# Problem

100 ants are randomly placed in a square box with each side as 1m. They move with an average speed of 1cm/s. If the ants collide with each other, they reverse their directions and start moving with the same speed. If ants reach the boundary, apply the periodic condition (i.e the ants will enter the square from the opposite side). Perform the simulations until each ant at least collides once with the other ant. The objective of the study is to obtain the average distance between the ants before they collide with one another

# Assumptions

1. each ant is a circle finite radius, say 5 mm => min 10 mm should be kept between each ant to avoid collision
2. each ant moves one at a time such that other ant stay still but after collision, collided ants reverse direction and then position of all ants get reset to initial. similarly iterated for all other non-collided ants

# Algorithm

1. create position matrix of 100 ants by randomly assigning values between 0 to 1000mm (two 1D arrays for both x and y directions). convert to complex no.
2. similarly create velocity matrix
3. assign size of ant because points cannot collide except perfectly head on collision which is rare since no. of ants are less
4. find position vectors of all ants w.r.t i th ant by subtracting position vector of i th ant form all other ant position vector,
5. find displacement of i th ant.
6. ant crossing happen when displacement vector of i th ant greater than position vector of all other ant w.r.t i th ant (i.e available distance is less compared to next displacement), also there is tolerance of 2\* radius of ant
7. if crossing change direction of velocity vector of that ant
8. otherwise add to new position of ant
9. check for periodic condition and apply correction if needed
10. add a counter to add distance travelled by each ant before collision

n = input('size of ant (assuming a circle - 0.5 to 10 mm)');

pos\_x = randi(1000,[1 100]);

pos\_y = randi(1000,[1 100]);

pos = (pos\_x + 1i\*pos\_y); % using complex no. for easy calculation

save\_pos = pos; % to reset positions

theta = rand([1 100])\*2\*pi -pi;

velocity =cos(theta) + 1i\*sin(theta);

%plot(pos,'o')

%compass(real(velocity),imag(velocity))

quiver(real(pos),imag(pos),real(velocity),imag(velocity),.3,'r')

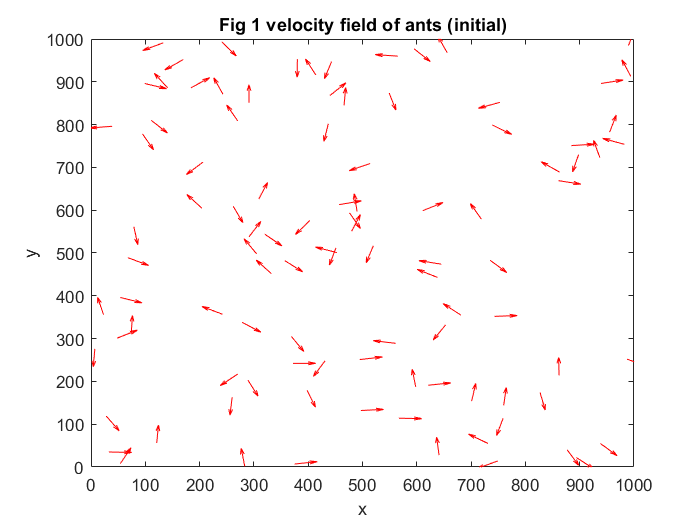
ylim([0 1000])

xlim([0 1000])

title('Fig 1 velocity field of ants (initial)')

xlabel('x')

ylabel('y')



dis = velocity\*1; % t = 1 sec

counter = 0; % to counts the no. of collitions taken by each ant

collide = zeros([1 100]); % collition matrix for each itration (dynamic)

Not\_collided = find(counter==0);

distance = zeros([1 100]);

while (~isempty(Not\_collided)) % run untill there is no element in Non\_collided matrix

v = Not\_collided(1); % take position of first element in non\_collided matrix

flag = 0; % to exit second loop

while(flag ~= 1)

change\_pos = pos - pos(v); % avilable displacement betwwen vth and all other ants (w.r.t vth ant)

collide = abs(change\_pos - dis(v)) < 2\*n; % crossing happen when available distance between v th ant compare to others is less than next displacement (also there is a verialbe tolerance of 2\*n, where n is size of ant)

collide(v) = false; % correction because above code always give this as true

collide\_pos = find(collide); % find position of collition if exist

if collide\_pos % run iff collition matrix has value

collide(v) = true;

counter = counter + double(collide); % add to counter, this helps to keep track of no. of collision for each ant

pos = save\_pos; % reset ant position after collision for next itration

velocity(collide) = -1\*velocity(collide); % but direction is changed

dis = velocity\*1; % t = 1 sec

flag = 1;

elseif isempty(collide\_pos) % runs if no collision

pos(v) = pos(v) + dis(v); % find the next position of v th ant

distance(v) = distance(v) + abs(dis(v)); % add distances

%following codes are to check periodcity and to correct it

pos\_x = real(pos(v));

pos\_y = imag(pos(v));

if pos\_x > 1000

pos\_x = pos\_x - 1000;

end

if pos\_x < 0

pos\_x = pos\_x + 1000;

end

if pos\_y > 1000

pos\_y =pos\_y - 1000;

end

if pos\_y < 0

pos\_y = pos\_y + 1000;

end

pos(v) = pos\_x + 1i\*pos\_y;

else

disp(['error'])

break

end

end

Not\_collided = find(counter==0);

end

avg\_distance = mean(distance);

disp(['avg distance travelled = ',num2str(avg\_distance),' mm'])

avg distance travelled = 361.15 mm

quiver(real(pos),imag(pos),real(velocity),imag(velocity),.3,'r')

ylim([0 1000])

xlim([0 1000])

title('Fig 2 velocity field of ants (final)')

xlabel('x')

ylabel('y')

