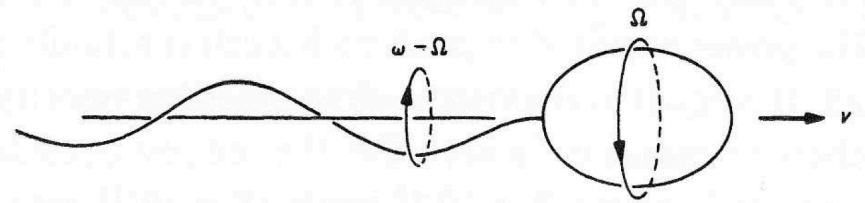
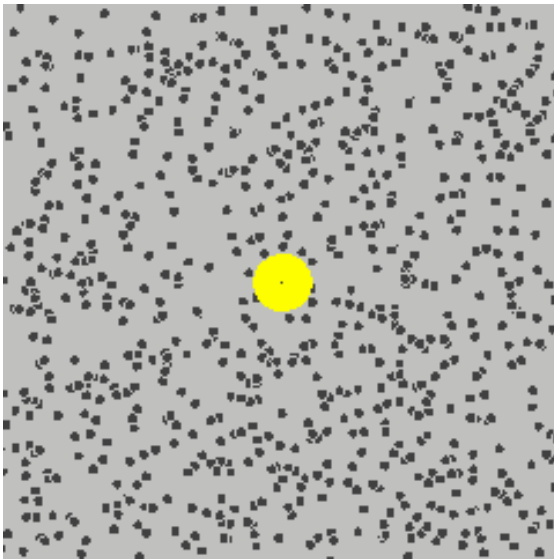
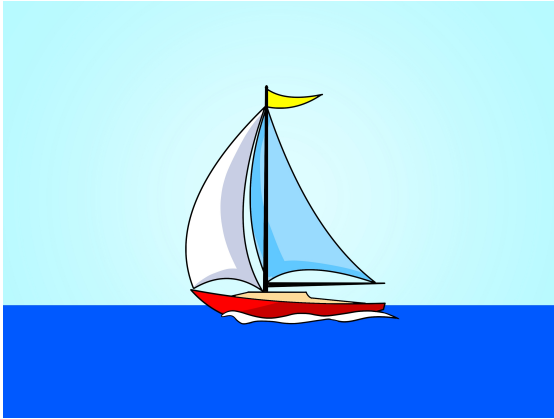


## Current Topics in Biophysics (BPHS 2090)

Instructor: Prof. Christopher Bergevin (cberge@yorku.ca)

Website: <http://www.yorku.ca/cberge/2090F2015.html>

Question: So how can we tell the difference between passive and active?



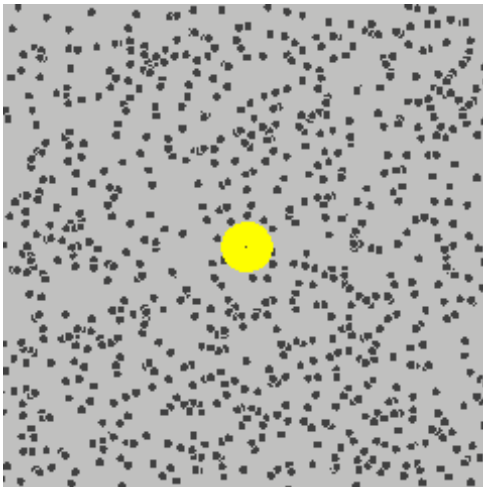
## “Anomalous diffusion”

### Question:

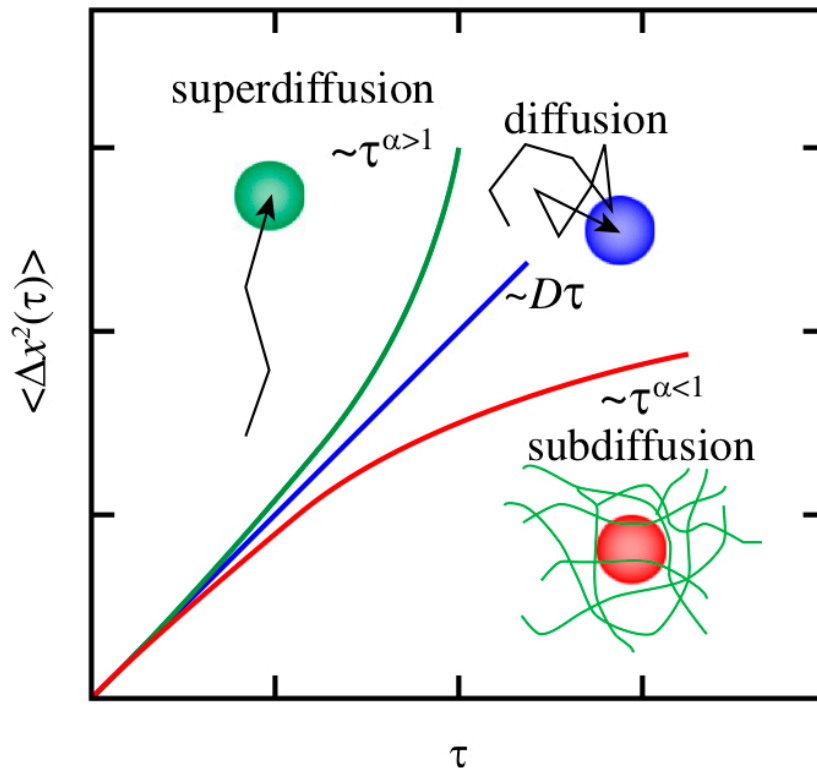
So how can we tell the difference between passive and active?

### Answer:

How “normal” is your diffusion?



$$\langle x^2 \rangle = D t$$



→ What precisely is being plotted here?

**Fig. 1.** Thermal diffusion (blue line) of a particle in a liquid is characterized by an MSD given by  $\langle \Delta x^2(\tau) \rangle = 2D\tau$ , where the displacement  $\Delta x(\tau) = x(t+\tau) - x(t)$  along one axis is measured over a time interval  $\tau$ . In equilibrium, this linear dependence on  $\tau$  is only expected for motion in simple liquids. In viscoelastic materials, such as polymer solutions, subdiffusive motion (red) is expected in equilibrium. By contrast, superdiffusive motion (green) often indicates partially or fully directed motion (e.g., for transport along a substrate) (11).

## Computational aside

- Notion of “mean squared distance” (MSD)

$$\langle x^2 \rangle = D t$$

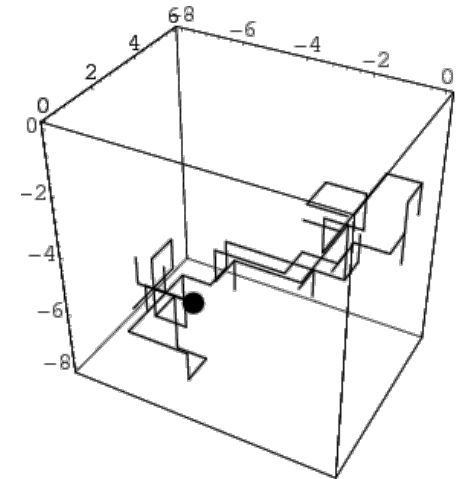
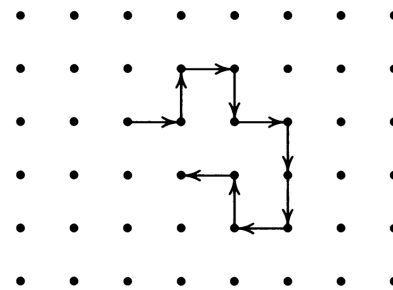
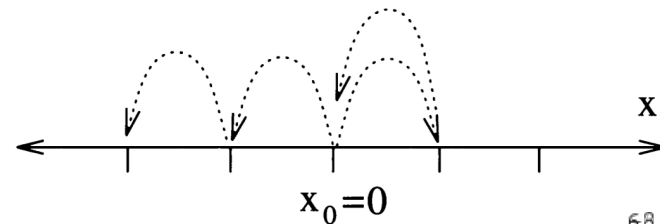
$\langle x^2 \rangle$  - Mean-squared distance

$D$  - ‘diffusion’ constant

$t$  - time allowed before ‘checking’  $\langle x^2 \rangle$

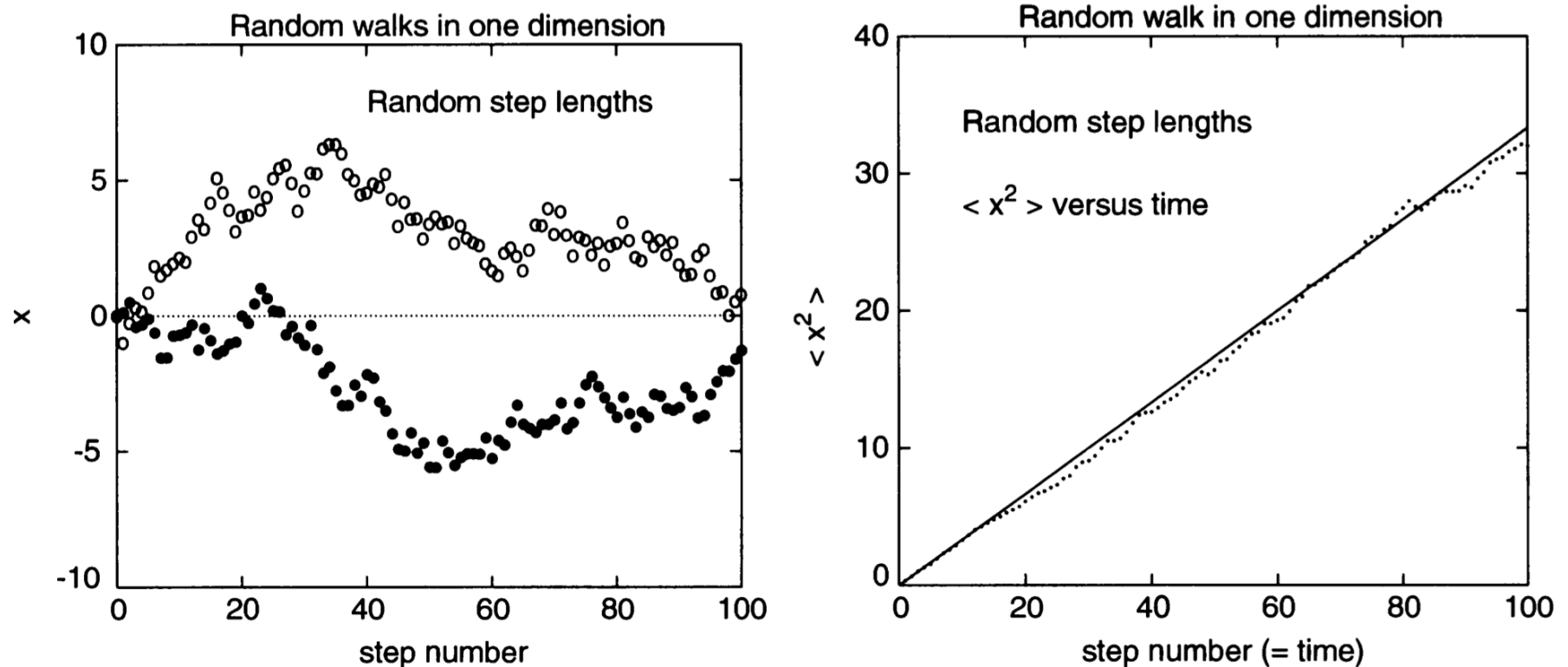
- We can do this in 1-D, 2-D, 3-D, or higher....

- Imagine a grid, upon which we take a step in a random direction. We can then trace out a path as time goes on



- If we consider an ‘ensemble’ of random walkers, each starting at the origin and independent of one another, computationally it’s easy for us to keep track of the *average net movement* (Mean Squared Distance, MSD)

## Computational aside

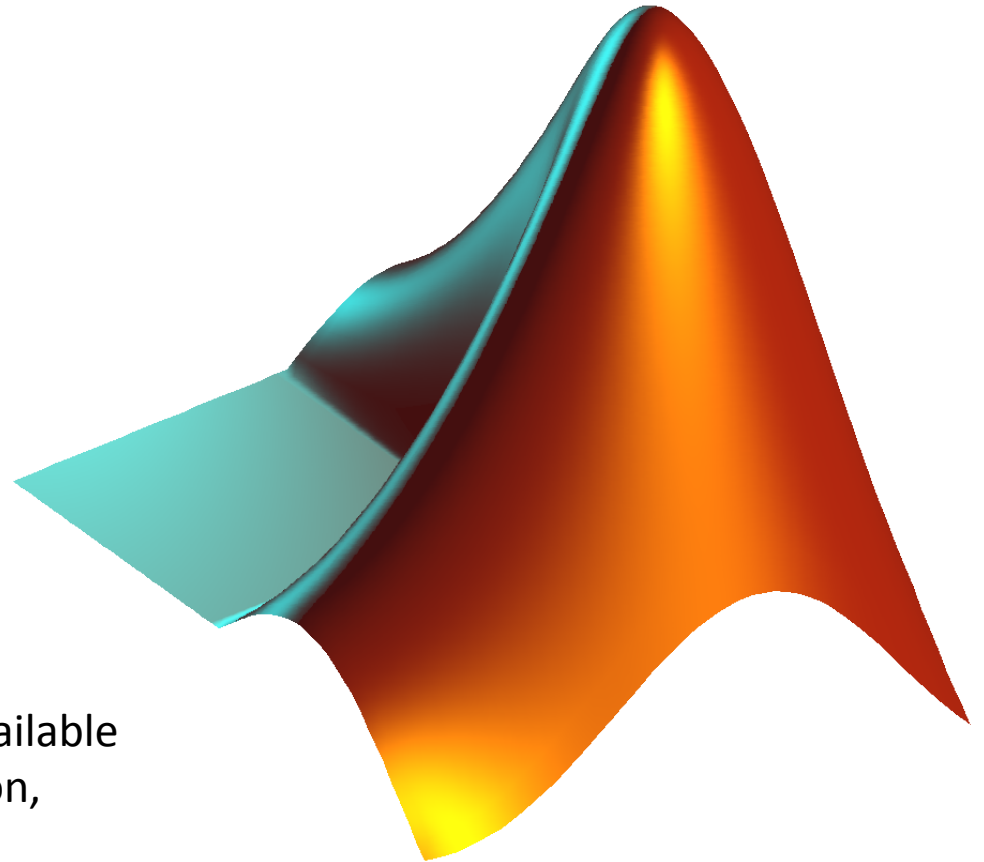


**Figure 7.10:** Left:  $x$  versus step number (that is, time) for two random walks in one dimension. Here the steps were of random lengths in the range  $-1$  to  $1$ . Right:  $\langle x^2 \rangle$  as a function of time for a collection of these one-dimensional random walks. The results for 500 walks were averaged.

→ So while each individual walker is random, the basic idea is that in an *ensemble average*, a repeatable/consistent trend emerges

## Computational aside

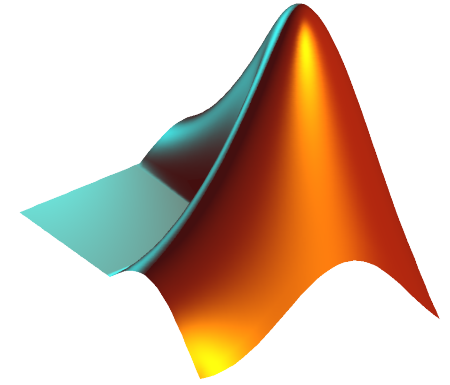
- Want some sort of programmable interface  
(take PHYS 2030!)
- Matlab is one option which we will use here....
- ... but there are many options available (e.g., Mathematica, Maple, Python, Octave, C, Java, etc....)



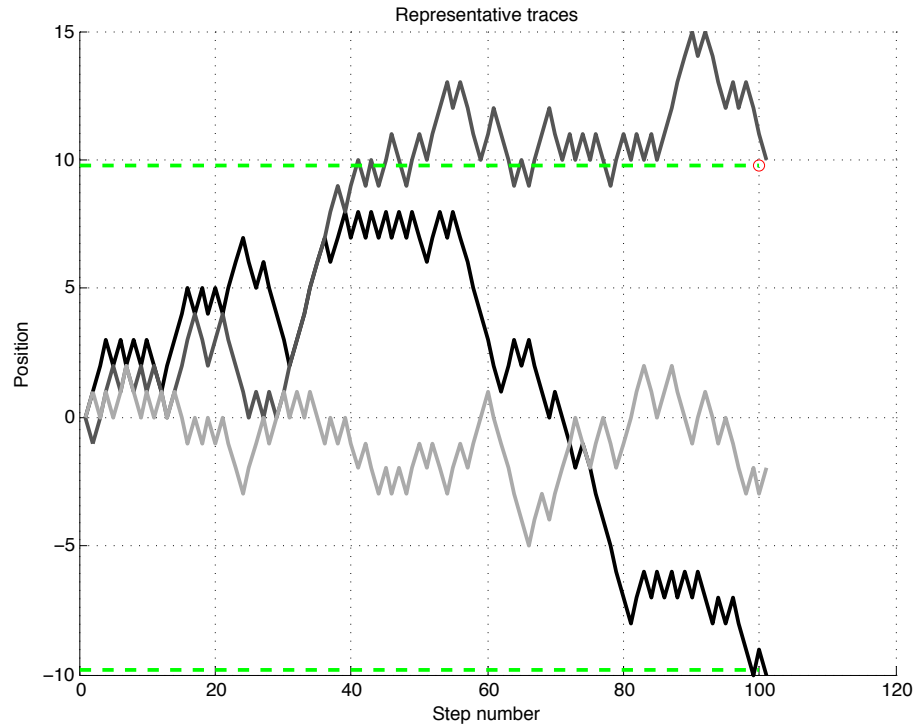
```

% ### EXrandomWalk1D.m ###      11.15.14
clear;
% -----
N= 200;      % Total # of (independent) walkers (each starts at x=0)
M= 100;      % Total # of steps for each walker
K= 3;        % # of walkers to show individual traces for [3]
bias= 0.5;   % number between [0,1] to indicate bias for left vs right (0.5= equal prob.)
% -----
% +++
step_number= zeros(1,M);      %
x2ave= zeros(1,M);           % allocate array to stored (sequentially averaged) MSD
step_number_array= [1:1:M];   %
% +++
%
% NOTE: the loop is set up in such a way to average x2ave across walkers
for r= 1:N
    x=0;      % initialize position for r'th walker
    position(r,1)= 0;
    % loop to go through M steps for r'th walker
    for nn=1:M;
        % conditional determines whether step is to the left or right
        if (rand<bias), x=x+1;
        else x=x-1; end;
        x2ave(nn)=x2ave(nn)+x^2;      % store squared displacement (handles averging across r)
        position(r,nn+1)= x;          % store displacment for each walker and step
    end;
end;
x2ave= x2ave/N;      % Divide by number of walkers
% plot MSD
figure(1);
plot(step_number_array, x2ave, 'k'); hold on;
title('MSD for 1-D random walk');
xlabel('Step number'); ylabel('Mean-Squared Distance (x^2)');
% plot a subset of individual traces
figure(2); clf; hold on; grid on;
for nn=1:K
    shade= 1-(nn-1)/K;
    plot(position(nn,:), 'Color',[1 1 1]-shade);
end
xlabel('Step number'); ylabel('Position'); title('Representative traces');
plot([0 M],[1 1]*sqrt(x2ave(end)), 'g--', 'LineWidth',2) % include MSD bounds at step M
plot([0 M],[-1 -1]*sqrt(x2ave(end)), 'g--', 'LineWidth',2)
plot(M,sqrt(mean(position(:,end).^2)), 'ro'); % reality check (another way to compute final MSD)
disp(['Final mean (non-squared) distance = ',num2str(mean(position(:,end)))]);

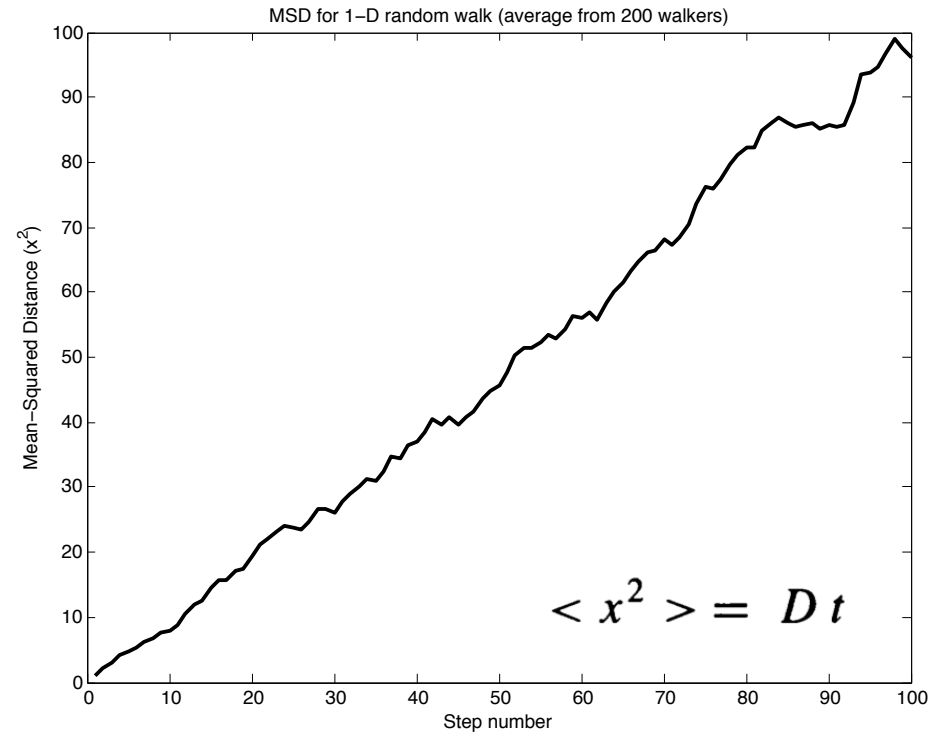
```



- Ensemble of  $N$  (independent) walkers
- Each takes  $M$  total steps, each step either left or right
- Note that the `for` loop averages as it goes

Computational aside

Representative traces from three different  
(independent) walkers

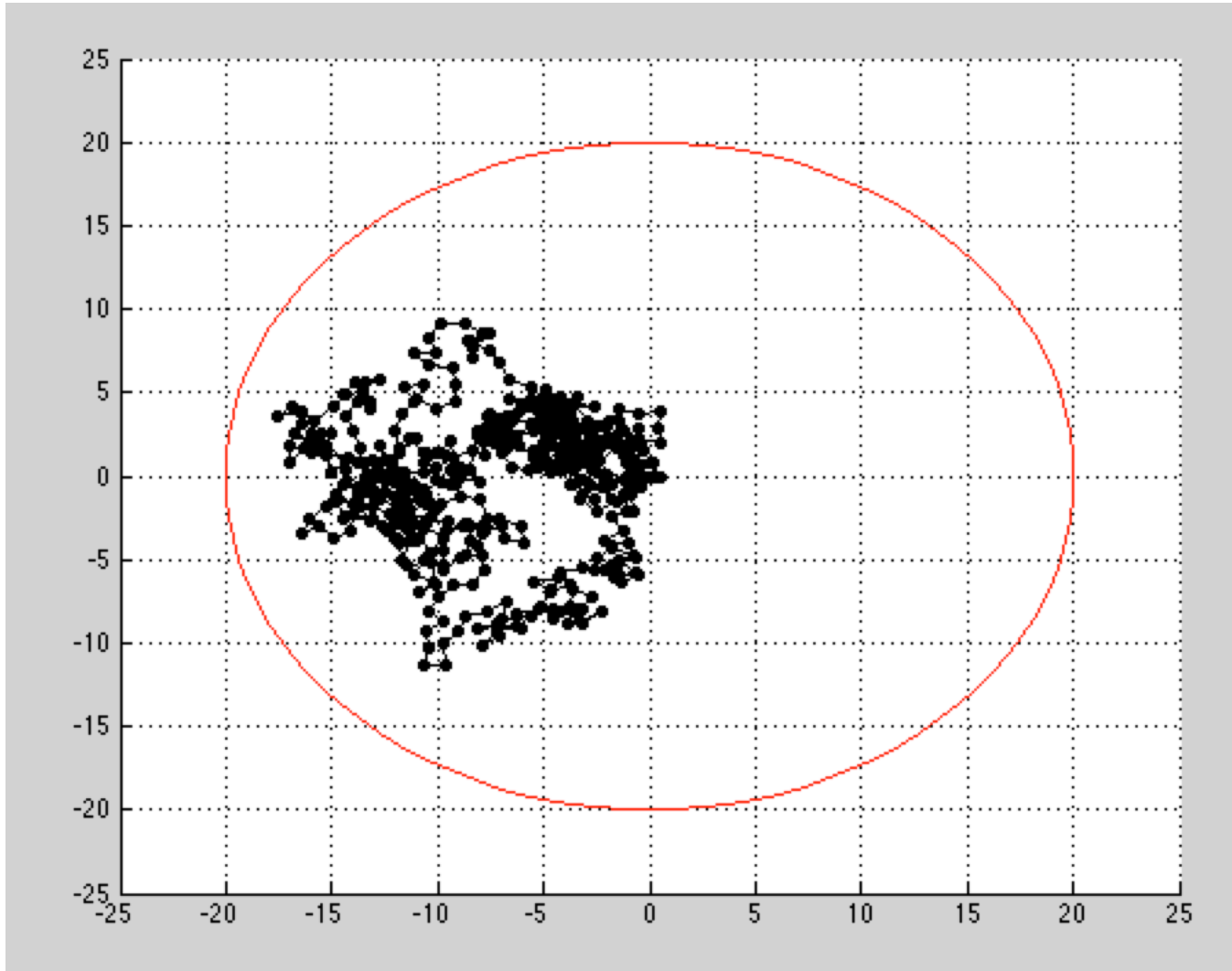


Mean-squared distance traveled by a large  
ensemble of walkers



Computational aside

- Can also compute in 2-D with a boundary (and maybe a bias) in place



```

% ### EXwalker2D.m ###          07.15.15 CB
% Simulates a 2-D Brownian walker
% - polar coords. for computing a step
% - step size is unit value with normally distrib. val.
% - direction-wise, allows for either (via A.Pbias):
% o step direction is uniformly distributed over circle
% o Gaussian-like directional bias (kinda kludgy, but works)
% - [IN PROGRESS] allow for a (circular) boundary condition (via A.Pbound)
% that either reflects or is periodic

clear
% =====
walkerNum= 200; % # of walkers to compute
steps= 500; % # of steps to take by walker
A0= 0.1; % limiting stochastic factor (re 1) for unit step size (A0=0 --> unit radial steps, A0>0 introduce Gaussian variance)
% ---
A.Pbias= 0; % boolean to create a directional bias
A.alpha= 0.2; % bias factor [0,1] --> small (~0.1 means stronger bias)
A.offset= 0.75; % offset direction for bias[cyc]
% ---
A.forcelD= 0; % boolean to force angle to be 0 or pi (thus making this 1-D)
% ---
A.Pbound= 1; % boolean to create circular boundary (i.e., walkers constrained)
A.bndR= 20; % radius of bounding wall (re origin)
A.boundType= 0; % boundary condition (req. A.Pbound=1): 0-"hard" (reflecting), 1-periodic
% ---
axLim= 25; % bounds for plotting (Fig.66)
kk= 1; % particle ID to visualize a single walker (Fig.66)
animate= 1; % boolean to turn on/off movie for an individual walker (Fig.66)
numWplot= 5; % # of walkers to plot individual (r^2) paths (Fig.4)
% =====

% ---
% if a constrained walk, force bounding condition (A.bndR) to be much
% larger than mean unit step size (helps avoid some coding headaches below)
if (A.Pbound==1 && A.bndR <= 5), disp('Make larger bounding condition'); end

for m= 1:walkerNum
% walker m initially at origin [i.e., cartesian (0,0)]
P(m).coord(1,:)= [0 0];

for nn=2:steps
% ----
P(m).A(nn)= 1+ A0*randn(1); % (radial) size of nn'th step for m'th walker
if (P(m).A(nn)<0), P(m).A(nn)=0; end % make zero size step if negative (introduces bias?)
% ----
% direction of nn'th step for m'th walker (allows possibility of bias)
if A.Pbias==0
P(m).theta(nn)= rand(1)*2*pi; % no bias
else
if (m==1 && nn==2), disp('Radial bias in effect'); end
P(m).theta(nn)= (A.offset + A.alpha.*randn(1))*2*pi; % w/ radial bias
end
end

```

```

% -----
% constrain angle such that movement is essentially 1-D
if A.forcelD==1
    P(m).theta(nn)= round(P(m).theta(nn)/(2*pi))*pi;
end
% -----
% update re last position and store away in Cartesian and radial coords.
P(m).coord(nn,:)= [P(m).coord(nn-1,1)+P(m).A(nn)*cos(P(m).theta(nn)) P(m).coord(nn-1,2)+P(m).A(nn)*sin(P(m).theta(nn))];
P(m).rsq(nn)= P(m).coord(nn,1)^2 + P(m).coord(nn,2)^2; % new radial position (squared)
P(m).phi(nn)= atan2(P(m).coord(nn,2),P(m).coord(nn,1)); % angle of new position re origin
% -----
% if constrained, check that new coords. aren't past wall (otherwise "reflect")
if A.Pbound==1
    if (m==1 && nn==2 && A.boundType==0), disp('Circular hard/reflecting boundary in effect'); end
    if (m==1 && nn==2 && A.boundType==0), disp('Circular periodic boundary in effect'); end
    temp1= sqrt(P(m).rsq(nn)); % dummy to reduce re-computation
    if temp1 >= A.bndR
        if A.boundType==0
            % ### HARD REFLECTION ###
            % angle stays the same, only radius changes (and in a simple way)
            temp2= 2*A.bndR- temp1; % reflected radial length
            %disp([temp1 P(m).A(nn) P(m).theta(nn) P(m).phi(nn) temp2]); % for debugging
        elseif A.boundType==1
            % ### PERIODIC B.C. ###
            % both radius changes and angle flips 180
            temp2= 2*A.bndR- temp1; % reflected radial length
            P(m).phi(nn)= mod(P(m).phi(nn)+pi,2*pi);
        end
        P(m).rsq(nn)= temp2^2; % squared version
        % revised Cartesian version
        P(m).coord(nn,1)= temp2*cos(P(m).phi(nn));
        P(m).coord(nn,2)= temp2*sin(P(m).phi(nn));
    end
end
% -----
% determine MSD
P(m).time(nn)= nn; % "time" is simply the step number (can rescale as needed)
%P(m).MSD(nn)= sqrt(P(m).coord(nn,1)^2 + P(m).coord(nn,2)^2); % radial position (not squared)
%P(m).MSD(nn)= P(m).coord(nn,1)^2 + P(m).coord(nn,2)^2; % squared to get the "S" in MSD
P(m).MSD(nn)= P(m).rsq(nn); % note that this is the radial position squared (hence "S" in MSD)
end
end
% -----
% compute mean MSD (across all walkers) --> KLUDGE (better way to do this sans loops??)
for nn=1:steps
    for m= 1:walkerNum
        val(m)= P(m).MSD(nn);
    end
    meanMSD(nn)= mean(val);
end
end

```

```

% -----
% plot vals. for a (specified) individual walker
if l==1
    figure(1); clf;
    subplot(211); plot(P(kk).coord(:,1),P(kk).coord(:,2),'k.-');
    xlabel('x'); ylabel('y'); grid on; hold on; title('Walker position')
    axis([-axLim axLim -axLim axLim])
    % --- (plot a bounding circle)
    if A.Pbound==1
        th= 0:pi/50:2*pi; xunit= A.bndR*cos(th); yunit= A.bndR*sin(th); h66= plot(xunit, yunit,'r-');
    end
    % --- (plot MSD for an individual walker)
    subplot(212); plot(P(kk).time,P(kk).MSD,'k-');
    xlabel('Time'); ylabel('Radial displacement (squared)'); grid on; hold on;
end

% -----
% plot MSD for the ensemble
figure(2); clf;
plot(P(m).time,meanMSD,'k-');
xlabel('Time'); ylabel('MSD'); grid on; hold on;
% if constrained, visualize effective bounding limit
if (A.Pbound==1), h2B= stem(A.bndR^2,max(meanMSD),'r--','LineWidth',1);
    legend(h2B,'Bounding radius (squared)','Location','SouthEast'); end

% -----
% plot distribution of angular values (polar histogram)
if l==0
    figure(3); clf;
    % == (single walker) directions taken for each step for an individual walker
    subplot(221); h3= rose(P(kk).theta,30);
    set(h3,'LineWidth',1.5); x = get(h3,'Xdata'); y = get(h3,'Ydata'); g=patch(x,y,'y');
    title('All steps for a single walker'); grid on; hold on;
    % == (all walkers) directions taken for all steps of all walkers
    subplot(223); h3= rose([P(:).theta],30);
    set(h3,'LineWidth',1.5); x = get(h3,'Xdata'); y = get(h3,'Ydata'); g=patch(x,y,'y');
    title('All steps for all walkers'); grid on; hold on;
    % == (all walkers) final position for all walkers [KLUDE: not sure how to do sans loop]
    for mm=1:numel(P) bank(mm)= P(mm).phi(end); end
    subplot(224); h3= rose(bank,floor(numel(P)/15));
    set(h3,'LineWidth',1.5); x = get(h3,'Xdata'); y = get(h3,'Ydata'); g=patch(x,y,'y');
    title('Final ang. position of all walkers'); grid on; hold on;
end

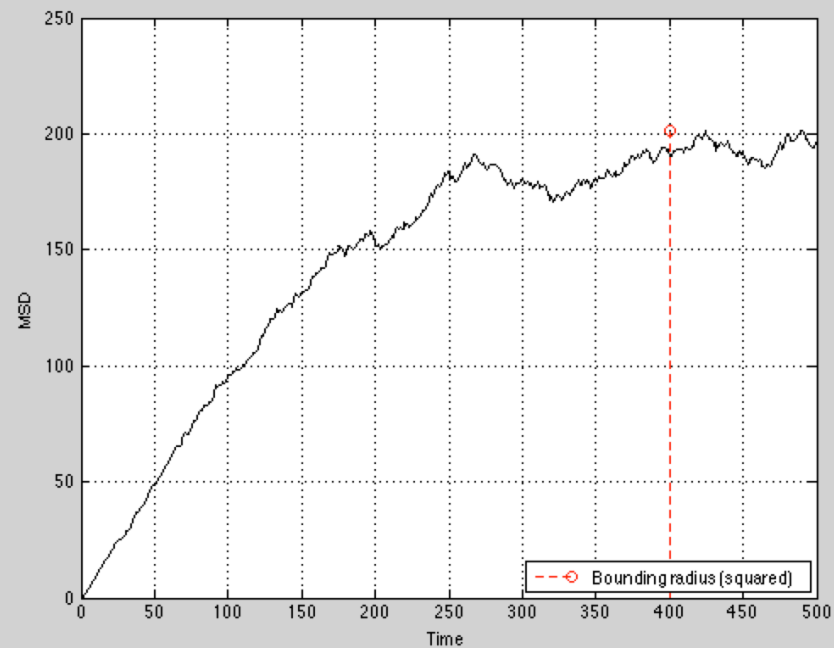
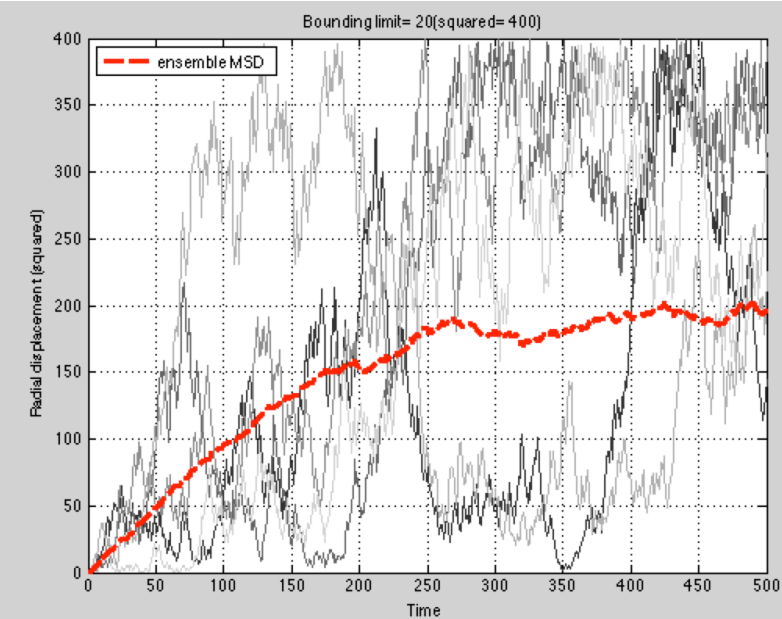
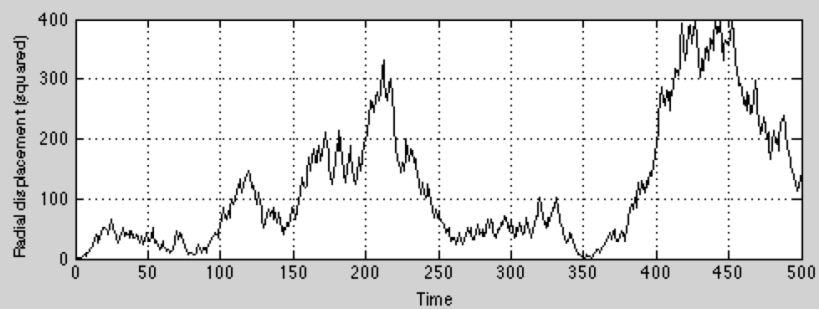
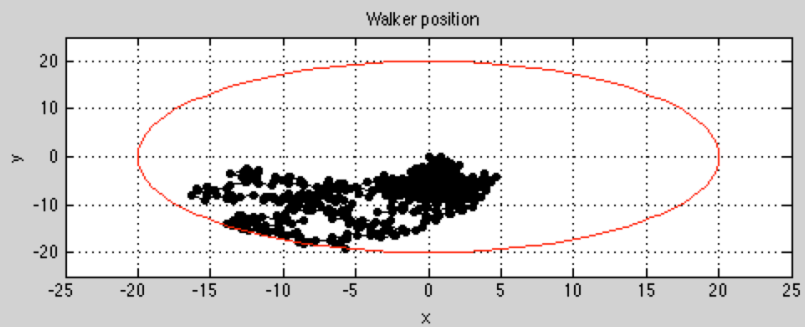
```

```

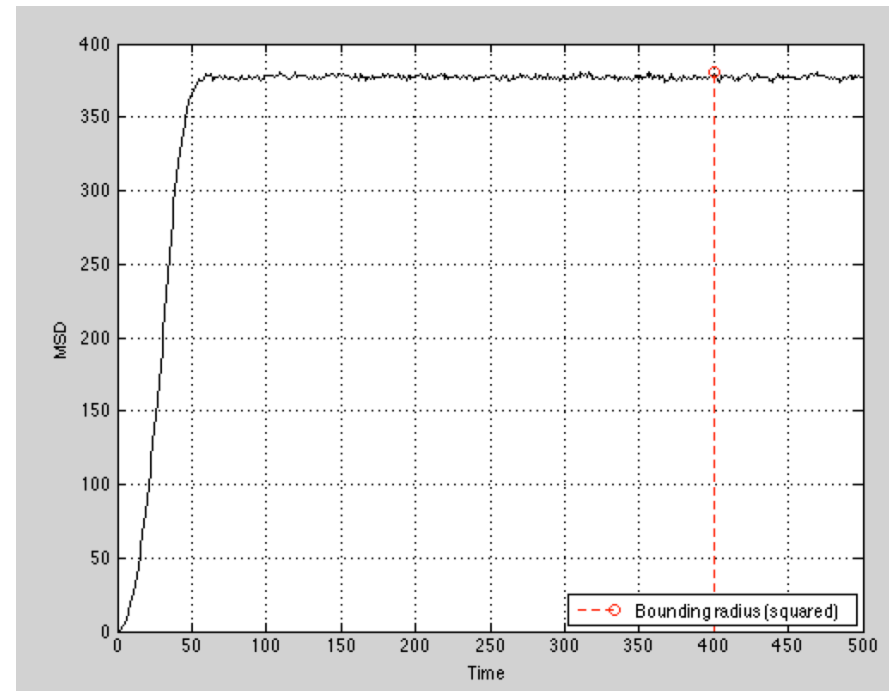
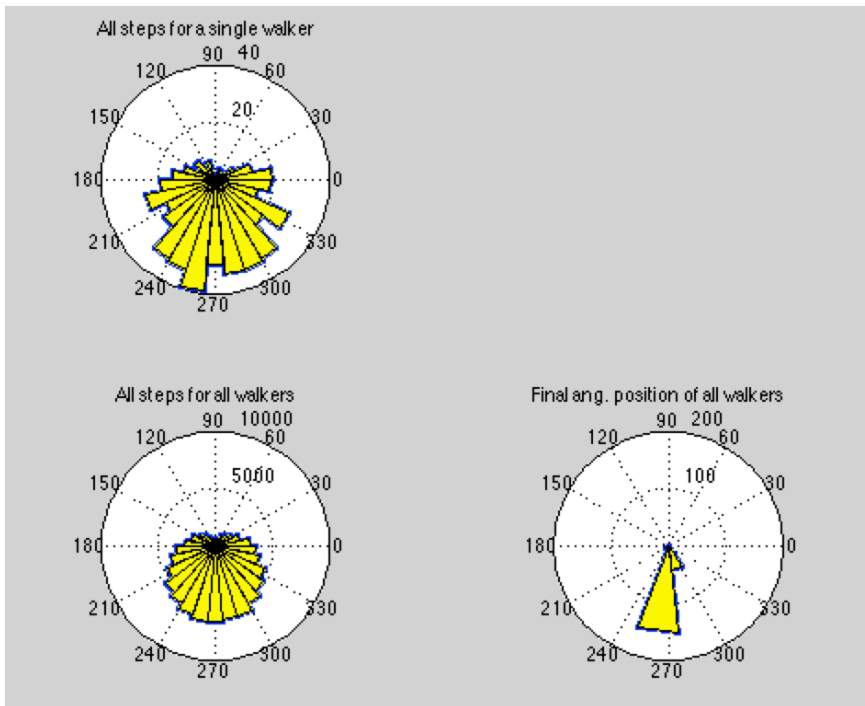
% -----
% plot time course (or r^2) for several walkers? (see also Fig.1B)
if l==1
    figure(4); clf;
    for n=1:numWplot
        hh= 0.8*n/numWplot; % shading factor (to discern different traces)
        plot(P(n).time,P(n).MSD,'-', 'Color',hh*[1 1 1]); grid on; hold on;
    end
    leg= plot(P(m).time,meanMSD,'r--','LineWidth',2); % also plot ensemble MSD
    xlabel('Time'); ylabel('Radial displacement (squared)');
    title(['Bounding limit= ',num2str(A.bndR),'(squared= ',num2str(A.bndR^2),')']);
    legend(leg,'ensemble MSD','Location','NorthWest');
end

% -----
% movie for an individual walker
if animate==1
    figure(66); clf; axis([-axLim axLim -axLim axLim]); grid on; hold on;
    for nn=2:steps
        % --- (plot a bounding circle)
        if A.Pbound==1
            th= 0:pi/50:2*pi; xunit= A.bndR*cos(th); yunit= A.bndR*sin(th); h66= plot(xunit, yunit,'r-');
        end
        % --- (plot/update the track)
        %plot(P(kk).coord(nn,1),P(kk).coord(nn,2),'ko-');
        plot([P(kk).coord(nn-1,1) P(kk).coord(nn,1)], [P(kk).coord(nn-1,2) P(kk).coord(nn,2)], 'k.-');
        pause(0.04); % {0.04}
    end
end
end

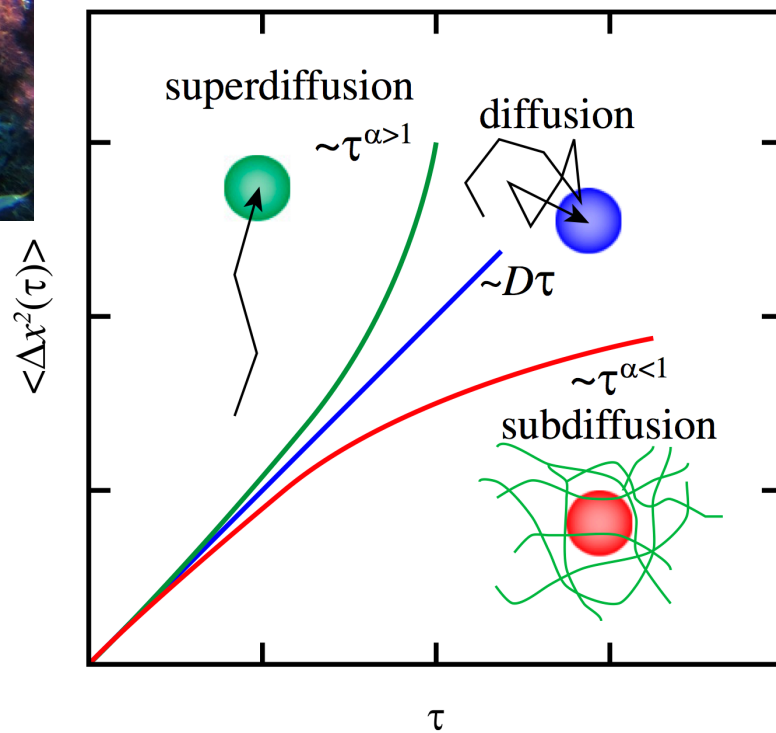
```



➤ Introducing a bias.....



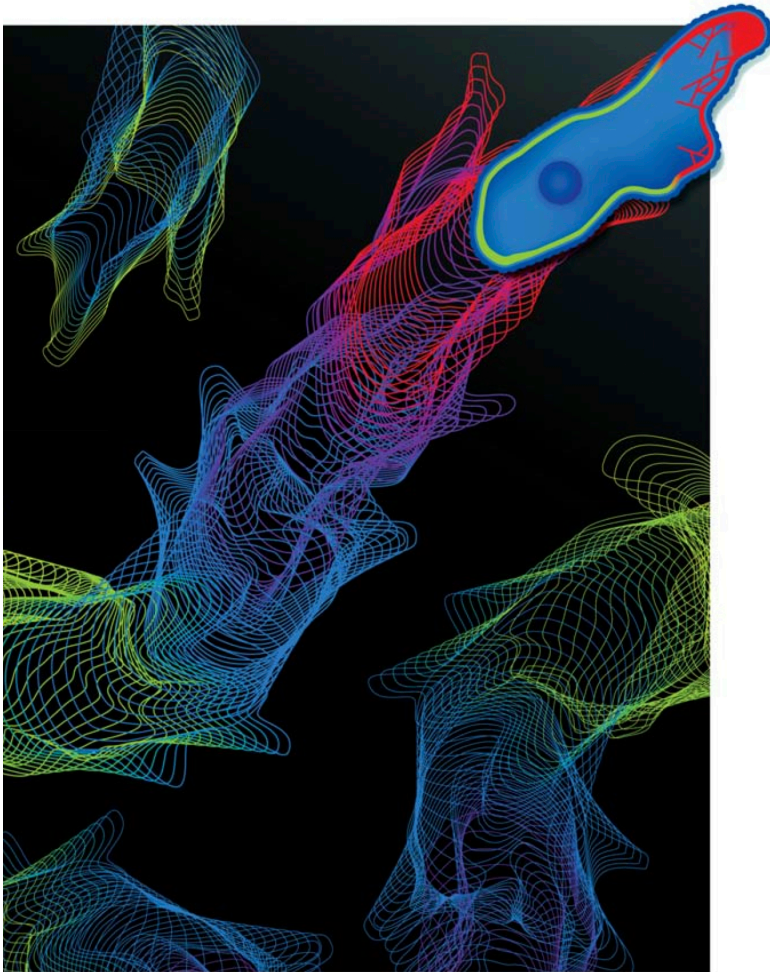
Rough analogy... (ignore convective currents)





So “why” might a bacteria move itself?

- Chemical environment may matter → chemotaxis

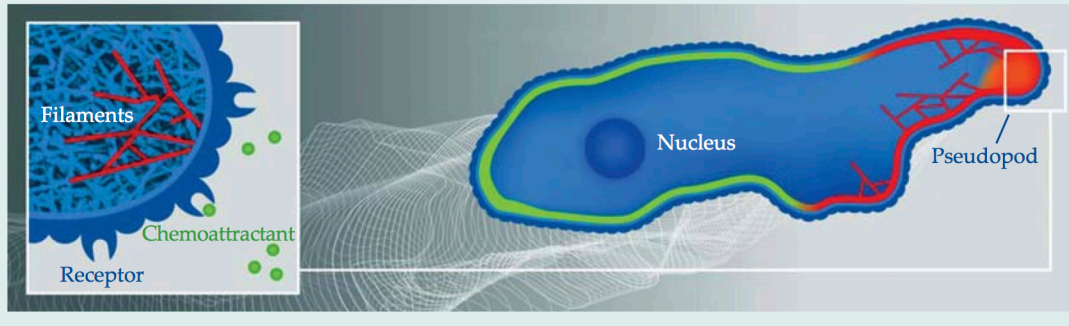


“Ooh, food over that way! Let’s go!”

“Ugh, poison! Let’s get out of here!”

## Case study: Chemotaxis

**Figure 1. Chemotaxis in a nutshell.** This illustration shows a cell moving to the right in response to a chemical gradient indicated by the gray-scaled background. The white lines show the outline of the cell membrane at earlier times. The motion-inducing gradient is detected by the cell through the binding of chemoattractants, shown as green balls in the inset, to receptors bound to the cell membrane. Changes in the shape of the receptors trigger internal signaling pathways; some pathway components are localized at the front of the membrane (red), others congregate at the back (green). Called pseudopods, membrane protrusions (orange) self-organize at the front of the cell. The inset also shows the branched actin network (light blue) that pushes the membrane forward, with several individual filaments highlighted in red.



$S$  – binding const. (0 or 1)

The receptor distribution conveys information about the gradient's direction. To see how, imagine a two-dimensional circular cell of diameter  $d$  with receptors located at equidistant angles  $\theta_j$  and consider the cell placed in a linearly varying field. The statistic  $\Phi$  defined through

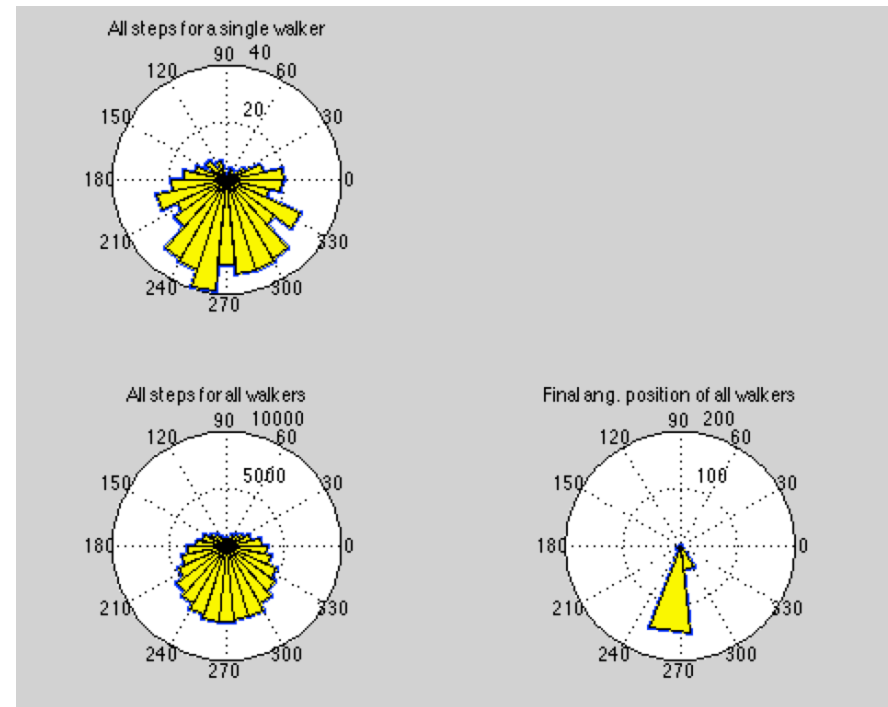
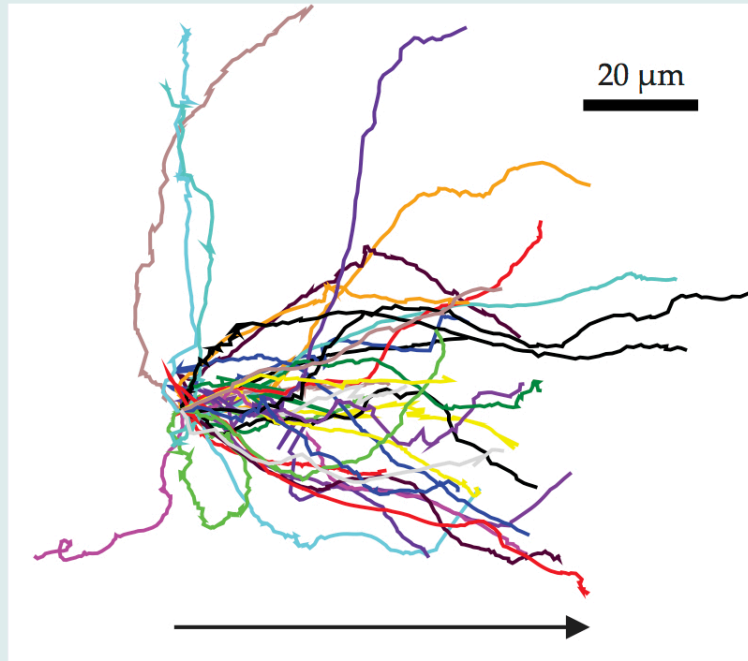
$$\Phi = \text{Arg} \left( \frac{1}{N} \sum_{j=1}^N e^{i\theta_j} S_j \right) \quad (2)$$

acts as an estimator of the gradient direction. (“Arg” stands for “argument,” the phase of the complex number that follows the term.) That is because, in the sum on the right-hand side of equation 2, the  $S$  values on the higher-concentration side of the cell are more likely to be 1 than those on the lower-concentration side; hence the phases on the higher-concentration side are weighted more heavily.

Idea: Sum together various contributions to get a net effective directionality

## Case study: Chemotaxis

**Figure 3. Cell tracks.** The colored lines, whose origins have been brought to a common point, depict the observed paths of *Dictyostelium discoideum* cells through a chemical solution whose concentration changes by 10% across the 10- $\mu\text{m}$  distance typical of a cell diameter. The tracks exhibit considerable motile variability; even for gradients steeper than those pertinent to the figure, a typical maximal chemotactic index, defined as the ratio of the distance traveled in the gradient direction to the total distance traveled, is at most about 0.7. The arrow indicates the direction of increasing chemoattractant concentration. (Adapted from ref. 4.)



## Life at low Reynolds number

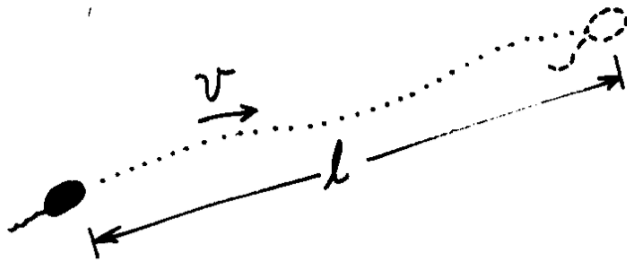
E. M. Purcell

Lyman Laboratory, Harvard University, Cambridge, Massachusetts 02138

(Received 12 June 1976)

American Journal of Physics, Vol. 45, No. 1, January 1977

Note: Purcell (1912-1997) won the 1952 Nobel Prize for discovering NMR



to out-swim diffusion:

$$l \geq D/v$$

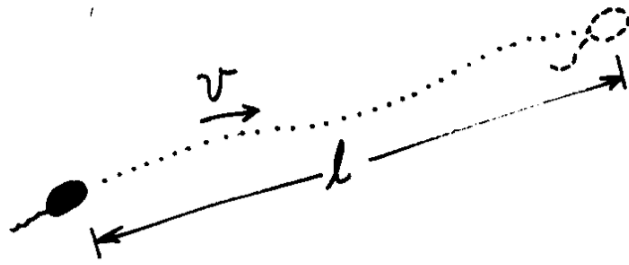
if  $D = 10^{-5} \text{ cm}^2/\text{sec}$ ,  $v = .003 \text{ cm}/\text{sec}$

$$l \geq 30 \mu$$

"If you don't swim that far you haven't gone anywhere."

- How far does a bacteria need to swim?
- Answer depends upon its speed relative to diffusion

Case study: Chemotaxis



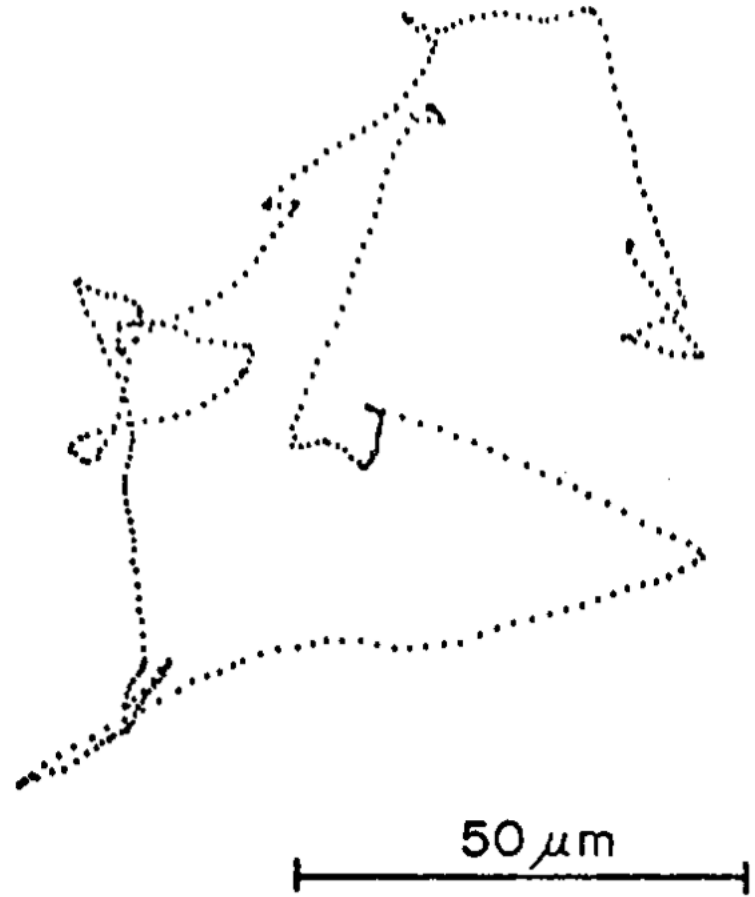
to out-swim diffusion:

$$l \geq D/v$$

if  $D = 10^{-5} \text{ cm}^2/\text{sec}$ ,  $v = .003 \text{ cm}/\text{sec}$

$$l \geq 30 \mu$$

"If you don't swim that far you haven't gone anywhere."

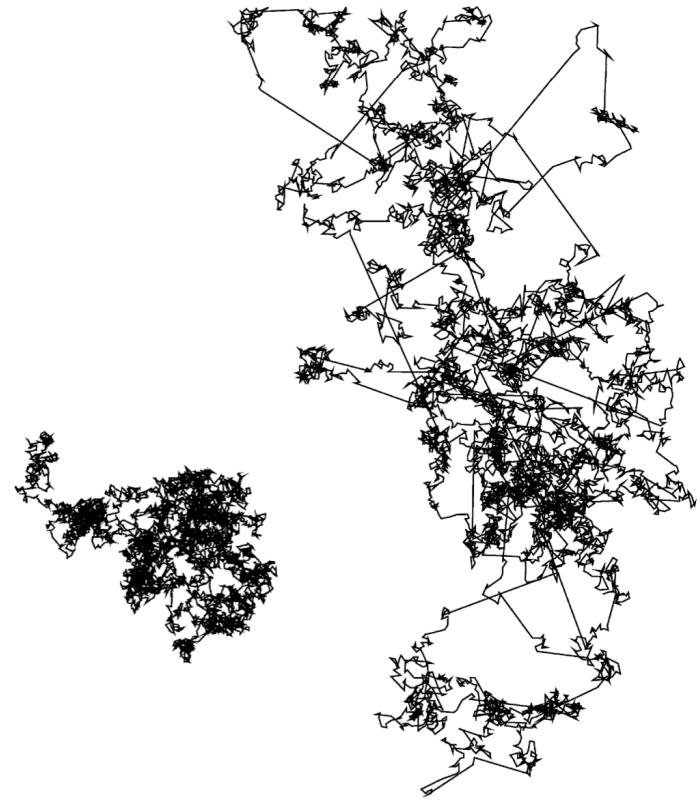


→ Explains some of the structure we saw in Berg's *E. coli* tracks....

## Collective dynamics

- Bacteria swimming alone versus.....
- ... lots of (**interacting**) bacteria swimming together

→ **Swarming**



BACTERIOLOGICAL REVIEWS, Dec. 1972, p. 478-503  
Copyright © 1972 American Society for Microbiology

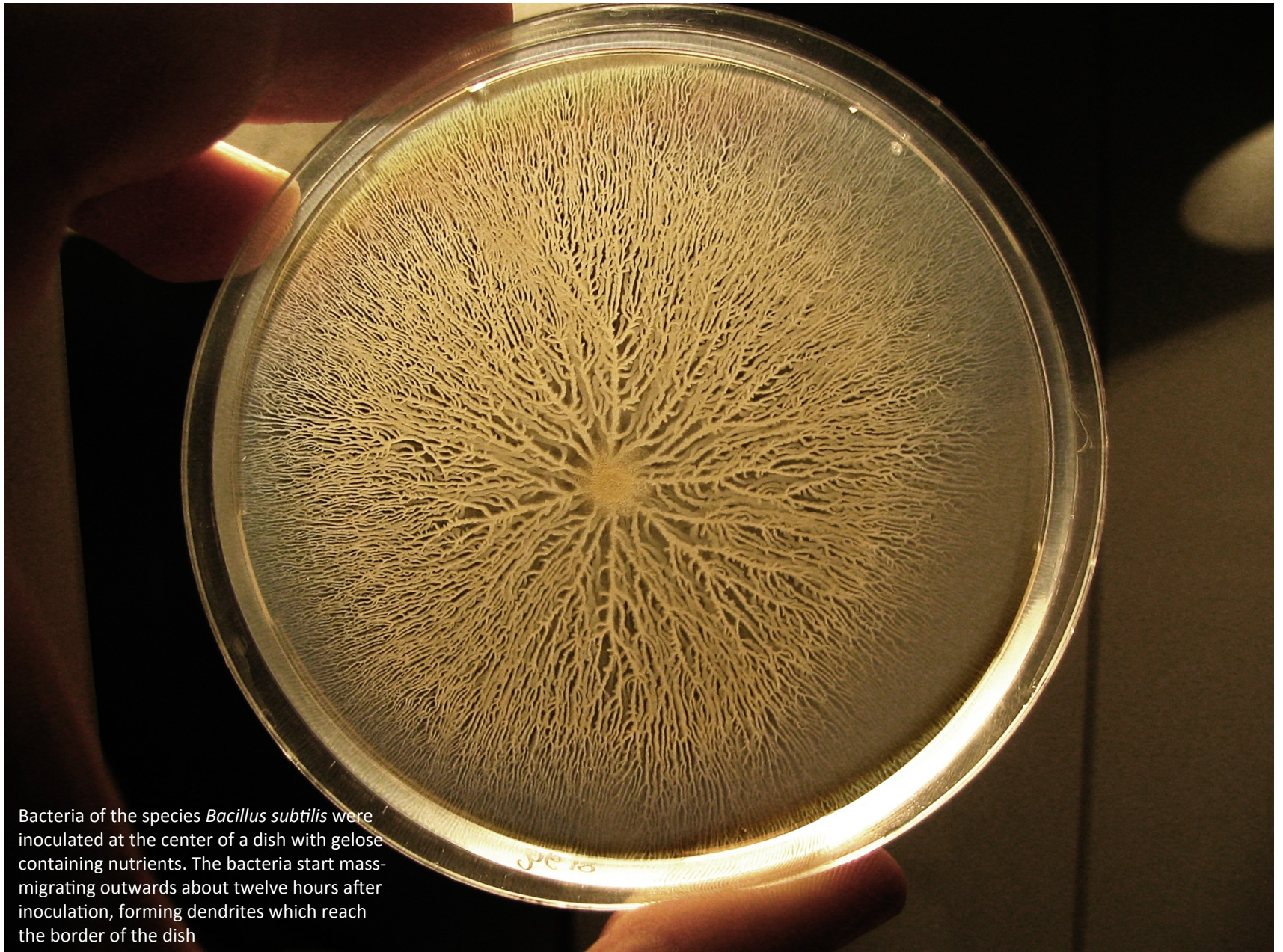
Vol. 36, No. 4  
Printed in U.S.A.

# Bacterial Surface Translocation: a Survey and a Classification

JØRGEN HENRICHSEN

*Department of Diagnostic Bacteriology, Statens Seruminstitut, Copenhagen, Denmark*

## Case study: Swarming

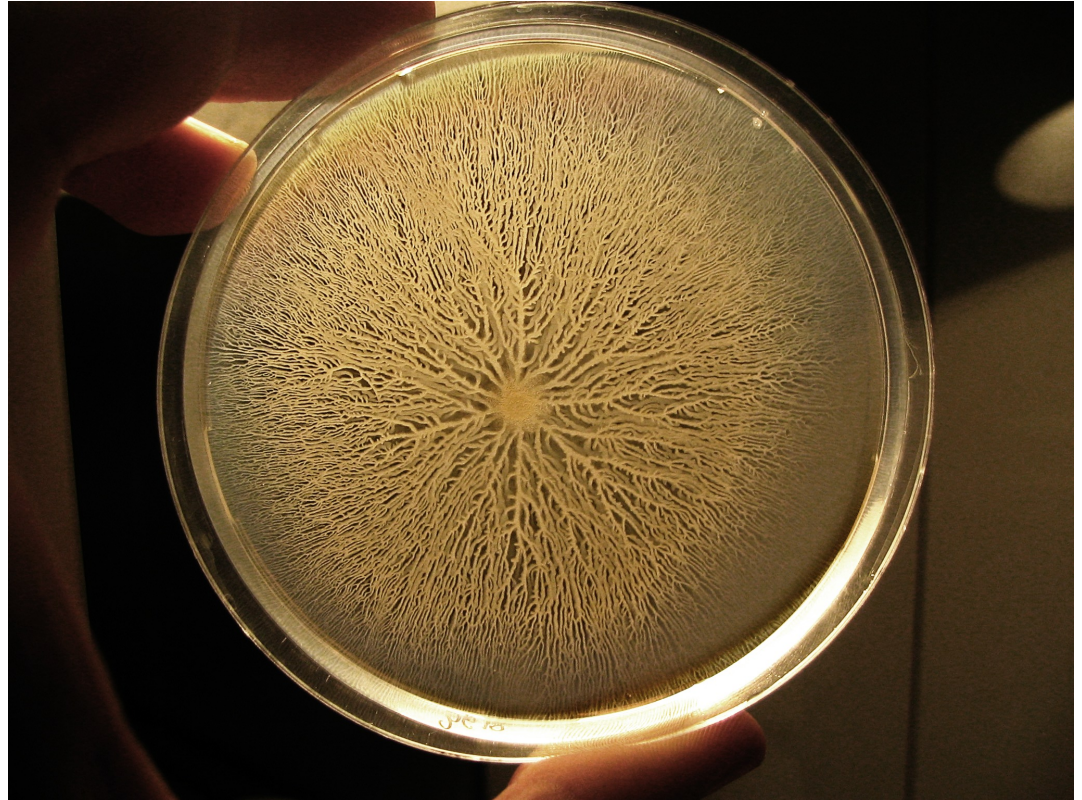


Bacteria of the species *Bacillus subtilis* were inoculated at the center of a dish with gelose containing nutrients. The bacteria start mass-migrating outwards about twelve hours after inoculation, forming dendrites which reach the border of the dish

## Case study: Swarming

- Key idea here is that the swimmers can interact
- Notion of collective dynamics

→ The “whole” is more/  
different from the sum of  
the parts



## Statistical mechanics for natural flocks of birds

William Bialek<sup>a</sup>, Andrea Cavagna<sup>b,c</sup>, Irene Giardinà<sup>b,c,1</sup>, Thierry Mora<sup>d</sup>, Edmondo Silvestri<sup>b,c</sup>,  
Massimiliano Viale<sup>b,c</sup>, and Aleksandra M. Walczak<sup>e</sup>

4786–4791 | PNAS | March 27, 2012 | vol. 109 | no. 13

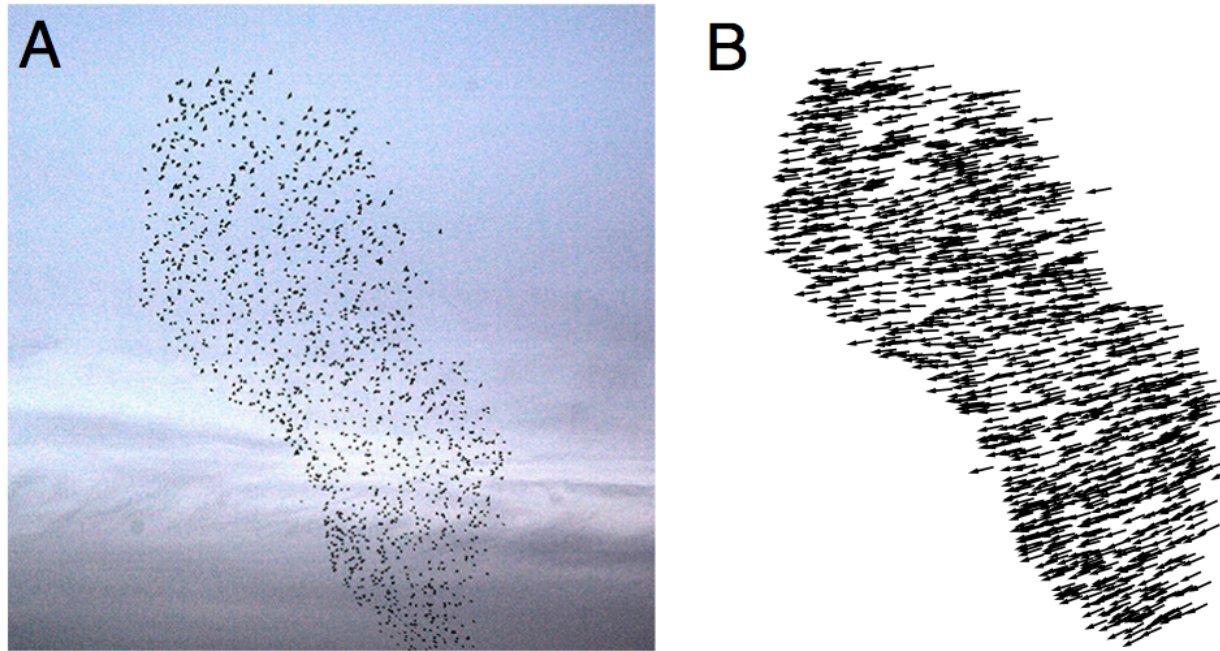
Flocking is a typical example of emergent collective behavior, where interactions between individuals produce collective patterns on the large scale. Here we show how a quantitative microscopic theory for directional ordering in a flock can be derived directly from field data. We construct the minimally structured (maximum entropy) model consistent with experimental correlations in large flocks of starlings. The maximum entropy model shows that local, pairwise interactions between birds are sufficient to correctly predict the propagation of order throughout entire flocks of starlings, with no free parameters. We also find that the number of interacting neighbors is independent of flock density, confirming that interactions are ruled by topological rather than metric distance. Finally, by comparing flocks of different sizes, the model correctly accounts for the observed scale invariance of long-range correlations among the fluctuations in flight direction.



Case study: Flocking



A swarm-like flock of starlings

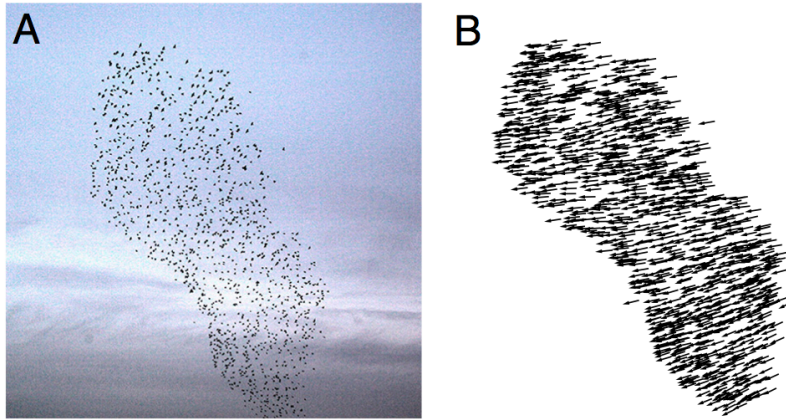


**Fig. 1.** The raw data. (A) One snapshot from flocking event 28 – 10,  $N = 1,246$  birds (see [SI Appendix, Table S1](#)). (B) Instantaneous vector velocities of all the individuals in this snapshot, normalized as  $\vec{s}_i = \vec{v}_i/|\vec{v}_i|$ .

The maximum entropy distribution consistent with the directional correlations  $C_{ij}$  is

$$P(\{\vec{s}_i\}) = \frac{1}{Z(\{J_{ij}\})} \exp \left[ \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij} \vec{s}_i \cdot \vec{s}_j \right], \quad [1]$$

## Case study: Flocking



**Fig. 1.** The raw data. (A) One snapshot from flocking event 28–10,  $N = 1,246$  birds (see [SI Appendix, Table S1](#)). (B) Instantaneous vector velocities of all the individuals in this snapshot, normalized as  $\vec{s}_i = \vec{v}_i/|\vec{v}_i|$ .

We consider flocks of European starlings, *Sturnus vulgaris*, as in Fig. 1A. At any given instant of time, following refs. 11–13, we can attach to each bird  $i$  a vector velocity  $\vec{v}_i$  and define the normalized velocity  $\vec{s}_i = \vec{v}_i/|\vec{v}_i|$  (Fig. 1B). On the hypothesis that flocks have statistically stationary states, we can think of all these normalized velocities as being drawn (jointly) from a probability distribution  $P(\{\vec{s}_i\})$ . It is not possible to infer this full distribution directly from experiments, because the space of states specified by  $\{\vec{s}_i\}$  is too large. However, what we can measure from field data is the matrix of correlations between the normalized velocities,  $C_{ij} = \langle \vec{s}_i \cdot \vec{s}_j \rangle$ . There are infinitely many distributions  $P(\{\vec{s}_i\})$  that are consistent with the measured correlations, but out of all these distributions, there is one that has minimal structure: It describes a system that is as random as it can be while still matching the experimental data. This distribution is the one with maximum entropy (10).

The maximum entropy distribution consistent with the directional correlations  $C_{ij}$  is

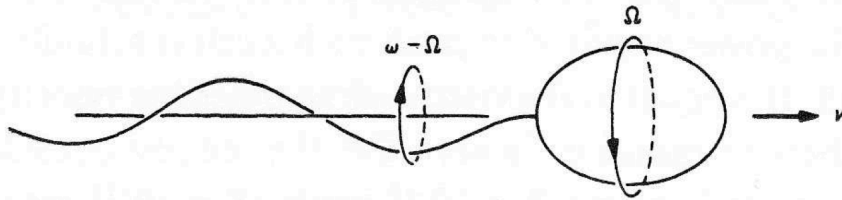
$$P(\{\vec{s}_i\}) = \frac{1}{Z(\{J_{ij}\})} \exp \left[ \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij} \vec{s}_i \cdot \vec{s}_j \right], \quad [1]$$

It should be emphasized that the maximum entropy principle is not a “modeling assumption;” rather it is the absence of assumptions. Any other model that accounts for the observed correlations will have more structure and hence (explicitly or implicitly) assumes something about the nature of the interactions in the flock beyond what is required to match the data. Of course the fact that the maximum entropy model is minimally structured does not make it correct. It could be, for example, that individual birds set their flight direction by computing a complicated non-linear combination of the velocities from multiple neighbors, in which case correlations among pairs of birds would be insufficient to capture the essence of the ordering mechanism. We view the maximum entropy model as a powerful starting point, from which, as we will see, we can generate detailed and testable predictions.

Moving on....

Question:

What differences are there for micro- vs. macro-scopic motors?



## Life at low Reynolds number

E. M. Purcell

*Lyman Laboratory, Harvard University, Cambridge, Massachusetts 02138*

(Received 12 June 1976)

**American Journal of Physics, Vol. 45, No. 1, January 1977**

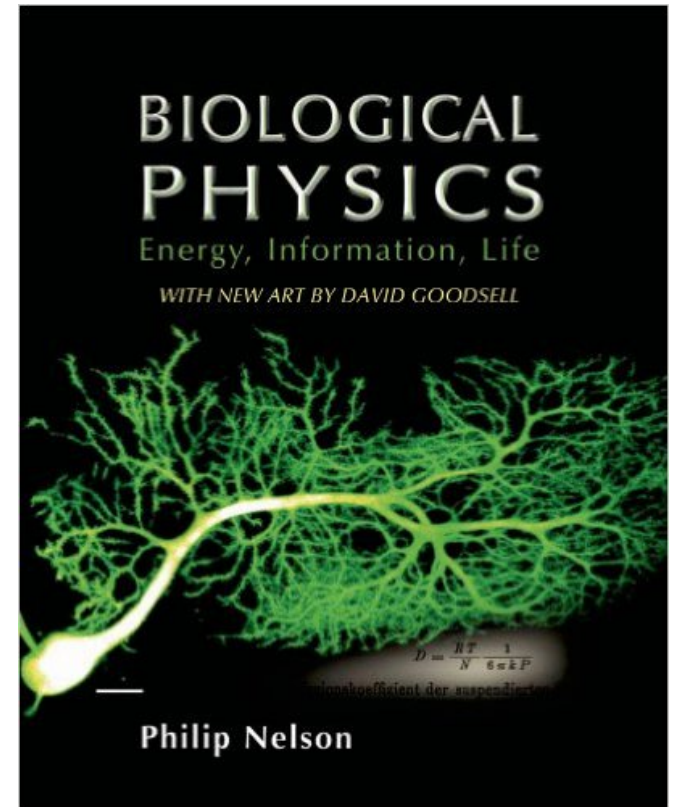
But I want to take you into the world of very low Reynolds number—a world which is inhabited by the overwhelming majority of the organisms in this room. This world is quite different from the one that we have developed our intuitions in.

**Note:** Purcell (1912-1997) won the 1952 Nobel Prize for his work on NMR

Reynolds #

## Chapter 5

# Life in the slow lane: the low Reynolds-number world

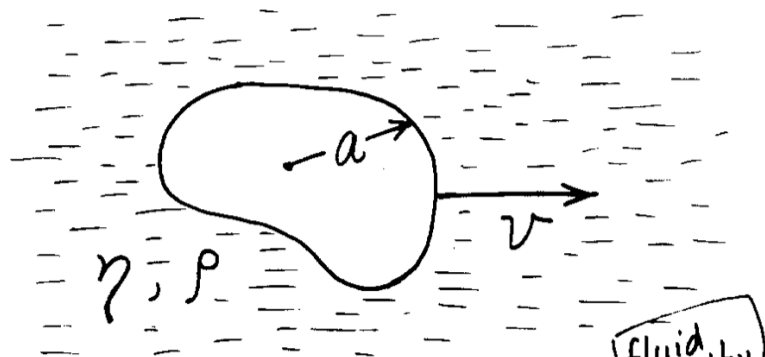


The Focus Question for this chapter is:

*Biological question:* Why do bacteria swim differently from fish?

*Physical idea:* The equations of motion appropriate to the nanoworld behave differently under *time reversal* from those of the macroworld.

## Reynolds #



$\frac{\text{inertial forces}}{\text{viscous forces}}$

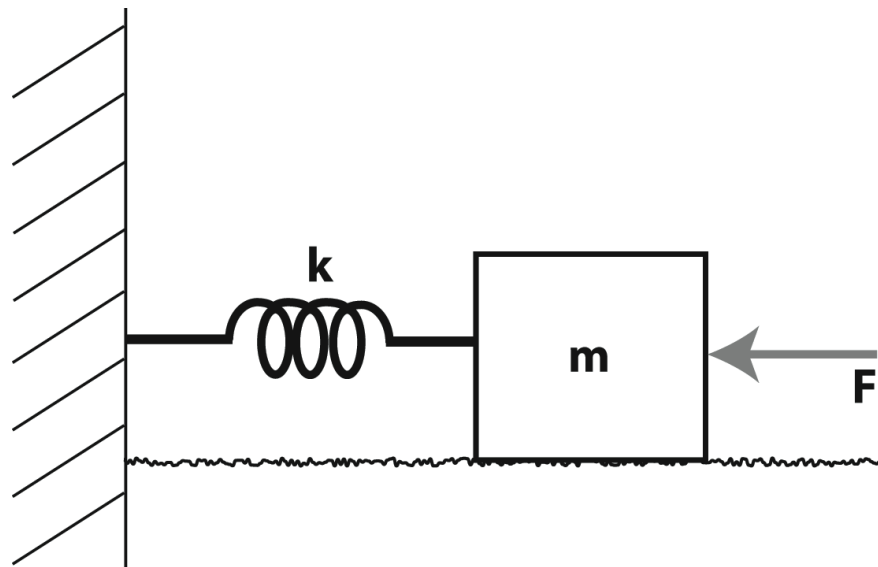
$$\approx \frac{av\rho}{\eta}$$

fluid density  $\rho$   
fluid viscosity  $\eta$

$$R = \frac{av\rho}{\eta} = \frac{av}{\nu}$$

$$= 10^{-2} \frac{\text{cm}^2}{\text{sec}} \text{ for water}$$

- Reynolds number ( $R$ ) is a dimension-less number that indicates the ratio of inertial to viscous forces



## Aside: Viscosity



Water



Oil



Maple syrup

- Viscosity ( $\eta$ ) deals how a liquid “deforms” due to stress (i.e., forces) applied to it
- Tied to how individual fluid molecules interact and friction arising from such

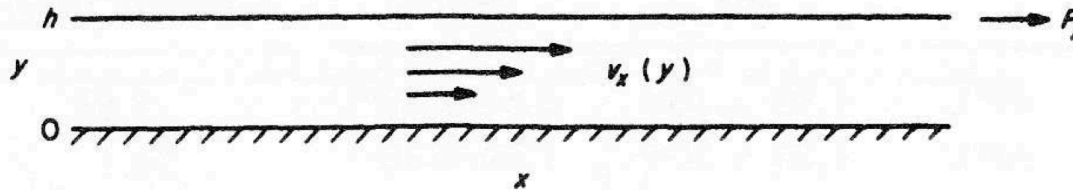
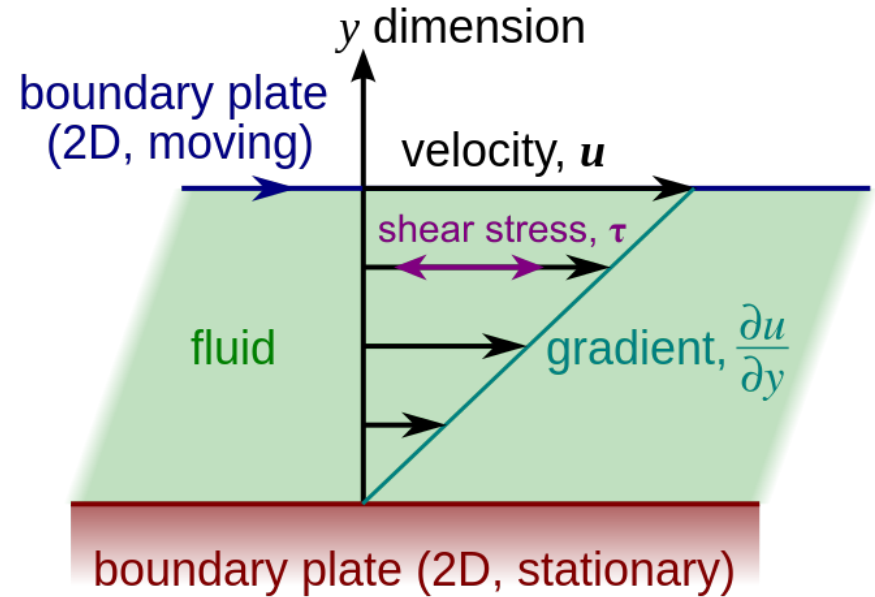
Table 5.1: Density, viscosity and viscous critical force for some common fluids at 25°C.

Fluid	$\rho_m$ (kg m <sup>-3</sup> )	$\eta$ (Pa · s)	$f_{\text{crit}}$ (N)
Air	1	$2 \cdot 10^{-5}$	$4 \cdot 10^{-10}$
Water	1000	0.0009	$8 \cdot 10^{-10}$
Olive oil	900	0.080	$4 \cdot 10^{-6}$
Glycerine	1300	1	0.0008
Corn syrup	1000	5	0.03

## Aside: Viscosity

“Laminar shear of fluid between two plates. Friction between the fluid and the moving boundaries causes the fluid to shear. The force required for this action is a measure of the fluid's viscosity.”

Let's firm this up a bit more....

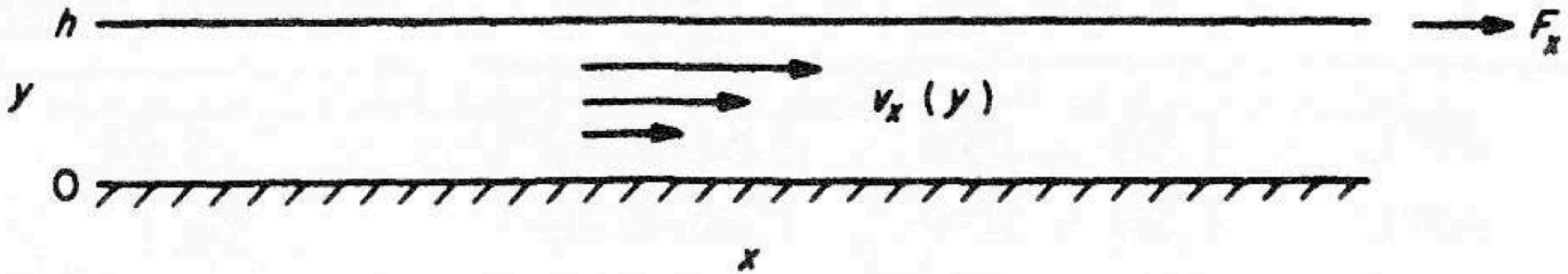


**Fig. 4.2.** A viscous fluid contained between flat plates at  $y = 0$  and  $y = h$ . The area of each plate is  $A$ . The bottom plate is fixed. The top one is propelled to the right by a force in the  $x$  direction,  $F_x$ . Arrows show the velocity of the fluid relative to the bottom plate at different distances from this plate,  $y$ .



## Aside: Viscosity

Let's firm this up a bit more....



$$v_x(y) = \frac{v_x(h)}{h} y$$

Linear relationship in fluid velocity  
(i.e., fluid flows in “layers” → laminar flow)

Shear “force”

$$\partial v_x / \partial y$$

$$F_x = \eta A \left. \frac{\partial v_x}{\partial y} \right|_{y=h}$$

→ Viscosity ( $\eta$ ) is the constant of proportionality  
between applied force and shear