Chaotic Mixing in a Shear Flow Environment

Daniel Kelleher
Undergraduate Student
Department of Physics
UMass Dartmouth
Dartmouth MA 02747
Email: dkelleher@umassd.edu

Dr. Amit Tandon
Associate Professor
Department of Physics
UMass Dartmouth
Dartmouth MA 02747
Email: atandon@umassd.edu

Abstract

The purpose of this research is to devise a simpler method to present chaotic mixing in a shear flow environment by modeling diffusion as a random walk and advection as a shear velocity field. A random walk leads to a Gaussian distribution in all dimensions. By creating a random walk for a large number of particles simultaneously, the resulting graphs mimic the process of mixing in a fluid at rest. We examine mixing in the presence of shear flows by a Lagrangian particle tracking approach. The results will be verified by examining the probability density function and comparing it with the tracer statistics of the advection-diffusion equation.

Introduction

The advection-diffusion equation (1) is an advanced vector calculus equation which is particularly challenging for undergraduate students to approach. The goal of this research was to develop a simple but computationally effective method of modeling dynamic systems using basic statistics. The results were then compared with results of similar situations solved with the advection-diffusion equation. A simple random walk was utilized to model diffusion in a fluid at rest. Once it was established that a simple random walk could properly model diffusion, a shear velocity field was applied to the system result in a model for shear dispersion.

$$\frac{\partial \phi}{\partial t} + v \cdot \nabla \phi = k \nabla^2 \phi \tag{1}$$

In addition to being challenging to solve for undergraduate students, the advection-diffusion equation is impossible to solve analytically except in extremely simple cases, such as shear dispersion. An analytical solution for the advection-diffusion equation can be found in both Thiffeault (2008) [1] and Young *et al* (1982) [2]. The paper by Thiffeault discuses and presents explicit solutions, with visuals for the simple situations using the advection-diffusion equation (1) including shear dispersion. In contrast, the paper by Young *et al* explored specific situations of shear dispersion using the advection-diffusion equation. The main purpose of Young *et al* is to explore "vertical shear of the wave field" [2] in more complex cases, such as oscillatory shear flows. In addition, Young addresses the process of shear dispersion in infinite boundary conditions i.e. large bodies, such as the ocean.

Simple Random Walk

As stated above, a simple random walk is characterized by a Gaussian probability density function (PDF). Random walk is a statistical algorithm, here taken with a constant step size where the point at one vertex may move to any directly adjacent vertex. For example, in four directions for a 2-dimensional graph, as shown in figure 1) or six directions for a 3-dimensional graph, as shown in figure 2. It is not immediately apparent from either of these graphs that a simple random walk trends to a gaussian curve but it is clear that the particles are more likely to stay in the near vicinity of the origin. As additional proof that the random walk trends to a Gaussian curve figure 3shows the PDF in

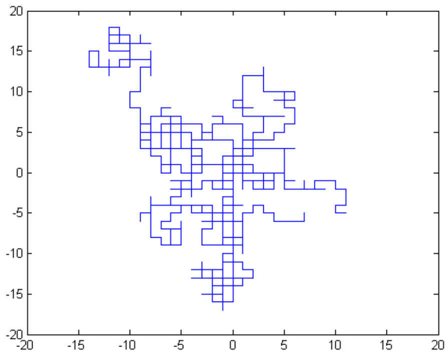

Figure 1: n=5 t=150

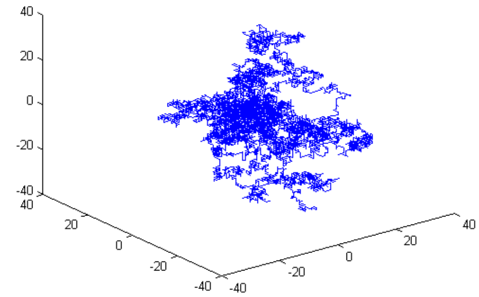

Figure 2: n=10 t=1000

X, Y and Z. In addition it is easy to see from the 3-dimentional graph that the particles expand slowly in all direction and mimic mixing in a static fluid. Figure 3 was generated with the code listed in appendix B.

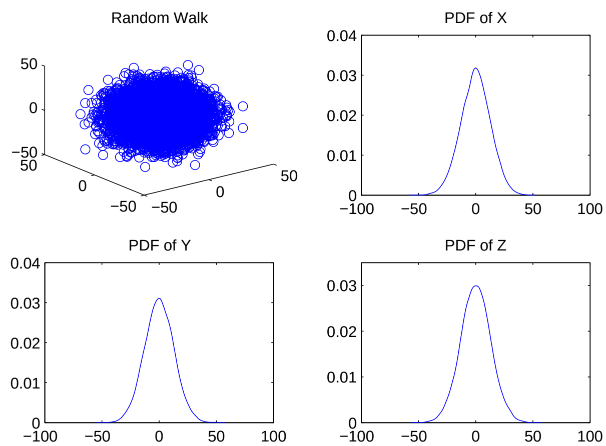

Figure 3: n=10,000 t = 500

Shear Dispersion

This report is specifically focused on the described model as a form of modeling shear dispersion in an incompressible fluid. A shear velocity field in an incompressible fluid is characterized by equation (2) and equation (3). Figure 4 is an example of a shear velocity field. Shear velocity is important in fluid mechanics because fluids near a surface act like particles at the origin of a shear velocity flow. As the particle moves farther away from the surface, friction with the surface and surrounding particles decreases, allowing the particle to speed up. As a result, the Y dependance of the X velocity is clear (3). Figure 4 is not drawn to scale with reference to the velocity field of this experiment.

$$\nabla \cdot v = 0 \tag{2}$$

$$\frac{dU}{dy} = C \tag{3}$$

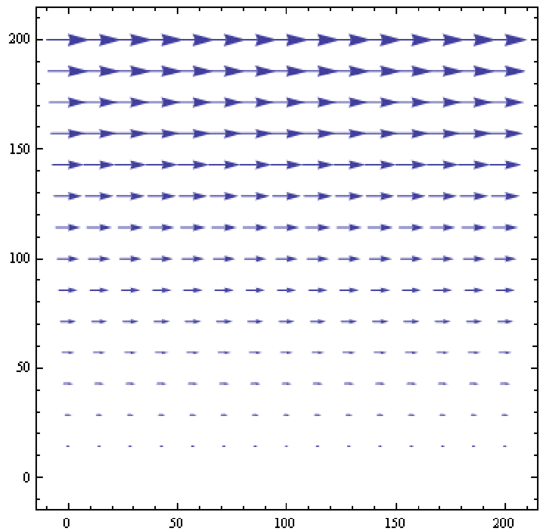

Figure 4: shear velocity flow

Results and Procedure

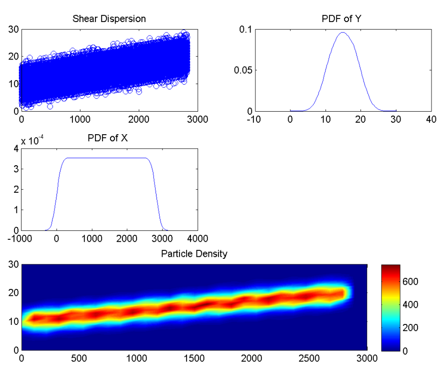

Figure 5: n=1,000,000 t=12

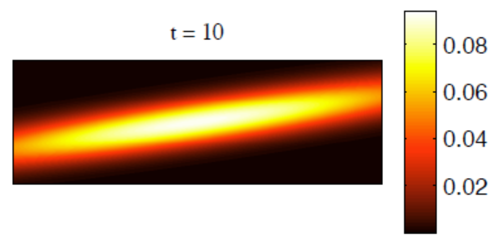

Figure 6: Comparative results from Thiffeault (2008) [1]

Combining a shear velocity field and a simple random walk resulted in a model for shear dispersion in an incompressible fluid. The results were then compared to generally accepted results from a similar situation which was solved using the advection-diffusion equation such as Thiffeault (2008) [1]. Figure 5 is an example of the shear dispersion model of 1 million particles evolved over 12 random walk time steps. In this experiment, the random walk time scale is longer than the shear time scale by a factor of 3. This implies in equation (3) $C = 3$. It is clear from visual inspection that the models are virtually identical. In addition, the particle density functions were exactly what was expected. As a result, it is clear that the model is effective for shear flows. The code used to generate figure 5 can be seen in appendix A.

Goals and Future Research

A goal for the next research project is to explore Levy Flights, or other non-gaussian governed random walks. This will help explore the most realistic random walk probability distribution and consider other aspects of the advection-diffusion equation, such as viscosity or stochasticity, and their effects on this model.

$$\sigma_{t_0}^{T} = \frac{1}{|T|} ln \sqrt{\lambda_{max}(\Delta)} \tag{4}$$

Another goal for the next research project is to incorporate finite time Lyapunov exponents (equation (4)) into the analysis of this model. Finite time Lyapunov exponents are a method of representing the Lagrangian coherent structures in a fluid by mapping the stretching of fluid particles into a contour map. This acts to solve a partial differential equation by converting it to an ordinary differential with discreet time steps.
Furthermore, the next research project should cover more advanced flow profile, which we hope includes data from an actual ocean current off the cost of the eastern United States, specifically an ocean eddy.

Acknoledgements

This research was supported by the National Science Foundation CSUMS program NSF-DMS-0802974 to S. Gottlieb, S. Leon, G. Davis and J. Jung and the University of Massachusetts Dartmouth Department of Mathematics.

References

[1] J.-L. Thiffeault. Scalar decay in chaotic mixing, in *Transport and Mixing in Geophysical Flows. Lecture Notes in Physics*, 744:3–35, 2008.

[2] Rhines P. Garrett C. Young, W.R. Shear-flow dispersion, internal waves and horizontal mixing in the ocean. *Journal of Physical Oceanography*, 12:515–527, 1982.

Appendix A: Matlab Shear Dispersion Code

```
%Function name sheardispersion with variables n, b, time, and dist
function [r] = sheardispersion(n,b,time,dist)

%Initialize the radius variable
r = zeros(n,2,time);

%Input initial conditions into the radius variable
r(:,2,1) = [10:((dist-10)/(n-1)):dist];

%For loop which evaluates the system from t equals 1 to time
for t = 1:time

    %A random number which is then converted into a step in one of four
    %directions for every particle. The step size is 1 unit of distance
    d = rand (n,1);
    up = d<0.25;
    down = -1*((d>0.25)&(d<0.5));
    left = -1*((d>0.5)&(d<0.75));
    right = d>0.75;

    %Initialize the random step variable and then input the random steps
    %for each particle.
    mod = zeros(n,2);
    mod(:,1) = left + right;
    mod(:,2) = up + down;

    %Initialize an calculate the shear velocity field for each particle
    shear = zeros(n,2);
    shear(:,1) = b*(r(:,2)-10);

    %Resulting position of the random variable and the shear field for
    %every particle
    r(:,:,t+1) = r(:,:,t) + mod + shear;

    %Analyze the resulting probability density curve in x and y
    [x,f] = ksdensity (r(:,1,t+1));
    [y,g] = ksdensity (r(:,2,t+1));

    %Analyze the density of particle in two dimensions to get a more
    %precise understanding of the distrobution of particles
    [xi,yi,z] = density(r(:,1,t+1),r(:,2,t+1),29);

    %Plot every particle on x and y
    subplot(3,2,1);
    plot(r(:,1,t+1),r(:,2,t+1),'o')
    axis([0 3000 0 30])
    title('Shear Dispersion')

    %Probability density function in X
    subplot(3,2,3);
    plot(f,x)
    title('PDF of X ')

    %Probability density function in Y
    subplot(3,2,2);
    plot(g,y)
    title('PDF of Y ')
```

```
    subplot(3,2,[5 6]);
    pcolor(xi,yi,z)
    shading interp;
    title('Particle Density');
    colorbar

    %Pause for animation
    pause(1)

    %Save plots
    filename = ['figure_' num2str(t) '.png'];
    saveas(gcf,filename)
    close
end
```

Appendix B: Matlab Random Walk Code

```
%Function name randomwalk3d with variables n and time
function [r] = randomwalk3d(n,time)

%Initialize the radius variable
r = zeros(n,3);

%For loop which evaluates the system from t equals 1 to time
for t = 1:time

    %A random number which is then converted into a step in one of six
    %directions for every particle. The step size is 1 unit of distance
    d = rand (n,1);
    up = d<(1/6);
    down = -1*((d>(1/6))&(d<(2/6)));
    left = -1*((d>(2/6))&(d<(3/6)));
    right = (d>(3/6))&(d<(4/6));
    forward = (d>(4/6))&(d<(5/6));
    back = -1*(d>(5/6));

    %Initialize the random step variable and then input the random steps
    %for each particle.
    mod(:,1) = forward + back;
    mod(:,2) = left + right;
    mod(:,3) = up + down;

    %Resulting position of the random variable
    r = r + mod;

    %Analyze the resulting probability density curve in X, Y and Z
    [x,f] = ksdensity (r(:,1));
    [y,g] = ksdensity (r(:,2));
    [z,h] = ksdensity (r(:,3));

    %Plot every particle on X and Y
    subplot(2,2,1)
    plot3(r(:,1),r(:,2),r(:,3),'o')
    axis([-50 50 -50 50 -50 50])

    %Probability density function in X
    subplot(2,2,2)
    plot(f,x)

    %Probability density function in Y
    subplot(2,2,3)
    plot(g,y)

    %Probability density function in Z
    subplot(2,2,4)
    plot(h,z)

    %Pause for animation
    pause(0.1)

end
```