Final Project Report

IMPROVING THE FLOCKING MODEL

10 May 2023

Carissa Mayo and Anna Hirschmann
University of Colorado Boulder
CSCI 4314: Dynamic Models in Biology
Dr. Orit Peleg

1 Introduction

Throughout the semester, we have studied emergent behaviors in populations of animals and explored how individuals interact to create larger phenomena. Through our class lectures and assignments, we have explored flocking—the grouping and travelling of animals together—and our goal in our project was to improve our existing model for this behavior.

There are many factors that govern animal interactions, a number of which were explored in homework 3. We were able to investigate how different levels of attraction, repulsion, and heading drove the simulated movement of the flock and studied how different inputs impacted the performance of the simulation.

Our existing model relies on constant user-specified parameters which indicate the influence of attraction, repulsion, heading, and randomness on the updated velocity of each boid—or agent in the model—at each timestep. Clearly, all four of these are important to the construction of a biologically accurate model of multi-agent interaction. We note, however, that many other factors play a role in large-group emergent behaviors.

Social agents are driven by a number of elements, including the decisions made by others [1]. In the existing models, the synchronous updates of the position and directional velocity of the boids took into account the status of all other individuals in the flock, but favored those who were closest. Thus, the simplified model placed emphasis on the pairwise interactions of nearby individuals, modulating the impacts of the various types of interactions based on the input parameters. Each agent experienced an attraction to those around it, counterbalanced by a level of repulsion that encouraged the individuals to not collide with one another. The heading factor drove the realignment of the direction of travel, ultimately encouraging the boids to travel in the same direction, while the randomness intended to give the model more biological merit.

This model allowed for the visualization of some of the collective behavior seen in nature, although it used an implementation that focused more on a static interaction between agents than a dynamic one. That is, once the flock formed successfully and the boids were headed the same direction, the model did not leave much to be observed. This unrealistic result left us wondering if there were simple updates to the code that would make the model more applicable to observed biological interactions.

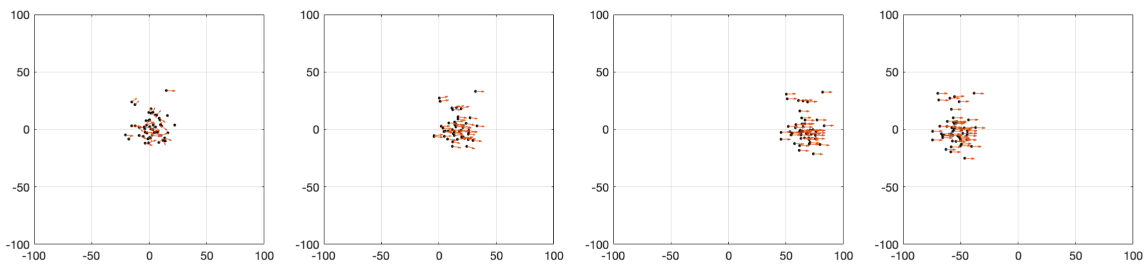
In our project, we will explore some updates to the simulation that could enhance the efficacy of the model and make it more true-to-life. We intend to investigate the results of giving each of the agents biological characteristics (i.e. a field of vision) as well as introducing "predator" agents to study how this impacts the model and its updates over time. It is our hope that this gives us a deeper understanding of the observed interactions between animals and the mathematical theory behind modelling these interactions in a computer simulation.

2 Background

Flocking is a collective behavior that has intrigued biologists for a long time. Many attempts have been made to model and simulate flocking as scientists try to understand the underlying mechanisms of the behavior [2]. Flocking has been proposed to provide a number of benefits for individuals, including increased time and success foraging as well as lower risks of predation [3]. Because flocking seems to provide benefits for animal survival, thorough and biologically valid models are necessary to fully understand and exploit this behavior, which has a wide range of biological applications.

While our existing code provided a rudimentary visualization of flocking behavior, there are many assumptions made by the model that threaten its suitability for biological application. Performing a parameter sweep (as we did in HW3) we were able to see interesting behaviors and dynamics emerge, which helped us to understand the impact each of the inputs had on the resulting agent behavior. In addition, we were able to integrate static "obstacle" agents in the model to see how boids individually responded under different parameter constraints. This investigation provided us with an in-depth physical understanding of simulated flocking behavior, though it left something to be desired as a biological model. This inspired us to study the consequences of adding biological characteristics—rather than just physical characteristics like attraction and repulsion—on the patterns that emerged in the simulation.

Some updates we considered applying to the model were: more realistic and biologically motivated random behavior (to mimic the lack of predictability of the environment), addition of biological characteristics (for example vision and hearing) to boid agents that contributed to their behavior in the simulation, and inclusion of predator agents to study the response of the flock. We intended originally to integrate the ideas about random walks that we discussed in lecture to the larger flocking model, but found this difficult. Instead, we decided to update the biological validity of the agents in the simulation. We were successfully able to integrate fields of vision for boids as well as predator agents.



(a) frame 1 (b) frame 5 (c) frame 15 (d) frame 32

Figure 1: Original flocking model.

We wanted to include a field of vision component because, obviously, flocking animals are responding to visual cues as they interact and group together. Without a high heading component, individuals will rely on their visible surroundings to group together. Something we saw in the original system were unrealistic instantaneous updates in direction of travel for individual boids, which we aimed to limit by restricting the amount of radial adjustment that could be made at each time step. This helped us see more realistic behaviors and dynamic updating of the flocking simulation if we lowered the influence of the heading component on the velocity updates of the individuals.

Likewise, avoiding predation is considered one of the benefits of forming flocks in prey animals [3]. For this reason, we wanted to study the impact of adding a predator agent to the updated simulation in order to study the response of individuals as well as the group. Because we implemented static obstacles in homework 3, we used a similar technique to introduce predator agents, which were moving agents with high repulsion.

The results of our updates to the simulation are described below.

3 Methods

As indicated, we were successfully able to extend the flocking model from class to incorporate two additional elements: a field of vision and a simulated predator (which was integrated similarly to the investigation of obstacles in HW3). Our code is included in the appendices for reference.

3.1 Field of Vision

All boids were given the same search radius and search angle values for the entirety of the simulation. While iterating through each boid in each frame of the simulation, the algorithm first checked if there were any boids within the defined radius, R, using the distance equation (1) where x_n, y_n iterates through all boid positions and x_0, y_0 is the current boid.

$$\sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} < R \tag{1}$$

The algorithm then further checked if there were any boids within the search angle, θ, of the current boid. The angle between each boid in the radius and the current boid was calculated with the dot product formula (2) where v_0 is the velcoity vector of the current boid and v_i is iterative vector of the position difference of each boid in the radius to the current boid.

$$\arccos\left(\frac{v_0 \cdot v_i}{|v_0||v_i|}\right) < \theta \tag{2}$$

The current boid's velocity is then updated to add the average heading of all other boids in its search radius and angle.

3.2 Predation

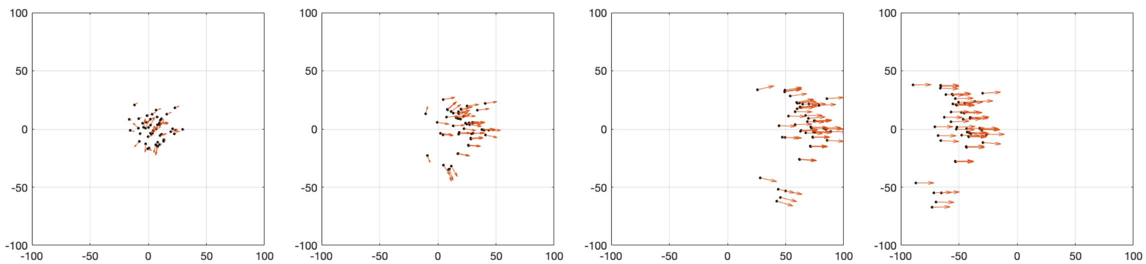
Next, we included the option to add a predator to the model. The predator was implemented like a moving obstacle and was also given a field of vision to add further biological complexity. The predator was introduced as an additional boid in the position and velocity arrays, but was given a separate repulsion value that was much higher than the other boids' repulsion.

The main difference in the calculation of the field of vision for the predator is the way it was updated. Rather than adding the average heading velocity of the boids in the field of vision and radius to the velocity, like the other boids, it was updated with the current heading in addition to the average of the other boids' position differences. This allowed for a chasing behavior of the predator rather than just going with the group. The final difference in the predator field of vision is the search angle and radius. The predator was given a much higher search radius and lower search angle to simulate a predator behavior of searching and narrowing in on prey.

4 Results

In order to evaluate our additions to the flocking model and to determine any improvements, the results of our project are evaluated qualitatively by comparing the updated model with field of vision to the original model in various parameter cases. Then, we will investigate the changes and reactions of the field of vision model when a moving obstacle, such as a predator, is introduced to the system.

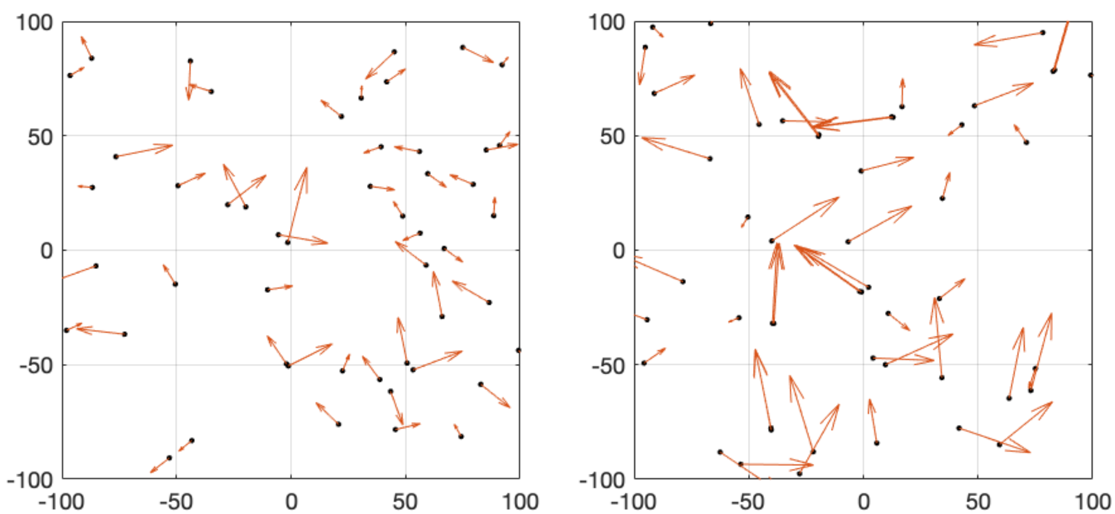
First, we will look at how the boids form a flock without field of vision and with field of vision. In both cases, the model defines the heading to be in the positive x-axis direction for each run and all scaling factors are set to be the same. The main differences are in the first 30 frames when the boids are starting in random directions and positions and begin moving towards the heading. The original flocking model, in Figure 1, shows the sudden change of direction of the boids towards the heading, especially in frames 1 through 5. The field of vision is introduced in Figure 2. Here, we can see the more gradual change in direction of the boids. We can see the boids on the bottom of Figure 2 move together throughout the frames and gradually turn in an arc motion towards the same heading as the rest of the flock.



(a) frame 2 (b) frame 6 (c) frame 15 (d) frame 32

Figure 2: Field of vision flocking model.

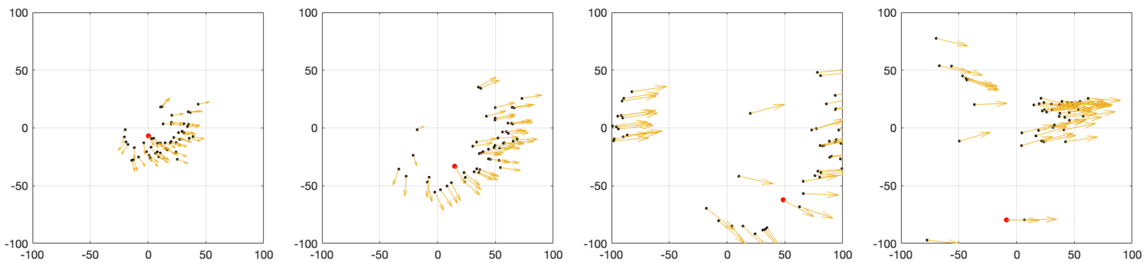
The main improvement of our model compared to the original model is in the case of a low heading. When there is no heading scaling factor in the original model, the boids are not able to form any collective movement as shown in Figure 3.a, but when field of vision is implemented, small collective behavior is formed in Figure 3.b as most boids were able to travel with other boids in their field of vision.



(a) Original model: frame 200 (b) Vision model: frame 200

Figure 3: No heading scaling factor for the original model and the model with field of vision implemented.

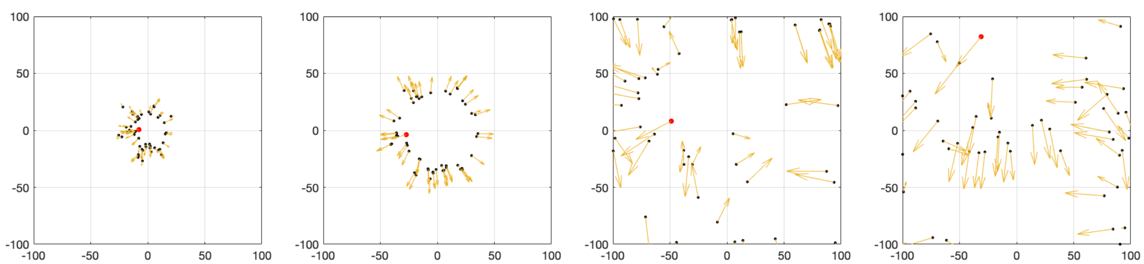
The final result from this project is the introduction of a predator to the field of vision model. To first investigate the dynamics of this implementation, we examine the predator in the case of the flocking model with a normal heading scaling factor of 1. From Figure 4.a we can see the effect of the high repulsion right from the start at frame 5 and the predator is chasing those in the field of vision. Figure 4.b shows the interesting behavior of the predator not yet singling one boid out, but it follows a group. Then in Figure 4.c we can see the predator single out one boid, due to its narrower search angle, and chase it for the rest of the time of the 50 frames. Figure 4.d shows another interesting behavior of all of the other boids forming a line outside of the repulsion area of the predator.



(a) frame 5 (b) frame 11 (c) frame 20 (d) frame 50

Figure 4: Field of vision flocking model with a predator and a heading scaling factor of 1. The predator boid is marked in red.

No heading factor is interesting to examine because it gets rid of the line formation shown in Figure 4.d since the boids have more freedom to move around. This adds complexity with predator-prey interactions. To briefly summarize this simulation, the predator seems to find one group of boids and ends up singling in on one, but after some time, it ends up passing another group of boids and singles another one out.



(a) frame 2 (b) frame 6 (c) frame 65 (d) frame 141

Figure 5: Field of vision flocking model with a predator and no heading scaling factor. The predator boid is marked in red.

5 Conclusion

In this project, we aimed to update the flocking model to improve its observational efficacy. We were able to implement a successful field of vision attribute that impacted the boid adjustment, as well as adding moving “obstacle” objects to simulate predators. Ultimately, we have found that field of vision provides the boids with a more collective motion.

5.1 Challenges

The main challenge faced in this project was determining which updates to our existing model were feasible to make the results more biologically significant. Originally, we had planned to introduce a random walk to influence the movement of each of the individual boids that comprised the flock. However, we quickly learned that this was not a complex problem as a random walk parameter is very similar to the heading parameter. Therefore, when this implementation was attempted, only one line of code was used and there was no significant improvement from the original flocking model.

This original exploration of updating randomness drove us to consider alternate ways to update the model that would produce visible results. We were able to include fields of vision for the flocking agents as well as introducing a predator, which was an extension of the obstacle exploration we did in homework 3.

5.2 Future Directions

One improvement that can be made for this model are the predator parameters to increase its biological validity. For example, when chasing prey, its field of vision most likely changes as it narrows in on its prey of choice over time. Also, there are more complex speed dynamics involved with chasing prey that could be incorporated, such as, speeding up once the predator chooses its prey or slowing down to choose a different boid. These dynamics would be interesting to explore in this model, however, this is slightly straying from the main objective of this project of improving the flocking model.

References

[1] R. P. Mann, “Collective decision making by rational individuals,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 44, pp. E10 387–E10 396, 2018. [Online]. Available: https://www.jstor.org/stable/26532464

[2] W. A. Thompson, I. Vertinsky, and J. R. Krebs, “The survival value of flocking in birds: A simulation model,” *Journal of Animal Ecology*, vol. 43, no. 3, pp. 785–820, 1974. [Online]. Available: http://www.jstor.org/stable/3537

[3] M. A. Elgar and C. P. Catterall, “Flocking and predator surveillance in house sparrows: Test of an hypothesis,” *Animal Behavior*, vol. 29, no. 3, pp. 868–872, 1981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000334728180022X

[4] Orit Peleg, “Code for hw3 csci 4314/5314 dynamic models in biology,” https://canvas.colorado.edu/courses/91860/assignments/1589336, 2015, accessed: 2023-05-10.

Project Contributions

Anna Hirschmann was responsible for doing outside research for the formulation of the project topic. Anna focused on the communication of the project through the Introduction, Background and Conclusion sections as well as communicating the planned approach during the oral presentation.

Carissa Mayo focused on implementing our updated model in MATLAB and generating figures that demonstrated the model performance. Carissa was responsible for writing up findings in the Methods, Results, and Conclusion sections of the paper, and presented background information orally.

Code

We include the code we developed for the project below. We note that this code built on the existing flocking model published by Orit Peleg [4].

```
function y_out = Flocking_FieldOfVision()

close all;
set(0,'Defaultlinelinewidth',5, 'DefaultlineMarkerSize',6,
   ...
    'DefaultTextFontSize',5, 'DefaultAxesFontSize',18);

% initialize parameters
N = 50;          %No. of boids
frames = 200;     %No. of frames in movie
limit = 100;        %Axis limits
L=limit*2;
P = 10;           %Spread of initial position (gaussian)
V = 10;           %Spread of initial velocity (gaussian)
delta = 1;        %Time step
c1 = .00001;     %Attraction scaling factor (.00001)
c2 = .01;        %Repulsion scaling factor (.01)
c3 = 0;           %Heading scaling factor (1)
```

```
c4 = 6;         %Field of vision scaling factor
c5 = .1;         % randomness scaling factor (.1)
vlimit = 5;       %Maximum velocity
predator = true;
c6 = 20;         % Predator repulsion
field_vision = true;

% Field of vision parameters
searchRadius = 1;
searchAngle = 180; %Degrees

%Initialize
if(predator)
    p = P*randn(2,N+1);
    v = V*randn(2,N+1);
    v = v./vecnorm(v);
    figure();

    PredsearchAngle = 20;
    PredsearchRadius = 50;
else
    p = P*randn(2,N);
    v = V*randn(2,N);
    v = v./vecnorm(v);
    figure();
end

%Main loop
for k=1:frames
    v1=zeros(2,N);
    v2=zeros(2,N);
    v4=zeros(2,N);
    v5=zeros(2,N);
    if(predator)
        agent = zeros(2,N);
    end

    %Compute heading [alignment] v3
```

```
%v3 = [sum(v(1 ,:))/N; sum(v(2 ,:))/N]*c3;
v3 = [1; 0]*c3;
%Limit max velocity
if(vecnorm(v3) > vlimit), v3 = v3*vlimit/vecnorm(v3);
   end
for n=1:N
    vCopy = v;
    for m=1:N
        if m ~= n
                                r = p(:, m) - p(:, n);

            if r(1)>L/2, r(1) = r(1)-L;
            elseif r(1)<-L/2, r(1) = r(1)+L;
            end

            if r(2)>L/2, r(2) = r(2)-L;
            elseif r(2)<-L/2, r(2) = r(2)+L;
            end

            %Compute distance between agents rmag
            rmag = sqrt(r(1)^2 + r(2)^2);
            %Compute Attraction v1
            v1(: ,n) = v1(: ,n) + c1*r;
            %Compute repulsion (non-linear scaling)
            v2(: ,n) = v2(: ,n) - c2*r/(rmag^2);

        end
    end
    if(predator)
        r1 = p(:, N+1) - p(:, n);
        if r1(1)>L/2, r1(1) = r1(1)-L;
        elseif r1(1)<-L/2, r1(1) = r1(1)+L;
        end
        if r1(2)>L/2, r1(2) = r1(2)-L;
        elseif r1(2)<-L/2, r1(2) = r1(2)+L;
        end
        rmag1 = sqrt(r1(1)^2 + r1(2)^2);
        agent(: ,n) = agent(: ,n) - c6*r1/(rmag1^2);
```

```
    inPredRadiusIndeces = find(sqrt((p(1,N+1) - p
       (1,:)).^2 + (p(2,N+1) - p(2,:)).^2) <
       PredsearchRadius & sqrt((p(1,N+1) - p(1,:))
       .^2 + (p(2,N+1) - p(2,:)).^2) ~= 0);
    PredcurrVel = vCopy(:,N+1);
    PredcurrPos = p(:,N+1);
    PredsumHead = [0,0];
    Pcount = 0;

    for PredsearchIndex = inPredRadiusIndeces
        PredtargetPos = p(:,PredsearchIndex);

        PredtoTargetVector = PredtargetPos -
           PredcurrPos;

        Predangle = acosd(dot(PredcurrVel,
           PredtoTargetVector)/(norm(PredcurrVel)*
           norm(PredtoTargetVector)));
        % 3) If the other boid is in the search
           radius, add it's
        %       heading to find the average of all
            boids in the search
        %       radius
        if Predangle < PredsearchAngle
            PredsumHead = PredsumHead +
               PredtoTargetVector';
            Pcount=Pcount+1;
        end
    end
     if(Pcount ~= 0)
        PavgHead = PredsumHead/Pcount;
        PavgHead = PavgHead/norm(PavgHead);
        v4(:,N+1) = PavgHead*c4;
    else
        v4(:,N+1) = 0;
    end
end
```

```
      if(field_vision)
  % Implementation of field of vision
          % 1) Check if there are boids in the radius
          inRadiusIndeces = find(sqrt((p(1,n) - p(1,:))
             .^2 + (p(2,n) - p(2,:)).^2) < searchRadius
             & sqrt((p(1,n) - p(1,:)).^2 + (p(2,n) - p
             (2,:)).^2) ~= 0);
          currVel = vCopy(:,n);
          currPos = p(:,n);
          sumHead = [0,0];
          count = 0;
          % 2) For all of the boids in the radius,
          %       determine the angle to it from the
             current boid
          for searchIndex = inRadiusIndeces
              targetPos = p(:,searchIndex);

              toTargetVector = targetPos - currPos;

              angle = acosd(dot(currVel,toTargetVector)
                 /(norm(currVel)*norm(toTargetVector)));
              % 3) If the other boid is in the search
                 radius, add its
              %       heading to find the average of all
                  boids in the search
              %       radius
              if angle < searchAngle
                  sumHead = sumHead + vCopy(:,
                     searchIndex)';
                  count=count+1;
              end
          end
          % 4) Make the heading of the current boid, the
              average of the
          %       others in the field of vision
          if(count ~= 0)
              avgHead = sumHead/count;
```

```
            avgHead = avgHead/norm(avgHead);
            v4(:,n) = avgHead*c4;
        else
            v4(:,n) = 0;
        end
    end

    v5(: ,n) = c5*randn(2,1);
    %Update velocity and position
    if(predator)
        v(: ,n) = v(: ,n) + v1(: ,n) + v2(: ,n) + v3 +
            v4(: ,n) + v5(: ,n) + agent(: ,n);
    else
        v(: ,n) = v(: ,n) + v1(: ,n) + v2(: ,n) + v3 +
            v4(: ,n) + v5(: ,n);
    end
    if(norm(v(:,n)) > vlimit)
        v(:,n) = (v(:,n)/norm(v(:,n))) * vlimit;
    end
end

if(predator)
    v(: ,N+1) = v(: ,N+1) + v4(: ,N+1);

    if(norm(v(:,N+1)) > vlimit)
            v(:,N+1) = (v(:,N+1)/norm(v(:,N+1))) *
               vlimit;
    end
end


%Update position
p = p + v * delta;

% Periodic boundary
tmp_p = p;

tmp_p(1,p(1,:)>L/2) = tmp_p(1,p(1,:)>L/2) - L;
```

```
    tmp_p(2,p(2,:)>L/2) = tmp_p(2,p(2,:)>L/2) - L;
    tmp_p(1,p(1,:)<-L/2)  = tmp_p(1,p(1,:)<-L/2) + L;
    tmp_p(2,p(2,:)<-L/2)  = tmp_p(2,p(2,:)<-L/2) + L;

    p = tmp_p;

    %Update plot:
    if(predator)
        plot(p(1,1:end-1),p(2,1:end-1),'k.','Markersize'
           ,10); hold on;
        plot(p(1,end),p(2,end),'r.','Markersize',20); hold
            on;
    else
        plot(p(1,1:end),p(2,1:end),'k.','Markersize',10);
           hold on;
    end
    quiver(p(1,:),p(2,:),v(1,:),v(2,:)); %For drawing
       velocity arrows
    axis([-limit limit -limit limit]); axis square;
    grid on
    drawnow;
    hold off;
    pause(0.1);
end
y_out=0;


end
```