Avian Flow: Continuum Boid Dynamics

Boids (‘Birds’ with a Brooklyn accent) refers to a certain type of algorithm for simulating the flocking of large numbers of animals – for example, the eponymous birds. The word refers both to the algorithm itself and to the individual elements being simulated, which are rendered onscreen and whose motion through simulated space evolves according to a set of three rules. The particular details of the algorithm may vary from implementation to implementation, but these three main features of the Boids’ are usually the same.

Boids, as entities, possess a position and a velocity, and obey the following physical principles:

1. **Alignment.** Boids wish to conform to the flight paths of their neighbours, and so will rotate to align their own velocity with the average velocity of neighbouring boids which are sufficiently close.
2. **Cohesion.** Boids like to fly in groups, and they like to be as close to the centre of the group as possible. For this reason they also steer towards the mean position of those sufficiently close neighbours.
3. **Separation.** Boids don’t, however, like to collide, so they will actively avoid any of their neighbours that come *too* close.

Why not make this a continuous system?

From Discrete ‘Boids’ to Continuous ‘Birdflow’

Firstly, let’s state exactly what we would like to do. We would like to model the time evolution of a bird-fluid which obeys a set of physical principles which are similar to those stated for the discrete boid system. We can think of what we are doing as taking the continuum limit of infinitely many infinitesimal boids, in much the same way that the continuum dynamics of real fluids arises from considering the bulk behaviour of many individual atoms. We’ll refer to our continuum model using the normal spelling, ‘bird’, to differentiate better between the two cases.

We will model our hypothetical birdfluid in $\mathbb{R}^n$ as a scalar bird-density field $\rho = \rho(t, \mathbf{x})$, for points $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, with a vector-valued velocity field $\mathbf{v}(t, \mathbf{x})$ taking values in $\mathbb{R}^n$. When convenient, we will also denote the velocity in component form by $\mathbf{v} = (v_1, v_2, \ldots, v_n)$. While fluid velocity may be more convenient to visualise, we will actually work in the pointwise fluid momentum $\rho(\mathbf{x})\mathbf{v}(\mathbf{x})$ as we (would like to) have a conservation law for this quantity. In fact, we need to be careful how we set up our interaction in order to make this a physically sensible system – i.e., we would also like energy conservation, and

we will try to make our interactions respect this. A lot of the discrete computational models that I have seen either have a fixed bird velocity, or hard-limit the maximum and minimum velocities. This is unphysical, and we would like to avoid it if possible.

We want to derive the equations of motion for our physical system. We'll be using some slightly modified Euler equations for an inviscid fluid – these are not complicated, but it's instructive to work through them, so we will do that here. Firstly we need expressions for the various forces that act on the birdfluid at each point. In the discrete case, these forces depend on the means of quantities like position and velocity within a certain neighbourhood of any given boid, and this is achieved by summing over only those boids within a certain separation distance from that boid. This may alternatively be considered as summing over *all* boids, but with contributions weighted by distance – in this case, the weighting is achieved through the application of a step-function mask, which is a radially symmetric function taking the value 1 or -1 between zero and some given constant radius, and taking the value 0 outside of this radius. We will of course be integrating rather than summing, but the principle will be the same: we will specify a radially symmetric weighting function for our integration. We could also choose a step function, but we will delay that choice and just work in general terms for the time being.

Alignment Force

This interaction causes the velocity of the birdfluid at a point to rotate towards the mean vector of the neighbouring birdfluid. What we must be careful about here is, as with all velocity-dependent forces in physical systems, that we do not break energy conservation. When we rotate a vector, its length will change unless that rotation is in a direction orthogonal to the direction of the vector. The

act of pointwise rotating our birdflow velocity based on a local mean is not an equal-and-opposite reaction; there is no potential associated to it, so if this operation changes the length of the velocity vector then it is creating and destroying kinetic energy. Thus in order to retain energy conservation we need this rotation to act everywhere orthogonally to the birdflow. We will construct a manifestly orthogonal force.

Firstly, pick a point $\mathbf{x}_0 \in \mathbb{R}^n$. Let $\rho_0 = \rho(\mathbf{x}_0)$ and $\mathbf{v}_0 = \mathbf{v}(\mathbf{x}_0)$. Define the *local mean velocity* $\mathbf{v}_{loc}$ to be the vector-valued function:

$$\mathbf{v}_{loc}(\mathbf{x}_0) = K_1 \int_{\mathbb{R}^n} F_1(\mathbf{x} - \mathbf{x}_0)\rho(\mathbf{x})\mathbf{v}(\mathbf{x})\mathrm{d}\mathbf{x}$$

where here K_1 is some constant inversely proportional to the total birdmass $\int_{\mathbb{R}^n} \rho(\mathbf{x})\mathrm{d}\mathbf{x}$, and $F_1(\mathbf{x} - \mathbf{x}_0)$ is our weighting function, centred on the point $\mathbf{x}_0$. Without going too much into the specifics of what this function should be, we will say: firstly, it should be a function of radius only; $F_1(\mathbf{x}) = F_1(\|\mathbf{x}\|) = F_1(r)$, secondly, it should satisfy $F_1(0) = F_1(\infty) = 0$ as we should not have the flow at a point interacting either with arbitrarily distant points or with itself (note that, even in the discrete case, we were only averaging over all *other* boids), and thirdly it should possess only one maximum at some positive radius, being monotonically increasing and decreasing for smaller and larger radii respectively.

Now that we have the (appropriately weighted) local mean velocity, we want to find the direction in which to rotate the velocity $\mathbf{v}_0$ in order to take it towards the local mean. Clearly then the vector difference $\mathbf{v}_{loc}(\mathbf{x}_0) - \mathbf{v}_0$, which is the direct, straight-line distance between the two, provides the necessary direction

for the rotation. We just need to ensure orthogonality to $\mathbf{v}_0$, and we do this by only subtracting a sufficient multiple of $\mathbf{v}_0$ to make the dot product $\mathbf{v}_{loc} \cdot \mathbf{v}_0$ qual to zero. We find:

$$(\mathbf{v}_{loc} - k\mathbf{v}_0) \cdot \mathbf{v}_0 = 0 \qquad \iff \qquad k = \tfrac{\mathbf{v}_{loc} \cdot \mathbf{v}_0}{\|\mathbf{v}_0\|^2}.$$

So let's set

$$\mathbf{v}_{Al}(\mathbf{x}_0) = \mathbf{v}_{loc}(\mathbf{x}_0) - \frac{\mathbf{v}_{loc}(\mathbf{x}_0) \cdot \mathbf{v}_0}{\|\mathbf{v}_0\|^2}\mathbf{v}_0$$

to be our *alignment* vector. Then since we would like the total contribution of this vector to depend on the bird density at $\mathbf{x}_0$, we set the alignment force $\mathbf{F}_{Al}$ acting at this point to be

$$\mathbf{F}_{Al} = \rho_0 \mathbf{v}_{Al}.$$

Recall that we have the freedom to scale this quantity by using the constant term K_1 in front of the integral. Also note how this force behaves: the acceleration is at its greatest magnitude when the velocity at $\mathbf{x}_0$ is orthogonal to the local mean radius about that point, decreasing to zero when the two are parallel. This includes the case when these velocities are *anti*parallel, which would seem to be a problem, but for this situation there is no unique rotation direction to choose to take one vector towards the other. We cannot choose an arbitrary direction, as this would break the generality of the model. We can't shrink or grow velocity

vectors in this situation, as we would lose energy invariance in the equations. This force will just be what it is.

CODE:

Updating velocity with respect to alignment factor between two particles.

```
self.velocity_x += (avg_velocity_x - self.velocity_x) * ALIGNMENT_FACTOR
self.velocity_y += (avg_velocity_y - self.velocity_y) * ALIGNMENT_FACTOR
```

Cohesion Force

We can actually combine the effects of the second and third boid principles into one single force term on the birdfluid. Both the attractive effects of neighbouring bird density and the repulsive effects of birdfluid which is too close can be modelled by the use of a single potential function. Imagine we are computing the behaviour of a collection of particles which experience a long-range attractive force, such as gravitation, but a shorter-range repulsive force such as an electrostatic barrier.

CODE:

If the particle is near to its neighbour, such that the distance between them is less than the Neighbour distance then, incrementing average position on both coordinates by individual position of the particle.

```
if self._distance(particle) < NEIGHBOR_DISTANCE:
    avg_pos_x += particle.position[0]
    avg_pos_y += particle.position[1]
    count += 1
```

Pick a point $\mathbf{x}_0 \in \mathbb{R}^n$. As before, let $\rho_0 = \rho(\mathbf{x}_0)$ and $\mathbf{v}_0 = \mathbf{v}(\mathbf{x}_0)$. The *local mean position* $\mathbf{x}_{loc}$ we define to be the vector-valued function:

$$\mathbf{x}_{loc} = K_2 \int_{\mathbb{R}^n} F_2(\mathbf{x} - \mathbf{x}_0)\rho(\mathbf{x})(\mathbf{x} - \mathbf{x}_0)\mathrm{d}\mathbf{x}$$

where K_2 is a constant, inversely proportional to the total birdmass $\int_{\mathbb{R}^n} \rho(\mathbf{x})\mathrm{d}\mathbf{x}$, and $F_2(\mathbf{x} - \mathbf{x}_0)$ is the weighting function for position, centred on $\mathbf{x}_0$. Again, we won't choose a particular form for this F_2, only demand that it satisfy some fundamental set of properties: firstly, like F_1, it should be a function of radius only, and should evaluate to zero at radius the origin and at infinity. F_2 should be negative for all positive radii less than some constant value, and should be nonnegative for radii above this value. The function should be monotonic everywhere.

$\mathbf{x}_{loc}$ already points in the direction along which we would like to accelerate the birdfluid at $\mathbf{x}_0$. Therefore, we relabel $\mathbf{x}_{loc} = \mathbf{x}_{Co}$, and our analysis of this part of the dynamics is done. We set the *cohesion* force to be:

$$\mathbf{F}_{Co} = \rho_0 \mathbf{x}_{Co}$$

and again, we have the freedom to scale this quantity using the constant K_2 on the integral. One thing to note about the behaviour of this force is that it obeys Newton's third law: pulling the factor of ρ_0 from the expression for the force into the integral giving $\mathbf{x}_{Co}(\mathbf{x}_0)$, we see that the integrand $F_2(\mathbf{x} - \mathbf{x}_0)\rho_0\rho(\mathbf{x})(\mathbf{x} - \mathbf{x}_0)$ acquires a factor of -1 upon the interchange of $\mathbf{x}$ and

$\mathbf{x}_0$. This means that the forces between pairs of distinct points of birdfluid must be equal and opposite.

CODE:

```
self.velocity_x += (avg_pos_x - self.position[0]) * COHESION_FACTOR
self.velocity_y += (avg_pos_y - self.position[1]) * COHESION_FACTOR
```

Updating velocity with respect to alignment factor between two particles.

Again,, updating the velocity with respect to the separation factor between 2 particles;

```
for particle in particles:
    if self._distance(particle) < SEPARATION_DISTANCE:
        move_x += (self.position[0] - particle.position[0])
        move_y += (self.position[1] - particle.position[1])
self.velocity_x += move_x * SEPARATION_FACTOR
self.velocity_y += move_y * SEPARATION_FACTOR
```

Algorithmically,

The Algorithm: (a = acceleration, v = velocity, p = position)

*v_next = v_current + a_current*timestep*
*p_next = p_current + v_current*timestep + 1/2 * a_current*timestep^2*

where we calculate a_current using the four acceleration factors below:

*a_velocity = (average velocity of neighbors - average velocity of boid) * scaling factor*
*a_center = (average position of neighbors - average position of boid) * scaling factor*
a_static = movement away from the closest obstacle involving both a turning factor and a direct movement away in the direction opposite of the displacement.
a_wall = movement away from the closest section of wall involving both a turning factor and a direct movement away in the direction opposite of the displacement.

*The first three factors are added in order of priority (static collision avoidance > velocity matching > flock centering) until either the maximum acceleration level of a boid MA is reached or until all three factors have been added in. The resulting quantity is our initial determination of a_current. If the boid is within range of a wall this value of a_current is averaged with a_wall to give us a new value of a_current. Finally, if a_current*timestep + v_current exceeds the boid's maximum velocity we cut back the acceleration accordingly so that at the next timestep the boid's velocity will be its maximum potential velocity MV.*

The Equations of Motion

Conservation of Mass

Consider the total mass M^V of birdfluid contained in an arbitrary volume V of $\mathbb{R}^n$. We have:

$$M^V = \int_V \rho \mathrm{d}V,$$

and so the rate of change of this birdmass must be

$$\frac{\mathrm{d}}{\mathrm{d}t}M^V = \int_V \frac{\partial}{\partial t}\rho \mathrm{d}V.$$

But if birdmass is conserved, then the rate of change of total birdmass must be given exactly by that which crosses the boundary of this volume:

$$\frac{\mathrm{d}}{\mathrm{d}t}M^V = \int_{\partial V} \rho \mathbf{v} \cdot \mathrm{d}\mathbf{S},$$

which, by an application of Stokes' theorem, becomes

$$-\int_V \nabla \cdot (\rho \mathbf{v}) \mathrm{d}V.$$

We equate this integral with the second one, and, since the volume V was arbitrary, we also equate their integrands. Thus we arrive at the first equation of motion:

$$\frac{\partial}{\partial t}\rho = -\nabla \cdot (\rho \mathbf{v}).$$

Conservation of Momentum

Consider the ith component Π_i^V of the total momentum Π^V contained in an arbitrary volume V of the birdfluid in $\mathbb{R}^n$. We have:

$$\Pi_i^V = \int_V \rho v_i \mathrm{d}V \,,$$

and so

$$\frac{\mathrm{d}}{\mathrm{d}t}\Pi_i^V = \int_V \frac{\partial}{\partial t}(\rho v_i)\mathrm{d}V \,.$$

But, due to conservation of momentum and to Newton's first law, the rate of change of the total momentum contained in the volume V is either due to the flow of momentum across the boundary of V, or it is caused by the action of forces on the birdfluid contained in V. Thus, componentwise, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\Pi_i^V = \int_{\partial V}(\rho v_i)\mathbf{v}\cdot\mathrm{d}\mathbf{S} + \int_V F_i \mathrm{d}V \,,$$

where F_i is the ith component of the net force as a function of position $\mathbf{x} \in V \subseteq \mathbb{R}^n$. Again, by Stokes' theorem, this becomes

$$\int_V (F_i - \nabla\cdot(\rho v_i \mathbf{v}))\mathrm{d}V \,.$$

As above, we equate this with the second integral of this section and argue that, since the volume V is arbitrary, we may equate the integrands. We therefore find the second equation of motion for the birdfluid:

$$\frac{\partial}{\partial t}(\rho v_i) = F_i - \nabla \cdot (\rho v_i \mathbf{v}).$$

Equations of Motion

We can substitute the derived expressions for the pointwise force acting on the birdfluid: $\mathbf{F} = \mathbf{F}_{Al} + \mathbf{F}_{Co} = \rho(\mathbf{v}_{Al} + \mathbf{x}_{Co})$. Without writing these out in their full integral form, we restate the full equations of motion in the one place.

$$\frac{\partial}{\partial t}\rho = -\nabla \cdot (\rho \mathbf{v})$$

.