$0.00105 - 0.00147\, g/(cm^2 \cdot sec)$
width of the film	$8.5\, cm$
length of the film	$17\, cm$

Table 5.1: Parameters of the simulation

The dynamical viscosity μ is larger by two orders of magnitude than in the experiment, which results of course in the Reynolds number in our computation

being lower by two orders of magnitude than in the experiment. The mass of filament stated in Table 5.1 is twice that of the experimental filament (saturated with soapy water). The extra mass is intended to model the bulges in the film that form around the filament as a result of surface tension, which increase the effective filament mass. Note that in the laboratory experiment the film thickness is about $3\,\mu m$ while the diameter of the filament is about $150\,\mu m$. See Fig. 3.1). We also did computations with different filament mass within the range $[0, 0.001]\,g/cm$; the specific value of filament mass density will be given in context for each of such computations. Where the filament mass is not specifically stated, it is understood to have the value in Table 5.1 i.e. 4×10^{-4}. The vertical length of the domain in our computation is shorter than the experimental soap film, but we do not believe that the length of the film is an important parameter, provided it is long enough not to interfere with filament motion. All the other parameters besides the Reynolds number, the filament mass, and the length of the film are the same as those in the experiment for most of simulations. We also did computations with different filament bending rigidity ranging from 0 to $10\,erg \cdot cm$, but most of our simulations use the experimental value as given in Table 5.1.

We have done a lot of computations; the main simulation results are reported here in the following three parts. Part 1 presents the simulation results of a *massless* filament, and as a contrast, the simulation results of a filament with nonzero mass. We found that with a *massless* filament the system does not have a sustained flapping state. We did computations with different filament mass (though we think the effective filament mass density is twice that of the experimental filament for the reason explained above), and found that, within a certain range, the more mass the filament has, the bigger flapping amplitude

the filament has. In Part 2 we report the simulation results concerning the bi-stable property of the film-filament system. We found numerically that the whole system has, within a certain parameter range, two distinct dynamically stable states: the static (rest) state (where the filament is motionless) and the sustained flapping state (where the filament executes a sinuous motion) depending on the filament initial condition (with all the other parameters fixed). We also did computations with different filament lengths and found that a short enough filament does not have the sustained flapping state and always returns to its static state. In Part 3, we address the effect of filament bending rigidity on the motion of the system. We did computations with different bending rigidity and found that the motion of the filament-film system depends on filament bending rigidity. In the above bi-stable case, if we set the bending rigidity to zero, the static state is no longer stable, so bistability is lost and only the sustained flapping state is seen. If we set the bending rigidity equal to twice the experimental value, on the other hand, the situation is just the opposite: the flapping state disappears and only the static state is seen. In addition, if we change the bending rigidity in the middle of the computation, we can make the filament change state in either direction, from flapping to static or from static to flapping, depending on whether the bending rigidity is increased or decreased sufficiently.

We will address these three parts in detail in the rest of this chapter. First, we say a few words about visualization of the simulation results. Our simulation results are visualized mainly by using two different techniques: the instantaneous positions of the fluid markers and the contours of vorticity. In the following figures where these visualizations are used, the left panel of each of the figures shows the instantaneous positions of fluid markers created in bursts

along the upper (inflow) boundary; the right panel of each figure shows the corresponding vorticity contours. In both (left and right) panels in each figure flow is from top to bottom (driven by gravity, falling against air resistance) at an inflow velocity equal to the film terminal velocity profile $V_0(x)$. (See Fig. 3.4.) The width of the channel is 8.5 cm, the height of the channel is 17 cm.

In all the computations with the *massless* filaments, the Navier-Stokes equations are discretized as (4.14) and (4.15), and the FFT is used to solve the resulting linear algebraic system. In all the computations with filaments of nonzero mass, the Navier-Stokes equations are discretized as (4.17), (4.18), and (4.20), and the multigrid method (seven-grid V-cycle) is applied to solve the resulting discretized Navier-Stokes equations.

5.1 *Massless* filament vs filament with nonzero mass

In this part, first we report the simulation results with a *massless* filament. We did computations with different filament lengths (2 cm to 6 cm), different inflow velocities (200 cm/sec to 280 cm/sec), and different initial perturbations (25% to 40% of the filament length). In each of these simulations, the filament returns to its rest state (stretched-straight aligned with the flow direction) after a few oscillations and remains in the rest state. We found that a *massless* filament in the flowing film can not exhibit sustained flapping, no matter how large the initial perturbation is. After a few oscillations, it always returns to its straight position pointing downstream. Thus the straight state is globally stable. This indicates that the filament mass plays a key role for the film-filament system to

have a bi-stable scenario.

Fig. 5.1 - 5.5 (top panels) show a typical simulation of a *massless* filament in a flowing soap film. The inflow velocity $\bar{V_0}$ is 280 cm/sec, the filament length is 3 cm, the Reynolds number Re = 210. The initial perturbation in filament position is a sine wave with amplitude equal to 25% of the filament length. Fig. 5.1 and Fig. 5.4 (top panels) are the instantaneous position of the fluid markers and vorticity contours at several selected times; Fig. 5.5 (top) shows the position of the free end of the *massless* filament against the time. See Fig. 5.6 to 5.8 for more results with different filament lengths, inflow velocities, and different initial magnitude of perturbations. As a final example of the computations with a *massless* filament, we set the filament bending rigidity to zero and keep all the other parameters unchanged in the computation whose result is shown in Fig. 5.8. We find that the filament is still in rest state. See Fig. 5.9

It appears that the lack of flapping state of a *massless* filament can be explained as follows. The filament has its velocity, but it can not have any momentum (mass times velocity), so it can not obtain work or energy from the surrounding flowing film, which seems to be necessary for the filament to have sustained flapping. This was not obvious before doing the simulation, however. Since the filament can only move by displacing the soap film in which it is immersed, one might have thought that the mass of the surrounding soap film would act qualitatively like filament mass and make sustained flapping possible. According to our simulation results, this is not the case.

Fig. 5.1 through Fig. 5.5 (bottom panels) display the results of simulation which has the same parameters as those in Fig. 5.1 - 5.5 (top panels) except that now the filament has nonzero mass. The bottom panels of Fig. 5.1 - 5.5

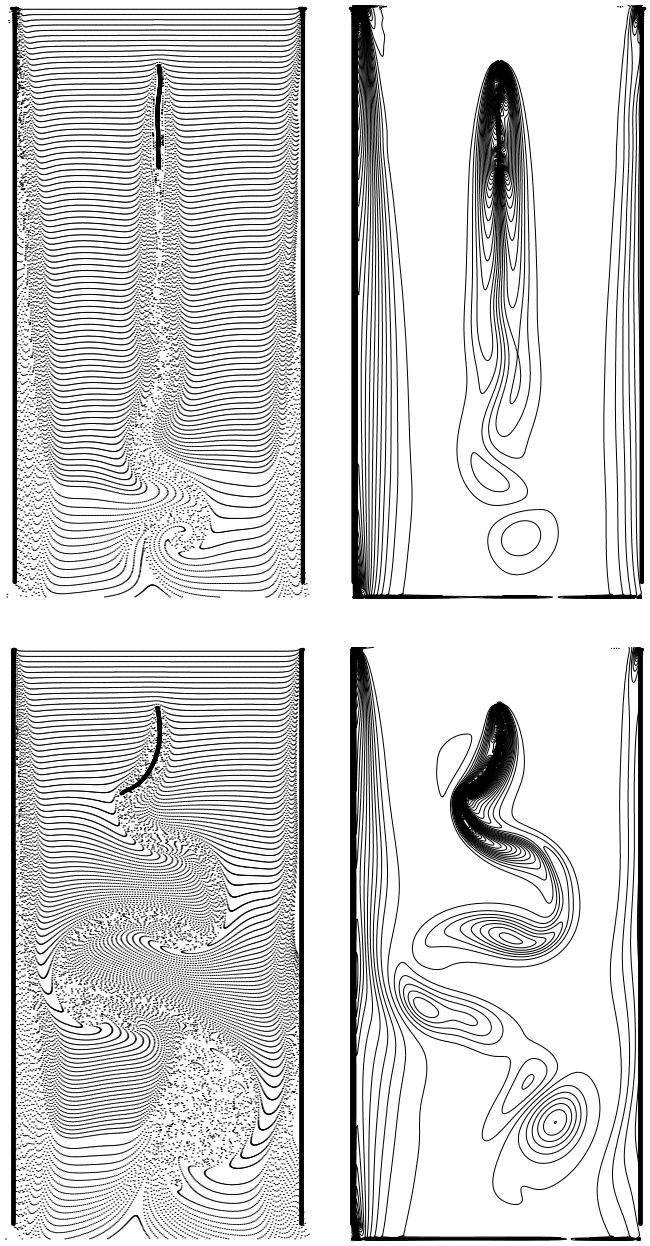

Figure 5.1: The motions of a *massless* filament (top panel) and a filament with nonzero mass (bottom panel) in a flowing soap film; $time = 0.0469 sec.$

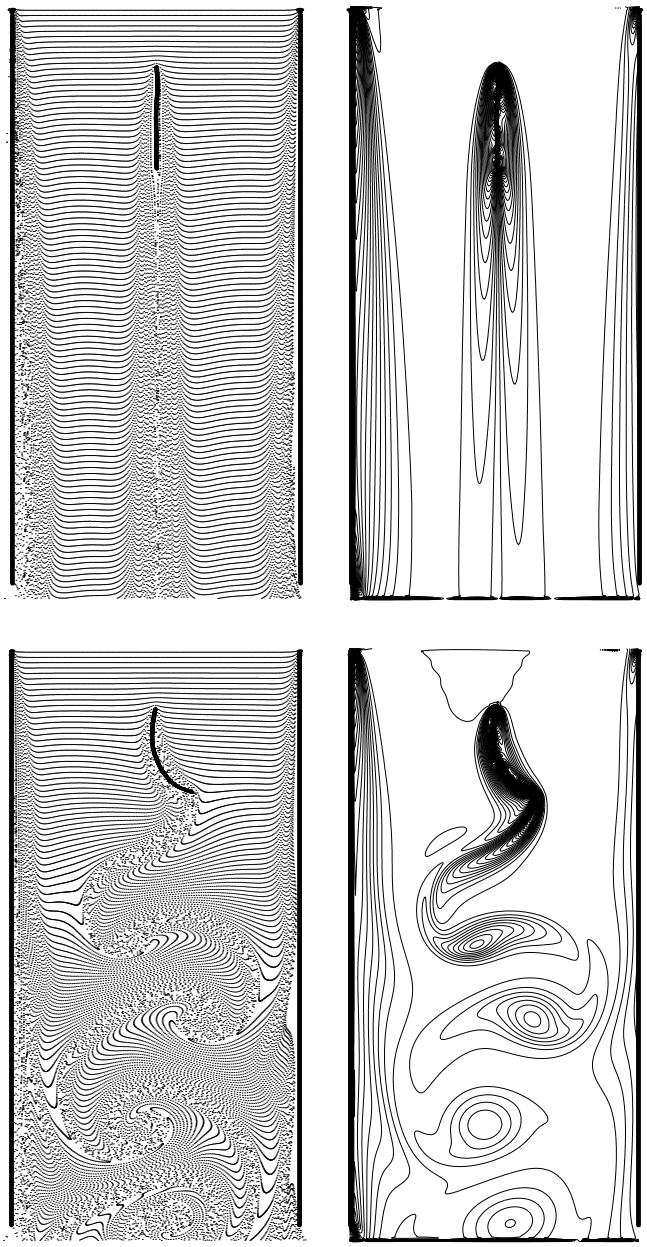

Figure 5.2: The motions of a *massless* filament (top panel) and a filament with nonzero mass (bottom panel) in a flowing soap film (continued from Fig 5.1); $time = 0.0968 sec.$

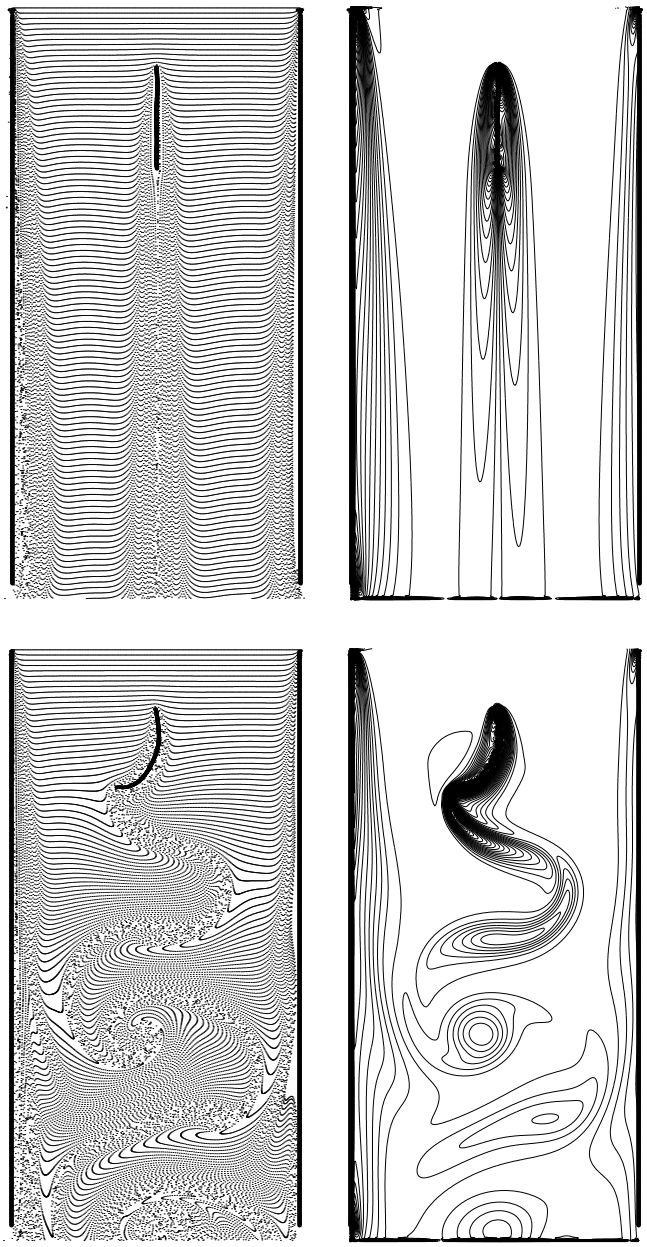

Figure 5.3: The motions of a *massless* filament (top panel) and a filament with nonzero mass (bottom panel) in a flowing soap film (continued from Fig. 5.2); $time = 0.1267 sec.$

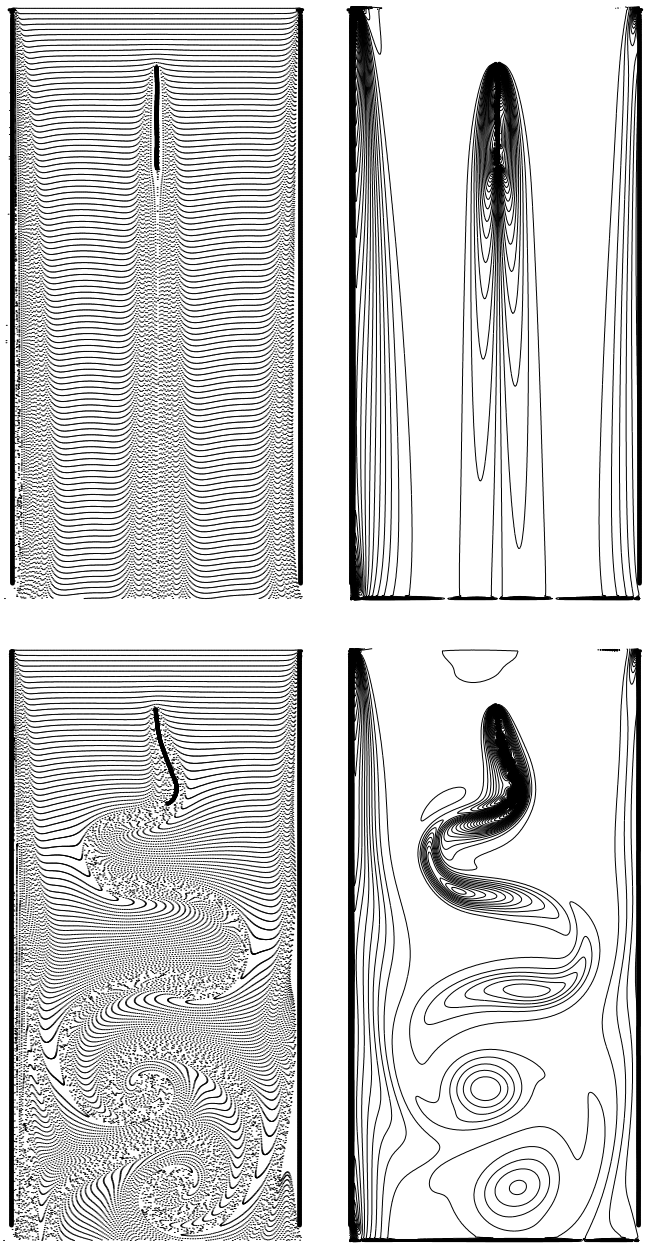

Figure 5.4: The motions of a *massless* filament (top panel) and a filament with nonzero mass (bottom panel) in a flowing soap film (continued from Fig. 5.3); $time = 0.1920 sec.$

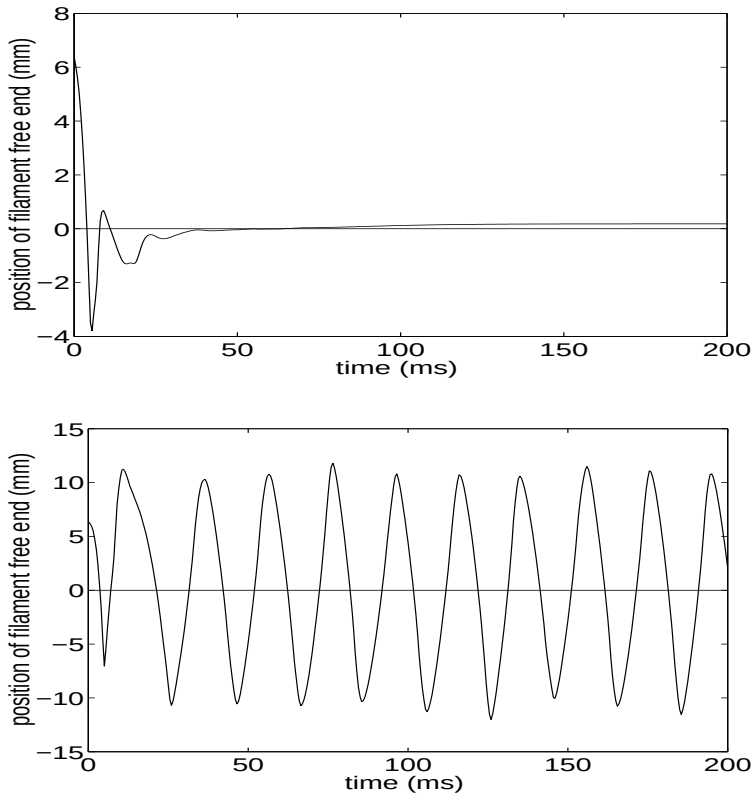

Figure 5.5: The position of the free end of the filament as a function of time: *massless* filament (top figure); filament with nonzero mass (bottom figure).

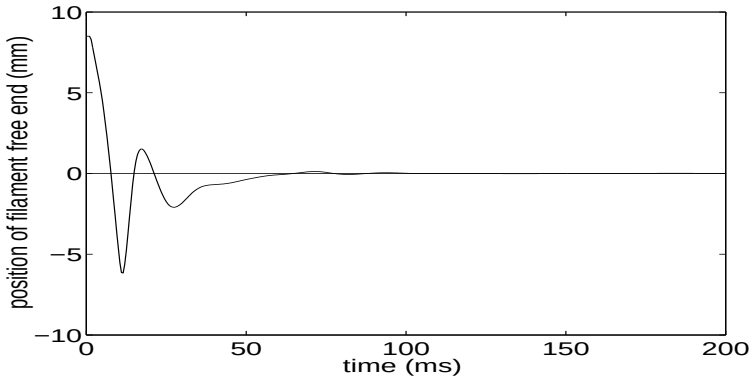

Figure 5.6: The position of the free end of a *massless* filament as a function of time: the inflow velocity is $200\,cm/sec$, the filament length is $4\,cm$,the initial perturbation is 25% of the filament length.

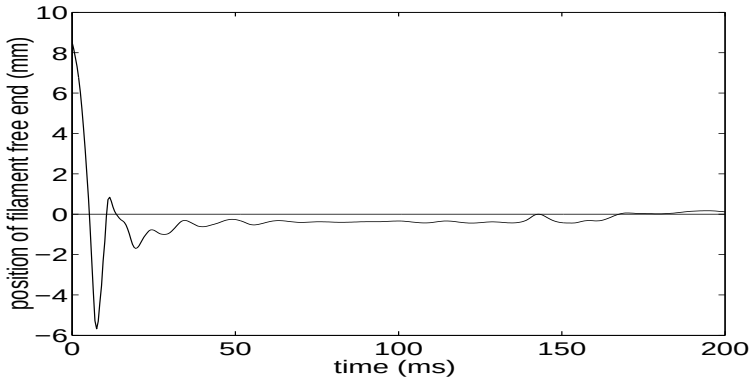

Figure 5.7: The position of the free end of a *massless* filament as a function of time: the inflow velocity is $280\,cm/sec$, the filament length is $4\,cm$,the initial perturbation is 25% of the filament length.

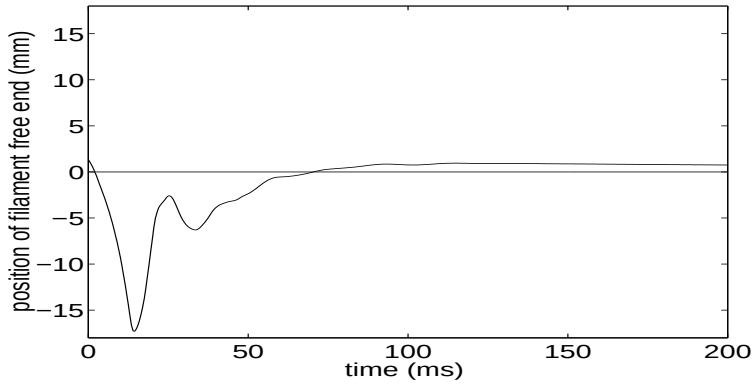

Figure 5.8: The position of the free end of a *massless* filament as a function of time: the inflow velocity is $220\,cm/sec$, the filament length is $6\,cm$,the initial perturbation is 40% of the filament length.

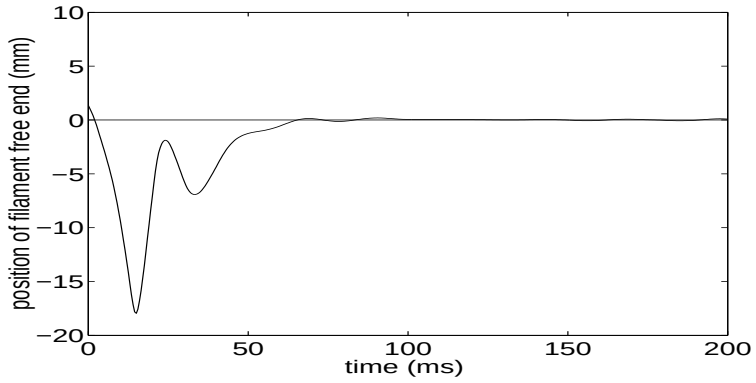

Figure 5.9: The position of the free end of a *massless* filament as a function of time: the inflow velocity is $220\,cm/sec$, the filament length is $6\,cm$,the initial perturbation is 40% of the filament length, the filament bending rigidity is zero.

shows the flapping state at different times of a filament whose mass is twice as that of the experimental one (for the reason explained above). The flapping frequency is about 50 Hz, which agrees very well with that observed in laboratory experiment even though the Reynolds number in simulation is much lower. The total excursion of the free end is about 2.1cm. The flapping is self-sustained and periodic in time.

Note that the time in the top panel and bottom panel in each figure of Fig. 5.1 - 5.4 is the same, respectively; thus we can see a clear contrast between the two obviously different scenarios resulting only from the filament mass.

We also did computations with different filament mass; (Even though we believe the effective filament mass density is about twice of that of experimental filament.) the flapping frequencies and the total excursions of the free end of the filament with different mass are listed in Table 5.2 (inflow velocity is 280 cm/sec,the filament length is 3 cm) and Table 5.3 (inflow velocity is 200 cm/sec, the filament length is 2 cm). From Table 5.2 and Table 5.3 we can see that the total excursion of the filament free end increases with the increasement of filament mass; but the flapping frequency does not show such a simple relationship. (In the case of 200 cm/sec inflow speed and 2 cm filament, however, the flapping frequency decreases with the increasement of mass.) Fig. 5.10 - 5.11 show the horizontal position of filament free end with the time. It seems that the more mass the filament has, the more violent the motion of the system is.

filament density (g/cm)	frequency (Hz)	amplitude (mm)
2×10^{-4}	52.9	9.8
4×10^{-4}	50	21
6×10^{-4}	53.6	24.5
8×10^{-4}	47	27
1×10^{-3}	41	28

Table 5.2: Flapping frequency and amplitude of the filament with different filament mass; the inflow velocity is $280\,cm/sec$, the filament length is $3\,cm$.

filament density (g/cm)	frequency (Hz)	amplitude (mm)
2×10^{-4}	0	0
4×10^{-4}	38	15
6×10^{-4}	36	18.9
8×10^{-4}	33.2	23.1
1×10^{-3}	29.9	25.2

Table 5.3: Flapping frequency and amplitude of the filament with different filament mass; the inflow velocity is $200\,cm/sec$, filament length is $2\,cm$.

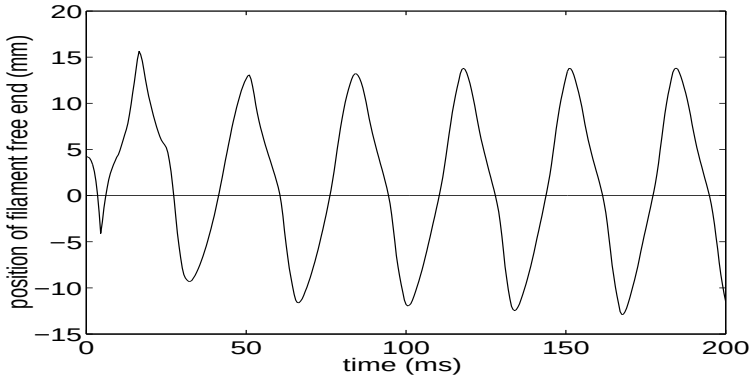

Figure 5.10: The position of the free end of the filament as a function of time; the filament mass is $1 \times 10^{-3}\, g/cm$ (the experimental value is $2 \times 10^{-4}\, g/cm$), the filament length is $2\, cm$, the inflow velocity is $200\, cm/sec$

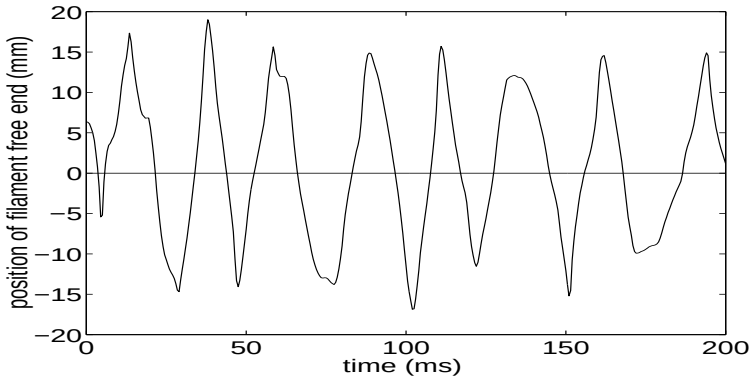

Figure 5.11: The position of the free end of the filament as a function of time; the filament mass is $1 \times 10^{-3}\, g/cm$ (the experimental value is $2 \times 10^{-4}\, g/cm$), , the filament length is $3\, cm$, the inflow velocity is $280\, cm/sec$

5.2 Bi-stability of the film-filament system

Fig. 5.12 through Fig. 5.15 exhibit the bi-stable property of the system. In these cases, the inflow film velocity $\bar{V_0}$ is 200 cm/sec, the filament length is 2 cm. All the other parameters are the same as those in Table 5.1. The only difference between the parameters in these two simulations is the initial perturbation: in the top panels of Fig. 5.12 - 5.15 the initial perturbation is 1% of the filament length, while in the bottom panels of Fig. 5.12 - 5.15 it is 25% of the filament length. In the case of a small initial perturbation, the filament returns to its rest state (straight position aligned with the flow direction) after a period of adjustment during which the filament oscillates with small amplitude. After settling down the flexible filament looks like a rigid body, and the resultant flow field resembles a two-dimensional flow passing a thin plate. (See top panels of Fig. 5.12 through Fig. 5.15) In the case of large perturbation (See the bottom panels of Fig. 5.12 through Fig. 5.15), the filament quickly settles into its sustained periodic flapping state after 1 or 2 oscillations. The flapping frequency is about 37 Hz, the peak-to-peak amplitude is about 1.3 cm. A vortex is shed from the free end of the filament by each stroke, and these form a "street" of vortices of alternating sign in the wake of the oscillating filament. Each vortex is washed away downstream by the flowing film and gets diffused because of the film viscosity. The vortex develops a mushroom-like structure which resembles those observed in interfacial instabilities (Rayleigh-Taylor instability, Richtmyer-Meshkov instability and Kelvin-Helmholtz instability). One feature of Zhang's experiment [3] that we do not yet capture is the *small-scale* structure of the vortex wake. In all of our simulation involving filament flappings, there is a sinuous line of highly sheared fluid connecting the large-scale shed vortices.

This is especially evident in the particle traces (left-hand panels of the Figures). In Zhang's experiment this line resolves itself into discrete *small-scale* vortices, which we do not see. This could be because our mesh is too coarse or because our Reynolds number is too low for this fine-scale structure to appear. Despite this, the large-scale structure of the vortex wake is very similar in our simulation and in Zhang's experiment.

We also did computations with variable filament length; the results are as follows. (All the parameters in these computations are the same as above bi-stable case except for the filament length, which changes from one computation to another.) The flapping frequencies and the total excursions of the free end of the filament with different lengths are listed in Table 5.4. The relation between flapping frequency and the filament length is plotted in the upper figure in Fig. 5.17, and the lower figure in Fig. 5.17 shows the relation between the total excursion of the free end of filament and the filament length. Note that in this case there is a critical length of about 1.6 cm below which the filament always returns to its static state independent of the magnitude of the initial perturbation.

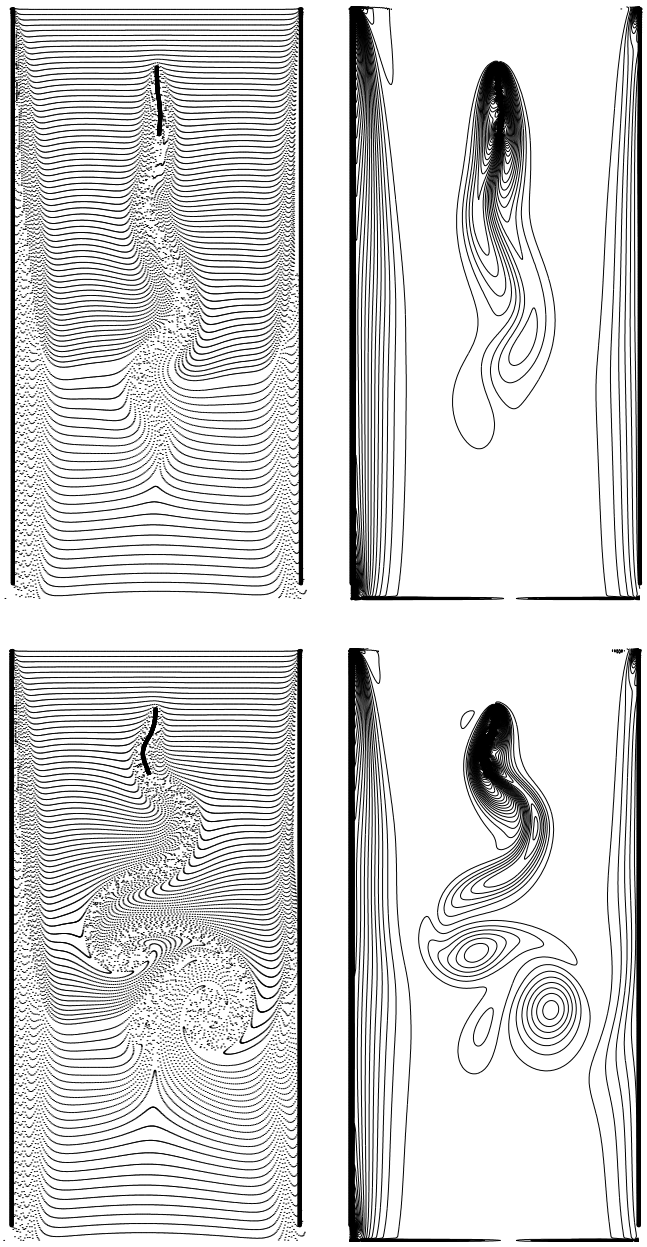

Figure 5.12: Bi-stability of the film-filament system; the only difference between the computations in top panel and bottom is the initial perturbation: small (top panel), large (bottom panel). $time = 0.05$ *sec.*

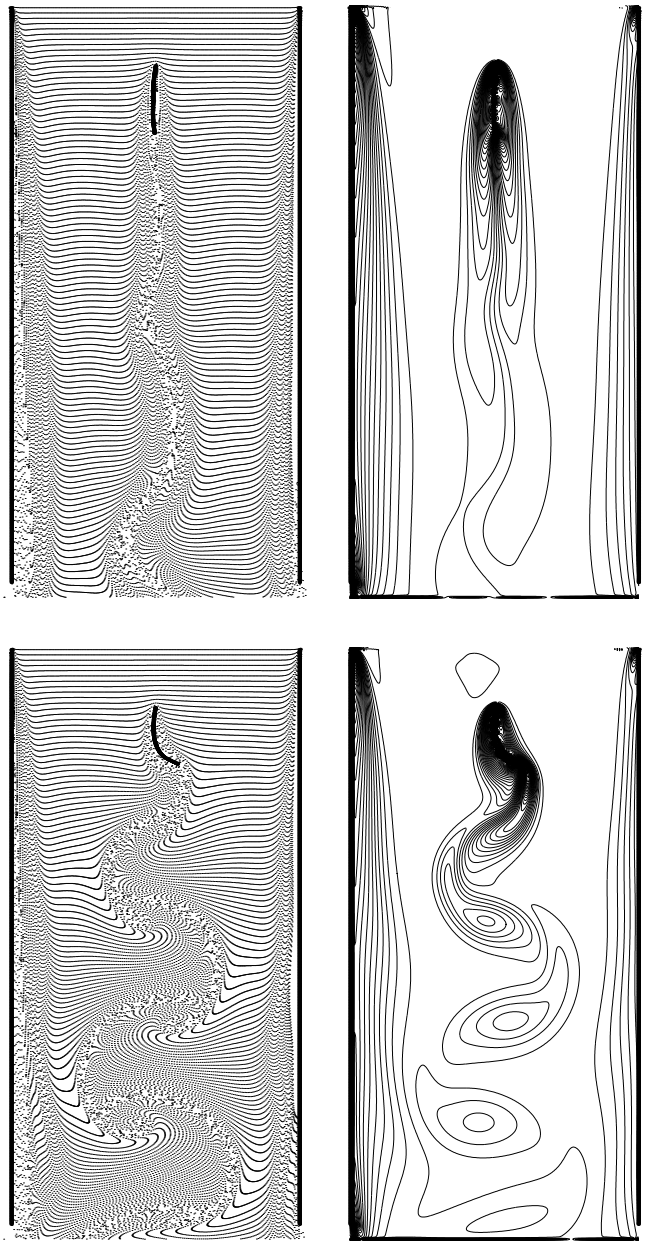

Figure 5.13: Bi-stability of the filament-film system (continued from Fig. 5.12); $time = 0.1\ sec.$

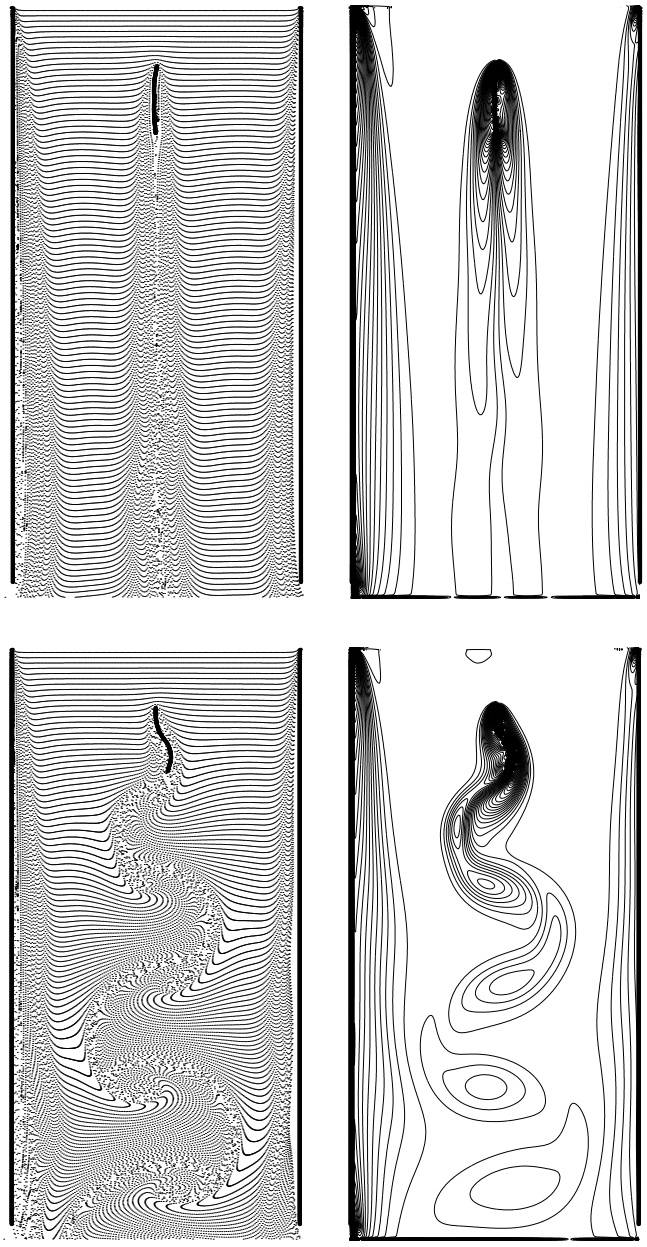

Figure 5.14: Bi-stability of the filament-film system (continued from Fig. 5.13); $time = 0.152\ sec.$

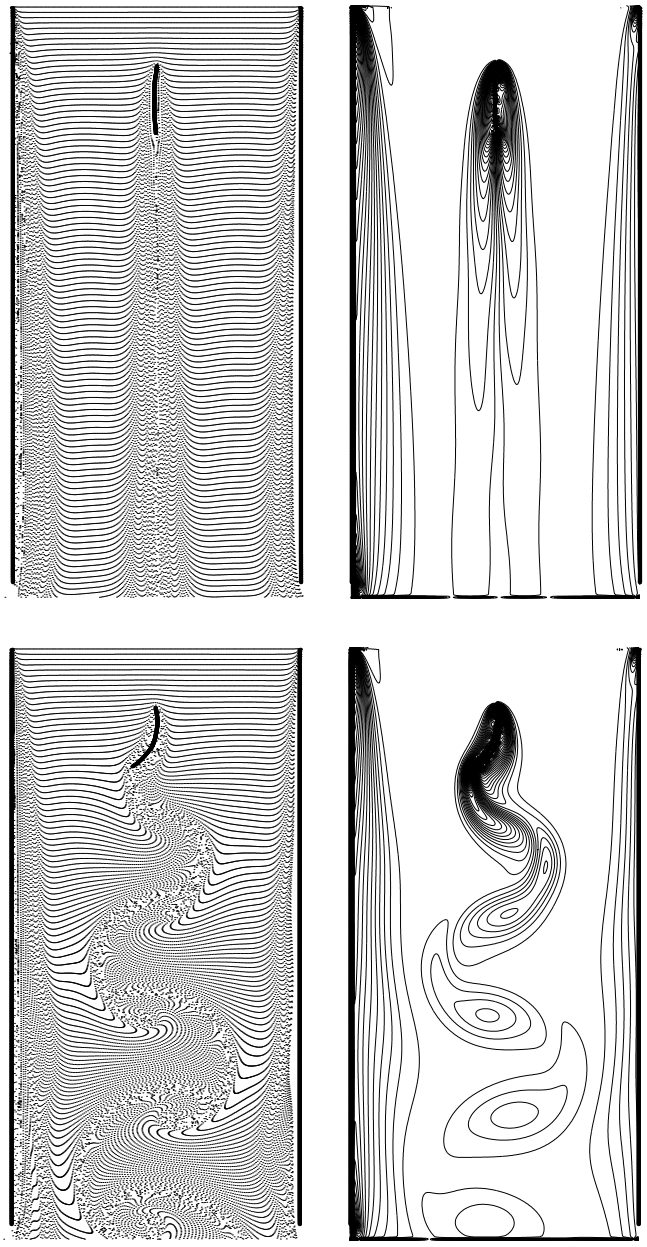

Figure 5.15: Bi-stability of the filament-film system (continued from Fig. 5.14); $time = 0.2\ sec.$

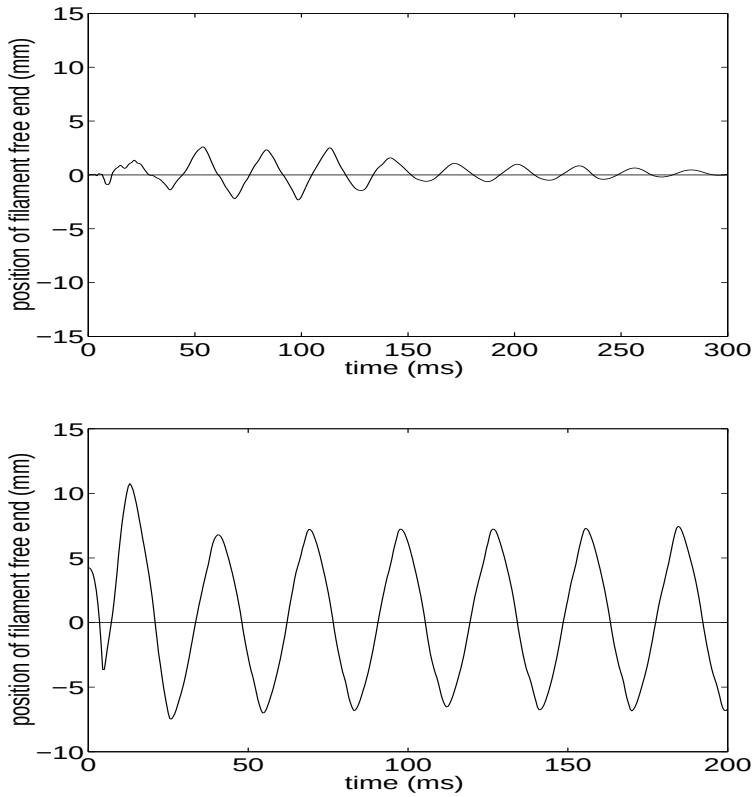

Figure 5.16: The position of the free end of the filament as a function of time: the static state (top figure) and the flapping state (bottom panel).

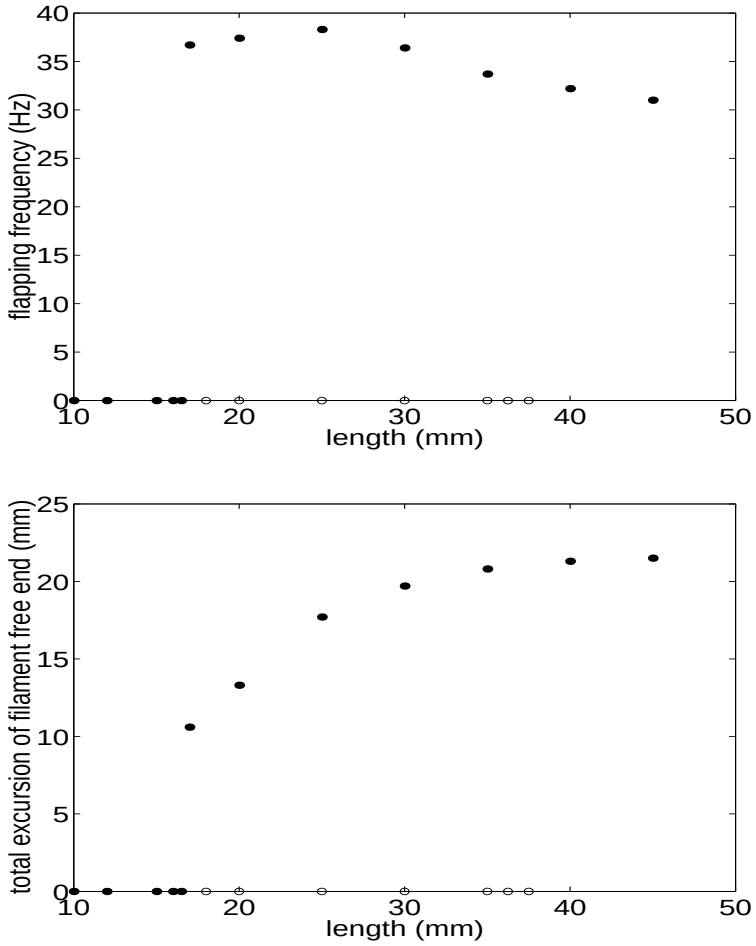

Figure 5.17: Flapping frequency vs filament length (top) and flapping amplitude vs filament length (bottom). The "•" represents the result from large initial perturbation, the "∘" represents result from small initial perturbation. Below the length of 1.8 cm, only the large amplitude perturbation results are shown, since even these approach the static rest state.

filament length (mm)	frequency (Hz)	amplitude (mm)
16	0	0
17	36.7	10.6
20	37.4	13.3
25	38.3	17.7
30	36.4	19.7
35	33.7	20.8
40	32.2	21.3
45	31	21.5

Table 5.4: Flapping frequency and total excursion of filament free end

5.3 Effect of filament bending rigidity

In this part we report the simulation results of the effect of filament bending rigidity on the motion of the film-filament system. First, starting from the previous simulation parameters of the bi-stable case in Part 2, if we change the bending rigidity from the laboratory measured value ($0.1\ erg \cdot cm$) to zero, we find that the static state loses its stability so that even the small amplitude initial perturbation leads to sustained flapping. (See Fig. 5.18.) In the case of the large amplitude perturbation, if we double the bending rigidity and keep all the other parameters the same, we find that the filament returns to its motionless state following a number of oscilations (See Fig. 5.19). Moreover, if we change the bending rigidity during the course of a computation, we find that the two distinct dynamical states can be switched depending on whether the bending rigidity is increased or decreased sufficiently. See Fig. 5.20 and

Fig. 5.21. Further computer experiments involving changes in bending rigidity are shown in Fig. 5.22 and Fig. 5.23. Note that in the case of inflow velocity 280 cm/sec, filament length 3 cm, when the bending rigidity is doubled from the laboratory value, the filament remains in its flapping state as if the bending rigidity was not changed at all. But when the rigidity is increased to 10 times as large as the laboratory value, the filament changes its state from flapping to static.

Note that even though the previous bi-stable scenario disappears when we change the filament bending rigidity from the value that it had in the above specific bi-stable case in Part 2, there could still be a bi-stable scenario for the filament-film system with different filament bending rigidity. We have not yet done the numerical experiments to determine this.

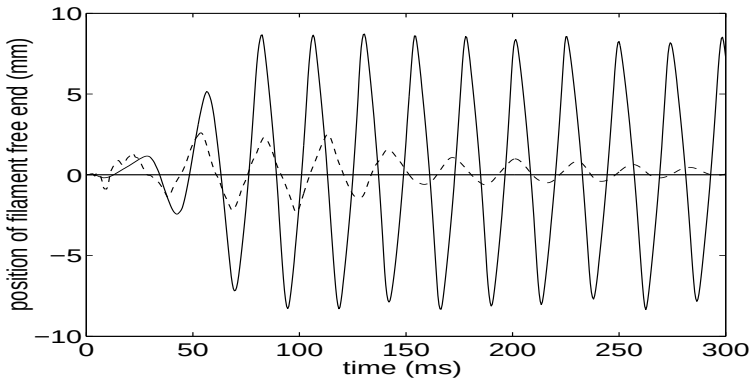

Figure 5.18: The position of the free end of the filament as a function of time; a filament in static state (dashed curve) changes to flapping state (solid curve) after its bending rigidity is set to zero.

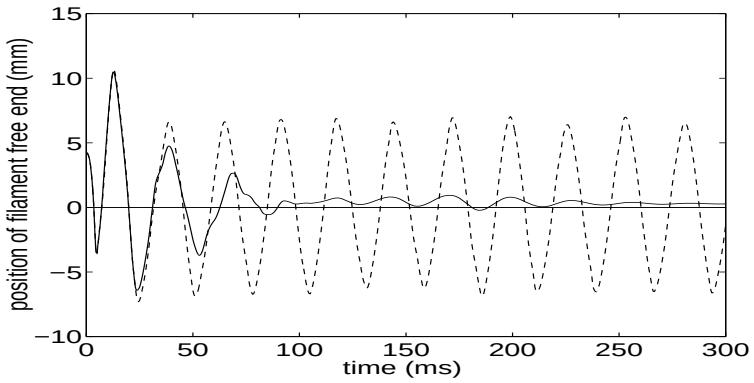

Figure 5.19: The position of the free end of the filament as a function of time; a filament in flapping state (dashed curve) changes to rest state (solid curve) after its bending rigidity is doubled.

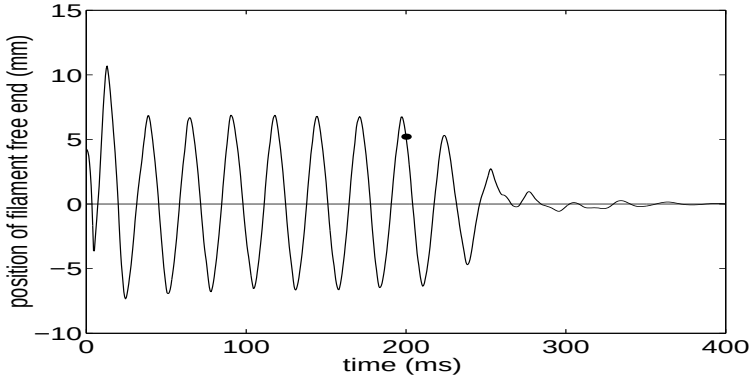

Figure 5.20: The position of the free end of the filament as a function of time: the inflow velocity is 200 cm/sec, the filament length is 2 cm, the "•" indicates the time when the filament bending rigidity is doubled. The flapping filament approaches its static state after its bending rigidity is doubled.

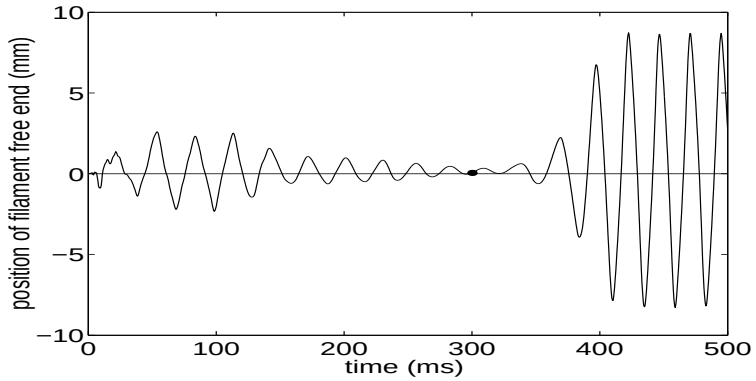

Figure 5.21: The position of the free end of the filament as a function of time: the inflow velocity is 200 cm/sec, the filament length is 2 cm, the "•" indicates the time when the filament bending rigidity is decreased to 0.001 $erg \cdot cm$. The filament in rest state becomes flapping after its bending rigidity is decreased.

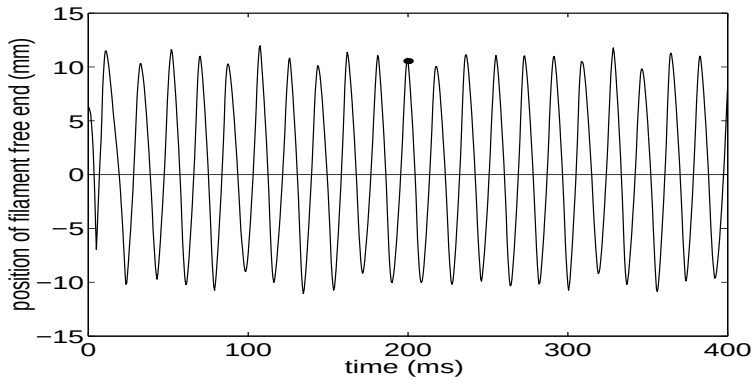

Figure 5.22: The position of the free end of the filament as a function of time: the inflow velocity is 280 cm/sec, the filament length is 3 cm, the "•" indicates the time when the filament bending rigidity is doubled. This flapping filament, unlike the one in Fig. 5.20, remains in its flapping state after its bending rigidity is doubled. The difference are that the filament is longer and the inflow velocity is higher.

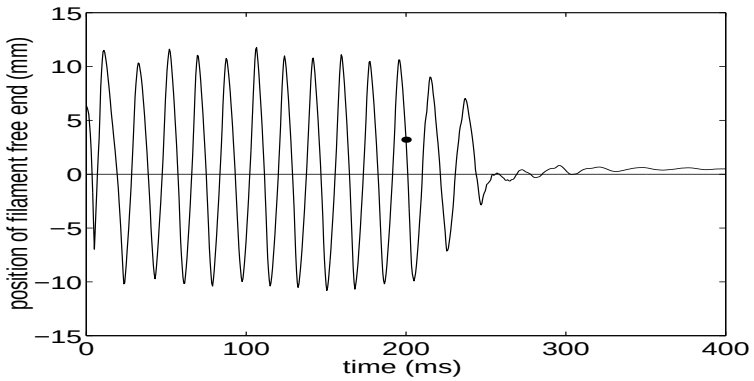

Figure 5.23: The position of the free end of the filament as a function of time: the inflow velocity is 280 cm/sec, the filament length is 3 cm, the "•" indicates the time when the filament bending rigidity is increased to 1 $erg \cdot cm$. This value is 5× the bending rigidity used in Fig. 5.22 and 10× the original bending rigidity. The flapping filament approaches its rest state after its bending rigidity is increased.

Chapter 6

Summary and Conclusion

The subject of this thesis is a flexible filament immersed in a flowing soap film. We have simulated this system by the Immersed Boundary Method, paying particular attention to its two dynamically stable states: a straight state in which the filament is at rest parallel to the flow, and a flapping state in which the filament oscillates from side to side periodically shedding vortices into the flowing soap film. Though the numerical research is still underway, some main conclusions have been reached:

1) The sustained flapping of the filament only occurs when filament mass is included in the formulation of the model; within a certain range of filament mass, the more mass of the filament the bigger the amplitude of flapping.

2) When the length of filament is short enough (below some critical length), the filament always approaches its straight state; but when the length is larger the system is bi-stable, which means that it can settle into either a straight state or a flapping state depending on the initial conditions. (There may be an upper critical length above which only the flapping state is stable, but we have not investigated this .)

3) The bi-stability of the film-filament system depends on the filament bending rigidity; the filament motion can be switched in either direction, from static to flapping, or from flapping to static, depending on whether the bending rigidity is decreased or increased sufficiently.

In choosing parameters for these computer simulations, we have closely followed the experimental data of [3], with one important exception: the Reynolds number of the computation is about 200, whereas the Reynolds number of the experiment is about 20,000.The fact that we get good agreement with experiment, not only qualitatively but even with regard to such quantitative measures as the flapping frequency (about 50 HZ in both the experiment and in the simulation in the case of 280 cm/sec inflow), suggests that the Reynolds number is not an important parameter for the bi-stability of the filament-film system. (Perhaps the Reynolds number has to be sufficiently high for the flapping to occur, but Re = 200 seems to be high enough.) This raises the question what other non-dimensionless parameters might be important. Two examples have been proposed by Shelley [4], which can be defined in our case as the follows:

$$\mathrm{S}_1 = \frac{ML}{\rho_0 L^2}$$

$$\mathrm{S}_2 = \frac{K_{bend}/L}{L^2 \rho_0 \bar{V}_0^2}$$

A good project for future work would be to study systematically the influence of these non-dimensional parameters (and also the Reynolds number) on the behavior of the filament-film system.

Other future works would be: 1) numerically study the interaction between two flexible filaments in the flowing soap film. (It was studied in laboratory

experiment by Zhang [3].) 2) use the formal second order Immersed Boundary Method [29] [31] to simulate the film-filament system. This would help to lift the Reynolds number in our simulation of such a system.

Bibliography

[1] M.P. Paidoussis, Fluid-Structure Interaction Vol. I (Academic San Diego, 1998).

[2] D.G. Crighton, New aspects of fluid-structure interaction theory,

[3] J. Zhang, S. Childress, A. Libchaber, and M. Shelley, Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind,*Nature* **408**,835 (2000).

[4] M. Shelley, S. Childress,and J. Zhang, Inertia dynamics of filaments, in preparation.

[5] P. Fast and W.D. Henshaw, Time-accurate computation of viscous flow around deforming bodies using overset grids, AIAA paper 2001-2604, 15th AIAA Computational Fluid Dynamics Conference, accepted.

[6] C.S. Peskin,Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* **25**,220(1977).

[7] C.S. Peskin and D.M. McQueen, A general method for the computer simulation of biological systems interacting with fluids,*Sympos. Soc. Exp. Biol.* **49**,265 (1995).

[8] 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,* edited by H.G. Othmer, F.R. Adler, M.A. Lewis, and J.C. Dallon (Prentice-Hall, Englewood Cliffs, NJ, 1996), p.309.

[9] D.M. McQueen and C.S. Peskin, A three-dimensional computer model of the human heart for studying cardiac fluid dynamics, *Computer Graphics* **34**,56 (2000).

[10] C.S. Peskin and D.M. McQueen, Computational biofluid dynamics, *Contemp. Math.* **141**, 161 (1993).

[11] D.M. McQueen, C.S. Peskin, and L. Zhu, The immersed boundary method for incompressible fluid-structure interaction, accepted by the First M.I.T. Conference on Computational Fluid and Solid Mechanics, June 2001.

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

[13] D.M. McQueen and C.S. Peskin, Shared memory parallel vector implementation of the immersed boundary method for the computation of the blood flow in the beating mammalian heart, *J. Supercomput.* **11**,213 (1997).

[14] D.M. McQueen, C.S. Peskin, and E.L. Yellin, Fluid dynamics of the mitral valve: physiological aspects of a mathematical model, *Am. J. of Physiol.,* **242**, 1095 (1982).

[15] L.J. Fauci, Interaction of oscillating filaments–A computational study,*J. Comput. Phys.*,**86**, 294 (1990).

[16] L.J. Fauci and A.L. Fogelson,Truncated Newton methods and the modeling of complex elastic structures,*Comm. Pure Appl. Math.*,**46**,787(1993).

[17] L.J. Fauci and C.S. Peskin, A computational model of aquatic animal locomotion,*J. Comput. Phys.* **77**,85 (1988).

[18] R.P. Beyer, A computational model of the cochlea using the immersed boundary method, *J. Comput. Phys.* **98**,145 (1992).

[19] E. Givelberg, Modeling elastic shells immersed in fluid, PhD thesis, Courant Institute of Mathematical Sciences, New York University, September 1997 (unpublished).

[20] A.L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, *J. Comput. Phys.* **56**,111 (1984).

[21] A.L. Fogelson and C.S. Peskin, A fast numerical method for solving three-dimensional Stokes equations in the presence of suspended particles,*J. Comput. Phys.***79**, 50 (1988).

[22] D. Sulsky and J.U. Brackbill, A numerical method for suspension flow,*J. Comput. Phys.* **96**, 339 (1991).

[23] E. Jung and C.S. Peskin, 2-D simulation of valveless pumping using the immersed boundary method, *SIAM J. Sci. Comput.*, to appear (2001).

[24] M.E. Rosar, A three-dimensional computer model for fluid flow through a collapsible tube, PhD thesis, Courant Institute of Mathematical Sciences, New York University, 1994 (unpublished).

[25] 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. Comput. Phys.* **147**,402 (1998).

[26] D.C. Bottino, Modeling viscoelastic networks and cell deformation in the context of the immersed boundary method, *J. Comput. Phys.*,**147**,86 (1998).

[27] C.D. Eggleton and A.S. Popel, Large deformation of red blood cell ghosts in a simple shear flow,*Phys. Fluids*,**10** ,1834 (1998).

[28] J.M. Stockie and S.I. Green, Simulating the motion of flexible pulp fibres using the immersed boundary method, *J. Comput. Phys.* **147**,147 (1998).

[29] 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 (2000).

[30] M.C. Lai, Simulations of the flow past an array of circular cylinders as a test of the immersed boundary method, PhD thesis, Courant Institute,Sept. 1998.

[31] D.M. McQueen and C.S. Peskin, Heart simulation by an immersed boundary method with formal second order accuracy and reduced numerical viscosity, *ICTAM 2000 Proceedings*, Kluwer, in press.

[32] A.L. Fogelson and J. Zhu, Implementation of a variable-density immersed boundary method, unpublished, http://www.math.utah.edu/fogelson.

[33] Y. Couder, J.M. Chomaz, and M. Rabaud, On the hydrodynamics of soap films, *Physica D*, **37**,384 (1989).

[34] M. Gharib and P. Derango, A liquid film (soap film) tunnel to study two-dimensional laminar and turbulent shear flows, *Physica D*, bf 37,406 (1989).

[35] H. Kellay, X-I. Wu, and W.I. Goldburg, Experiments with turbulent soap films, *Physical Review Letters*, **74**(20), 3975 (1995).

[36] M.A. Rutgers, X-I. Wu, R. Bhagavatula, A.A. Peterson, and W.I. Goldburg, Two-dimensional velocity profiles and laminar boundary layers in flowing soap films,*Phys. Fluids* **8** (11), 2847 (1996).

[37] A. J. Chorin, Numerical solution of the Navier-Stokes equations, *Math. Comp.,***22**,745 (1968).

[38] A. J. Chorin, On the convergence of discrete approximations to the Navier-Stokes equations, *Math. Comp.,***23**,341 (1969).

[39] W.L. Briggs, V.E. Henson, and S.F. McCormick, A multigrid tutorial, 2nd edition, SIAM, 2000.

[40] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran: the art of scientific computing, 2nd edition, Cambridge University Press, p.862, 1992.

[41] A. Brandt, Multigrid Techniques: 1984 Guide with Application to Fluid Dynamics, GMD-Studien Nr. 85, Gesellschaft fur Mathematik und Datenverarbeitung, St. Augustin, Bonn, 1984.