Simon Fraser University

APMA 990

Immersed Boundary Method and Applications

Vocal fold dynamics using the penalty immersed boundary method

Author:

Reynaldo J. Arteaga

Professor:

Dr. John Stockie

April 12, 2013

Contents

1 Abstract

In this paper, a proposal to model human vocal folds in two-dimensions is accomplished by applying the penalty immersed boundary (pIB) method which was originally introduced by Peskin [3] and further detailed by Kim and Peskin [2]. Modification of an existing immersed boundary (IB) solver originally written by Froese and Wiens [4] is performed to implement the pIB method. The main results will consist of properly implementing the pIB method, along with a convergence study to verify the model indeed solves the underlying set of equations of the model for an ellipse structure. Simulation of the results shown by Kim and Peskin [2] along with vorticity graphs illustrated by Duncan et. al [1] is confirmed. Complementing the vorticity graphs, a two-dimensional simulation of air passing through the vocal folds replicating true physical characteristics is established.

2 Introduction

The purpose for studying this problem is to simulate human vocal folds and study the different effects of varying volumes of air that pass through the vocal chords. The effects of speaking loudly is illustrated through vocal chords that become thick and swollen due to a large continuous force colliding with the vocal folds. Moreover, a comparison of a healthy set of vocal folds against a defective pair gives an indication of how the speech is affected. The study of human vocal folds has various alternative methods to compute its solutions, such as the finite element method, lumped parameter models and the immersed boundary (IB) method. With regards to this course, IB Method and Applications, the study of vocal folds is an excellent employment of a variation of the method, i.e. pIB. The ease of creating different geometries makes it simple to simulate various models of vocal folds that have some minor differences. It is challenging from the perspective of the course, since the fluid of the model is air, which has a high Reynolds number ranging from 100 to 1000. This in consequence leads to taking a very small time step, or adjustments in the code to compensate for the high Reynolds number. Careful adjustments such that the previous IB code, now applies the pIB method and the two massless and massive boundaries properly communicate is discussed in § 3. This is in turn related to the previous structure of the code and its flexibility. Lastly, the force is an aggregate of new forces discussed in § 3.3.1 that are introduced due to correspondence with physical characteristics of vocal folds. Hence, there are simple and complex adjustments needed to replicate various realistic human vocal folds with different structures.

3 Model

In this section there is a discussion of the theory behind the pIB method and its physical significance of its parameters.

3.1 Summary of pIB method

The pIB method when applied to the vocal fold model involves three major components: (a) the immersed material, (b) air and (c) the interface between

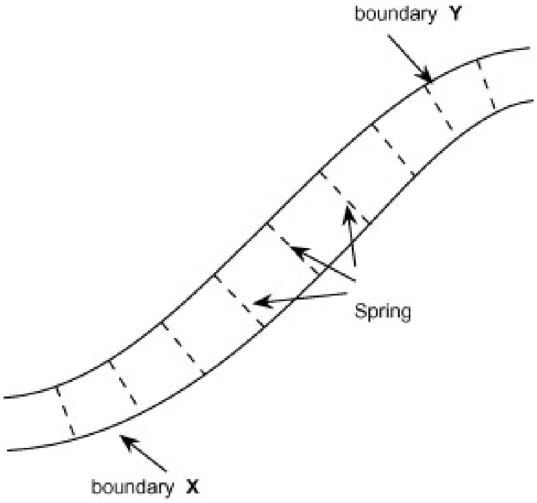


Figure 1: Massive (Y) and massless (X) boundaries are linked together by a system of very stiff springs of which the rest length is zero (left).

them. The immersed material involves the pIB method which behaves very closely to the usual IB method, however its defining characteristic is the separation of the single boundary into two components: massless boundary, and massive boundary. A pictorial view of the two boundaries are shown through a fiber in Fig. 1 .

The application of interest (vocal folds) involves a very small viscosity in which case, the mass of the immersed boundary cannot be considered negligible. The purpose for introducing a fictious boundary is an alternative approach to simply placing the mass of the immersed boundary directly onto the fluid grid points. Incorporating the mass as mentioned involves having a variable density and the previous IB solvers would no longer be applicable since the density is taken to be constant. This approach is still possible to uphold but does not take advantage of previously written IB solvers. However, by introducing an artificial boundary that simply drags the mass along close to the massless boundary, this allows the system to take advantage of previously existing IB solvers and only need to include the data for storing the new boundary. A more in depth description of the model accentuates these distinctions.

3.2 Theory of continuous model

The air component of the model is taken on by the Navier-Stokes equations. The Navier-Stokes equations for a viscous, forced, incompressible fluid along with the other equations for the pIB method consists of the following:

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

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

$$\mathbf{f}(\mathbf{x},t) = \int_\Gamma \mathbf{F}(r,s,t)\delta[\mathbf{x}-\mathbf{X}(r,s,t)]drds, \tag{3}$$

$$\frac{\partial \mathbf{X}}{\partial t}(r,s,t) = \mathbf{u}[\mathbf{X}(r,s,t),t] = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta[\mathbf{x}-\mathbf{X}(r,s,t)]d\mathbf{x}, \tag{4}$$

$$\mathbf{F} = \mathbf{F}_E + \mathbf{F}_K, \tag{5}$$

$$\mathbf{F}_E = -\frac{\partial E}{\partial \mathbf{X}}, \tag{6}$$

$$\mathbf{F}_K(r,s,t) = K[\mathbf{Y}(r,s,t) - \mathbf{X}(r,s,t)], \tag{7}$$

$$M(r,s)\frac{\partial^2 \mathbf{Y}}{\partial t^2} = -\mathbf{F}_K(r,s,t), \tag{8}$$

where $\mathbf{x}, \rho, \mathbf{u}, p$ and μ are respectively the fixed Cartesian coordinates, density, velocity, pressure and viscosity of the air. The term $\mathbf{f}$ is the force per unit volume applied by the immersed boundary to the fluid (force per unit area in the current two-dimensional model), and E is the energy functional taken from Kim and Peskin [2].

$$\mathbf{f}(\mathbf{x},t) = \int_{\Gamma} \mathbf{F}(r,s,t)\delta^{(2)}\left[\mathbf{x}-\mathbf{X}(r,s,t)\right] drds, \tag{9}$$

$$E(\mathbf{X},t) = \frac{1}{2}c_s \int \left(\left| \frac{\partial \mathbf{X}}{\partial s} \right| - 1 \right)^2 ds + \frac{1}{2}c_b \int \left| \frac{\partial^2 \mathbf{X}}{\partial^2 s} \right|^2 ds, \tag{10}$$

where $\delta^{(2)}$ is the two-dimensional Dirac delta distribution. $\mathbf{X}(r,s,t), \mathbf{Y}(r,s,t)$, represents the position vector of a point on the vocal fold for the massless and massive boundary respectively, parametrized by r and s at time t. $M(s,t)$ is the mass density of the massive boundary $\mathbf{Y}(r,s,t)$. $\mathbf{F}_E$ is composed of two elastic properties: 1) stretching and compression force and 2) a bending force for the massless boundary, and $\mathbf{F}_k$ is the resting force on the massless boundary, c_s, c_b are constants, and K is a large spring constant that determines the stiffness between the massive and massless boundary. Note that the energy functional changes depending on which forces are taken into consideration for the vocal folds. The third component of the pIB method involves the forcing terms which is the major component of the interface communication between the air and the vocal fold and tissue forces.

3.3 Theory of discrete model

The discrete model of the pIB method constitutes of the forcing terms in the immersed boundary, and the proper implementation of the various steps to solving the discretization of the system of equations (1) -(8).

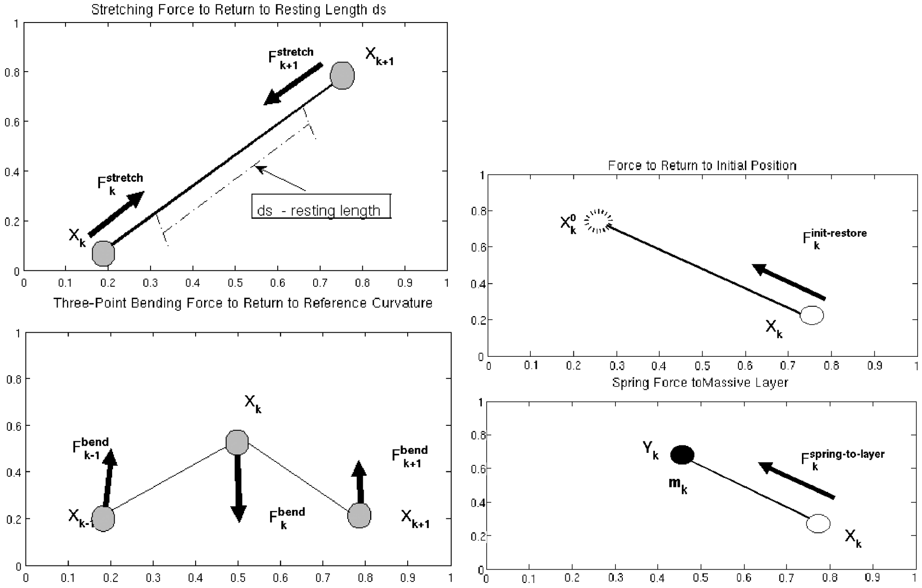


Figure 2: Forcing Terms

3.3.1 Forces

There are different forces acting on the vocal folds which aggregate to the total force. In this model, these forces consist of:

a) a stretching force (Fig. 2):

$$\mathbf{F}_{stretch}(k) = \frac{[\tau(k+1) * \hat{e}(k+1/2) - \tau(k) * \hat{e}(k-1/2)]}{ds},$$

$$\tau(k) = s_{stretch}(k) * [\ell(k) - ds],$$

$$\ell(k) = |\mathbf{X}(k) - \mathbf{X}(k-1)|,$$

where τ is the tension at particle $k, \hat{e}(k+1/2)$ is a unit vector centered between the particles pointing from particle $k-1$ to particle k, and $\ell(k)$ is the distance between particles $k-1$ and k. The parameter $s_{stretch}(k)$ is the stretching force constant. $\hat{e}(k+1/2)$ is a unit vector centered between the particles pointing from particle $k-1$ to k.

b) a bending force (Fig. 2):

$$\begin{aligned}\mathbf{F}_{bending}(k) &= s_{bending}(k) * [curv[\mathbf{X}(k)] - curv[\mathbf{X}_0(k)]]\,, \\ curv[\mathbf{X}(k)] &= \frac{[\mathbf{X}(k-1) - 2 * \mathbf{X}(k) + \mathbf{X}(k+1)]}{ds^2},\end{aligned}$$

where $s_{bending}(k)$ is the bending force constant, $curv[\mathbf{X}(k)]$is an approximation to the second derivative of the kth particle position along the boundary and $\mathbf{X}_0(k)$ is the initial position on the kth particle.

c) a fixed-point restorative force (Fig. 2):

$$\mathbf{F}_{restore}(k) = -s_{restore}(k) * [\mathbf{X}(k) - \mathbf{Y}(k)],$$

where $s_{restore}(k)$ is the force constant, $\mathbf{X}(k)$ is the kth massless particle position, and $\mathbf{Y}(k)$ is the position of the inner massive particle. The restoration force essentially counters the spring force between the massive and massless boundary. The restoration force decreases the oscillation of the massive boundary about the massless boundary. Since the computation has the same form of the spring force, numerically, these two steps are completed in one.

d) a spring force:

$$\mathbf{F}_M(k) = -s_M(k) * [\mathbf{X}(k) - \mathbf{Y}(k)],$$

where $s_M(k)$ the spring constant per unit length of the spring connecting the kth massless particle at $\mathbf{X}(k)$ to the kth massive particle at $\mathbf{Y}(k)$.

Totaling the forces gives us:

$$\mathbf{F}(k) = \mathbf{F}_{stretch}(k) + \mathbf{F}_{bending}(k) + \mathbf{F}_{restore}(k) + \mathbf{F}_M(k).$$

The manner in which both the IB and pIB method account for the forcing terms is to spread the forcing over the fluid domain (by means of the delta distribution) once it has calculated the sum of the forces generated by the immersed boundary. A step by step description of the overall scheme of the pIB method can be given now that the forcing term has a precise definition.

3.3.2 Numerical method

McQueen and Peskin [1] have an outline for solving the system ((1)- (8)) using the pIB method. A single step in time from t^n to $t^{n+1} = t^n + dt$, is outlined as follows:

General Outline of Numerical Implementation

1. given the position of all boundary points and the velocity field at time t^n
2. update particle positions $\mathbf{X}(k)$ to time level $t^{n+1/2}$
3. compute the force per unit length $\mathbf{F}(k)$ on each vocal fold point k due to adjacent vocal fold points given by $\mathbf{F}_{internal}[\mathbf{X}(k) + \mathbf{F}_M(k)]$
4. spread this force to the Eulerian grid giving the force density exerted on the fluid
5. update the Eulerian fluid velocity and pressure to time $t^{n+1/2}$
6. use the no-slip condition to advance all the vocal fold point positions to time t^{n+1}

7. solve the forced Navier Stokes equations to find the velocity and pressure field at time $t^{n+1} = t^n + dt$ and advance the massive points at $\mathbf{Y}(k)$ to t^{n+1}

8. advance the massive points at $\mathbf{Y}(k)$ to t^{n+1}

Recall that within each step lies various numerical methods and procedures that are being applied in order to obtain the results needed at the current time step. I will describe these underlying numerical computations now. Update the particle positions ($\mathbf{X}^n(r, s) \equiv \mathbf{X}(r, s, n\Delta t)$ where Δt is the duration of the time step) of both massive and massless boundaries at $n + 1/2$ by:

$$\begin{aligned}
\frac{\mathbf{X}^{n+1/2} - \mathbf{X}^n}{\Delta t/2} &= \sum_{\mathbf{x}} \mathbf{u}^n(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^n(k))h^2, \\
\frac{\mathbf{Y}^{n+1/2} - \mathbf{Y}^n}{\Delta t/2} &= \mathbf{V}^n,
\end{aligned}$$

where $\mathbf{u}, \mathbf{V}$ is the velocity vector of the massless and massive boundary respectively. The Delta distribution is defined in [2]:

$$\delta_h(\mathbf{x}) = \frac{1}{h^2}\phi\left(\frac{x_1}{h}\right)\phi\left(\frac{x_2}{h}\right),$$

where $\mathbf{x} = (x_1, x_2)$ and

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

Next step involves calculating the force density $\mathbf{F}^{n+1/2}$ applied by the immersed boundary to the fluid. The force is composed of the sum of different forces that define the structure of the immersed boundary detailed in §3.3.1. After computing the sum of the forces, multiply this with the delta distribution which spreads the forces along the fiber as follows:

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

Then solve the Navier-Stokes equations:

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

for $i = 1, 2$ and $\mathbf{D}\cdot\mathbf{u}^{n+1/2} = 0$. When solving the Navier-Stokes equations, a crucial approach is through the use of the fast Fourier transform (FFT) algorithm since the equations have constant coefficients and can be uncoupled. This is a very fast and efficient method for solving the Navier Stokes equations. Lastly, calculate the velocity $\mathbf{V}^{n+1/2}$ of the massive boundary:

$$M\left(\frac{\mathbf{V}^{n+1/2} - \mathbf{V}^n}{\Delta t/2}\right) = -\mathbf{F}_M^{n+1/2}. \tag{11}$$

This concludes the half time step calculation of the model. Afterwards we proceed to computing the other half time step to evolve from $t^n \to t^{n+1}$. The numerical equations in consequence of the second portion of the time step are identical to the first half except for the time intervals change slightly.

$$\begin{aligned}
\frac{\mathbf{X}^{n+1}-\mathbf{X}^{n}}{\Delta t} &= \sum_{x}\mathbf{u}^{n+1/2}(x)\delta_h(\mathbf{x}-\mathbf{X}^{n+1/2}(k))h^2,\\
\frac{\mathbf{Y}^{n+1}-\mathbf{Y}^{n}}{\Delta t} &= \mathbf{V}^{n+1/2},\\
&\quad \rho\left[\frac{u_i^{n+1}-u_i^{n}}{\Delta t}+\frac{1}{2}[\mathbf{u}\cdot\mathbf{D}u_i+\mathbf{D}\cdot(\mathbf{u}u_i)]^{n+1/2}\right]+D_i\tilde{p}^{n+1/2}\\
&= \mu L(u_i^{n+1}+u_i^{n})+f_i^{n+1/2} \text{ for } i=1,2,\\
\mathbf{D}\cdot\mathbf{u}^{n+1/2} &= 0\\
M\left(\frac{\mathbf{V}^{n+1}-\mathbf{V}^{n}}{\Delta t}\right) &= -\mathbf{F}_M^{n+1/2}.
\end{aligned}$$

After this detailed description of the pIB numerical implementation, the coupling of the numerical and physical model is carefully depicted.

3.4 Physical model

In this section, a description of the physical properties of the vocal folds and how it is properly modeled by the pIB method is given. An illustration of the vocal folds with the pIB method is shown in Fig. 3. There are two walls that represent the vocal folds, and for each wall there is a massive and massless set of particles. The massless set of particles interact with the fluid, whereas the massive set of particles (which is very rigid) only interacts with the massless particles. Hence, the massless particles act as the vocal folds and the fluid, which is air in this case, passes through them leading to some vorticies in the narrow region of the vocal folds. The massive particles are used to contain the mass of the massless immersed boundary. The pIB method performs almost identically to that of the “usual” IB method

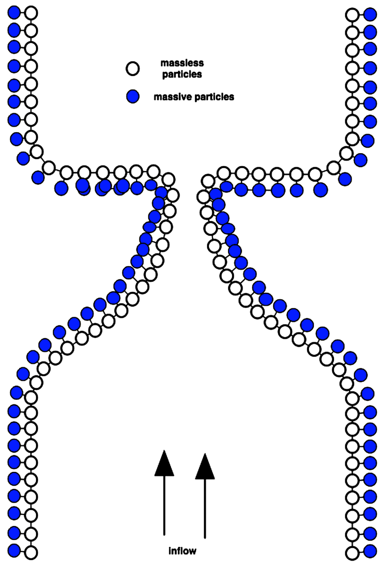


Figure 3: Vocal Fold Configuration under the pIB method

except the method also keeps track of the massive boundary. We can easily transform the structure of the vocal folds, along with adding necessary forces that simulate the true characteristics. We now move forward to the results of the pIB method on both a test case and the application of vocal fold simulation in two dimensions.

4 Results

I have adjusted the code so that the pIB method is being applied, and not the "usual" IB method. The adjustments involved incorporating the new spring force connecting the massless and massive boundaries together, along with solving for the velocity and position of the massive boundary (which previously did not exist in the code). Let us note that since the massive boundary consists of only a spring force connecting it to the massless boundary, we should expect the massive boundary to oscillate between the massless boundary. There is no tension spring force connecting the particles along the massive boundary together since these points are artificial and introducing this characteristic does not have any physical significance. (see §4.4) This explains the oscillatory behavior of the massive boundaries solution.

4.1 Ellipse test case

There are different standard test cases used to verify the pIB method is indeed computing the correct solution. In the literature [2], Kim and Peskin use the ellipse as a test problem to verify the pIB is correct and here the same trajectory is followed. The initial shape of the ellipse [2] is in the form:

$$\begin{aligned} \mathbf{X}(r,s,0) &= \frac{1}{2} + \left[\alpha + \gamma\left(r - \frac{1}{2}\right)\right]\cos(2\pi s), \\ &\quad \frac{1}{2} + \left[\beta + \gamma\left(r - \frac{1}{2}\right)\right]\sin(2\pi s). \end{aligned} \tag{12}$$

With constants $\alpha = 0.2, \beta = 0.25$ and $\gamma = 0.0625$. An image of the ellipse at an earlier time is seen in Fig. 4. The constants that pertain to the simulation are listed in Table 1.

From Table 1 we have taken a bending, stretching, and spring force into our problem. These forces aggregate to form the total sum of the forces, and use this within solving the pIB method. Theoretically we expect the ellipse to oscillate within the fluid and after some time T, begin to reach equilibrium, which is in the form of a circle. The pIB method does in fact

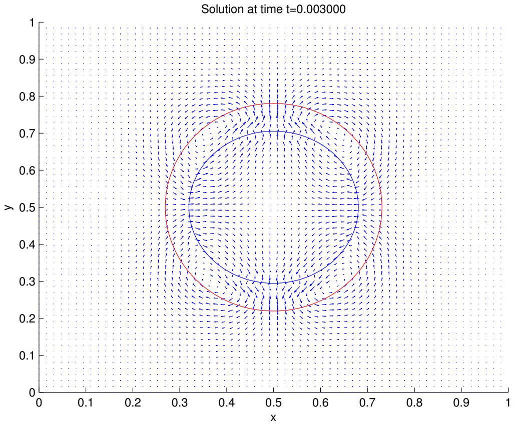


Figure 4: Initial data for pIB method with an ellipse structure

Parameter	Symbol (units)	Value
Number of grid points	Nx, Ny	64, 64
Number of IB points	$N_b(= 3 * Nx = 3 * Ny)$	192
Time step	dt(s)	1.0e-7
Final Time	T (s)	0.026
Bending force constant	s_{bend} (dyne)	10.0
Stretching force constant	$s_{stretch}$ (dyn/cm)	1.0e6
Spring force constant	s_M (dyn/cm^2)	1.0e6
Spring tension force constant	s_T (dyn/cm^2)	1.0e4

Table 1: List of initial conditions and parameters

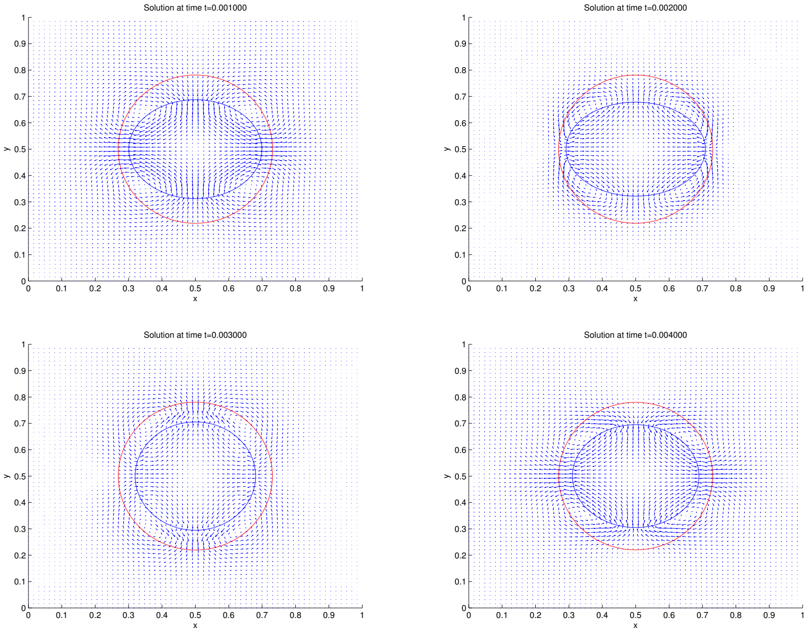


Figure 5: pIB method applied to ellipse test case

hold true to the theory as one can see in the sequence of subfigures in Fig. 5. The ellipse at first, oscillates due to the fluid, and eventually begins to settle and head towards equilibrium.

4.2 Convergence study

Since we have altered the formulation of the IB method, we verify the computational solution of the method, and confirm it remains convergent to the appropriate solution. Although Kim and Peskin [2] perform a rigorous convergence study taking $N = 64, 128, 256, 512$ the computational cost and time is far too large for the system available. In consequence, an application of Froese and Wiens [4] convergence test which illustrates the radius of the

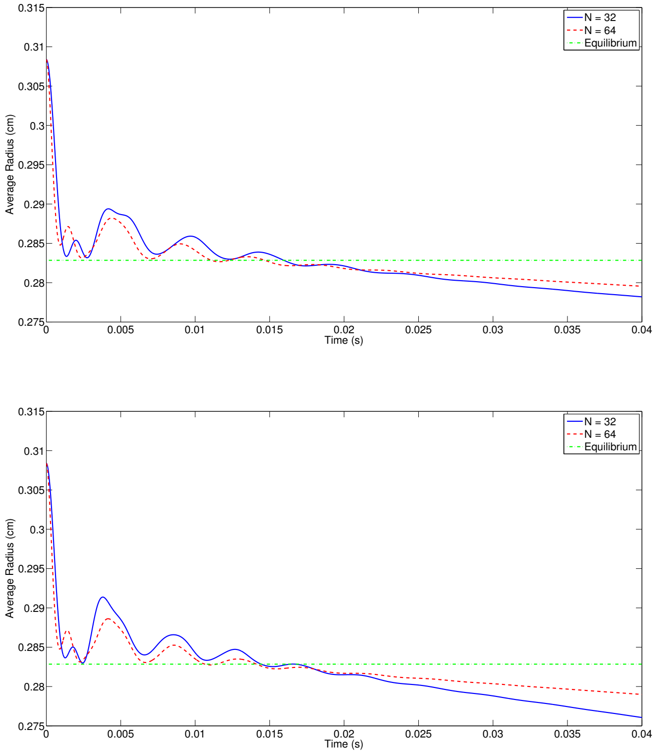


Figure 6: Top: IB method. Bottom: pIB method.

ellipse over time and the equilibrium state of the ellipse is implemented. Initial parameters are listed in Table 2.

The conclusions that are drawn from the graphs 6 are that the pIB method indeed converges to the equilibrium state. The two convergence plots of the IB and pIB method are almost identical. Given that the IB method is convergent, the graph illustrates the correctness of the convergence of the pIB. The explanation for why the ellipse begins to decay and not reach equilibrium is from the forcing terms pushing tangentially inward

Parameter	Value
density	1.0
viscosity	1.0
number of grid points	32,64
time step	0.04/N
total time	0.04
penalty constant	$10{\times}N^2$

Table 2: Data used in convergence study

into the center of the immersed boundary and the pressure is very high within the object as well. These two terms against each other causes a volume loss in the system and in consequence, the graph demonstrates a decay in the radius. However, the graph also demonstrates that with a finer mesh, the ellipse tends to equilibrium. Moreover, an implicit scheme of the IB method would fix this numerical instability.

4.3 Varying mass test case

In this section we give a deeper discussion of the main objective and relevance of the pIB method. The purpose of this test is to give some numerical results that demonstrate that the mass cannot be taken to be negligible as in the "usual" IB method. This is accomplished by testing the pIB method on an object with almost zero mass, ideal mass, and very large mass, with expectations of observing differences in the results. This test case is inspired by the analysis in [2], however here it is not comparing to equilibrium since

Parameter	Density	Viscosity	Δt	total T
Value	1.0	0.005	1e-5	0.1
Forcing Variables	Tether	Penalty Spring	Bend	Stretch
Value	1e3	10 $\times N^2$	10	1e5

Table 3: Data used in mass case study

the code does not conserve volume after a certain time T. In the simulations, $N = 32$ and the values taken for the mass of the immersed boundary is $M = \{10^{-4}, 0.5, 1.0, 10^4\}$ along with other initial conditions given in Table 3.

The results from Fig. 7 demonstrate that the value of the mass is an important characteristic of the problem. Changing the mass of the immersed boundary has a large effect on the solution. The graph demonstrates that for a large mass, the immersed boundary oscillates and has a larger radius than in equilibrium and does not settle. Looking at a smaller perturbation where $M = 1.0$ we obtain a larger volume loss than our ideal case of $M = 0.5$. The case of $M = 1.0$ derives from the tapered case taken in [2] at its maximum value. Lastly, take note that if the mass is almost negligible (this is the replicating the case of the "usual" IB method) the volume loss is much higher. This volume loss is a direct result of the low viscosity or high Reynold's number. Hence, the graphs illustrate the importance of monitoring the mass of the immersed boundary properly. When not done correctly, this can lead to drastic changes in the solution graphs.

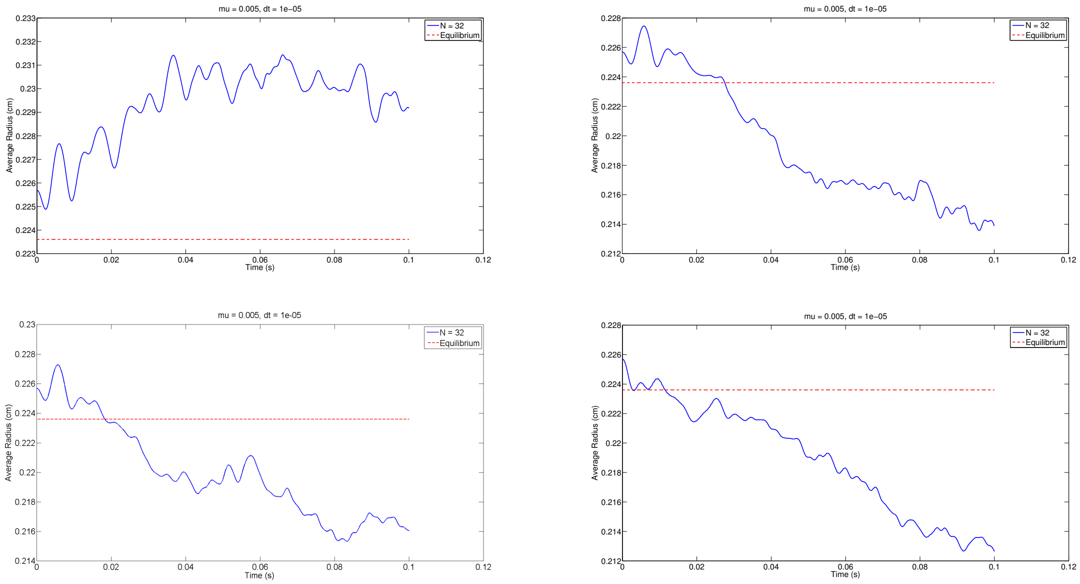
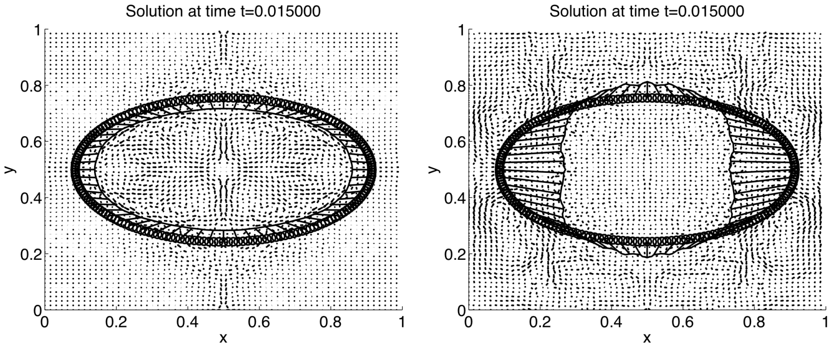


Figure 7: Top Left: $M = 10^4$. Top Right: $M = 1.0$. Bottom Left: $M = 0.5$. Bottom Right: $M = 10^{-4}$



Solution at time t=0.015000

Solution at time t=0.015000

Figure 8: Left: without tension. Right: with tension

4.4 Study of added characteristic to the massive boundary

The massive boundary has one property and that comes from the spring force attaching the boundaries together. Looking at some numerical results, we look closely at whether there should be some added component to the massive boundary to more accurately find the solution.

Firstly does the artificial boundary need to have tether points between its particles? The particles carry the mass of the true immersed boundary, so does it make sense that these added masses are dependent on their respective neighbouring particles? We take an implementation of this idea to see what the results will give. Fig. 8 and 9 both run simulations in the cases where there is and is not tension in the massive boundary, along with the condition of a moving immersed boundary or not to accentuate the differences. The plot shows the massive and massless boundary, and the connecting springs between the two boundaries.

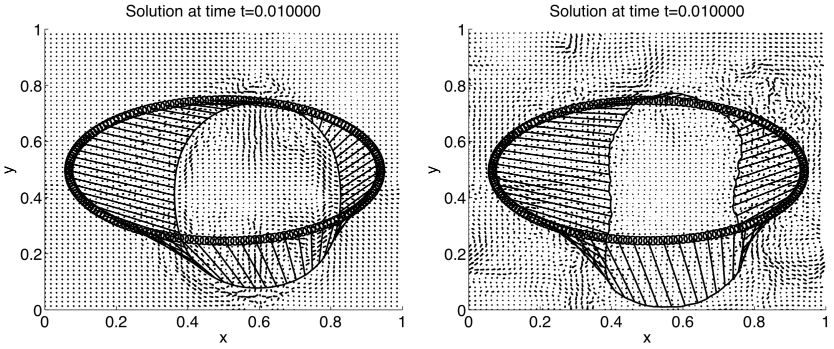


Figure 9: Left: without tension. Right: with tension

Observations from the plots that can be made are that attaching springs along the massive boundary, numerically, leads to an incorrect computation of the immersed boundary. The conjecture for this phenomenon derives firstly from the fact that this boundary is completely artificial and has no true physical significance. Hence, adding this tether property has no true meaning. Moreover, by attaching spring forces along the massive boundary, the mass of each point along the boundary begins to drag in other directions due to neighbouring forces and the corresponding mass no longer stays close to the fluid point. In conclusion, since this immersed boundary is artificial, it is not logical to place other fictitious properties on an artificial boundary.

4.5 Vocal fold simulation 1

Now we take the configuration to be two fibers that act as the vocal folds in the human throat. For simplicity I have modelled the vocal folds in two dimensions using the Matlab code of Froese and Wiens [4]. Fig. 3 gives the initial formation of the vocal folds, rotated counterclockwise 90, degrees and the explicit form is given by a gaussian function:

$$\mathbf{X}(r,s,0) = \left(r, D + A * \exp(B * (r-0.5)^2)\right), \tag{13}$$

with $D = 0.14, A = 1/\sqrt{2\pi}$ and $B = -20$. The airflow passes from left to right of the graph. The other wall is simply a reflection over the x-axis and translated upward.

Moreover, we have taken σ representing the spring force between particles in the vocal folds to vary on space. This choice of σ is justified because the horizontal part of the vocal folds does not move, however there is much more movement at the bump in the vocal folds. Hence, in order to replicate physical characteristics, we implement the choice of spring tension to be the same as the initial configuration, i.e. a Gaussian function.

Pressure and vorticity plots are given:

Fig. 10 and 11 show the vocal folds beginning to bend and stretch as the airflow passes through the vocal folds. This simulates physical characteristics and the results seen in Duncan et al. [1]. In all cases, the vocal folds portion in the center is expected to oscillate in the x-direction as the airflow passes through.

4.6 Vocal fold simulation 2

Moreover, in order to obtain a more concrete grasp of the air flow through the vocal folds, we will consider a new configuration. This configuration will mimic that of Duncan et al. [1]. The configuration allows us to study the affects of airflow passing through the vocal folds (left to right) more clearly

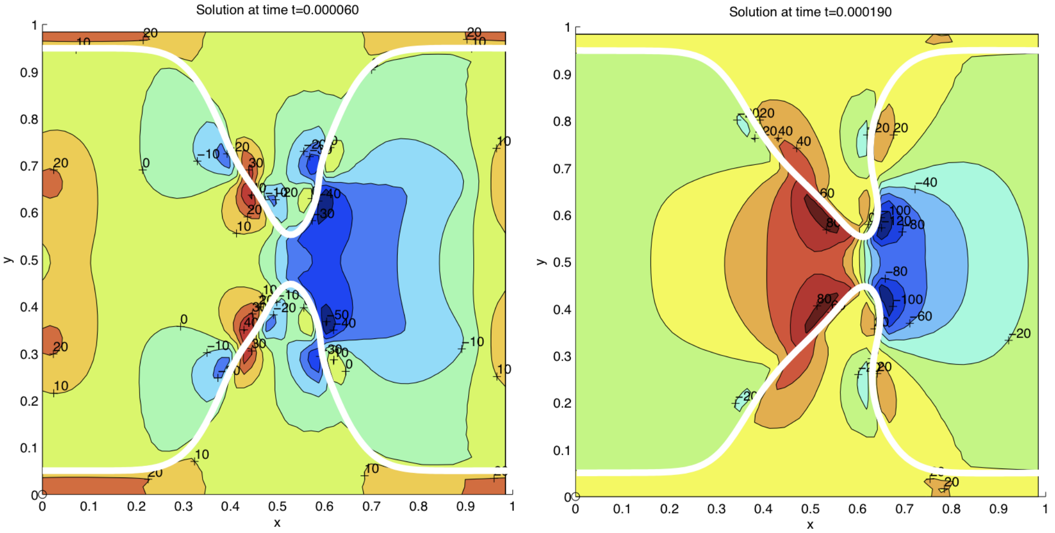


Figure 10: Pressure pIB method

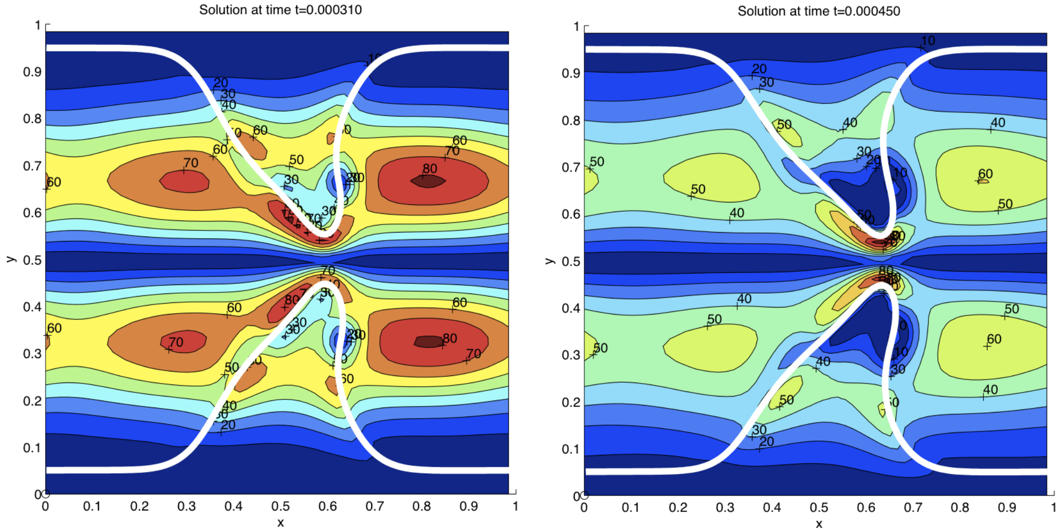


Figure 11: Vorticity from pIB method

since there is a larger tunnel through which the air can pass. This is in contrast to the previous simulation, where the air passes only for a short period of time. We expect to see small vorticies to form along the narrow slit of the vocal folds. The entrance and exit of the vocal narrow region should have vorticies forming since not all the airflow will pass through the center but rather collide with the vocal folds themselves. Similarly, we expect some vorticies to form within the narrow region since the airflow will collide with the vocal folds and cause a temporary congestion from the air coming in from behind. The plots in Fig. 12 and 13 demonstrate the velocity, and vorticity respectively. The column of particles represents the wall of airflow passing through the vocal folds. Note that the airflow is only non-zero within the channel of the vocal folds. Due to computational restrictions, the constants used are not the same as in Duncan et al. [1]. However, the vorticity and velocity plots do align with those in the literature [1]. The time step is taken to be very small due to the forcing terms that restrict the method, explaining why the overall structure of the vocal folds does not change drastically.

4.7 Code

The previously written code corresponded to applying the Immersed Boundary Method to an ellipse structure. Necessary changes to the previous code is within the forces being applied to the system. After these transformations, numerical results can be taken and analyzed. These consist of looking at the vorticity of the model and the convergence of the system as well. There are numerous small changes in the original code [4] however, only the

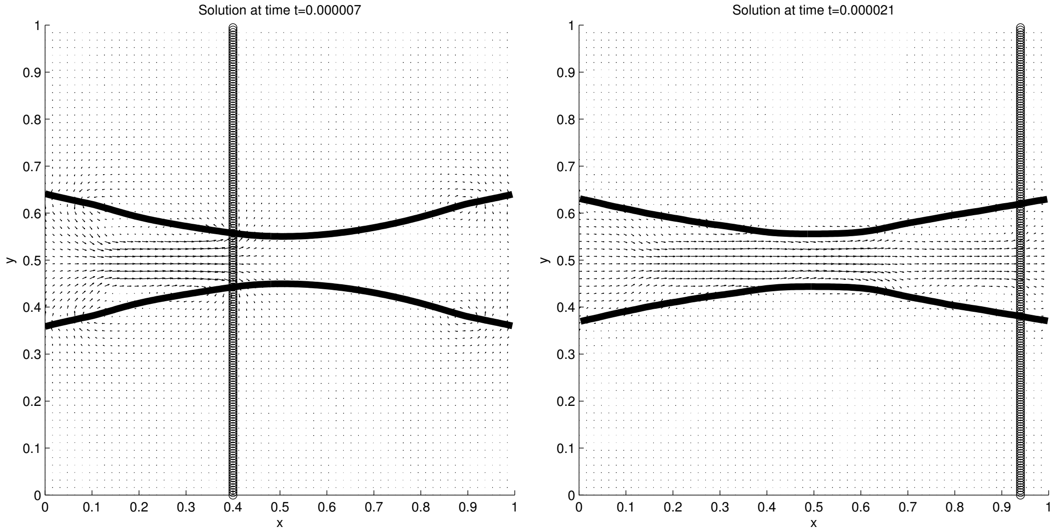


Figure 12: Velocity graphs of second vocal fold configuration

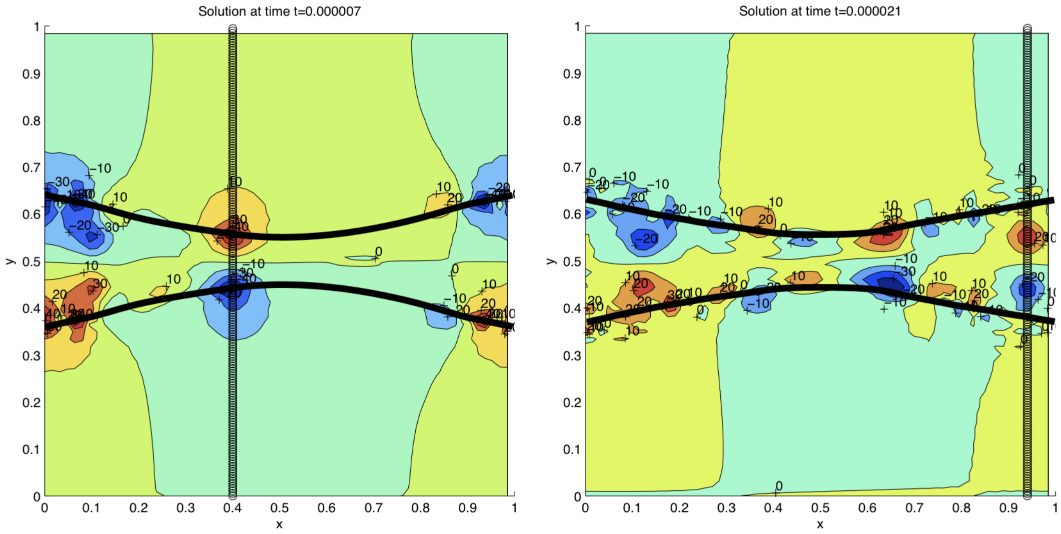


Figure 13: Vorticity graphs of second vocal fold configuration

computation of the forcing terms shall be written explicitly.

```
%Forcing term from the massive and massless spring terms
FspringX = -spring*(chiX-chiMX);
FspringY = -spring*(chiY-chiMY);

%Forcing term for the stretch in the immersed boundary
lenX(2:Nb) = abs(chiX(2:Nb) - chiX(1:Nb-1));
lenY(2:Nb) = abs(chiY(2:Nb) - chiY(1:Nb-1));
lenX(1) = abs(chiX(1) - chiX(Nb));
lenY(1) = abs(chiX(1) - chiX(Nb));
taoX = stretch*(lenX - ds);
taoY = strech*(lenY - ds);
FstretchX(2:Nb-1) = (taoX(3:Nb) - taoX(2:Nb-1))/ds;
FstrecthY(2:Nb-1) = (taoY(3:Nb) - taoY(2:Nb-1))/ds;

%Calculation of the bending force for the immersed boundary
e = ones(Nb, 1);
Dxx = spdiags([e -2*e e],[-1 0 1], Nb, Nb);
Dxx(1,end) = 1; Dxx(end,1) = 1;
FbendX = sBend.*(Dxx*chiX./(ds^2) - x0Bend);
FbendY = sBend.*(Dxx*chiY./(ds^2) - y0Bend);
```

5 Conclusion and further work

The outcome of applying the pIB method to the vocal fold configuration is an accurate and simplistic way for modeling. Due to the flexibility of the pIB method, we can create different configurations very easily and simulate the flow of air passing through the vocal folds. These simulations allow us to test problems without spending large amounts of technology in the laboratory to view the vocal folds. The results from the tests demonstrate that the method is indeed convergent for the ellipse problem. Moving to the application of the vocal folds, the vorticity graphs illustrate the flow of air passing through the folds. I learned that the pIB method applied a simple concept of introducing a fictitious massive boundary which keeps the same code for the "usual" IB method, and with a few extra functions can solve this system very easily. In the programming perspective it is very flexible and can be easily used on previously written IB methods. The IB method in general is very flexible in the sense of introducing new forces to the immersed boundary. By writing ones own force computation, they merely need to add to the previous sum and the remaining code does not need any adjustments. Although the literature previously used very small times, it appears that the code is stable for a less strict time step. This is of great importance, because if the time step is too small, then the computations are still valid, however is no longer as interesting since it requires too much time and power from the computer. Due to this constraint, and working on aa much smaller framework of computers than in the literature, direct comparisons were not possible, but obtaining results with similar characteristics was accomplished.

Further work has multiple possible facades. Firstly, encountering a way to counter act the small time step would be a major improvement to the method. This could be done by implementing an implicit IB scheme or finding a way to include a low viscosity to not drastically decrease the time step. Secondly, looking at other approaches to this problem, i.e. finite element method, and compare and contrast the two results. Moreover, bringing this method to the three dimensional case, and being able to have a rotating view or splice of the trachea and view the vorticities from this perspective. Lastly, an interesting topic to pursue would be to simulate the process of choking, and being to see the best manners to remove the lodged object. Comparing our results to those of common and accepted solutions when adhering choking in today's generation. This could be accomplished through placing a foreign object into the trachea to simulate a food particle and seeing the effects of the flow rate and necessary pressure and airflow to remove the object.

In conclusion, the pIB is not the only method to simulating vocal fold dynamics, however it is a very simple and flexible approach to the problem. It's implementation allows for various configurations, and gives an accurate solution with numerous data variables to observe.

References

[1] Comer Duncan, Guangnian Zhai, and Ronald Scherer. Modeling coupled aerodynamics and vocal fold dynamics using immersed boundary methods. *The Journal of the Acoustical Society of America*, 120(5):2859–2871, 2006.

[2] Yongsam Kim and Charles S. Peskin. Penalty immersed boundary method for an elastic boundary with mass. *Physics of Fluids*, 19(5):053103, 2007.

[3] Charles S Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252 – 271, 1972.

[4] Jeffrey Wiens and Brittany Froese. Two dimensional-immersed boundary method matlab code.