

# A Computational Model of Caudal Fin Undulations Using Impulse Methods

**Jamie N. Bailey**

Westminster College, Salt Lake City

**Apolinar Gallardo, Jr.**

Los Angeles Valley College

**Tiffany Psemeneki**

Boston University

## **Abstract**

The motion of a fish's caudal fin is modeled as a flexible boundary immersed in a three-dimensional fluid. Due to the high Reynolds number, the fluid can be represented by Euler's equations for incompressible flow. A two-dimensional plate of points is used to simulate the fin's membrane, where external forces and impulse variables drive the fin's motion. The impulse field is approximated by a summation of cutoff functions, or blobs. The results of computational simulations are presented.

## **1 Introduction**

The goal of this research is to create a computer simulation of the swimming motion of a fish. We model caudal fin undulations in which thrust is generated by significant lateral movements that occur only at the caudal fin. This allows the fish to maintain high cruising speeds for long periods of time. We will represent the fin by a rectangular plate of points in a three-dimensional fluid flow. The idea for this research was motivated by [3], where Cortez mentions that very little work has been done in the study of various swimming motions of undulating creatures, especially large creatures. This fluid flow is known as high Reynolds number flow and is represented by the Euler equations, in which the inertial force exerted by the creature is sufficiently larger than the viscous force of the fluid.

To find numerical solutions to this kind of fluid flow problem, we take advantage of the impulse method introduced in [1]. This method is based on the Lagrangian vortex method for general Navier-Stokes flows. Impulse methods introduce additional impulse variables, which are a function of time and space. Therefore, every point in the plate moves with a velocity that is derived from impulse variables and the forces that lie within them. The forces we use make our model as realistic as possible. We want the plate of points to act as the thin flexible membrane of the fin, so we need forces due to the elasticity of the fin's tissue (spring forces), the rigidity and bending of the fin (curvature forces), and the swimming motion (external forces). The most

challenging part of this model is balancing the magnitudes of these three types of forces and the constants associated with each.

We begin by presenting background information on the equations for fluid flow, with a particular emphasis on Euler flow. Then we introduce blob functions and how we use them to approximate the impulse field in the impulse methods. We provide a discussion of all of the forces that must be used to create an accurate model and the challenges associated with their correct implementation. Examples of our simulations from *Matlab* are presented along with a description of the program implementation and the time-consuming task of running the simulations.

## 2 Fluids Background

Fluid flows are represented by the Navier-Stokes equations

$$\rho(\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u}) = -\nabla p + \mu\Delta\mathbf{u} + \mathbf{f} \quad (1)$$

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

where  $\mathbf{u}$  is the velocity,  $p$  is pressure,  $\mu$  is the viscosity of the fluid,  $\mathbf{f}$  is the external force, and equation (2) insures the incompressibility of the flow. Attempting to evaluate the various terms of the Navier-Stokes equations at a particular point in the flow can be a difficult task since the terms may have different orders of magnitude. Depending on the flow, some terms may be huge compared to others and some may be so small that they can be ignored completely. In order to take advantage of the possibility of eliminating some terms, the dimensionless form of equations (1) and (2) is derived by scaling all of the terms by appropriate factors and comparing their relative magnitudes [6]. Let the dimensionless variables be

$$\tilde{x} = \frac{1}{L}\mathbf{x}, \quad \tilde{u} = \frac{1}{U}\mathbf{u}, \quad \tilde{t} = \frac{U}{L}t, \quad \tilde{p} = \frac{1}{P}p, \quad \tilde{f} = \frac{1}{F}\mathbf{f}$$

where the relative scales are  $L$  for length,  $P$  for pressure,  $U$  for velocity, and  $F$  for external force. Then the dimensionless form of the Navier-Stokes equations is

$$\tilde{u}_{\tilde{t}} + (\tilde{u} \cdot \tilde{\nabla})\tilde{u} = -\left(\frac{P}{\rho U^2}\right)\tilde{\nabla}\tilde{p} + \frac{1}{Re}\tilde{\Delta}\tilde{u} + \left(\frac{LF}{\rho U^2}\right)\tilde{f}. \quad (3)$$

$$\tilde{\nabla} \cdot \tilde{u} = 0 \quad (4)$$

The Reynolds number,  $Re = \frac{\rho LU}{\mu}$ , expresses the ratio between the inertial force and viscous force. If  $Re$  is sufficiently large then the quantity  $\frac{1}{Re}$  will be small and the  $\frac{1}{Re}\tilde{\Delta}\tilde{u}$  term can be ignored. The Reynolds number is large for a flow that involves a large length scale or small viscosity, for example, a large swimming creature in water. Modeling a fish involves a flow that has a high Reynolds number. Thus, we can neglect the  $\frac{1}{Re}\tilde{\Delta}\tilde{u}$  term. If we then choose  $P$  and  $F$  so that the coefficients of the other terms are equal to 1, then (3) and (4) become the Euler equations

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \mathbf{f} \quad (5)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (6)$$

which describe high Reynolds number flows.

### 3 Blobs

There are several methods available to approximate an unknown function with known initial conditions. The numerical method we are using involves the use of functions that smoothly approximate the Dirac delta function, which is commonly used in impulse forcing. These functions are called cutoff functions, or blobs (see [4]). We will use blobs to approximate the impulse field,  $\mathbf{m}$ .

First, we let  $f$  be a smooth, radially symmetric function satisfying  $\int_{\mathbb{R}^3} f = 1$ . In spherical coordinates, this can be rewritten as  $4\pi \int_0^\infty r^2 f(r) dr = 1$ . Some examples of these functions include

$$f(r) = \frac{3}{4\pi} e^{-r^3} \quad \text{and} \quad f(r) = \frac{A_p}{(r^2 + 1)^{p/2}},$$

where  $A_p$  and  $p$  are constants chosen such that  $f(r)$  satisfies the above conditions. A blob  $f_\delta(r)$  in three dimensions is defined as a scaled version of  $f(r)$  and they are related by

$$f_\delta(r) = \frac{1}{\delta^3} f\left(\frac{r}{\delta}\right),$$

where  $\delta$  is a small scaling parameter between 0 and 1. Figure 1 graphs the blob  $f_\delta(r) = \frac{3\delta^2}{4\pi(r^2 + \delta^2)^{5/2}}$ . Notice that by varying  $\delta$ , the height and width of the blob change. Increasing  $\delta$  will result in a smaller function value for  $f_\delta(0)$  and decreasing  $\delta$  will result in a larger function value for  $f_\delta(0)$ . Although the blob changes with  $\delta$ ,  $f_\delta(r)$  still satisfies,  $4\pi \int_0^\infty r^2 f_\delta(r) dr = 1$ , which by definition makes it a blob.

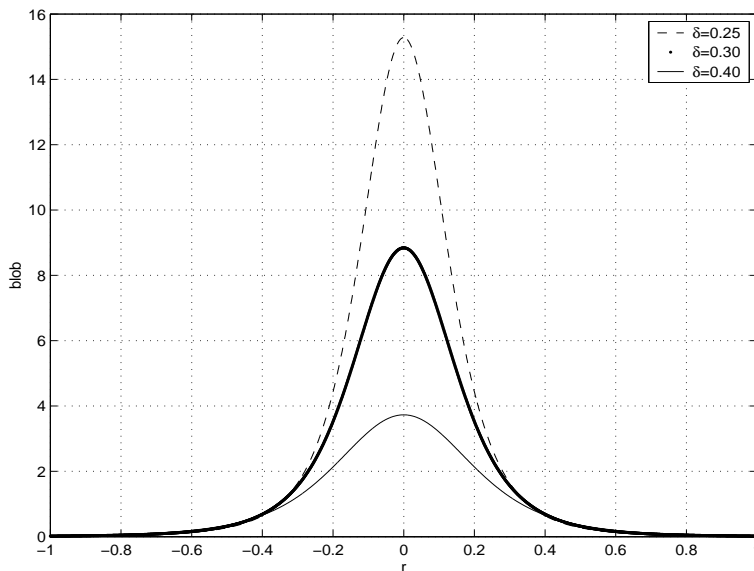


Figure 1: A graph in two dimensions of the three-dimensional blob  $f_\delta(r) = \frac{3\delta^2}{4\pi(r^2 + \delta^2)^{5/2}}$  for  $\delta = 0.25, 0.30, 0.40$ . The height and width of the blobs are determined by  $\delta$ .

One way to approximate an unknown function with known initial conditions is through the use of blobs. The convolution of a smooth function  $T$  with a blob  $f_\delta$  (in

three dimensions) is defined as

$$(T \star f_\delta)(\mathbf{x}) = \int_{\mathbb{R}^3} T(\mathbf{y}) f_\delta(\mathbf{x} - \mathbf{y}) d\mathbf{y} \quad (7)$$

For a sufficiently small  $\delta$ ,  $f_\delta(\mathbf{x})$  is concentrated most where  $x = 0$ . So the integral

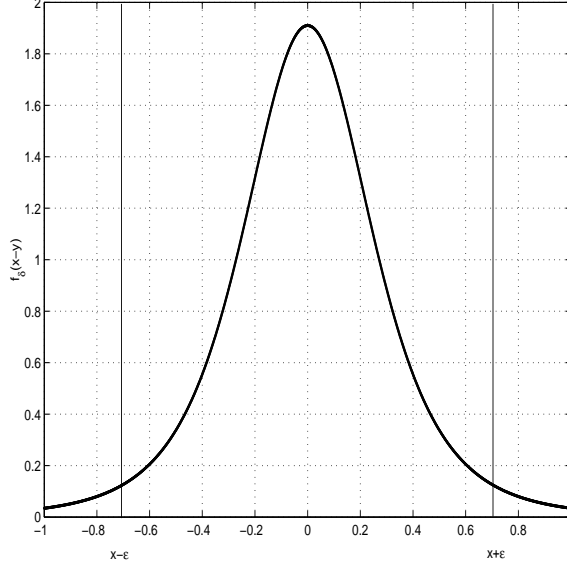


Figure 2: For  $y < x - \varepsilon$  and  $y > x + \varepsilon$ , the blob  $f_\delta(x - y)$  is approximately zero, so we do not have to integrate over  $\mathbb{R}^3$ .

in (7) will have contributions mostly from values of  $y$  near  $x$  (see Figure 2), and the convolution should be a good approximation of  $T$ . To see this, we can rewrite the convolution as

$$(T \star f_\delta)(\mathbf{x}) \approx \int_{x-\varepsilon}^{x+\varepsilon} T(\mathbf{y}) f_\delta(\mathbf{x} - \mathbf{y}) d\mathbf{y}$$

If  $T(\mathbf{y})$  is approximately constant for  $\|\mathbf{y} - \mathbf{x}\| < \varepsilon$  then  $T(\mathbf{y}) \approx T(\mathbf{x})$  and

$$(T \star f_\delta)(\mathbf{x}) \approx T(\mathbf{x}) \int_{x-\varepsilon}^{x+\varepsilon} f_\delta(\mathbf{x} - \mathbf{y}) d\mathbf{y} \approx T(\mathbf{x}) \cdot 1 = T(\mathbf{x}).$$

This indicates that  $(T \star f_\delta)(\mathbf{x})$  approximates  $T(\mathbf{x})$  with only a small amount of error. For numerical methods we approximate (7) by a summation of blobs,

$$T(\mathbf{x}) \approx \sum_{k=1}^N T(\mathbf{x}_k) f_\delta(\mathbf{x} - \mathbf{x}_k) V_k,$$

where  $V_k$  is the volume represented by the  $k$ -th particle. In the next section we will approximate the impulse,  $\mathbf{m}(\mathbf{x})$  in this way.

For our model we have chosen the blob

$$f_\delta(r) = \frac{3\delta^2}{4\pi(r^2 + \delta^2)^{5/2}}.$$

However, choosing a particular blob is arbitrary (so long as the degree of the small amount of error associated with the blob is ignored) because all blobs are the same qualitatively, except for the value of  $\delta$ . All blobs satisfy  $4\pi \int_0^\infty r^2 f(r) dr = 1$ . So the area under the curve will always be 1, but the height and width of the curve will have to adjust as  $\delta$  is changed.

## 4 Impulse and Velocity

Recall that Euler flow is given by equations (5) and (6). We take the curl of (5) and get that

$$\nabla \times \mathbf{u}_t + \nabla \times (\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \times (-\nabla p) + \nabla \times \mathbf{f}$$

This will give us the Euler equation in terms of the vorticity and also eliminate the pressure term, since  $\omega = \nabla \times \mathbf{u}$  and  $\nabla \times (-\nabla p) = 0$ ,

$$\omega_t + (\mathbf{u} \cdot \nabla) \omega = \nabla \times \mathbf{f} + (\omega \cdot \nabla) \mathbf{u}.$$

Impulse can be thought of as change in linear momentum and can be defined as a vector field whose curl is the vorticity in the flow, assuming constant fluid density [3]. The impulse field,  $\mathbf{m}$ , is related to the velocity by

$$\mathbf{m} = \mathbf{u} + \nabla \phi. \tag{8}$$

To simplify and eliminate  $\mathbf{u}$ , we take the divergence and get  $\nabla \cdot \mathbf{m} = \Delta \phi$ . We would like to approximate  $\mathbf{m}$  by a convolution of  $\mathbf{m}$  with a blob  $f_\delta$ , because, for small enough  $\delta$ , the convolution of  $\mathbf{m}$  with a blob  $f_\delta$  approximates  $\mathbf{m}$  itself.

$$\mathbf{m}(\mathbf{x}) = \sum_{k=1}^N \mathbf{m}_k(t) f_\delta(\mathbf{x} - \mathbf{x}_k). \tag{9}$$

From the relationship between the impulse and the velocity in (8), we have  $\mathbf{u} = \mathbf{m} - \nabla \phi$ . Substituting this into (5), we get

$$\mathbf{m}_t - \nabla \phi_t + (\mathbf{u} \cdot \nabla) \mathbf{m} - (\mathbf{u} \cdot \nabla) \nabla \phi = -\nabla p + \mathbf{f}.$$

Substituting (8) into the equation and letting  $p = \phi_t + \mathbf{u} \cdot \nabla \phi + \frac{1}{2} |\mathbf{u}|^2$  (see [1] for more detail) we get the Euler equation in terms of impulse,

$$\mathbf{m}_t + (\mathbf{u} \cdot \nabla) \mathbf{m} = -(\nabla \mathbf{u})^T \mathbf{m} + \mathbf{f} \tag{10}$$

where  $\nabla \mathbf{u}$  is a 3x3 matrix of partial derivatives and  $T$  denotes the transpose. Noticing that the left hand side of equation (10) is the material derivative of  $\mathbf{m}$ , we can define the rate of change of the impulse strengths as

$$\frac{d\mathbf{m}_j}{dt} = -(\nabla \mathbf{u})^T \mathbf{m}_j + \mathbf{f}_j.$$

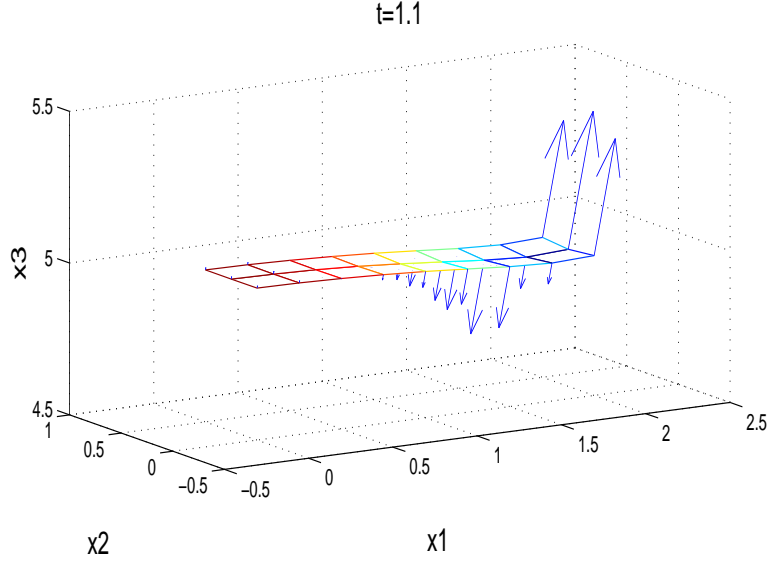


Figure 3: Impulse vectors

See Figure 3 for a graphical representation of the impulse vectors applied to a plate of points.

To find the velocity, we begin with the relationship  $\mathbf{u} = \mathbf{m} - \nabla\phi$  and solve for  $\phi$  by taking the divergence of both sides. After this calculation we have

$$\nabla \cdot \mathbf{u} = 0 = \nabla \cdot \mathbf{m} - \nabla \cdot (\nabla\phi) \implies \Delta\phi = \nabla \cdot \mathbf{m}.$$

Suppose  $\mathbf{m}$  is given by a convolution with a blob, then  $\mathbf{m} = \mathbf{m}_0 f_\delta(x - x_k)$ . Let a function  $g$  satisfy  $\Delta g = f_\delta$ , then  $\phi = \sum \mathbf{m}_0 \cdot \nabla g$  and  $\nabla\phi = (\mathbf{m}_0 \cdot \nabla)\nabla g$ . Therefore

$$\mathbf{u} = \mathbf{m} - (\mathbf{m}_0 \cdot \nabla)\nabla g$$

where  $g$  is determined by our particular choice of blob. In spherical coordinates, the equation for  $g$  can be written in partial derivatives as

$$\frac{1}{r^2}(r^2 g_r)_r = f_\delta(r). \quad (11)$$

Since we do not need to find  $g$  explicitly (only  $g_r(r)$ ), we solve (11) and get

$$g_r(r) = \frac{1}{r^2} \int_0^r y^2 f_\delta(y) dy = \frac{F(r)}{4\pi r^2}$$

where  $F(r) = 4\pi \int_0^r s^2 f_\delta(s) ds$ .  $F(r)$  is called the shape factor and is specific to the chosen blob. Now the velocity (from [1]) can be written in terms of  $F(r)$  as

$$\mathbf{u}(\mathbf{x}) = \frac{1}{4\pi} \sum_{j=1}^N \mathbf{m}_j \left[ \frac{rF'(r) - F(r)}{r^3} \right] - \left( \frac{\mathbf{x} - \mathbf{x}_j}{r} \right) \left( \mathbf{m}_j \cdot \frac{\mathbf{x} - \mathbf{x}_j}{r} \right) \left[ \frac{rF'(r) - 3F(r)}{r^3} \right]$$

where  $r = |\mathbf{x} - \mathbf{x}_j|$ .

We still need to complete our impulse equations by finding the entries of the matrix  $(\nabla \mathbf{u})^T$ . This can be achieved by simply evaluating the partial derivatives of our expression for  $\mathbf{u}(\mathbf{x})$ . The  $k$ -th column of the matrix (from [1]) will be given by

$$\begin{aligned} \frac{\partial \mathbf{u}(\mathbf{x})}{\partial x_k} &= \frac{1}{4\pi} \sum_{j=1}^N \mathbf{m}_j \left( \frac{x_k - x_j}{r} \right) \left[ \frac{r^2 F'' - 3rF' + 3F}{r^4} \right] \\ &\quad - m_k^j \left( \frac{\mathbf{x} - \mathbf{x}_j}{r} \right) \left[ \frac{rF' - 3F}{r^4} \right] - \mathbf{e}^k \left( \mathbf{m}_j \cdot \frac{\mathbf{x} - \mathbf{x}_j}{r} \right) \left[ \frac{rF' - 3F}{r^4} \right] \\ &\quad - \left( \frac{x_k - x_j}{r} \right) \left( \frac{\mathbf{x} - \mathbf{x}_j}{r} \right) \left( \mathbf{m}_j \cdot \frac{\mathbf{x} - \mathbf{x}_j}{r} \right) \left[ \frac{r^2 F'' - 7rF' + 15F}{r^4} \right] \end{aligned}$$

where  $\mathbf{e}^k$  is a unit vector in the  $k$ -th direction.

We have now arrived at a system of ordinary differential equations that can be solved numerically to find the velocity and impulse strengths for the motion of a particle at a particular location:

$$\frac{d\mathbf{x}_j}{dt} = \mathbf{u}(\mathbf{x}_j) \quad (12)$$

$$\frac{d\mathbf{m}_j}{dt} = -(\nabla \mathbf{u})^T \mathbf{m}_j + \mathbf{f}_j. \quad (13)$$

These equations are solved simultaneously, for  $j = 1, 2, \dots, N$ .

## 5 Forces

We simulate the fin using a rectangular plate comprised of a series of points. When forces are introduced to stimulate its motion, there is the possibility that some or all of the points which make up the fin will wander off unless there exist forces to cause the points to act as a cohesive unit. We use three types of forces: spring forces, curvature forces, and external forces. The spring forces, governed by Hooke's law, are placed between the points in the plate to keep them connected at all times. The curvature forces prevent the plate of points from bending excessively when the spring forces are inactive. The external forces create sinusoidal motion in the plate that is weaker at the base of the fin but gets stronger towards the tip of the fin. We explain these forces in more detail next.

### 5.1 Spring Forces

To maintain the fin's cohesiveness, we introduce ideal, Hooke's law springs, that are used to connect all of the points in our plate together. All of the fin's points are connected by a spring to its neighboring points. Interior points are connected to four other points; points along the edges of the plate, with the exception of the corners, are connected to their three neighboring points; and the corner points are connected

Figure 4: A schematic representation of how the points are connected by Hooke's law springs

to their two neighboring points. Figure 4 shows a 3x3 example of how the points are connected.

These springs, which help form the elastic membrane of our fin, are analogous to the bonds that hold atoms together. These springs, or "fibers", are designed to obey Hooke's law. Thus, when the spring is unstretched and in its relaxed state, the spring is in its equilibrium position and exerts no force; when the spring is stretched or compressed, it exerts a force  $\mathbf{F}^s$  that opposes change from the equilibrium position. The spring force can also be described as a restoring force since  $\mathbf{F}^s$  is always trying to restore the spring to its relaxed, equilibrium state. The further away (or closer) that two connected points are from its equilibrium state, the greater the restoring force will be. Through experiments [7], it has been shown that the restoring force varies linearly with the distance that the spring is stretched or compressed. Hooke's law is represented by the relationship

$$\mathbf{F}^s = -k\mathbf{x}$$

The proportionality constant,  $k$ , is the spring constant and has units of Newtons per meter (N/m). We use Hooke's law to apply force to  $\mathbf{x}_a$  in the following way

$$\mathbf{F}^s(\mathbf{x}_a) = -k \left( \frac{\|\mathbf{x}_a - \mathbf{x}_b\|}{L} - 1 \right) \frac{\mathbf{x}_a - \mathbf{x}_b}{\|\mathbf{x}_a - \mathbf{x}_b\|},$$

where  $L$  is the resting length of the spring.

## 5.2 Curvature Forces

When the springs between the points in the plate are in their equilibrium positions, there is no force to keep the plate from bending. Curvature forces are needed to

prevent the deformation of the plate, when no other restoring forces are acting on it.

All of the fin's points, except for the corners and edges, can be thought of as having four vector tails attached to them (two on the right and left, and two on the top and bottom). See Figure 4. The curvature forces are zero if the vectors lie in a plane. However, once bending starts to take place, the restoring force tries to bring the points to their initial unbent state. The force identifies bending when the dot product between the vectors changes. The curvature force is related to a type of energy equation, denoted as  $E$ , which is discussed in [3],

$$E = \frac{K_0}{2} \sum_{b=1}^{nx} \sum_{a=2}^{ny-1} \left[ \frac{(\mathbf{x}_{a,b} - \mathbf{x}_{a-1,b}) \cdot (\mathbf{x}_{a+1,b} - \mathbf{x}_{a,b})}{dy^2} - 1 \right]^2 dx dy$$

$$+ \frac{K_0}{2} \sum_{a=1}^{ny} \sum_{b=2}^{nx-1} \left[ \frac{(\mathbf{x}_{a,b} - \mathbf{x}_{a,b-1}) \cdot (\mathbf{x}_{a,b+1} - \mathbf{x}_{a,b})}{dx^2} - 1 \right]^2 dx dy$$

where  $K_0$  is a constant related to the spring constant,  $nx$  and  $ny$  are the dimensions of the plate,  $a$  and  $b$  denote the position of a point in the plate, and  $dx$  and  $dy$  are the resting lengths between the points. The energy increases from zero as the curvature of the fin deviates from the desired curvature. The curvature force for a point in the plate located at  $(a, b)$  is defined as

$$\mathbf{F}_{a,b}^c = -\frac{\partial E}{\partial \mathbf{x}_{a,b}}.$$

### 5.3 External Forces

The aforementioned spring and curvature forces are designed with the intention of holding the creature's fin together. However, these forces are not designed to stimulate any type of swimming motion. In order to produce the desired sinusoidal motion, external forces are added. Without the external forces, the points lie motionless in the fluid.

To obtain the maximum possible motion, the external forces are designed in a such a way that they are always pointing in a direction that is normal to the plate of points that represents the fin. The normal vector at a particular point can be approximated by first, developing two vectors that are parallel to the surface of the plate and second, taking their cross product to obtain the desired normal vector,  $\mathbf{n}$ . The final step is to then rewrite  $\mathbf{n}$  as a unit vector since its direction is our only concern. See Figure 5 for a pictorial representation of how we compute the normal vectors.

Knowing the direction of the external forces at each point, we can now begin to apply the forces. To produce forward swimming motion in nature, a wave moves through the fin, with greatest amplitude at the tip. Therefore we need the force to be applied in such a way that it is very weak at the base of the fin, but is continuously increasing along the fin towards the tip. Functions with these properties can be linear, quadratic, or exponential (see Figure 6 for an example of how a quadratic force is

Figure 5: We obtain the unit normal vector,  $\mathbf{n}$ , at the point  $O$  by taking the cross product of the vectors  $\mathbf{AB}$  and  $\mathbf{CD}$  and normalizing it.

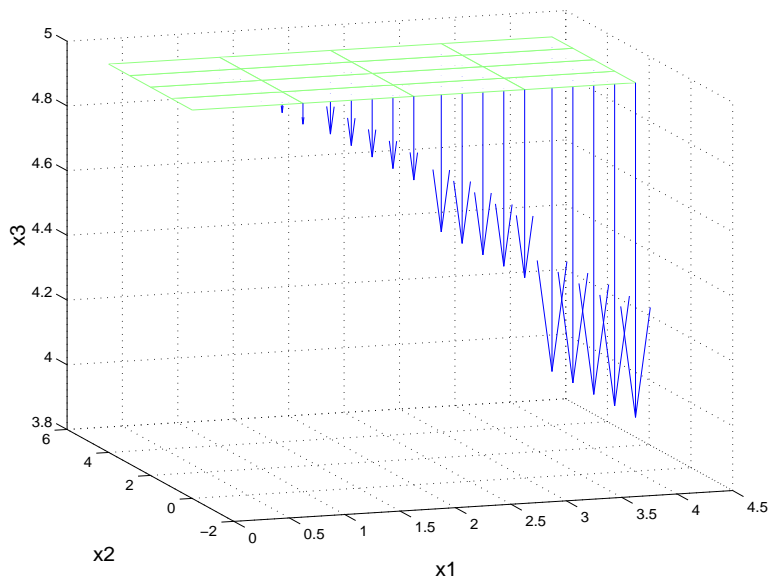


Figure 6: The graph of the vectors that represent the quadratic external forces using *Matlab*

applied). The coefficient of such a function needs to be sinusoidal in time since we desire caudal fin undulations and horizontal translation. Now we have obtained sufficient information to define our external forces as

$$\mathbf{F}^e = \alpha(t)\mathbf{n}g(\mathbf{x})$$

where we select  $\alpha(t)$  to have the form  $\alpha = A \cos(Bt + C)$  for constants  $A$ ,  $B$ , and  $C$ . Where  $\mathbf{n}$  is the normal vector and  $g(\mathbf{x})$  is either  $\mathbf{x}$ ,  $\mathbf{x}^2$ , or  $e^{\mathbf{x}}$ , depending on whether we choose to vary the force linearly, quadratically, or exponentially. We need  $\alpha$  to symmetrically alternate the sign of  $\mathbf{n}$  between positive and negative values so that the motion of the fin will also alternate about a fixed horizontal axis. The previously described forces are used in conjunction with these forces to keep the fin's shape as close as possible to the desired shape.

## 6 Results

We use *Matlab* to find a numerical solution to equations (12) and (13) and to plot the motion of our model. We design our program so that each of the three forces are in separate files, which are used in the function file, where we define our system of ordinary differential equations. In our main file we use the built-in numerical solver *ode45* to solve (12) and (13) using a 4th-order Runge-Kutta method. These equations are solved in three-dimensions, for each point in the plate. We plot the motion of the fin for different time steps and in different views.

One of the biggest challenges in designing this model is determining the values for many different constants: the constants for the various forces, dimensions of the rectangular plate, distance between points, and the small blob parameter  $\delta$ . With so many constants that need to be initialized, determining a right combination can be difficult. We start by trying to discover a relationship between the constants. The strengths of the forces should be comparable so that one force does not overpower the others. If the spring and curvature constants ( $k$  and  $K_0$ ) are too small compared with the external forces, then the fin "explodes". The spring forces are too weak to hold the plate of points together and the curvature forces are also too weak to maintain the stable shape of the plate. If the spring constant is increased, the fin may encounter stability problems. If the spring constants are too large, then the springs holding the plate of points together become extremely rigid and the fin collapses in upon itself because the restoring force is so great.

Through trial and error, we found that the spring and curvature constants are related: the curvature constant should be around ten times the spring constant. We also discovered a relationship between  $\delta$  and the distance between the points: the value of  $\delta$  should be about twice that of the distance between the points. If the dimensions of the rectangular plate or distance between points are changed, the force constants will need to be changed as well. The more points that are in the plate, the more delicate it is to keep the plate from falling apart or collapsing. The spring and curvature constants need to increase as the number of points increase. The adjustment

of the many different constants in this model can be described as a delicate balancing act.

## 6.1 Simulations

The fin is initially discretized with 27 points in a thin rectangular strip so that the distance between the points is 0.25. The small blob parameter is chosen to be  $\delta = 0.65$  for the blob,  $f_\delta(r) = \frac{3\delta^2}{4\pi(r^2 + \delta^2)^{5/2}}$ . The spring constant is initialized to 5 and the curvature constant is initialized to 50. The external force is applied quadratically with an initial  $\alpha = 1.5 \cos(\frac{\pi}{2}t + C)$ , where the value of  $C$  is allowed to vary.

When the simulation runs with these constants, the movement of the fin is asymmetric. The fin moves up and down through the initial horizontal position, but after a period of time it starts to move in only one direction (up or down, not both). See Figure 7 for a series of stills of the motion for different times in a two-dimensional view. Also, see Figure 8 for a three-dimensional view of the same. The asymmetric

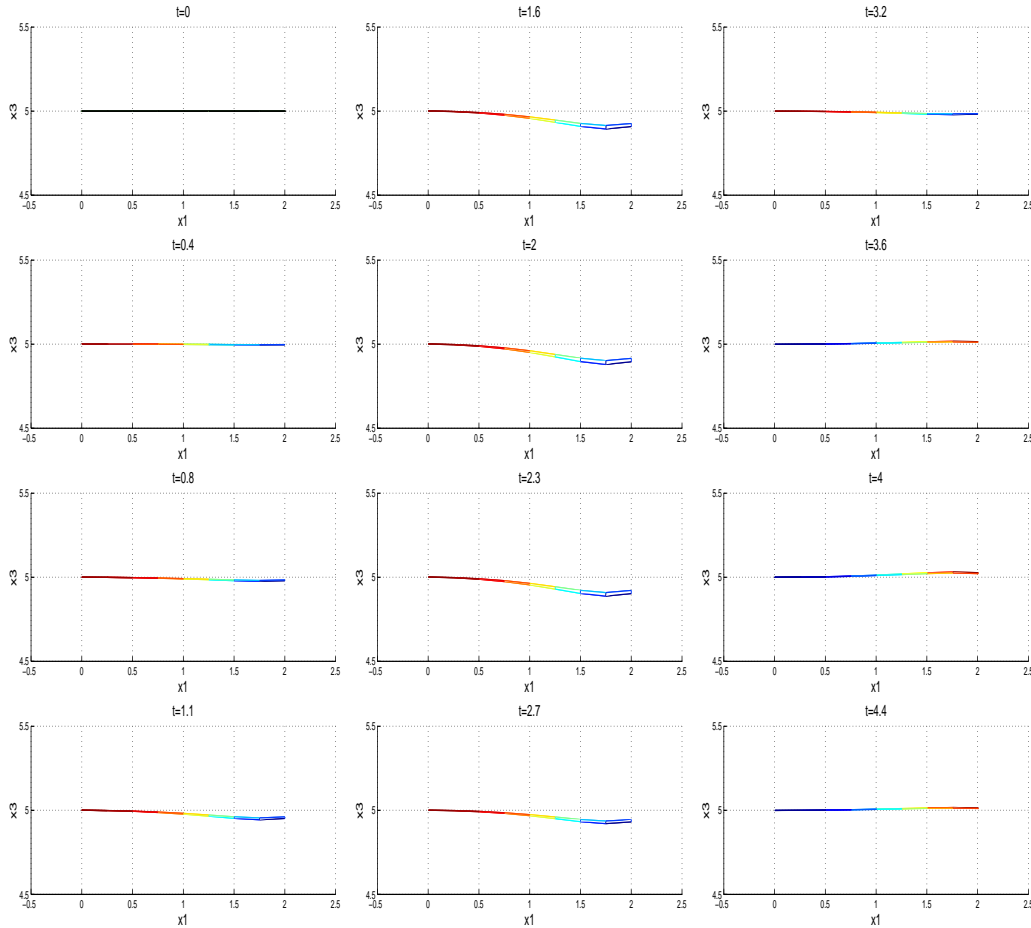


Figure 7: Swimming motion of one undulation of the fin from  $t = 0$  to  $t = 4.4$ , at intervals of 0.4 (2-dimensional, side view).

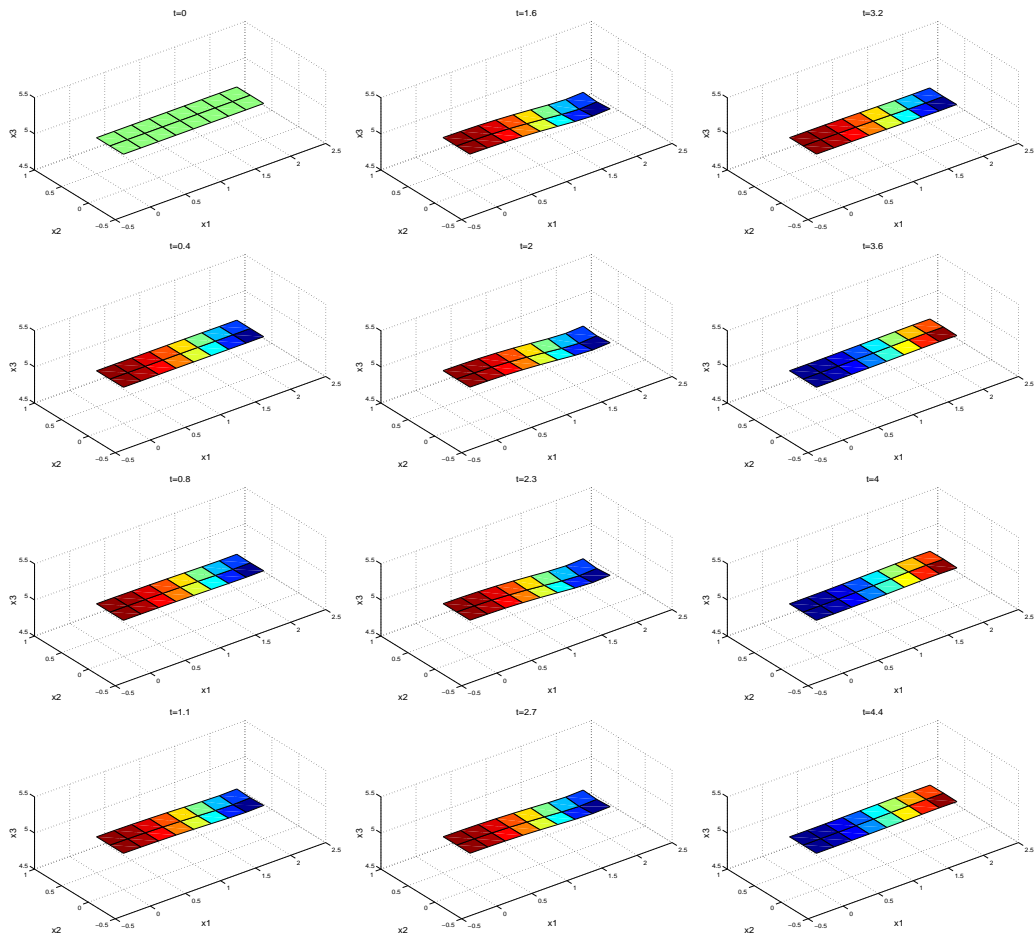


Figure 8: Swimming motion of one undulation of the fin from  $t = 0$  to  $t = 4.4$ , at intervals of 0.4 (3-dimensional view).

movement does successfully produce swimming motion, but at an angle and in the backward direction (see Figure 9).

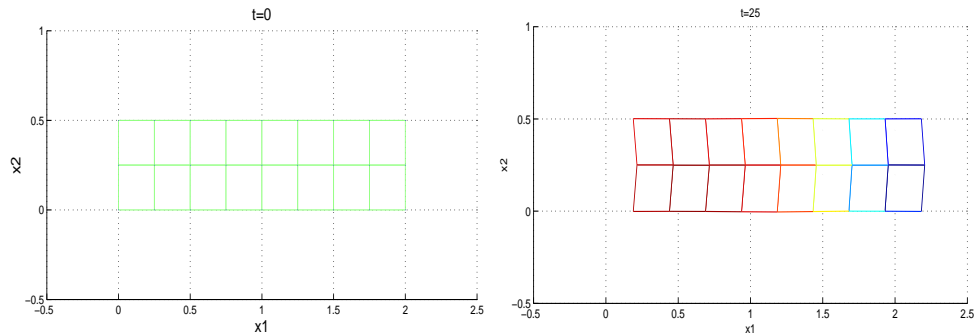


Figure 9: Top view of the fin at  $t = 0$  and then at  $t = 25$ . The fin moves backwards in the negative  $x_1$ -direction.

We would like to determine the motion of the fluid around the fin. For this purpose, we compute the fluid velocity on a set of points lying on a grid perpendicular to the initial plate position. The initial plate was the set  $\{(x_1, x_2, x_3) : 0 \leq x_1 \leq 2, 0 \leq x_2 \leq 0.5, x_3 = 5\}$  and the flow was computed on part of the plane  $x_2 = 0.25$  (See Figure 10). As the tip of the fin moves up, it drags fluid with it, creating a clockwise flow behind the fin. The opposite is true as the tip of the fin moves down. A vortex of smaller magnitude forms near the base (left side) as well. However, the center of this vortex lies either above or below the fin depending on the position of the fin. When the vortex lies above the fin it rotates counterclockwise, and when the vortex lies below the fin it rotates clockwise. The centers of the vortices lie a distance away from the fin (as opposed to lying on the same plane as the fin), which used correctly, could be conducive to forward swimming motion. Currently the external force model generates a flow that has a backward component on the points in the fin. We need to redesign the forces in a way that generates a flow that makes the vortex near the base reverse the direction of rotation. If we can force this vortex to rotate clockwise when it is above the fin and counterclockwise when it is below the fin, this will lead to forward motion.

We also examine the velocity field from a different perspective. The previous view is rotated so we can study the fluid velocity on either side of the fin, while looking at it from behind (see Figure 11). In doing this we notice two areas of vorticity, one on the right side of the fin and another on the left. They spin in different directions, meaning that one always rotates clockwise while the other rotates counterclockwise. As the tip of the plate moves up, it drags fluid with it and creates clockwise flow on the right, and counterclockwise flow on the left. The opposite is true when the tip of the plate moves down. These two vortices behave according to dipole flow. We try to correct this backward motion by slightly adjusting the external forces, in the hopes that the fin will move forward. We change the external force by redefining the force in the  $x_1$ -direction. This force is altered by subtracting the sum of the force in the  $x_1$ -direction and dividing by the number of points,  $N$ . This adjustment does

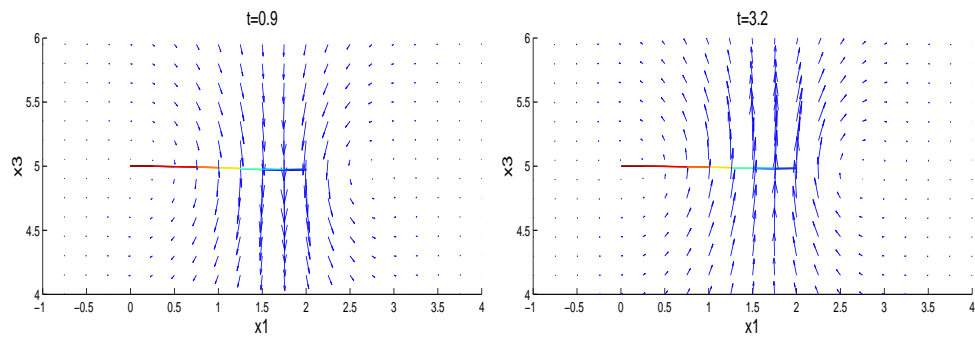


Figure 10: Velocity fields, side view of the fin

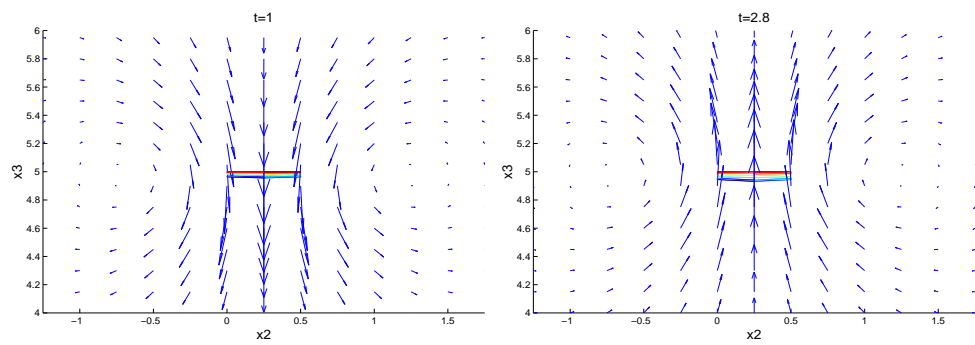


Figure 11: Velocity fields, back view of the fin

make the fin move forward (see Figure 12), but this problem is not completely solved. Another thing we want to achieve is to get a symmetric motion that causes the fin

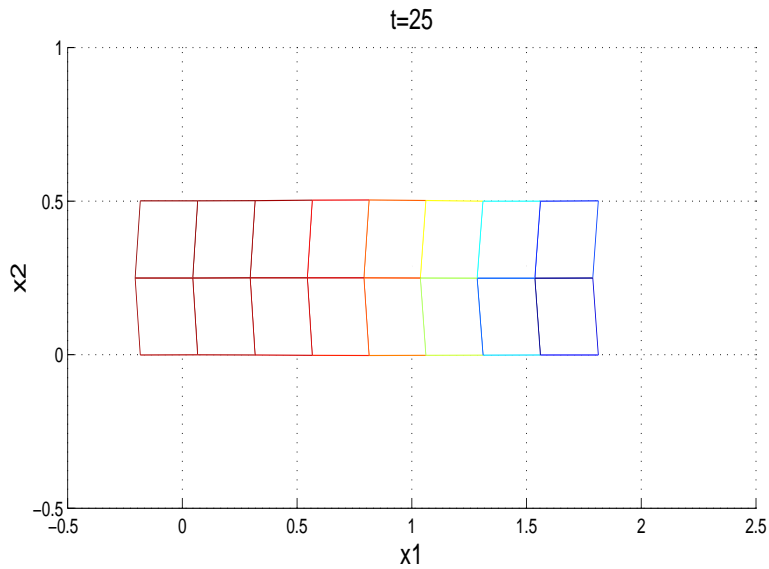


Figure 12: Top view of the fin at  $t = 25$ . The fin moves forwards in the positive  $x$ -direction.

to move forward horizontally, and not at an angle. One way that we found to do this is by alternating the signs of  $\alpha$  after a period of time. This produces symmetry in the fin (see Figure 13), but the motion is not smooth. The fin slightly pauses as it passes through the horizontal axis, which creates a discontinuous and unrealistic motion. More research is needed to perfect our model.

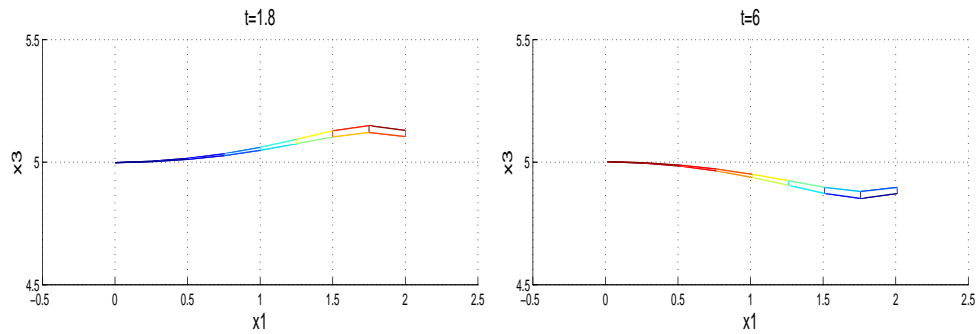


Figure 13: Shows the peaks for the symmetric motion in the up and down direction.

## 7 Conclusions

Taken together, the four separate vortices that we found reveal an encouraging result: forward swimming motion is possible. The combined rotations caused by the four vortices trap the fin along a predetermined path that will always move backwards. This happens because when the fin is up, all the vortices rotate towards the tip. As the fin moves down some of the vortices change orientation, but they do this in a backwards direction. So every component is working together to push the fin backward: the rotations of the four vortices and how they relate to the positions of the fin in time.

This is not what we set out to create, but it means that, given the right external forces, the directions of the rotations could be changed or the motion of the fin could be slightly altered so that it is up and down at the right times for the vortices to push it forward. We suspect that the change needs to take place in the definition of the external forces. All of the vortices are in place to provide the forward swimming motion, we just need to find a way to adjust the external forces so that the fin will move with them and not against them.

After finding the appropriate external force then the next step would be to get the motion of the fin to be smoothly symmetric. Currently, our model moves asymmetrically, but it can be made to move symmetrically with the problem of discontinuity. We suspect that the right combination of external forces and symmetric motion would produce the desired results.

## 8 Further Research

Not all of the objectives of this project were accomplished due to time constraints. Running a program of this length on a personal computer results in excessive run time, which could be anywhere from 5 minutes to 1 hour. Often the program must run for a long period of time, just to discover if the fin will collapse or "explode". Also, using *Matlab* extends the running time; the program could run faster in *Fortran* or *C++*. The run time significantly increases as the number of points increases to make the fin larger, so we use a thin 2x8 strip to avoid extended run times.

Given more time and better resources, we would like to extend our research in the following ways.

1. Find the exact relationship between the various constants.
2. Increase the dimensions of the plate to include more points.
3. Find a way to create a smooth symmetric motion.
4. Find the external force needed to move the fin forward.
5. Add a body to the fin to make the model more realistic.
6. Study other forms of undulations that may result in swimming.

## Acknowledgements

We would like to thank Ricardo Cortez for directing this research. The staff of the Summer Institute in Mathematics for Undergraduates (SIMU) is thanked for giving us this opportunity. We would like to thank National Science Foundation grant No. DMS-9987901 and National Security Agency grant No. MDA904-02-1-0006 for providing funding and support.

## References

- [1] R. Cortez, *Impulse-based particle methods for fluid flow*, Ph.D. thesis, University of California, Berkeley, 1995.
- [2] R. Cortez, *On the accuracy of impulse methods for fluid flow*, SIAM Jour. Sci. Comput., **19**, 1290-1302, 1998.
- [3] R. Cortez, *A vortex/impulse method for immersed boundary motion in high reynolds number flows*, Jour. Comput. Phys., **160**, 385-400, 2000.
- [4] R. Cortez, *Week 3: numerical methods*, SIMU 2002 Lecture Notes, 1-10, 2002.
- [5] C.S. Peskin & D.M. McQueen, *A general method for the computer simulation of biological systems interacting with fluids*, Biological Fluid Dynamics, 265-276, Cambridge, 1995.
- [6] C. Pozrikidis, *Introduction to Theoretical and Computational Fluid Dynamics*, Oxford University Press, New York, 1997.
- [7] R. Resnick, D. Halliday, & K. Krane, *Physics- Volume One*, Fifth Edition, John Wiley & Sons Inc., New York, 2002.