MOLECULAR DIFFUSIVITY AND DECAY RATE IN DETERMINISTIC AND NON-DETERMINISTIC FLOWS

LUKE APPLEBY

Abstract. An investigation of the relationship between molecular diffusivity and decay rate of a passive scalars in both deterministic and non deterministic flows. In both cases an exponential relationship was found with greater decay rates for the non-deterministic case in comparison with the deterministic case. This is an important result that has application in processes in which rapid mixing is required.

1. Introduction

We are looking at the action of the alternating sine map on passive scalars. The alternating sine map describes a type of flow of fluid. It is defined iteratively which enables us to easily model the action of such a flow. We are interested in how the decay rate γ of concentration varies with κ molecular diffusivity. We will look at both deterministic and non-deterministic variations of the alternating sine map. The non-deterministic flow includes a degree of randomness into the action of the map which is to account for the chaotic nature of fluid motion.

This is a special case of a more general question which is how well do fluids mix given a certain flow and degree of molecular diffusivity. This has particular application in many sectors especially combustion technologies in aerospace as a jet engine requires kerosene and air to mix properly in order for ignition and complete combustion to occur. Additionally many commercial industrial chemical process require efficient and proper mixing of fluids in order for production to take place at a rate conducive with profitability.

2. Background Information

The alternating sine map is a map that describes a particular flow. Given a position of a fluid particle (x_n, y_n) its position after a small unit of time t is given by:

$$\begin{aligned} x_{n+1} &= x_n + \mu \sin(y_n + \alpha_n) \\ y_{n+1} &= y_n + \mu \sin(x_{n+1} + \beta_n) \end{aligned} \tag{1}$$

Where $\mu = At$ for some constant A. To approximate the position of a fluid particle after a larger time period we apply several iterations of this map. In general more iterations with a smaller unit of time will result in a better approximation. In the deterministic case we will let $\alpha_n = \beta_n = 0$ for all n and in the non-deterministic case we let $\alpha_n, \beta_n \in [0, 2\pi)$ with uniform distribution.

Secondly a passive scalar is a scalar field meant to represent the concentration of one fluid within another. It is called passive since we assume that it does not affect the existing flow of the fluid. This is a fairly safe assumption if we are adding a low volume of low viscosity fluid to a flow however, this assumption may be problematic if we are dealing with a large volume of more viscus fluid. In the later case it would be unreasonable to expect the flow to be unchanged.

$C_0(x, y)$ is the initial concentration at the point (x, y). We impose the condition

$$\int_0^{2\pi} \int_0^{2\pi} C_0(x, y) dx dy = 0 \tag{2}$$

Which can be met by adding a constant to $C_0(x, y)$. If we want to know the concentration at a point (x, y) after n iterations of the sine map $C_n(x, y)$. Notice it will be the same as the concentration of the fluid particle which ends up at the point (x, y) after n iterations. Thus we have $C_{n+1}(x_{n+1}, y_{n+1}) = C_n(x_n, y_n)$ and from 1 see that

$$x_n = x_{n+1} - \mu \sin(y_n + \alpha_n), y_n = y_{n+1} - \mu \sin(x_{n+1} + \beta_n) \tag{3}$$

3 does not take into account molecular diffusion. This is better modeled by

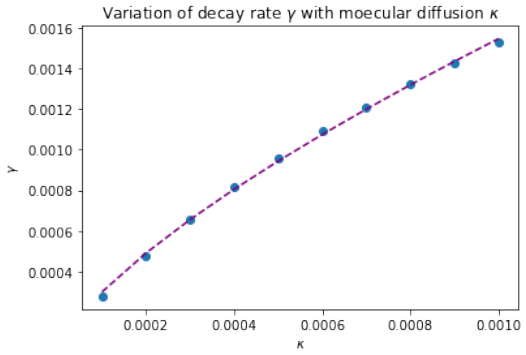
$$C_{n+1}(x, y) = \frac{1}{4\pi\kappa} \int \int C_n(x'', y'') e^{-[(x'-x'')^2+(y'-y'')^2]/4\kappa} dx'' dy''. \tag{4}$$

where κ is a value that accounts for molecular diffusion. A greater κ value corresponds to a greater degree of molecular diffusivity which is the mean squared difference travelled by particles over a time t due to diffusion. In fact we expect the concentration to decay exponentially

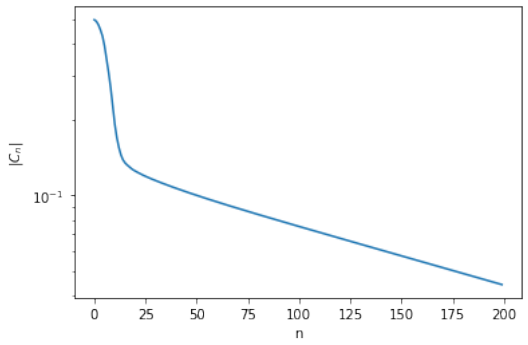
$$||C_n|| \asymp e^{-n\gamma}. \tag{5}$$

3. Deterministic action of the alternating sine map on passive scalars

To model the deterministic flow the angles α_n, β_n in 1 were set to be 0 for all n. Using the code in A.1 adapted from the given notebook [2] we looked at the decay rate γ for different κ in the interval $[10^{-4}, 10^{-3}]$. To estimate the decay rate the difference in concentration after 100 iterations and after 200 iterations was divided by 100 to get the mean decay per iteration. A simulation was done for 10 evenly spaced κ values in the interval. A best fit line was fitted to the data with an assumed relationship $\gamma = a\kappa^b$. The function curve_fit from scripy was used in A.2 to approximate the values of a and b. This function chooses the values of a and b that minimises the sum of squares of the residuals. The observed data along with the best fit curve are seen in figure 1a. As we can see the decay rate increases as the molecular diffusivity increases. The values a, b in $\gamma = a\kappa^b$ were calculated to be 0.211 and 0.712 respectively. In figure 1b we can see that the concentration $|C_n|$ decreases rapidly over the first 20 iterations and then the decay rate slows significantly.



(a) Decay rate γ plotted against strength of molecular mixing κ.

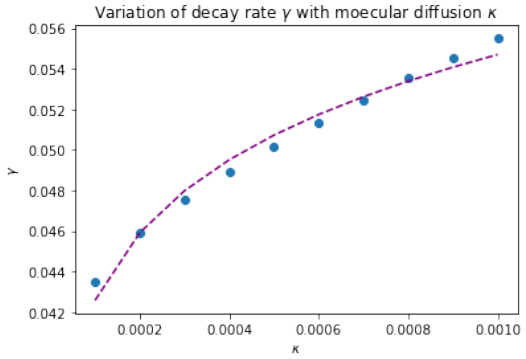


(b) Concentration $|C_n|$ against number of iterations of the deterministic sine map with $\kappa = 0.0005$.

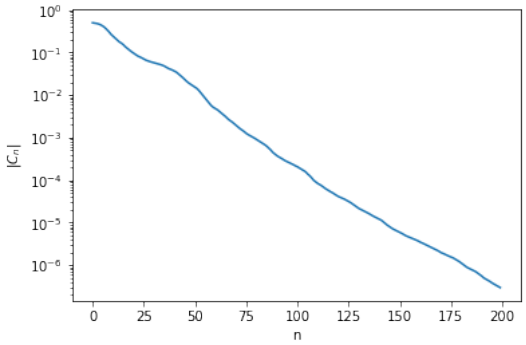
Figure 1

4. Non-deterministic action of the alternating sine map on passive scalars

To model the non-deterministic case we let the angles α_n and β_n be uniformly distributed in the interval $[0, 2\pi]$ A.3. The action of this map was simulated for 200 iterations for each value of κ in the interval $[10^{-4}, 10^{-3}]$. In figure 2a we can see that the decay rate γ increases as κ the molecular diffusivity increases. A best fit curve $\gamma = a\kappa^b$ is also shown with a, b calculated to be 0.116 and 0.109 respectively A.4. In figure 2b we can see the concentration decreases by a power of 10 approximately every 25 iterations.



(a) Decay rate γ plotted against strength of molecular mixing κ.



(b) Concentration $|C_n|$ against number of iterations of the non deterministic sine map with $\kappa = 0.0005$.

Figure 2

5. Discussion and conclusions

In the simulations it was observed that grater values of molecular diffusivity κ resulted in greater decay rates γ. This was expected as it makes sense that greater molecular diffusivity which is the mean squared difference travelled by particles due to diffusion would result in a greater degree of mixing and thus a greater decay rate especially when the boundaries between areas of high concentration and low concentration are large. What was also observed was that the decay rate in the non-deterministic case was significantly greater than that of the deterministic case across all κ values used in this investigation. This may be because the non-deterministic motion disturbs the long filaments thus creating larger boundaries across which molecular diffusion to occurs. Further it was observed that the best fit curve fitted for the deterministic case had lower sum of squares of residuals compared with the non-deterministic case. With the sum of squares of residuals being 0.0055 and 0.0260 for the deterministic and non-deterministic cases respectively A.2, A.4. This could be due to the higher degree of randomness in the non-deterministic case resulting in each γ value having a larger degree of variance from simulation to simulation. This in turn could make the relationship between κ and γ more difficult to model. In future investigations we could do several simulations for each κ value and calculate the mean of these and then fit a curve to these means which will reduce the random variance. This was not done as the computation time would be prohibitive. It is also possible that the relationship $\gamma = a\kappa^b$ is not a good approximation for the non-deterministic case.

References

[1] Maths in Action lecture notes.
[2] Maths in Action Jupyter Notebook

Appendix A. Codes

A.1. **Deterministic Simulation.** Code for simulating the deterministic case of the alternating sine map on scalar flow. 10 simulations are done. One for each value of κ.

```
import numpy as np
import numpy.fft as fft
import matplotlib.pyplot as plt
from IPython.display import clear_output

# Deterministic flow with several kappa values

pi = np.pi

m = 256 # number of grid points in x and y directions,
        # can be decreased for faster runs, or increased for higher accuracy
dx = 2*pi/m # grid spacing

x = np.linspace(0,2*pi-dx,m)
y = np.linspace(0,2*pi-dx,m)
xx, yy = np.meshgrid(x,y) # grid

k = np.linspace(-m/2,m/2-1,m) # grid in Fourier space (for smoothing)
k = fft.ifftshift(k)
l = k
kk, ll = np.meshgrid(k,l)

kappas = np.linspace(0.0001, 0.001, num = 10)
# an array of values in the interval [0.0001, 001]

N = 200 # number of iterations of the map
mu = 1
phi = np.zeros((N,2))  # angles  determinitic case

rates = []

index = -1
# Loop over iterations
for kappa in kappas:
    decay = np.exp(-kappa*(kk**2+ll**2))
    C = np.sin(xx)*np.sin(yy) # initial concentration
    Cnorm = np.zeros(N)
    for n in range(N):
        Cnorm[n] = np.sqrt(np.sum(C[:]**2)/m**2) # norm of the concentration
        for j in range(m):
            yshift = - mu * np.sin(x[j]+phi[n,1])
            yshift = np.mod(yshift,2*pi)
            indshift = round(yshift/dx)
            C[:,j] = np.roll(C[:,j],indshift)
```

```
        for i in range(m):
            xshift = - mu * np.sin(y[i]+phi[n,0])
            xshift = np.mod(xshift,2*pi)
            indshift = round(xshift/dx)
            C[i,:] = np.roll(C[i,:],indshift)

        # smoothing
        C = np.real(fft.ifft2(decay*fft.fft2(C)))

    # estimation of the decay rate from decay of Cnorm between iteration N/2 and N
    rate = -(np.log(Cnorm[N-1])-np.log(Cnorm[int(N/2)-1]))/(N/2)
    rates.append(rate)
```

A.2. **Deterministic Plot.** Code for fitting the best fit curve and making the plot.

```
from scipy.optimize import curve_fit
x = kappas
y = np.array(rates)

def func(x, a, b):
    return a * (x**b)

popt, pcov = curve_fit(func, x, y) # Fit the best fit curve

plt.scatter(x,y)
plt.plot(x, func(x, *popt), color='purple', linestyle='--', label="Fitted Curve")
plt.xlabel(r'$\kappa$')
plt.ylabel(r'$\gamma$')
plt.title(r'Variation of decay rate $\gamma$ with moecular diffusion $\kappa$')
plt.show()

# Calculating sum of squares of residulas
sum_squares = ((x - func(x, *popt))**2).sum()
sum_squares
```

A.3. **Non-Deterministic Simulation.** Code for simulating the action of the non-deterministic sine map on the scalar flow. 10 simulations are done for each value of κ.

```
# Non-deterministic flow with several kappa values

pi = np.pi

m = 256 # number of grid points in x and y directions,
        # can be decreased for faster runs, or increased for higher accuracy
dx = 2*pi/m # grid spacing

x = np.linspace(0,2*pi-dx,m)
y = np.linspace(0,2*pi-dx,m)
```

```
xx, yy = np.meshgrid(x,y) # grid

k = np.linspace(-m/2,m/2-1,m) # grid in Fourier space (for smoothing)
k = fft.ifftshift(k)
l = k
kk, ll = np.meshgrid(k,l)

kappas = np.linspace(0.0001, 0.001, num = 10)
# array of values of kappa

N = 200 # number of iterations of the map
mu = 1
phi = np.random.rand(N,2)*2*np.pi # angles, random case


C = np.sin(xx)*np.sin(yy) # initial concentration
Cnorm = np.zeros(N)

Rates = []

index = -1
# Loop over iterations
for kappa in kappas:
    decay = np.exp(-kappa*(kk**2+ll**2))
    C = np.sin(xx)*np.sin(yy) # initial concentration
    Cnorm = np.zeros(N)
    for n in range(N):
        Cnorm[n] = np.sqrt(np.sum(C[:]**2)/m**2) # norm of the concentration
        for j in range(m):
            yshift = - mu * np.sin(x[j]+phi[n,1])
            yshift = np.mod(yshift,2*pi)
            indshift = round(yshift/dx)
            C[:,j] = np.roll(C[:,j],indshift)
        for i in range(m):
            xshift = - mu * np.sin(y[i]+phi[n,0])
            xshift = np.mod(xshift,2*pi)
            indshift = round(xshift/dx)
            C[i,:] = np.roll(C[i,:],indshift)

        # smoothing
        C = np.real(fft.ifft2(decay*fft.fft2(C)))

    # estimation of the decay rate from decay of Cnorm between iteration N/2 and N
    Rate = -(np.log(Cnorm[N-1])-np.log(Cnorm[int(N/2)-1]))/(N/2)
    Rates.append(Rate)
```

A.4. **Deterministic Plot.** Code for generating plot of non-deterministic case.

```
from scipy.optimize import curve_fit
x_ = kappas
y_ = np.array(Rates)

# a function that has the form of the assumed realtionship between
# gamma and kappa
def func(x, a, b):
    return a * (x**b)

# curve fit function finds the optimal values of a and b such that the
# sum of the squares of residulas is minimised
popt_, pcov_ = curve_fit(func, x_, y_)

# The points are ploted along with the best fit line
plt.scatter(x_,y_)
plt.plot(x_, func(x_, *popt_), color='purple', linestyle='--' ,  label="Fitted Curve")
plt.xlabel(r'$\kappa$')
plt.ylabel(r'$\gamma$')
plt.title(r'Variation of decay rate $\gamma$ with moecular diffusion $\kappa$')
plt.show()

# calculating sum of squares of residulas
sum_squares = ((x_ - func(x_, *popt_))**2).sum()
sum_squares
```