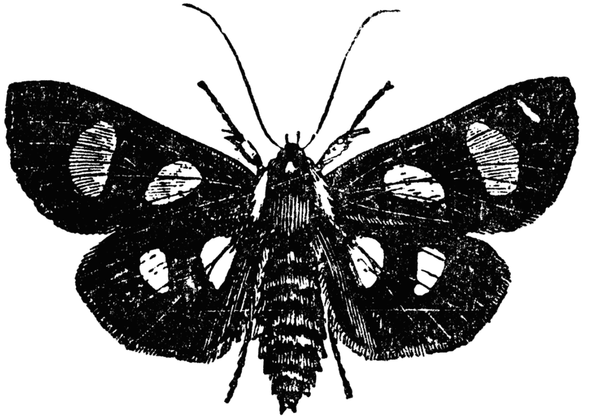
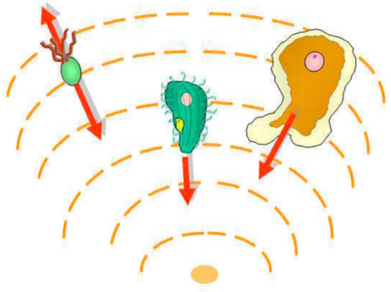
Olfactory Search in Turbulent Flows

Andrew Burns, Jonathan Kroc, Antony Pearson, and Alexia Tatem



Outline

Purpose of Paper

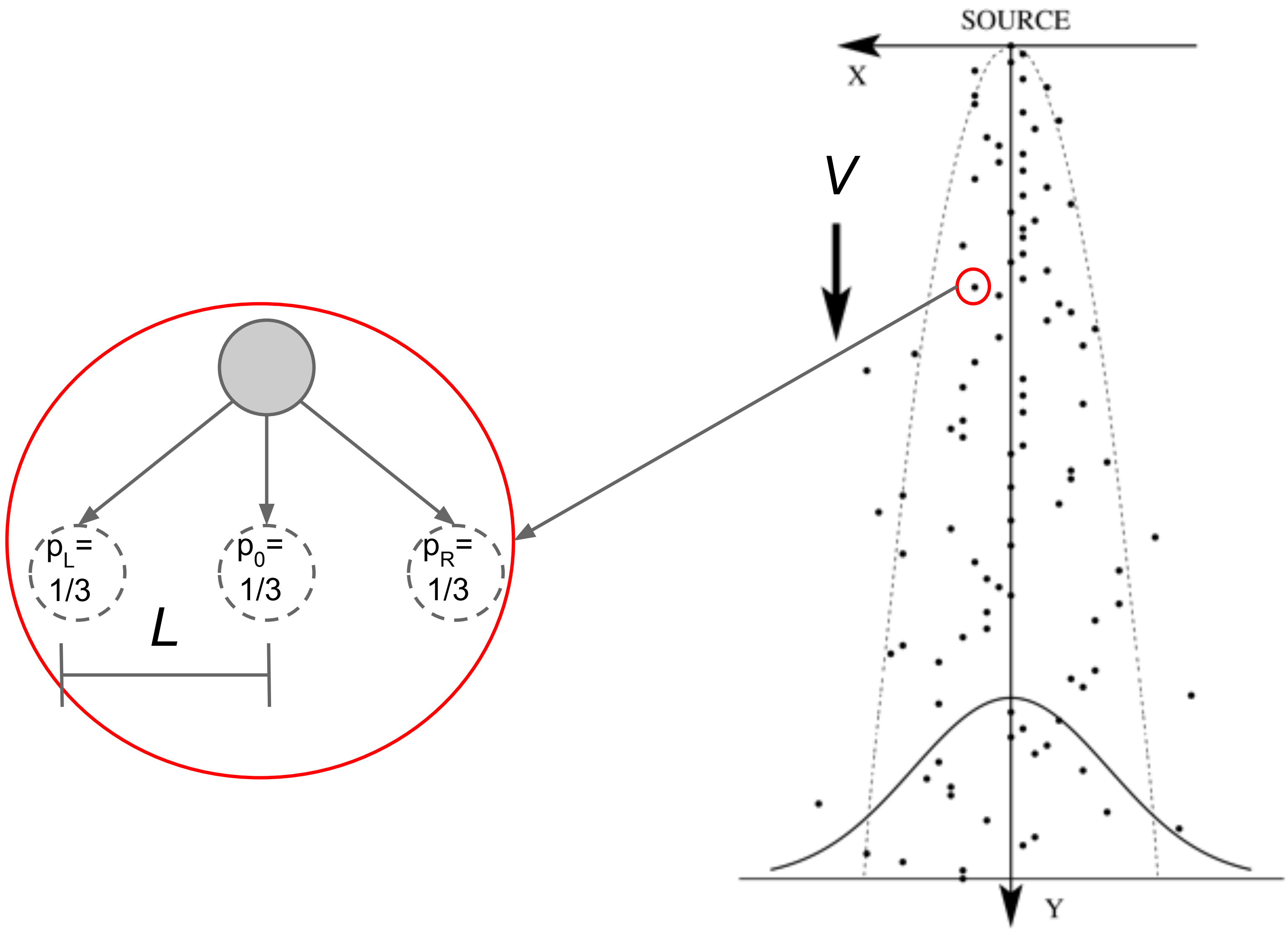


- Typical approach of locating the source of a substance is chemotaxis- inappropriate for the problem of olfactory search in high Reynold's number flow regimes.
- Turbulent flow regimes lack the uniformity or "smoothness" of flow that would make them amenable to chemotaxis.
- Balkovsky and Shraiman outline "[a] more complex strategy involving, in addition to the sense of smell, the ability to determine the wind direction."

Balkovsky and Shraiman's turbulent flow model

- Flow characterized by a global mean velocity, *V*.
- Odor molecules move with a local velocity equal to the sum of the global mean velocity, *V*, and local fluctuations.
- At scales larger than the lattice constant, *L*, the particle motion is Brownian.

Balkovsky and Shraiman's turbulent flow model



Fully developed distribution of plume

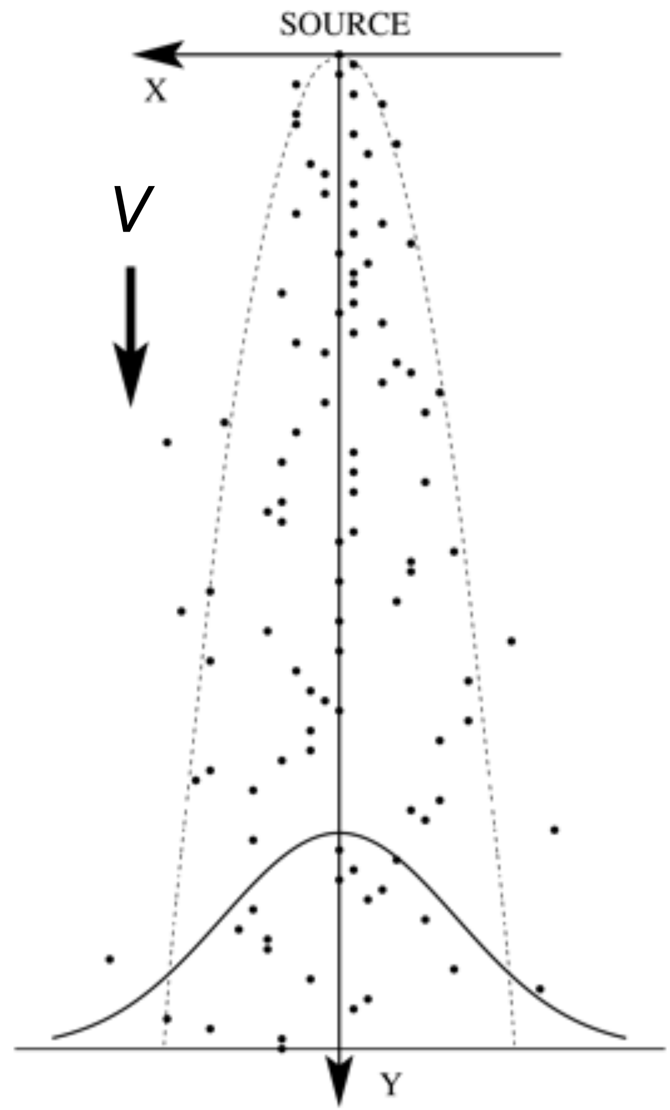
For y≫1:

$$p(\mathbf{r}) = \frac{1}{\sqrt{4\pi D y}} \exp\left[-\frac{x^2}{4Dy}\right]$$

where

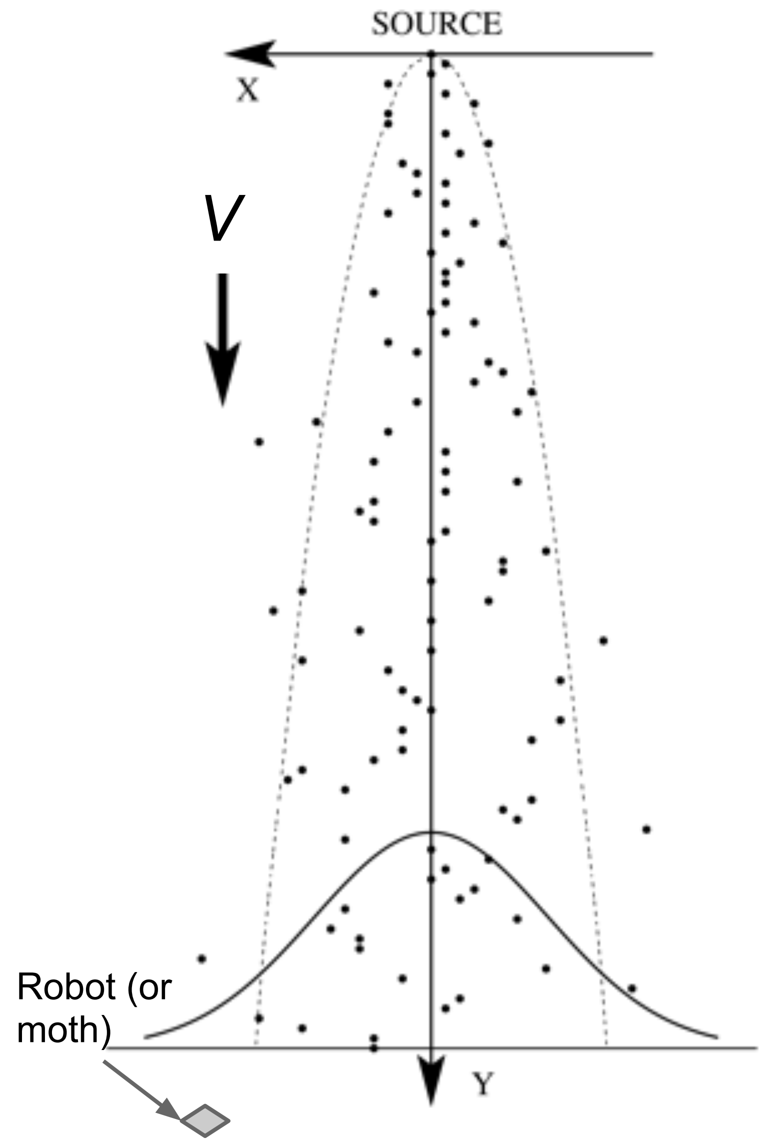
$$D = \frac{p_R + p_L}{2}$$

is the diffusivity coefficient.



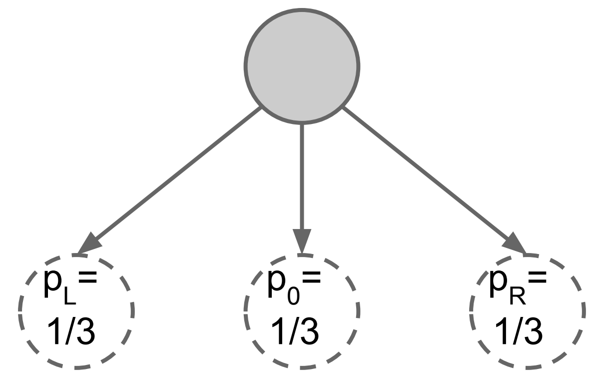
Problem setup

- A robot (or moth) located a distance y_0 can detect:
  - the event of an odor patch arriving at its (the robot's) current location
  - the direction from which the odor patch arrived
- Each time step, the robot is able to move at most one lattice step along the y-axis and/or one step along the x-axis

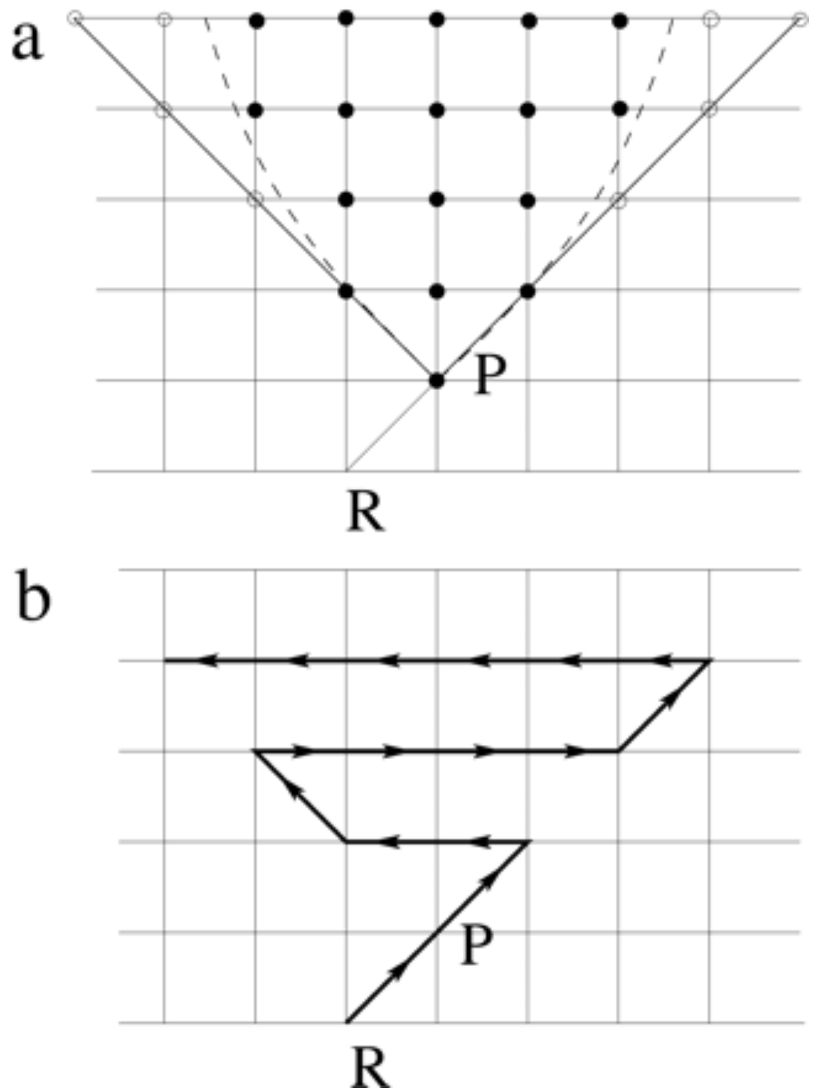


Problem setup

- Each time step, the source, located at (0,0), releases a new "odor patch" which is advected by the "wind."
- The robot search doesn't start until it encounters patch



The Causality Cone

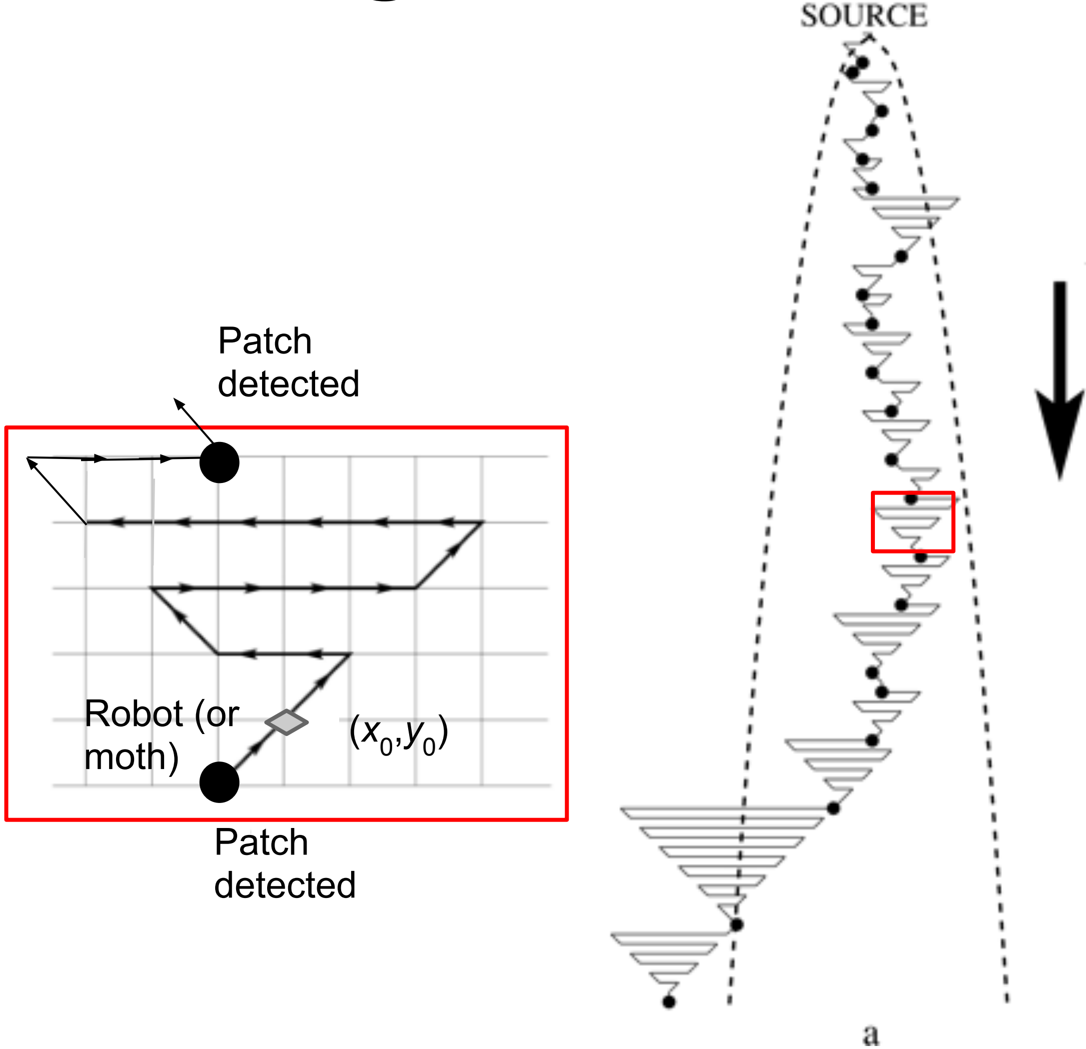


(x_0, y_0) = the source of the odor patch one time step ago

$y - y_0 = \pm (x - x_0)$,
$y < y_0$

Conical Search Algorithm

- The robot actively explores the space in the interior of the cone described by $y-y_0=\pm(x-x_0)$
- When a patch is detected, robot moves to the position from which the patch originated, then restarts search.
- *Typical search time:* $t_s \propto y_0^{5/4}$

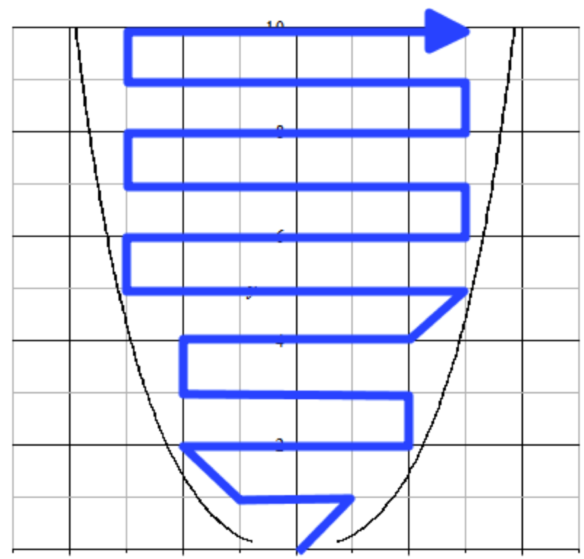


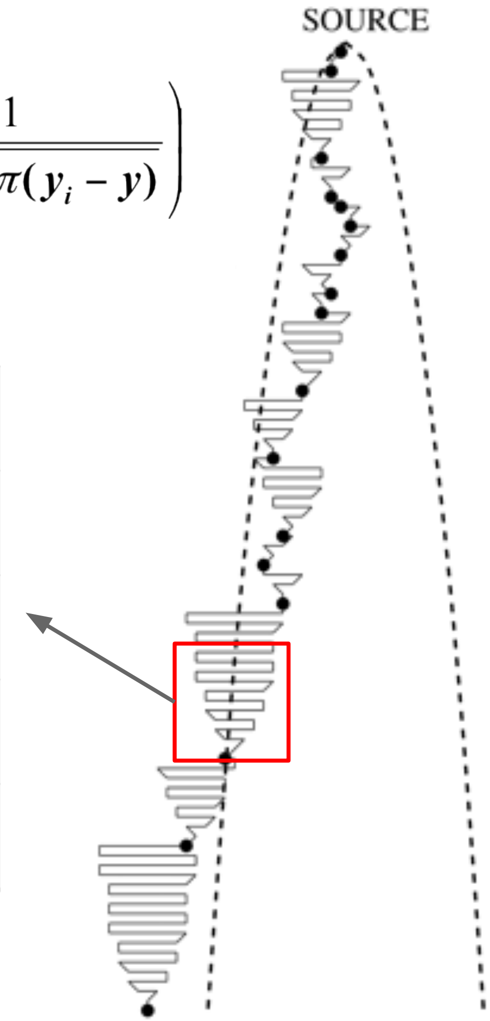
Parabolic Search Algorithm

- Alters the conical search algorithm to omit points of low probability. This high-likelihood region is parabolic.

- Typical search time: $t_s \propto y_0^{7/6}$

$$(x - x_i)^2 \le \frac{4}{3}(y_i - y)\ln\left(\frac{1}{0.05\sqrt{\frac{4}{3}\pi(y_i - y)}}\right)$$



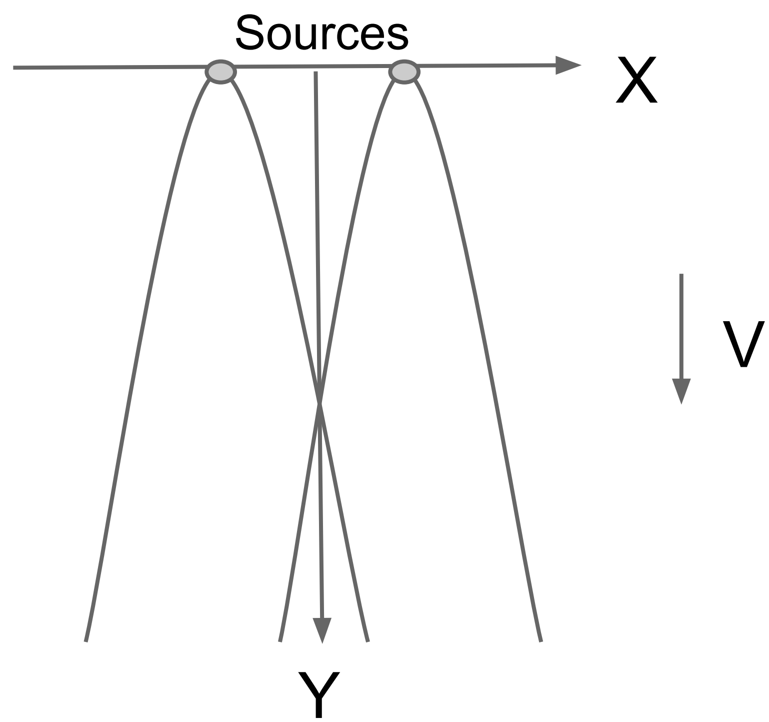


Assessing algorithm effectiveness

- Algorithms were judged by the search time.
- Balkovsky and Shraiman evaluated algorithms and plotted the probability that the source is found during a *t*, *t*+1 interval as a function of time, ρ(t).
- Algorithms with means closer to zero were deemed more effective.
- With respect to this definition, the parabolic search algorithm is the most effective

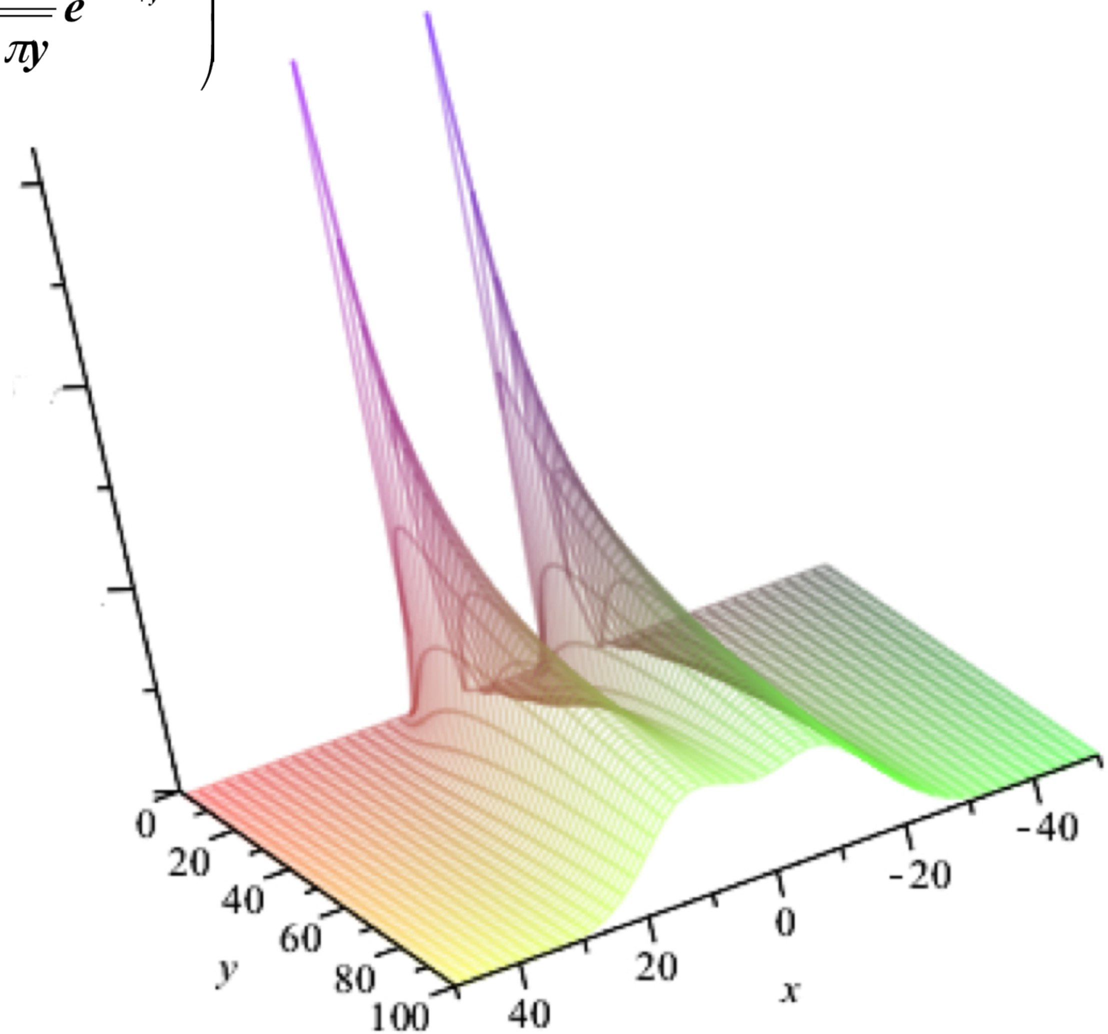
Motivation and project description

- Study impact of adding another source using similar active search
- Many real-world scenarios have multiple sources
- Useful for the design of robots used to find small gas leaks or explosives.



$\boldsymbol{P}(\text{particle at x} \mid \text{fixed y} >> 1,\ p_R = p_L = 1/3, \text{sources at (-10,0) and (10,0)})$

$$= 1 - \left(1 - \frac{1}{\sqrt{\frac{4}{3}\pi y}} e^{-\frac{3(x-10)^2}{4y}}\right)\left(1 - \frac{1}{\sqrt{\frac{4}{3}\pi y}} e^{-\frac{3(x+10)^2}{4y}}\right)$$

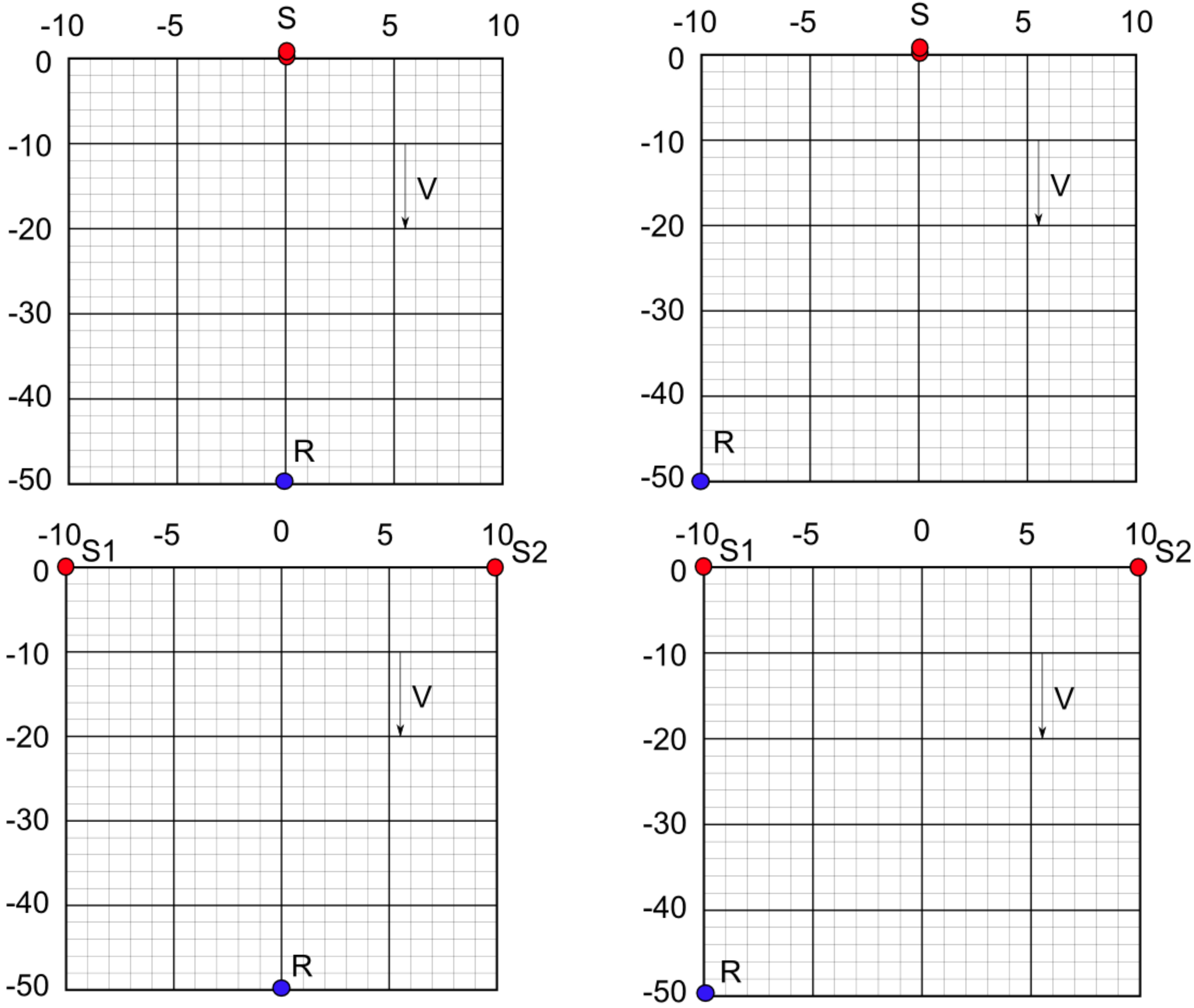


Methods

- The parabolic search algorithm described in the paper was reproduced using Monte Carlo simulations in Python.
- Simulations were performed for five different initial conditions
- Histograms created for each set of simulations
- Percentage of misses by the seeker calculated for each run

Simulations

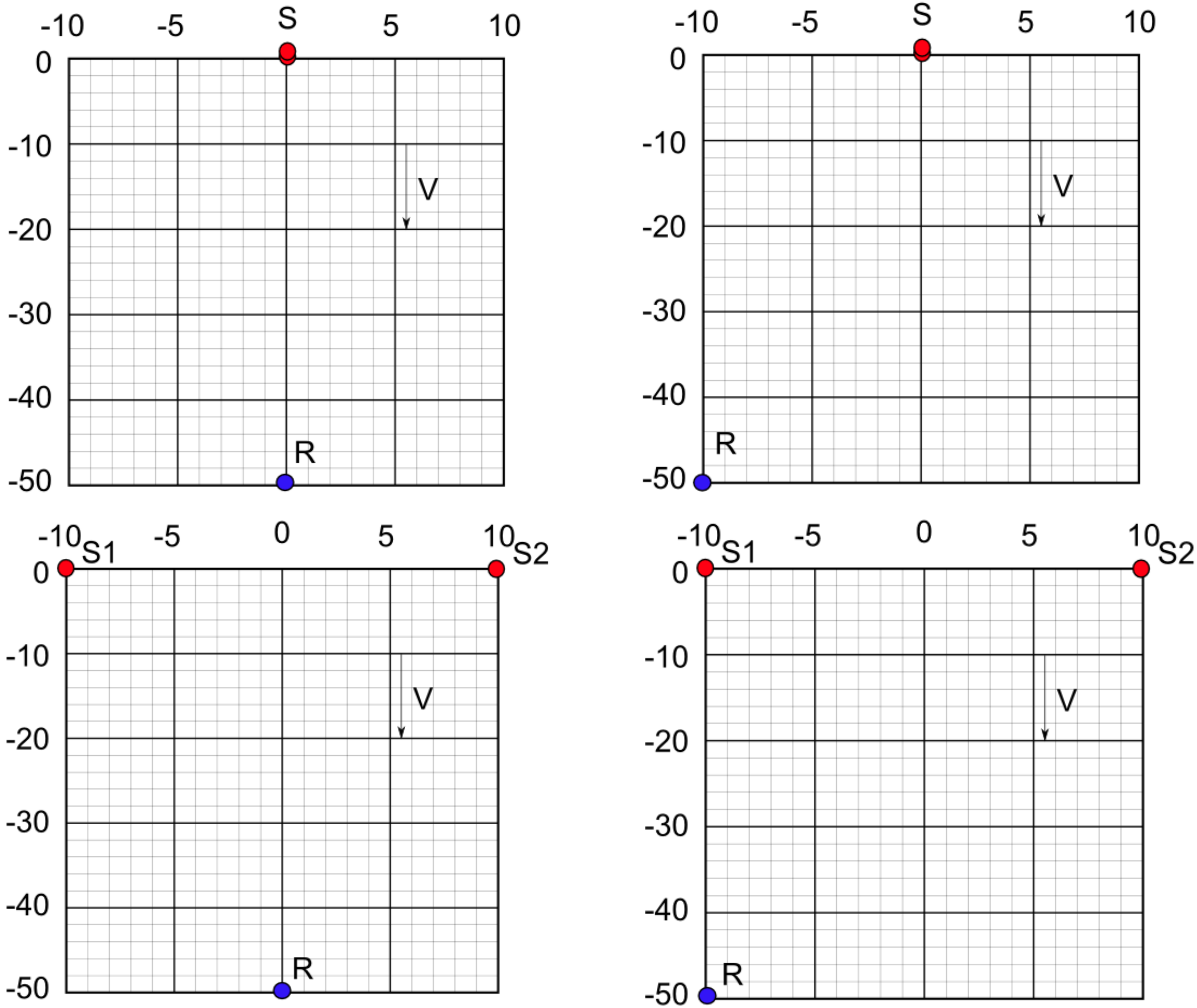
- In the first two simulations:
  - both sources located at (0,0)
  - first run with initial robot position at (0,50)
  - second run with initial robot position at (10,50)
- In the final two simulations:
  - two sources placed at (-10,0) and (10,0)
  - first run, the initial robot position at (0,50)
  - second run with initial robot position at (10,50)
- Simple case with:
  - one source at origin
  - robot initially at (0,50)



-10
-5
S
5
10
0
-10
-20
-30
-40
-50
V
R
-10
-5
0
5
10
S1
S2

Results

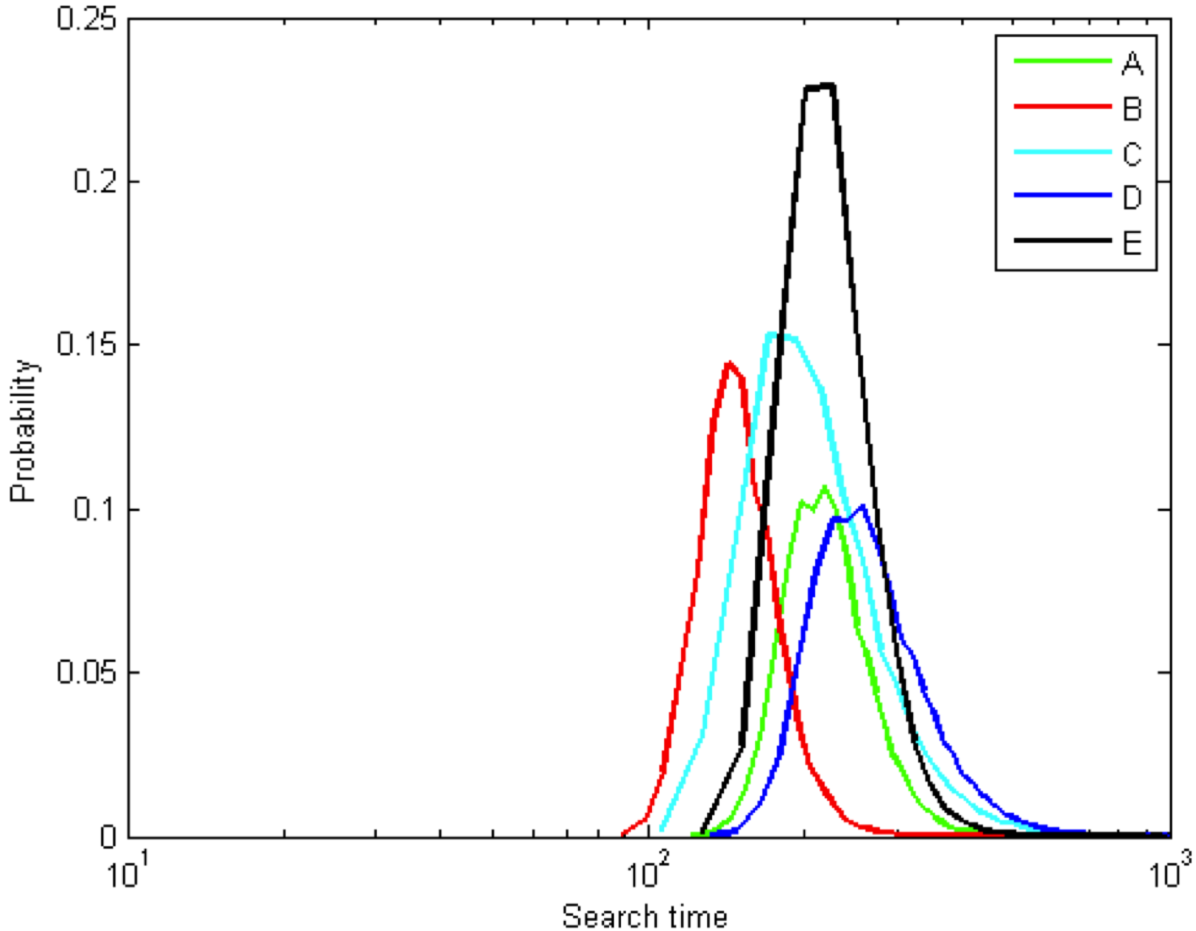
Simulation	**Miss rate**
(A) One source at (0,0); robot initially at (0,50)	5282 out of 100,000 5.282% misses
(B) Both sources at (0,0); robot initially at (0,50)	2588 out of 100,000 2.588% misses
(C) Both sources at (0,0); robot initially at (10,50)	10392 out of 100,000 10.392% misses
(D) Sources at (-10,0) and (10,0); robot initially at (0,50)	12160 out of 100,000 12.16% misses
(E) Sources at (-10,0) and (10,0); robot initially at (10,50)	5318 out of 100,000 5.318% misses



-10
-5
S
5
10
0
-10
-20
-30
-40
-50
V
R
S1
S2
0

Results

(A) One source at (0,0); robot initially at (0,50)
(B) Both sources at (0,0); robot initially at (0,50)
(C) Both sources at (0,0); robot initially at (10,50)
(D) Sources at (-10,0) and (10,0); robot initially at (0,50)
(E) Sources at (-10,0) and (10,0); robot initially at (10,50)



Potential applications and future work

- A strategy for locating multiple sources in turbulent flows would be useful for the design of robots to find gas leaks or explosives.
- Further research
  - Effect of changing the width between the sources -- There may be a particular source separation distance that dramatically increases misses.
  - Adding multiple seekers that can interact with each other
  - Altering the search algorithm with respect to the time between encounters
  - Analytically describe the search times with multiple sources

References

Balkovsky E. and Shraiman, B.I. "Olfactory search at high Reynolds number", Proc Natl Acad Sci USA.99,12589-93 (2002).