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

Luoding Zhu and Charles S. Peskin

Courant Institute of Mathematical Sciences, New York University

251 Mercer Street, New York, NY 10012 USA

Email: zhuld@cims.nyu.edu    peskin@cims.nyu.edu

running title: flapping filament in flowing film

current address of correspondence:

Mail: Department of Computer Science

University of California at Santa Barbara

Santa Barbara, CA 93106

Email: zhuld@cs.ucsb.edu

Phone: (805)893-5728(O), (805)968-1805(H)

**Abstract**

This paper 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 need to use a multigrid method for solving the discretized fluid equations. This replaces the FFT based method that is commonly used in the uniform-density case.

Our main 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 (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.

This numerical result we observed in computer simulation is the same as that of the laboratory experiment even though the Reynolds number of the computations is lower than that of the laboratory experiment by two orders of magnitude.

**Keywords**

Immersed Boundary Method, flapping filament, flag-in-wind, bi-stability, computational fluid dynamics, multigrid method

**Classification numbers:** 65-04, 65M06, 65M55, 76D05, 76D17, 76D27, 76E17 , 76E30, 76M25

## 1. Introduction

Many problems in biofluid dynamics involve interactions between deformable elastic bodies and incompressible viscous fluids, for instance, the swimming motions of eel, sperm and flagella. As a model of hydrodynamic interaction of deformable bodies with surrounding fluid flows, Zhang [1] studied experimentally the dynamics of flexible filaments in a flowing soap film. See Fig. 1 for the experimental setup: 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 [1] have shown that the flexible filament in the flowing soap film is actually bi-stable. In addition to that, the system itself (a free boundary problem) is very interesting: we have a one-dimensional immersed 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. A vortex street is formed downstream.

Currently Shelley [2] is working on the instability analysis of this system; Fast [3] 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 practical and efficient way to simulate fluid-structure interaction in the incompressible case. It has been applied successfully to a wide range of problems, particularly in computational biofluid mechanics : blood flow in the human heart [4] [5] [6] [7] [8] [9] [10] [11] , the design of prosthetic cardiac valves [12], aquatic animal locomotion [13] [14] [15], wave propagation in the cochlea [16] [17], platelet aggregation during blood clotting [14] [18] , flow of suspensions [19] [20], valveless pumping [21], flow in a collapsible tube[22], flow and transport in a renal arteriole [23], cell and tissue deformation under shear flow [24] [25] [26]. At present there exist several versions of the Immersed Boundary Method. The version we use here is different from most existing versions [5] [7] [9] [11] [27] [28] in two aspects: 1) the discretization of the Navier-Stokes 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 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-Stokes 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 [29], here we report on another such example.

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 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 need to use a multigrid method for solving the discretized fluid equations. This replaces the FFT based method that is commonly used in the uniform-density case.

Our main 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 (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. This numerical result we observed in computer simulation is the same as that of the laboratory experiment even though the Reynolds number of the computations is lower than that of laboratory experiment 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, etc.

**2. Mathematical Formulation**

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 **x**=(x,y) and the time $t$, and the independent Lagrangian variables are the curvilinear material coordinate s and 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 dependent Lagrangian variables are the position of 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:

$$\rho(\mathbf{x},t)(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \mu\Delta\mathbf{u} + \mathbf{f}(\mathbf{x},t) - \lambda\mathbf{u} - \rho(\mathbf{x},t)g\hat{\mathbf{e}}_2 \qquad (1)$$

$$\nabla\cdot\mathbf{u} = 0 \qquad (2)$$

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) \qquad (3)$$

$$\mathbf{f}(\mathbf{x},t) = \int \mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))ds \qquad (4)$$

$$\rho(\mathbf{x},t) = \rho_0 + \int M\delta(\mathbf{x}-\mathbf{X}(s,t))ds \qquad (5)$$

$$\mathbf{U}(s,t) = \int \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(s,t))d\mathbf{x} \tag{6}$$

$$\mathbf{F}(s,t) = \mathbf{F}_s(s,t) + \mathbf{F}_b(s,t) = \frac{\partial T\hat{\boldsymbol{\tau}}}{\partial s} - \frac{\partial E_b}{\partial \mathbf{X}} \tag{7}$$

$$T = K_s(|\frac{\partial \mathbf{X}}{\partial s}| - 1) \tag{8}$$

$$\hat{\boldsymbol{\tau}} = \frac{\frac{\partial \mathbf{X}}{\partial s}}{|\frac{\partial \mathbf{X}}{\partial s}|} \tag{9}$$

$$E_b = \frac{1}{2}K_b \int |\frac{\partial^2 \mathbf{X}(s,t)}{\partial s^2}|^2 ds \tag{10}$$

where the constant $\mu$ is the soap film viscosity; the $\lambda$ 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 film terminal velocity, $\rho_0$ is the mass per unit area of the soap film. (Note unusual units: this is a 2-D problem!) and g is the gravitational acceleration. 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 over the majority of the span about the midline (rather than a parabola) due to air resistance. M is the uniform Lagrangian mass density of the filament, $\hat{\mathbf{e}}_2$ denotes the unit vector in the vertical direction

($y$ direction). The function $\delta(\mathbf{x})$ is the Dirac $\delta$ function. The Lagrangian force density $\mathbf{F}(s,t)$ consists of two terms: the stretching and compression force $\mathbf{F}_s(s,t)$, and the bending force $\mathbf{F}_b(s,t)$. $T$ is the tension in the filament which is computed by Hook's law (Equation 8). $\hat{\boldsymbol{\tau}}$ is the unit tangent vector defined at each point of the filament. $K_s$ is filament stretching coefficient which is chosen in computation so that the filament has almost no stretch. $K_b$ is the bending rigidity which was measured in the laboratory experiment. The bending force density is obtained by taking the Frechet derivative of the bending energy $E_b$ which is defined by equation (10). This is essentially the principle of virtual work.

These equations (without the viscous and air-resistance terms) can be derived formally from the principle of least action, see [9] for details. The equations (1)-(2) are the incompressible Navier-Stokes equations with multiple forcing terms (gravity, air resistance and the forces applied by the filament); equations (3) are the equations of motion of the filament, where $\mathbf{U}(s,t)$ is the velocity of the filament. The above two systems of partial differential equations (equations (1),(2) and equations (3)) are coupled through the three integrals (4)-(6).

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:

$$\begin{cases} \mu V_{xx} - \lambda V - \rho_0 g & = 0 \\ V(a) = V(b) & = 0 \end{cases} \tag{11}$$

Where $a$ and $b$ are the x-coordinates of the position of the two wires. Equation (11) is obtained by setting $\mathbf{u} = (0, V_0(x))$, $\frac{\partial}{\partial t} = 0$, $p(\mathbf{x}, 0) = constant$ in the incompressible Navier-Stokes equations.

The value of $\mathbf{X}(s, 0)$ is specified as initial condition for the filament and the boundary condition is that $\mathbf{X}(0, t)$ 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. Numerical Method

The above system of differentio-integral equations are numerically solved by the Immersed Boundary Method. Our computational box (rectangle here) is slightly larger than the physical domain in both $x$ and $y$ directions, and periodical boundary conditions are used in both sides of the $x$ direction. (Note the film velocity at the two wires within the computational rectangle is 0 in the immersed boundary computation.) At inflow and outflow, the same film velocity profile $(0, V_0(x))$ (solution to problem (11) ) is specified. Here we assume that the film-filament 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. As we mentioned in the Introduction, the incompressible viscous Navier-Stokes equations 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 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 $\Delta t$ be the duration of time steps, 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 filament tension and unit tangent are defined at the "half-integer"points given by $s = (m + 1/2)\Delta s$. For any function $\phi(s)$ , let

$$(D_s\phi)(s) = \frac{\phi(s + \frac{\Delta s}{2}) - \phi(s - \frac{\Delta s}{2})}{\Delta s} \tag{12}$$

Then make the definitions:

$$T^n = K_s(|D_s\mathbf{X}^n| - 1) \tag{13}$$

$$\hat{\boldsymbol{\tau}}^n = \frac{D_s\mathbf{X}^n}{|D_s\mathbf{X}^n|} \tag{14}$$

Both of which hold for $s = (m + 1/2)\Delta s$. Finally, we define $\mathbf{F}^n$ at the points $s = m\Delta t$ using $T^n$ and $\hat{\boldsymbol{\tau}}^n$:

$$\mathbf{F}^n_s = D_s(T^n\hat{\boldsymbol{\tau}}^n) \tag{15}$$

Note that $\mathbf{F}^n$ is defined at the same points as $\mathbf{X}^n$. We discretize the bending energy and the corresponding bending force as follows:

$$E_b = \frac{1}{2}K_b\sum_m |D_sD_s\mathbf{X}|^2\Delta s = \frac{1}{2}K_b\sum_{m=2}^{n_f-1}[\frac{|\mathbf{X}_{m+1} + \mathbf{X}_{m-1} - 2\mathbf{X}_m|^2}{(\Delta s)^4}]\Delta s \tag{16}$$

$$(\mathbf{F}_b)_l = \frac{K_b}{(\Delta s)^4}\sum_{m=2}^{n_f-1}(\mathbf{X}_{m+1} + \mathbf{X}_{m-1} - 2\mathbf{X}_m)(2\delta_{ml} - \delta_{m+1,l} - \delta_{m-1,l}) \tag{17}$$

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

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

Note that bending energy $E_b$ is not defined at the filament endpoints. This is why the two endpoints are left out in the summation of equations (16) and (17). But the contribution of the endpoints to the total bending energy is *not* left out: $\mathbf{X}_1$ and $\mathbf{X}_{n_f}$ appear in the computation of $E_b$.

The three integral relations can be discretized as follows:

$$\mathbf{f}^n(\mathbf{x}) = \sum_s \mathbf{F}^n(s)\delta_h(\mathbf{x} - \mathbf{X}^n(s))\Delta s \tag{18}$$

$$\rho^n(\mathbf{x}) = \rho_0 + \sum_s M\delta_h(\mathbf{x} - \mathbf{X}^n(s))\Delta s \tag{19}$$

$$\mathbf{U}^{n+1}(s) = \sum_{\mathbf{x}} \mathbf{u}^{n+1}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^n(s))h^2 \tag{20}$$

Here the notation $\sum\limits_s$ means the sum over all the discrete collection of points of the form $s = m\Delta s$, where $m$ is integer. The notation $\sum\limits_{\mathbf{x}}$ 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 $\delta_h$ is a smoothed approximation of the two dimensional Dirac $\delta$ function. In our computation, we choose the following

$\delta_h$:

$$\delta_h(\mathbf{x}) = h^{-2}\phi(\frac{x}{h})\phi(\frac{y}{h}) \tag{21}$$

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

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

See [7] for details regarding choosing $\phi(r)$. Note the support of the $\delta_h$ is a square with width $4h$ at each point instead of a circle with diameter of $4h$.

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

$$\frac{\mathbf{X}^{n+1}(s) - \mathbf{X}^n(s)}{\Delta t} = \mathbf{U}^{n+1}(s) \tag{22}$$

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 discretization of Navier-Stokes equations. First we state the following definitions:

$$(D^0_\alpha)(\mathbf{x}) = \frac{\phi(\mathbf{x} + h\hat{\mathbf{e}}_\alpha) - \phi(\mathbf{x} - h\hat{\mathbf{e}}_\alpha)}{2h} \tag{23}$$

$$(D_\alpha^+)(\mathbf{x}) = \frac{\phi(\mathbf{x} + h\hat{\mathbf{e}}_\alpha) - \phi(\mathbf{x})}{h} \tag{24}$$

$$(D_\alpha^-)(\mathbf{x}) = \frac{\phi(\mathbf{x}) - \phi(\mathbf{x} - h\hat{\mathbf{e}}_\alpha)}{h} \tag{25}$$

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 $\nabla$, and $\sum_{\alpha=1}^{2} D_\alpha^+ D_\alpha^-$ is a 5-point difference approximation to the Laplace operator $\boldsymbol{\Delta}$.

There are many numerical schemes for the incompressible Navier-Stokes equations; here we employ a projection method, which is a fractional step scheme. Since the pioneering work by Chorin [30] [31], a lot of work has been done on projection methods. We refer readers interested in these methods to the following papers and references therein: [32] [33] [34] [35] [36] [37] [38]. Our scheme differs from the original projection method [30] [31] in the treatment of the nonlinear term, which is explicit and skew-symmetric. The motivation for using skew-symmetric method instead of upwind differencing (as was used in many immersed boundary computations.) is that the skew-symmetric scheme has the desirable property that $\frac{d}{dt}||\mathbf{u}||_{L_2}^2 = 0$, (See [27]) , which guarantees conservation of kinetic energy of the soap film.

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:

$$\rho^n(\frac{\tilde{\mathbf{u}}_k^{n+1}-\mathbf{u}_k^n}{\triangle t}+\frac{1}{2}(\mathbf{u}\cdot\mathbf{D}^0\mathbf{u}_k+\mathbf{D}^0\cdot(\mathbf{u}\mathbf{u}_k))^n) = \mu\sum_{\beta=1}^{2} D_\beta^+ D_\beta^- \tilde{\mathbf{u}}_k^{n+1}+\mathbf{f}_k^n-\lambda\tilde{\mathbf{u}}_k^{n+1}-\rho^n g\hat{\mathbf{e}}_2 \tag{26}$$

for $k = 1$ and 2; where $\mathbf{u}_k$ is the $k^{th}$ component of velocity $\mathbf{u}$ and similarly for the components of any vectors. Note 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. Then we update (project) the velocity field with the pressure gradient and make the velocity be divergence free.

$$\rho^n(\frac{\mathbf{u}^{n+1}-\tilde{\mathbf{u}}^{n+1}}{\Delta t}) = -\mathbf{D}^0 p^{n+1} \tag{27}$$

$$\mathbf{D}^0\cdot\mathbf{u}^{n+1} = 0 \tag{28}$$

To see the relationship between this scheme and the Navier-Stokes equations, and in particular to see the meaning of $p^{n+1}$, add equations (26) and (27), and note that in the time-derivative term the intermediate velocity field $\tilde{\mathbf{u}}^{n+1}$ cancels out. The summed equation (i.e., the equation that is the result of adding equations 26 and 27) is in fact a discretization of the first (momentum) equation of the Navier-Stokes equations, with the slightly peculiar feature that the viscous (and air-resistance) terms on the right-hand side are evaluated neither at $\mathbf{u}^n$ nor at $\mathbf{u}^{n+1}$ but instead at $\tilde{\mathbf{u}}^{n+1}$, which however is within $O(\Delta t)$ of either $\mathbf{u}^n$ or $\mathbf{u}^{n+1}$, as can be seen directly from equations (26) and (27), respectively. In the summed equation, $p^{n+1}$ appears in the usual way as the pressure. Also note the variable coefficient $\rho^n$. Equations (27)-(28) define an orthogonal projection in the norm which uses $\rho$ as a weight function. To solve for pressure, we apply the central difference operator $\mathbf{D}^0$ on both sides of equations (27), then use the divergence free condition (28), thus we obtain a system of difference equations for pressure which is decoupled from the velocity field.

$$\mathbf{D}^0 \cdot (\frac{1}{\rho^n}\mathbf{D}^0 p^{n+1}) = \frac{\mathbf{D}^0 \cdot \tilde{\mathbf{u}}^{n+1}}{\Delta t} \tag{29}$$

Note that equations (29) 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 (26) and (29), both of which contain the non-constant coefficient $\rho^n$ (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 (26) are solved for $\tilde{\mathbf{u}}^{n+1}$ , and then with $\tilde{\mathbf{u}}^{n+1}$ in hand, equation (29) is solved for $p^{n+1}$ . Finally the velocity field $\mathbf{u}^{n+1}$ is calculated from equations (27). This completes the computations at each time step.

The multigrid technique solves a problem on a series of gradually coarsened grids instead of on a single grid. In our computation a 7-grid V-cycle is used with the finest grid 256 by 512 and the coarsest grid 4 by 8. We use red-black Gauss-Seidel ordering , apply the full-weighting scheme for residual restriction from fine grid to the next coarse grid, and employ linear interpolation to transfer data back from coarse grid to the next fine grid. We want to point out that using the simple injection for density in transferring $\rho^n(\mathbf{x})$ from $\Omega^h$ (the finest grid) to $\Omega^{lh}$ ($l = 2, 4, 8, 16, 32, 64$) results in a rather slowly convergent multigrid algorithm; instead, it is much better to use the following way to define $\rho^n_{lh}$, the density on $\Omega^{lh}$ at time n:

$$\rho^n_{lh}(\mathbf{x}) = \rho_0 + \sum_s M\delta_{lh}(\mathbf{x} - \mathbf{X}^n(s))\Delta s \tag{30}$$

Here $lh$ means the grid whose meshwidth is $l$ times $h$, where $h$ is the meshwidth of the finest grid. The reason is that in the case of using the simple injection, the discretized residual equations (defined as $L_h e_h = r_h$, where the residual $r_h = f_h - L_h\tilde{u}_h$, the error $e_h = u_h - \tilde{u}_h$; here $u_h$ and $\tilde{u}_h$ are the exact and computed solutions to $L_h u_h = f_h$ respectively, which is some discretization of a linear PDE $Lu = f$.) 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 the solutions of discretized N-S equations on a sufficiently coarse grid may not have any physical meaning at all; it is the solutions to the residual equations on coarse grids that help convergence.) Note that the smoothed approximation of delta function in equation (30) 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.

In our multigrid solver the relative residual in $L_2$ norm can converge to $10^{-13}$ i.e. $\frac{||r_h||_{L_2}}{||f_h||_{L_2}} = C \times 10^{-13}$, here $1.0 \leq C < 10.0$. In our simulation, the convergence criterion $\frac{||r_h||_{L_2}}{||f_h||_{L_2}} \leq 10^{-6}$ is used instead, because of the existence of discretization errors, which we believe are of the order $O(\Delta t, h^2)$. The numerical solutions are plugged back into the discretized N-S equations and it is found that in each time step we have $||L_h u_h - f_h||_\infty \leq 10^{-6}$. This ends the description of our numerical method.

## 4. Simulation Results

The parameters of our simulation are shown in Table 1. The dynamical viscosity $\mu$ 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 primary reason for doing this is to avoid computing in a regime in which numerical viscosity (which is mesh-width and flow dependent) completely dominates the physical viscosity so that the equations are, in effect, no longer the Navier-Stokes equations.

At the meshwidths used in our computations, we can reliably resolve flows in which the Reynolds numbers are on the order of a few hundred, and this seems to be high enough to reproduce the flapping behavior of the filament with considerable fidelity. We do not investigate the effect of Reynolds number on the film-filament system in this paper. The mass of filament is twice that of 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 $\mu m$. The length of the film in our computation is shorter than in the experiment, 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.

Most of the simulations ran up to 0.2 second (about 10 times of the characteristic time scale of the system, which can be estimated as the reciprocal of the flapping frequency.) except for some runs, which were used to check the flapping is truly periodic in time, lasted for 2 seconds.

Figure 2 through Figure 5 show our computer simulation of the system consisting of a flexible filament in a flowing soap film. Two different visualization techniques 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 panels in each figure flow is from top to bottom (driven by gravity, working against air resistance) at an inflow velocity equal to the film terminal velocity profile $V_0(x)$. The width of the channel is 8.5 cm, the height of the channel is 17 cm.

Figure 2 (top panel) shows the 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. 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.

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.

Figure 2 (bottom panel) has the same parameters as in Figure 2 (top panel) except that now the filament has nonzero mass. Figure 3 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 2.1 cm. The flapping is self-sustained and periodic in time.

We also did computations with different filament mass. The peak-to-peak amplitude of the free end of filament with different filament mass are listed in Table 2 for two cases: A) inflow velocity 200 $cm/sec$, filament length 2 $cm$; B) inflow velocity 280 $cm/sec$, filament length 3 $cm$. We can see that within a certain range of mass, the flapping amplitude increases with the filament mass.

Figure 4 exhibits 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 1. The only difference of the parameters in these two simulations is the initial perturbation: on the top panel in Figure 4 the initial perturbation is 1% of the filament length, while on the bottom panel in Figure 4 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" oscillation 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 panel in Figure 4.) In the case of large perturbation (bottom panel in Figure 4 and Figure 5), the filament quickly sets into its sustained periodical flapping state after 1 or 2 oscillations. The flapping frequency is about 38 Hz, the amplitude is about 1.5 cm. Each vortex is shed from the free end of the filament by each stroke, and this forms a "street"of alternating vortices 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). Note that in this case (inflow velocity equals 200 $cm/sec$), there is a critical length of about 16 $mm$ below which the filament always returns to its static state independent of the magnitude of the initial perturbation. (The critical length has not yet been determined in the laboratory for these particular flow conditions.) One feature of Zhang's experiment [1] 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 resolve 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.

## 5. Summary and Conclusion

Though the 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 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 (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.

There may be an upper critical length above which only the flapping state is stable. We have not investigated this since it is not likely possible to generate numerically an arbitrarily small perturbation, which is needed to determine the possible upper critical length. The laboratory experiment does not yet determine the possible upper critical length either.

In choosing parameters for these computer simulations, we have closely followed the experimental data of [1], 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 the same result as that of experiment, not only qualitative 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 of this problem. (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-dimensional parameters might be important. Although we have not investigated this in detail, the necessity of filament mass for flapping suggests that the dimensionless filament mass is an important parameter of the problem. One way to express this is in terms of

$$Fm = \frac{ML}{\rho_0 L^2} \tag{31}$$

where $M$ is the filament mass density (mass per unit length), $L$ is the filament length and $\rho_0$ is the density (mass per unit area) of the flowing soap film. This parameter has already been used by Shelley [2], who calls it $S1$, in the stability analysis of filament flapping. In our simulation $Fm$ is in the range

$[0.445, 0.667]$. There could be other non-dimensional parameters which may be important for this problem. We have not yet studied how the behavior of the film-filament system depends on these non-dimensional parameters, such as Reynolds number and $F_m$. Numerically testing the sensitivity of the system to Reynolds number and determining the influence of the non-dimensional parameters on the system would be a nice future work.

**Acknowledgment**

The authors are indebted to the National Science Foundation (USA) for support of this work under KDI research grant DMS-9980069. We also thank Jun Zhang for many discussions of the experiment, thank Peter Schmid for discussion of multigrid method, thank Michael Shelley, Stephen Childress, and Daniel Forger for helpful discussions of the problem, and David McQueen for help in using computers.

# References

[1] 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).

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

[3] 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.

[4] 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).

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

[6] 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).

[7] 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.

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

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

[10] 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.

[11] 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).

[12] 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).

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

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

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

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

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

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

[19] 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).

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

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

[22] 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).

[23] 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).

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

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

[26] 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).

[27] 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).

[28] 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.

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

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

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

[32] J. B. Bell, P. Colella, and H. M. Glaz, A second order projection method for the incompressible Navier-Stokes equations, *J. Comput. Phys.,* **85**, 257 (1989).

[33] J. B. Bell, P. Colella, and L. H. Howell, An efficient second-order projection method for viscous incompressible flow, in *Proceedings of the Tenth AIAA Computational Fluid Dynamics Conference, AIAA,*June 1991, pp. 360.

[34] O. Botella, On the solution of the Navier-Stokes equations using Chebyshev projection schemes with third-order accuracy in time, *Computer & Fluids*, **26**, 107 (1997).

[35] W. E and J. Guo Liu, Projection method I: Convergence and numerical boundary layers, *SIAM J. Numer. Anal.*, **32**, 1017 (1995).

[36] W. E and J. Guo Liu, Projection method II: Godunov-Ryabenki analysis, *SIAM J. Numer. Anal.*, **33**, 1597 (1996).

[37] J. B. Perot, An analysis of the fractional step method, *J. Comput. Phys.*, **108**, 51 (1993).

[38] D. L. Brown, R. Cortez and M. L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, **168** (2), 464 (2001).

[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.

**Figure Legends:**

Figure 1: The experimental setup (by courtesy of the experimentalist).

Figure 2: A *massless* filament (top panel) and a filament with mass (bottom panel) in a flowing soap film; $time = 0.192\ sec$.

Figure 3: A filament with mass in a flowing soap film: in top panel,$time = 0.097\ sec$; in bottom panel, $time = 0.127\ sec$.

Figure 4: Bi-stability of the filament in a flowing soap film; $time = 0.152\ sec$. The only difference between the computations shown in the top and bottom panels is the filament initial condition: A small perturbation from equilibrium was used in the top panel, and the result was a return to equilibrium. In the bottom panel, a large perturbation was used, and the result was sustained filament flapping.

Figure 5: The flapping filament of the bottom panel of Figure 4 at two later times: $0.175\ sec$ in top panel and $0.185\ sec$ in the bottom panel.

| | |
|---|---|
| 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 | $2 - 3\ cm$ |
| 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 of the film | $17\ cm$ |

Table 1: Parameters of the simulation

| filament density | amplitude (case A) | amplitude (case B) |
|---|---|---|
| $2 \times 10^{-4}$ | 9.8 | 0 |
| $4 \times 10^{-4}$ | 21 | 15 |
| $6 \times 10^{-4}$ | 24.5 | 18.9 |
| $8 \times 10^{-4}$ | 27 | 23.1 |
| $1 \times 10^{-3}$ | 28 | 25.2 |

Table 2: Flapping amplitude of the filament with different filament mass; case A: inflow velocity is 280 $cm/sec$, the filament length is 3 $cm$ ; case B: inflow velocity is 200 $cm/sec$, the filament length is 2 $cm$. The units for filament density and flapping amplitude are $g/cm$ and $mm$, respectively.

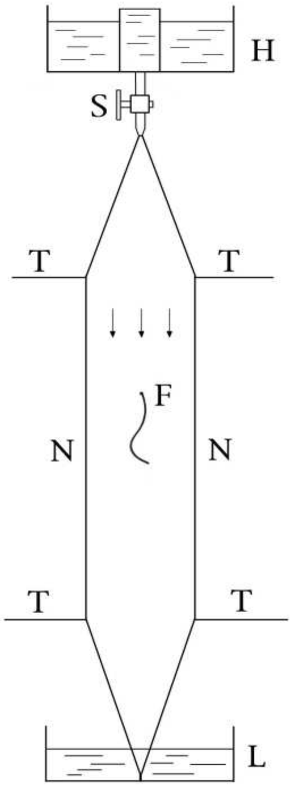


Figure 1: The experimental setup (by courtesy of the experimentalist)

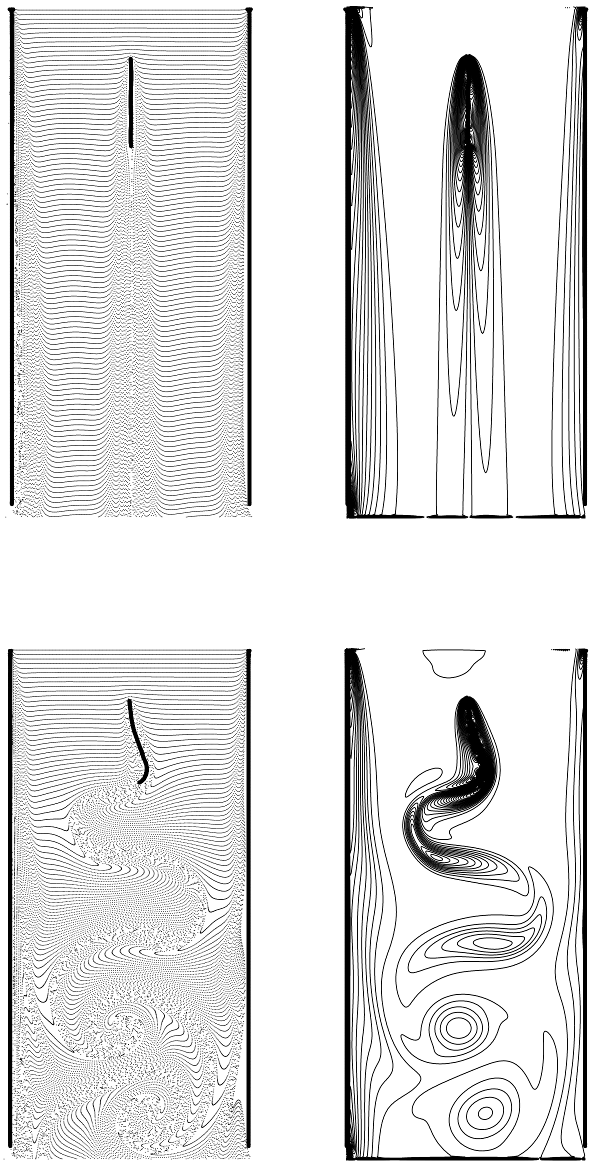

Figure 2: A *massless* filament (top panel) and a filament with mass (bottom panel) in a flowing soap film; $time = 0.192$ *sec.*

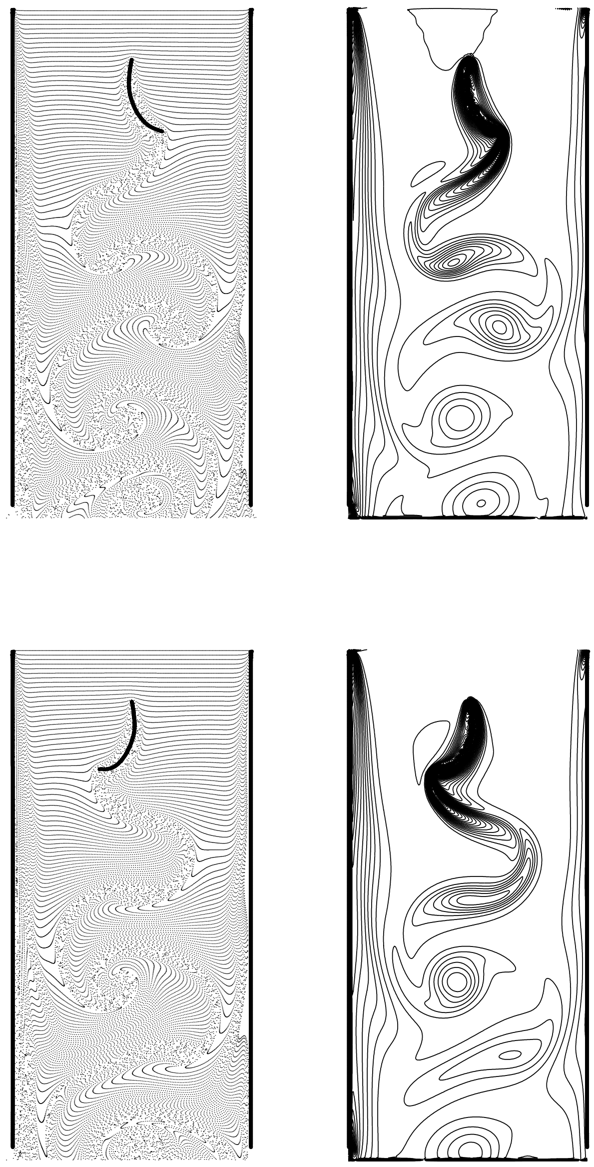

Figure 3: A filament with mass in a flowing soap film: in top panel, $time = 0.097\ sec$; in bottom panel, $time = 0.127\ sec$.

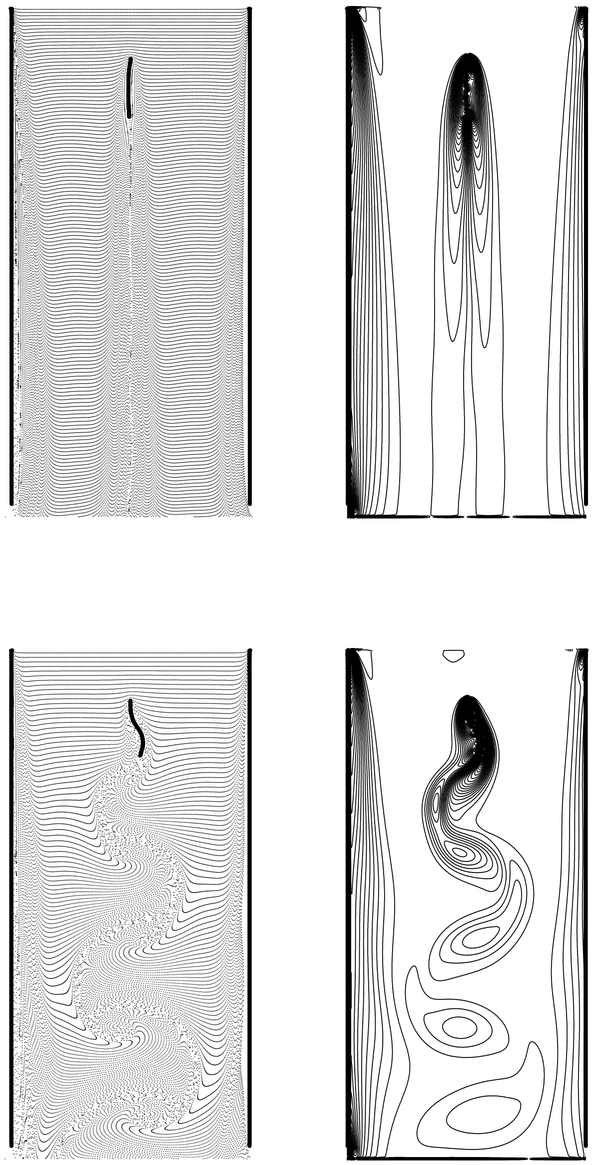

Figure 4: Bi-stability of the filament in a flowing soap film; $time = 0.152$ *sec.* The only difference between the computations shown in the top and bottom panels is the filament initial condition: A small perturbation from equilibrium was used in the top panel, and the result was a return to equilibrium. In the bottom panel, a large perturbation was used, and the result was sustained filament flapping.

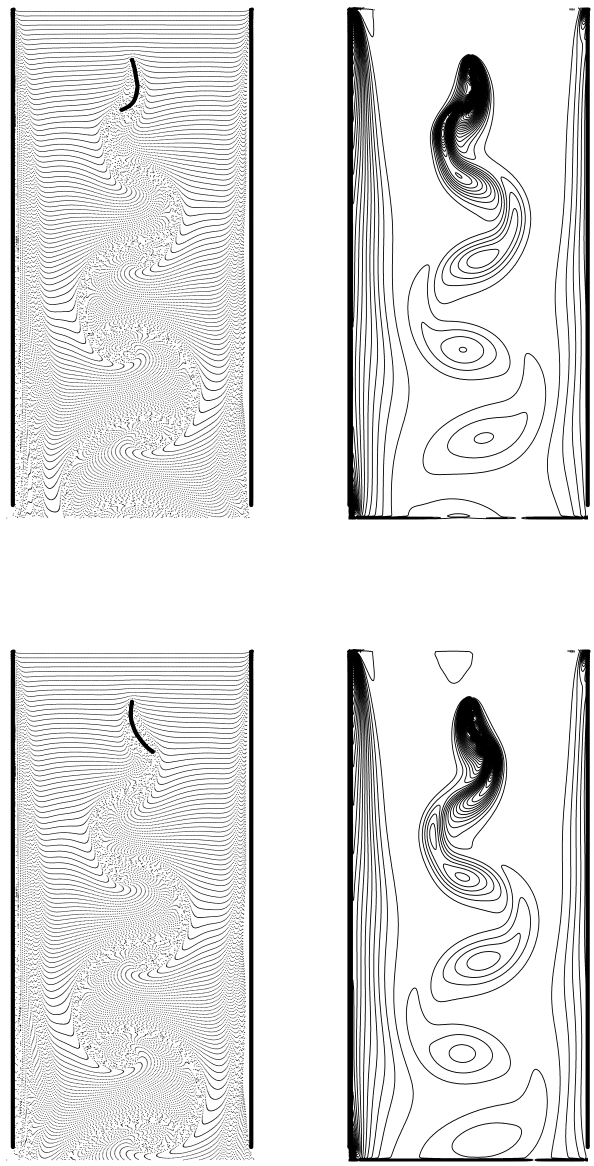

Figure 5: The flapping filament of the bottom panel of Figure 4 at two later times: 0.175 *sec* in top panel and 0.185 *sec* in the bottom panel.