

AN INVESTIGATION INTO DISTRIBUTION
OF RANDOM CURVES

by

Hoang Long Nguyen

Submitted in partial fulfillment of the
requirements for Departmental Honors in
the Department of Mathematics
Texas Christian University

Fort Worth, Texas

December 12, 2022

AN INVESTIGATION INTO DISTRIBUTION
OF RANDOM CURVES

Project Approved:

Supervising Professor: Ken Richardson, Ph.D.

Department of Mathematics

Rhonda Hatcher, Ph.D.

Department of Mathematics

Antonio Sanchez, D.Sc.

Department of Computer Science

ACKNOWLEDGEMENTS

First and foremost, I am very appreciative to Dr. Ken Richardson, my research supervisor, for mentoring me through this project. His guidance, support, and patience have carried me through all the stages of writing my project.

I would also like to thank Dr. Rhonda Hatcher and Dr. Antonio Sanchez for agreeing to be committee members of my research project, and for your brilliant comments. Your feedbacks certainly will help me a lot in finalizing this thesis in its most complete form.

In closing, I want to express my gratitude to all of the professors at TCU Mathematics department for making my four years at this university so memorable. Every single one of the discussions I had with you all helped to mold me into the math student I am today and I will become in the future.

ABSTRACT

This research project focuses on random paths with bounded curvature, using differential geometry and statistical mechanics. It is known from the work of Einstein that random walks are connected to Brownian motion and diffusion. We will examine random curves that are not merely continuous but that are smooth and have prescribed bounds on curvature. We examine the distribution of a finite number of endpoints of such random curves. Using Python, we obtain 2-D histograms, graphs, and charts to examine the diffusion resulting from particles moving along these random curves. A central goal in statistical mechanics is to describe the large-scale behavior of systems with the distribution of randomly generated data; we compare the distributions of curve endpoints to the Gaussian (normal) distribution.

CHAPTER 1. DEFINITIONS AND PRELIMINARIES

In this research, I denote by \mathbb{R}^2 the set of doubles (x, y) of real numbers.

Definition 1: A parametrized differentiable curve is a differentiable map $\alpha : I \rightarrow \mathbb{R}^2$ of an interval $I = (a, b)$ of the real line \mathbb{R} into \mathbb{R}^2 .

The word *differentiable* in this definition means that α is a correspondence which maps each $t \in I$ into a point $\alpha(t) = (x(t), y(t)) \in \mathbb{R}^2$ in such a way that the functions $x(t)$ and $y(t)$ are differentiable. The variable t is called the parameter of the curve. The word *interval* is taken in a generalized sense, so that we do not exclude the cases $a = -\infty$, $b = +\infty$. If we denote by $x'(t), y'(t)$ the first derivative of x and y at the point t , the vector $(x'(t), y'(t)) = \alpha'(t) \in \mathbb{R}^2$ is called the *tangent vector* (or *velocity vector*) of the curve α at t .

Let $\alpha : I \rightarrow \mathbb{R}^3$ be a parametrized differentiable curve. For each $t \in I$ where $\alpha'(t) \neq 0$, there is a well-defined straight line, which contains the point $\alpha(t)$ and the tangent vector $\alpha'(t)$ of the curve α at t . Conversely, any point t where $\alpha'(t) = 0$ is called a *singular point* of α , and they will be excluded from my research.

Definition 2: A parametrized differentiable curve $\alpha : I \rightarrow \mathbb{R}^2$ is said to be regular if $\alpha'(t) \neq 0$ for all $t \in I$.

Thus, from now on I only focus on regular parametrized differentiable curves, and I usually will omit the word "differentiable" when mentioned later.

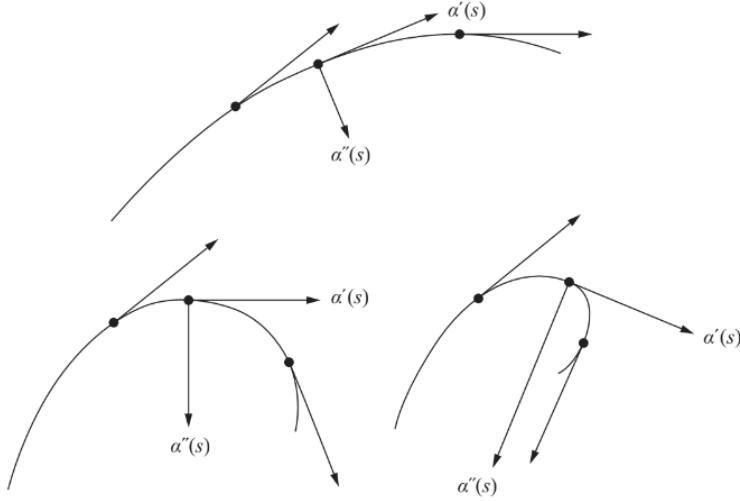
Given $t_0 \in I$, the *arc length* of a regular parametrized curve $\alpha : I \rightarrow \mathbb{R}^2$, from the point t_0 , is by definition

$$s(t) = \int_{t_0}^t |\alpha'(t)| dt,$$

where

$$|\alpha'(t)| = \sqrt{(x'(t))^2 + (y'(t))^2}$$

is the length of the vector $\alpha'(t)$. Since $\alpha'(t) \neq 0$, the arc length s is a differentiable function of t and $\frac{ds}{dt} = |\alpha'(t)|$.



Let $\alpha : I = (a, b) \rightarrow \mathbb{R}^2$ be a curve parametrized by arc length s . Since the tangent vector $\alpha'(s)$ has unit length, the norm $|\alpha''(s)|$ of the second derivative measures the rate of change of the angle which neighboring tangents make with the tangent at s . In other words, $|\alpha''(s)|$ gives a measure of how rapidly the curve pulls away from the tangent line at s , in a neighborhood of s (see some figures above). This suggests the following definition.

Definition 3: Let $\alpha : I \rightarrow \mathbb{R}^2$ be a curve parametrized by arc length $s \in I$, then the number $|\alpha''(s)| = k(s)$ is called the curvature of α at s .

If $n(s)$ is a unit vector perpendicular to $\alpha'(s)$ in the direction $\alpha''(s)$, $k(s)$ is well defined by the equation $\alpha''(s) = k(s) \cdot n(s)$. This way $k(s)$ can be positive or negative, if we declare $n(s)$ to be the vector $\alpha'(s)$ rotated by 90° counter clockwise. Moreover, $\alpha(s)$ is normal to $\alpha'(s)$, because by differentiating $\alpha'(s) \cdot \alpha'(s) = 1$ we obtain $\alpha(s) \cdot \alpha(s) = 0$. Thus, $n(s)$ is normal to $\alpha(s)$ and is called the normal vector at s .

Problem: Given a differentiable function $k(s)$, $s \in I$, show that the parametrized plane curve

having $k(s) = k$ as curvature is given by

$$\alpha(s) = \left(\int \cos \theta(s) ds + a, \int \sin \theta(s) ds + b \right),$$

where

$$\theta(s) = \int_0^s k(t) dt + \varphi,$$

and the curve is determined up to a translation of the vector (a, b) and a rotation of the angle φ .

Proof:

Assume the curvature of $\alpha(s)$ is $k(s)$. We have:

$$\begin{aligned} \alpha(s) &= \left(\int_0^s \cos \theta(t) dt + a, \int_0^s \sin \theta(t) dt + b \right) \\ \Rightarrow \alpha'(s) &= (\cos \theta(s), \sin \theta(s)) \end{aligned}$$

Moreover, $\theta(s) = \int k(s) ds + \varphi$ for a constant φ

$$\begin{aligned} \Rightarrow \alpha'(s) &= \left(\cos \left(\int_0^s k(t) dt + \varphi \right), \sin \left(\int_0^s k(t) dt + \varphi \right) \right) \\ \Rightarrow \alpha''(s) &= \left(-\sin \left(\int_0^s k(t) dt + \varphi \right) \cdot k(s), \cos \left(\int_0^s k(t) dt + \varphi \right) \cdot k(s) \right) \\ \Rightarrow \alpha''(s) &= k(s) \cdot (-\sin(\theta(s)), \cos(\theta(s))) = k(s) \cdot n(s) \end{aligned}$$

Assume that $\beta : I \rightarrow \mathbb{R}^2$ is another unit speed curve with curvature f . Let $\phi(s) = \theta(s) + \varphi$ be the smooth turning angle for β , then

$$\phi'(s) = k(s) \Rightarrow \phi = \int_{s_0}^s k(s) ds + \phi(s_0)$$

Since $\beta'(s) = (\cos \phi(s), \sin \phi(s))$ then $\beta(s) = \left(\int_{s_0}^s \cos \phi(s) ds, \int_{s_0}^s \sin \phi(s) ds \right) + \beta(s_0)$. Thus

$$\begin{aligned}\beta(s) &= \left(\int_{s_0}^s \cos(\theta(s) + \varphi) ds, \int_{s_0}^s \sin(\theta(s) + \varphi) ds \right) + \beta(s_0) \\ &= \left(\int_{s_0}^s (\cos \theta(t) \cos \varphi - \sin \theta(t) \sin \varphi) ds, \int_{s_0}^s (\sin \theta(t) \cos \varphi + \cos \theta(t) \sin \varphi) ds \right) + \beta(s_0) \\ &= \left(\cos \varphi \cdot \int_{s_0}^s \cos \theta(s) ds - \sin \varphi \cdot \int_{s_0}^s \sin \theta(s) ds, \right. \\ &\quad \left. \sin \varphi \cdot \int_{s_0}^s \cos \theta(s) ds + \cos \varphi \cdot \int_{s_0}^s \sin \theta(s) ds \right) + \beta(s_0) \\ &= \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \cdot \alpha(s) + \beta(s_0)\end{aligned}$$

Thus the curve is determined up to a translation and the rotation of angle φ .

These results are essential factors in my process of generating the curve $\alpha(s)$ from random provided curvature $k(s)$.

CHAPTER 2. DIFFERENT METHODS OF GENERATING RANDOM SMOOTH CURVES IN \mathbb{R}^2

My generating curves process is determining curves from its corresponding curvature $k(s)$. Given a function $k(s)$ to be the prescribed curvature of a unit speed curve starting with direction $a \in [0, 2\pi]$, the curve is obtained as follows.

Assume the angle of tangent vector = $\theta(s)$, then $\theta'(s) = k(s)$

$$\Rightarrow \theta(s) = a + \int_0^s k(t)dt$$

Given the angle of the tangent vector of the unit speed curve $(x(s), y(s))$, and $x'(s) = \cos(\theta(s))$, $y'(s) = \sin(\theta(s))$, so that if the curve starts at $(0, 0)$, then

$$\begin{aligned} x(s) &= \int_0^s \cos(\theta(t))dt \\ y(s) &= \int_0^s \sin(\theta(t))dt \end{aligned}$$

These examples below help me test different methods of obtaining "random" curvature functions.

2.1 Taylor Polynomial Curves with random coefficients in interval [-1, 1]

In this method, we let $K(s)$ be a 2nd degree polynomial with random coefficients.

$$K(s) = a_0 + a_1 \cdot s + a_2 \cdot s^2$$

Python Code to simulate curves:

We randomly assign values to coefficients a_0, a_1, a_2 in the interval $[-1, 1]$ with the help of the

“random.uniform()” function. Next, we get the function of $\theta(s)$ from integrating $K(s)$.

$$\begin{aligned}\theta(s) &= \int K(s)ds = \int (a_0 + a_1 \cdot s + a_2 \cdot s^2) ds \\ &= a_0 s + a_1 \cdot \frac{s^2}{2} + a_2 \cdot \frac{s^3}{3} + c\end{aligned}$$

In there, c also can randomly be assigned by using “random.random()” function times 2π . Next, we integrate $(\int \cos \theta(s)ds, \int \sin \theta(s)ds) = (x, y)$ by using the function “integrate.quad()” and hence get the plot of a generated curve (x, y) .

```
"""
Created on Thu Dec 30 11:31:29 2021
@author: Hoang Long Nguyen
"""

import matplotlib.pyplot as plt
import numpy as np
from scipy import integrate as IN
import random as ra

# Getting random values of coefficients
a0 = ra.uniform(-1,1)
print(a0)
a1 = ra.uniform(-1,1)
print(a1)
a2 = ra.uniform(-1,1)
print(a2)
randoma = 2*np.pi*ra.random()
print(randoma)

# Creating curve from provided curvature
def theta(s):
    return a0*s + a1*s**2/2 + a2*s**3/3 + randoma
def f(x):
    return np.cos(theta(x))
```

```

def f2(x):
    return np.sin(theta(x))

def integrateF(x):
    return IN.quad(f, 0, x)[0]

def integrateF2(x):
    return IN.quad(f2, 0, x)[0]

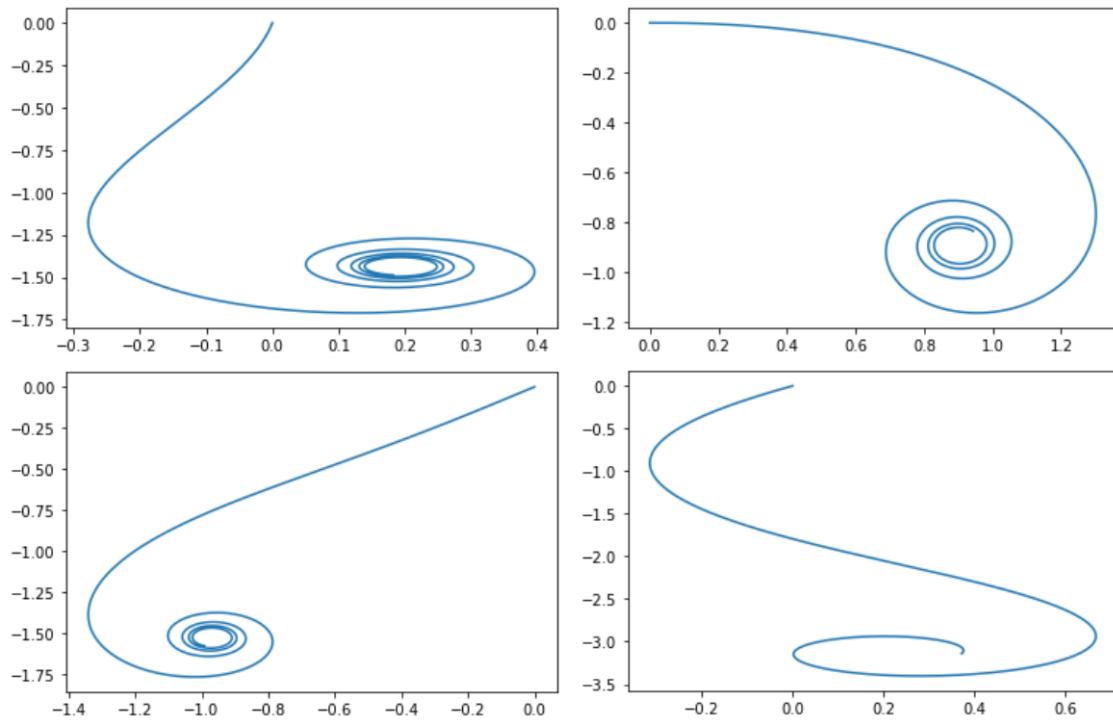
t = np.linspace(0, 5, num = 500)
x = np.array([integrateF(a) for a in t])
y = np.array([integrateF2(a) for a in t])

# Plotting the Curve
plt.plot(x,y)
plt.show()

```

Result:

Below, I provide some different Taylor polynomial curves generated by this code.



2.2 Fourier Series curves with random coefficients in interval [-1,1]

For this generating method, we let $K(s)$ be a Fourier Series with finite random coefficients.

$$K(s) = a_0 + a_1 \cdot \cos(s) + a_2 \cdot \cos(2s) + b_1 \cdot \sin(s) + b_2 \cdot \sin(2s)$$

Python Code to simulate curves:

Similar to the previous example, we also randomly assign values to the coefficients a_0, a_1, a_2, b_1, b_2 in the interval $[-1, 1]$ with the help of “random.uniform()” function. Next, we get the function of $\theta(s)$ from integrating $K(s)$.

$$\begin{aligned}\theta(s) &= \int K(s)ds = \int (a_0 + a_1 \cdot \cos(s) + a_2 \cdot \cos(2s) + b_1 \cdot \sin(s) + b_2 \cdot \sin(2s)) ds \\ &= a_0 \cdot s + a_1 \cdot \sin(s) + a_1 \cdot \frac{\sin(2s)}{2} - b_1 \cdot \cos(s) - b_2 \cdot \frac{\cos(2s)}{2} + c\end{aligned}$$

In there, c also can randomly be assigned by using “random.random()” function times 2π . Next, we integrate $(\int \cos \theta(s)ds, \int \sin \theta(s)ds) = (x, y)$ by using the function “integrate.quad()” and hence get the plot of a generated curve (x, y) .

” ” ”

Created on Thu Dec 30 11:39:41 2021

@author: Hoang Long Nguyen

” ” ”

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import integrate as IN
import random as ra

# Getting random values of coefficients
a0 = ra.uniform(-1,1)
print(a0)
a1 = ra.uniform(-1,1)
print(a1)
```

```

a2 = ra.uniform(-1,1)
print(a2)

b1 = ra.uniform(-1,1)
print(b1)

b2 = ra.uniform(-1,1)
print(b2)

randoma = 2*np.pi*ra.random()
print(randoma)

# Creating curve from provided curvature

def K(s):
    return a0+ a1*np.cos(s)+ a2*np.cos(2*s)+ b1*np.sin(s)+ b2*np.sin(2*s)

def theta(s):
    return IN.quad(K, 0, s) [0] + randoma

def f(x):
    return np.cos(theta(x))

def f2(x):
    return np.sin(theta(x))

def integrateF(x):
    return IN.quad(f, 0, x) [0]

def integrateF2(x):
    return IN.quad(f2, 0, x) [0]

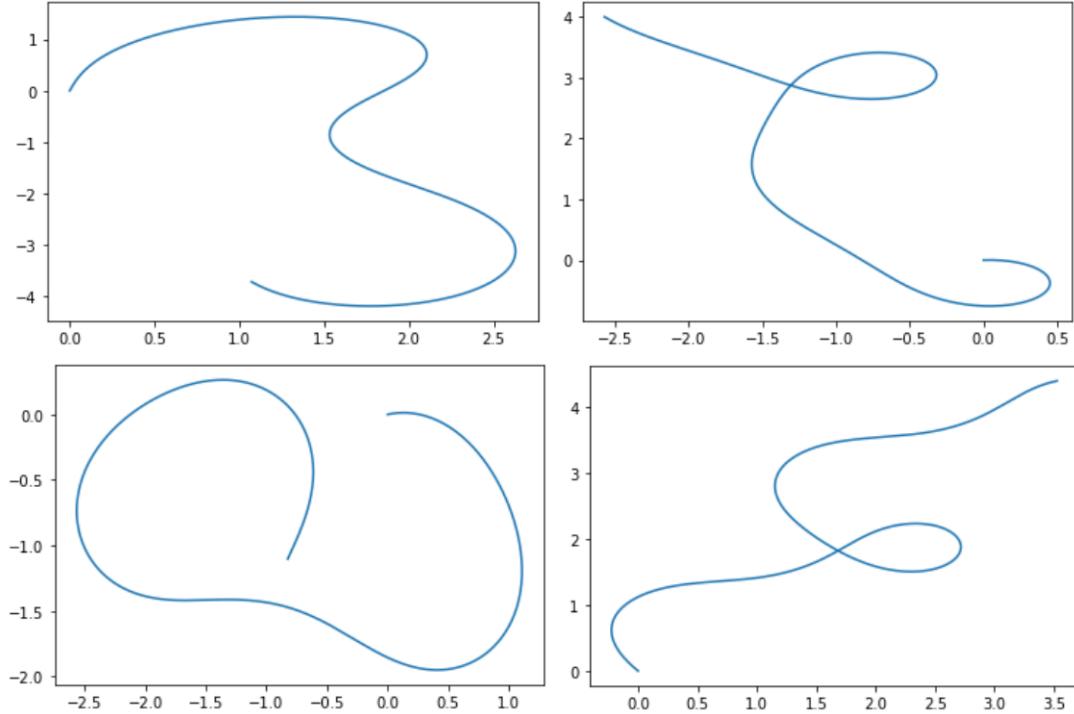
t = np.linspace(0, 10, num = 500)
x = np.array([integrateF(a) for a in t])
y = np.array([integrateF2(a) for a in t])

# Plotting the Curve
plt.plot(x,y)
plt.show()

```

Result:

Below, I provide some different Fourier Series curves generated by this code.



2.3 Sample curves generated from any curvatures $K(s)$

Letting $K(s)$ be a piecewise linear function with random value assigned between two curvature bounds. Choose $N = \text{number of subdivision of } [0, 2]$, so there are N random points a_1, a_2, \dots, a_n such that $-b \leq a_j \leq b \forall j$, and $K(s)$ is a piecewise linear function that connects these N random points. Also, I get the function of $\theta(s)$ from integrating $K(s)$. Next, we integrate

$$\left(\int \cos \theta(s) ds, \int \sin \theta(s) ds \right) = (x, y)$$

by using the function “`integrate.quad()`” and hence get the plot of a generated curve followed (x, y) .

Python code to simulate curves:

```
"""
Created on Thu Dec 30 11:39:41 2021
@author: Hoang Long Nguyen
"""

import pylab
```

```

import numpy as np
from scipy import integrate as IN
import random as ra
import itertools

# Getting random points and intervals
textinterval = input('Enter an interval b [-b, b]:')
nu = input('Enter number points:')
num = 1000
last_posx = []
n = eval(nu)
a = np.random.uniform(-eval(textinterval), eval(textinterval), n)
randoma = 2*np.pi*ra.random()

x = []
y = []
sum1 = 0
for i in range(n):
    x.append(i/(n-1)*2*np.pi)
    y.append(a[i])

# Creating the curvature of connecting random points
def K(s):
    jm = max([j for j in range(n) if x[j]<s])
    return (s-x[jm])*(y[jm+1]-y[jm])/((x[jm+1]-x[jm]) + y[jm])
def TRAP(s):
    difx = 1/(n-1)*2*np.pi
    jm = max([j for j in range(n) if x[j]<s])
    sum1 = sum([y[j] for j in range(jm+1)])
    return (sum1 - 1/2*(y[0]+y[jm]))*difx + 1/2*(y[jm] + K(s))*(s - x[jm])
def theta(s):
    return TRAP(s) + randoma
t = np.linspace(0.001, 2*np.pi, num = 500)

```

```

theta1 = np.array([K(s) for s in t])

# Plotting the Curvature
pylab.title('K(s) with N = ' + str(nu) + ' and b = ' + str(textinterval))
pylab.plot(t,theta1)
pylab.show()

# Creating the curve from curvature above
def f(u):
    return np.cos(theta(u))

def f2(v):
    return np.sin(theta(v))

def integrateF(u):
    return IN.quad(f,0,u)[0]

def integrateF2(u):
    return IN.quad(f2,0,u)[0]

h = 0.01
N = 100
t = [e*h for e in range(1,N)]
x1 = [f(z) for z in t]
y1 = [f2(z) for z in t]
vecx = [(x1[j]+x1[j+1])*h/2 for j in range(len(x1)-1)]
vecy = [(y1[j]+y1[j+1])*h/2 for j in range(len(y1)-1)]
intx = [sum(vecx[:j]) for j in range(1,len(vecx)-1)]
inty = [sum(vecy[:j]) for j in range(1,len(vecy)-1)]

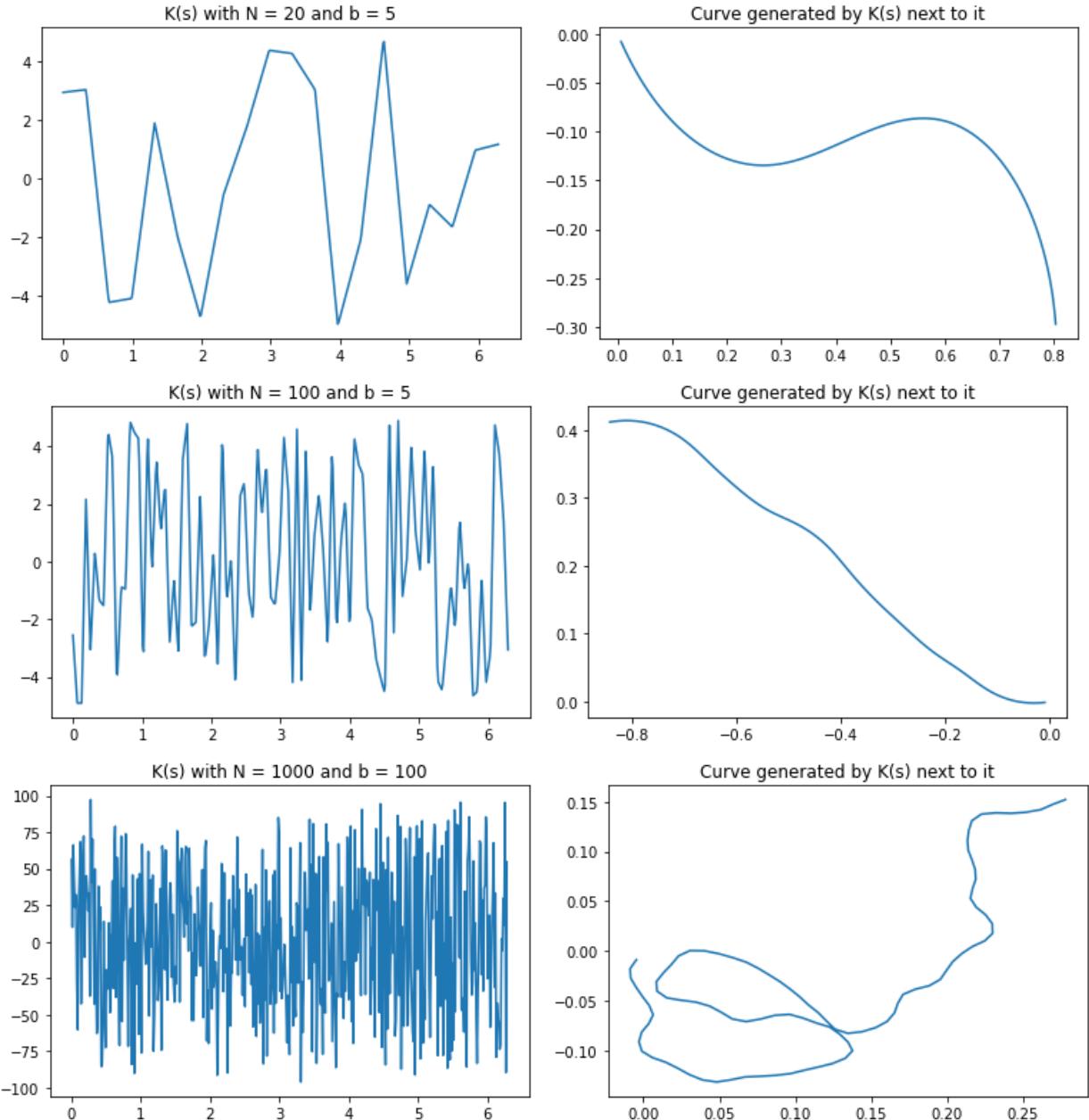
# Plotting the Curve
pylab.title('Curve generated by K(s) next to it')
pylab.plot(intx,inty)
pylab.show()

```

Result:

Here I provided some random curves along with its corresponding curvature $K(s)$, which is

generated by choosing an arbitrary number of random points in provided interval $K(s) \in [-b, b]$.



CHAPTER 3. ONE DIMENSIONAL RANDOM WALKS

3.1 Python Code to simulate curves:

Python code:

For a 1D random walk, we consider that the motion is going to be in just two directions i.e. either up or down, or left or right. Such a situation can be implemented as:

```
"""
Created on Sun Nov 27 18:50:57 2022
@author: Hoang Long Nguyen
"""

# Python code for 1-D random walk.

import random
import numpy as np
import matplotlib.pyplot as plt

# Probability to move up or down
prob = [0.5, 0.5]

# statically defining the starting position
start = 2
positions = [start]

# creating the random points
rr = np.random.random(100)
downp = rr < prob[0]
upp = rr > prob[1]

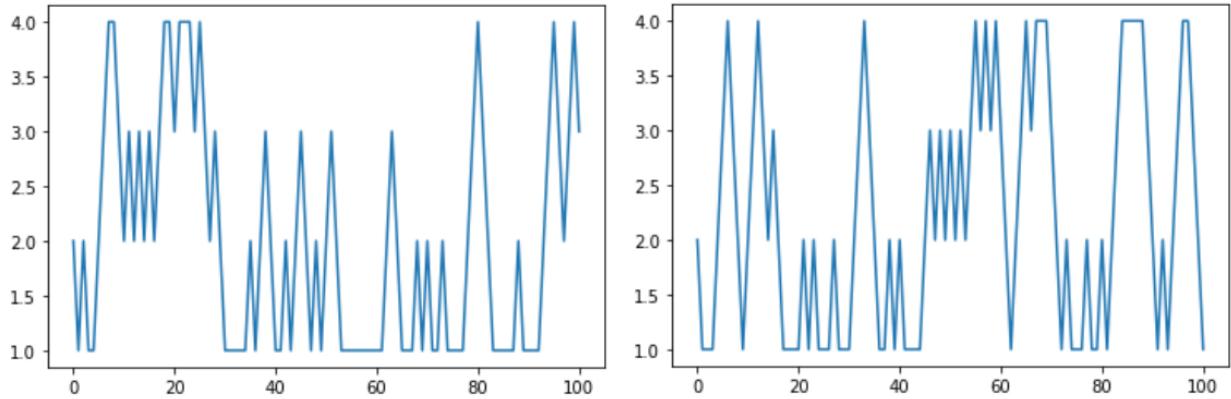
for idownp, iupp in zip(downp, upp):
    down = idownp and positions[-1] > 1
    up = iupp and positions[-1] < 4
    positions.append(positions[-1] - down + up)
```

```
# plotting down the graph of the random walk in 1D
plt.plot(positions)
plt.show()
```

We randomly assign a number to the “step” variable between 0 and 1 with the help of “random.uniform()” function. Then a threshold of 0.5 is set to determine the next step of the point (we can change the threshold to whatever value we want). And hence we get a plot of a random path on the 1-D plane.

Result:

I provided below two examples of random walks generated by this Python code.



3.2 Obtaining diffusion equations from random walks

In this section, we describe how Einstein showed that random walks in one dimension result in the density of particles satisfying the one-dimensional diffusion equation.

Let $F(x, t)$ = density of particles at position x at time t . This is assumed to be a C^4 function. Suppose that each particle has a 50% chance of moving right or left at each time interval (it moves Δ_x units in each Δ_t), then

$$F(x, t + \Delta_t) = \frac{1}{2}F(x - \Delta_x, t) + \frac{1}{2}F(x + \Delta_x, t) \quad (3.1)$$

Using the Taylor polynomial with Lagrange remainder,

$$F(x + \Delta_x, t) = F(x, t) + \Delta_x \cdot \frac{\partial F(x, t)}{\partial x} + \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} + \frac{\Delta_x^3}{3!} \cdot \frac{\partial^3 F(c_1, t)}{\partial x^3} \quad (3.2)$$

$$F(x - \Delta_x, t) = F(x, t) - \Delta_x \cdot \frac{\partial F(x, t)}{\partial x} + \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} - \frac{\Delta_x^3}{3!} \cdot \frac{\partial^3 F(c_2, t)}{\partial x^3} \quad (3.3)$$

for some c_1 between x and $x + \Delta_x$ and some c_2 between x and $x - \Delta_x$.

Plug results (3.2) and (3.3) into the equation (3.1), we have

$$\begin{aligned} F(x, t + \Delta_t) &= \frac{1}{2} \cdot \left(F(x, t) + \Delta_x \cdot \frac{\partial F(x, t)}{\partial x} + \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} + \frac{\Delta_x^3}{3!} \cdot \frac{\partial^3 F(c_1, t)}{\partial x^3} \right) \\ &\quad + \frac{1}{2} \cdot \left(F(x, t) - \Delta_x \cdot \frac{\partial F(x, t)}{\partial x} + \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} - \frac{\Delta_x^3}{3!} \cdot \frac{\partial^3 F(c_2, t)}{\partial x^3} \right) \\ \Rightarrow F(x, t + \Delta_t) &= \frac{1}{2} \cdot \left(2F(x, t) + 2 \cdot \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} + \frac{\Delta_x^3}{3!} \cdot \frac{\partial^3 F(c_2, t)}{\partial x^3} - \frac{\Delta_x^3}{3!} \cdot \frac{\partial^3 F(c_1, t)}{\partial x^3} \right) \\ \Rightarrow F(x, t + \Delta_t) &= F(x, t) + \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} + \mathcal{O}(\Delta_x^4) \quad (\text{odd terms cancel out}) \\ \Rightarrow F(x, t + \Delta_t) - F(x, t) &= \frac{\Delta_x^2}{2!} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} + \mathcal{O}(\Delta_x^4) \\ \Rightarrow \frac{F(x, t + \Delta_t) - F(x, t)}{\Delta_t} &= \frac{\Delta_x^2}{2\Delta_t} \cdot \frac{\partial^2 F(x, t)}{\partial x^2} + \frac{\mathcal{O}(\Delta_x^4)}{\Delta_t} \\ \Rightarrow \frac{\partial F(x, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_x^2}{2\Delta_t} \right) \frac{\partial^2 F(x, t)}{\partial x^2} \end{aligned}$$

The constant $D = \lim_{\Delta_t \rightarrow 0} \frac{\Delta_x^2}{2\Delta_t}$ is called as diffusivity. Assume that it exists. Then

$$\frac{\partial F(x, t)}{\partial t} = D \cdot \frac{\partial^2 F(x, t)}{\partial x^2} \quad (3.4)$$

This is the one-dimensional diffusion equation.

CHAPTER 4. TWO DIMENSIONAL RANDOM WALKS

In this chapter, we consider random walks through curves in two dimensions.

4.1 Normal Two Dimensional Random Walks:

4.1.1 Python code to simulate random walks:

Python code:

Two-dimentional Random Walk is propagated in a 2-D($x-y$) plane. At each instant, the particle moves in one of four possible directions: Up, Down, Left or Right.

To visualize the two-dimensional case, we can think about a person in the imagination who is walking randomly around a city. The city is effectively infinite and also arranged in a square grid of sidewalks. At every intersection, the person randomly chooses one of the four possible routes (including the one originally traveled from and previously crossed by).

Such a situation can be implemented as:

```
"""
Created on Sun Nov 27 19:30:17 2022
@author: Hoang Long Nguyen
"""

# Python code for 2D random walk.

import numpy
import pylab
import random

# defining the number of steps
n = 2000

#creating two array for containing x and y coordinate
#of size equals to the number of size and filled up with 0's
```

```

x = numpy.zeros(n)
y = numpy.zeros(n)

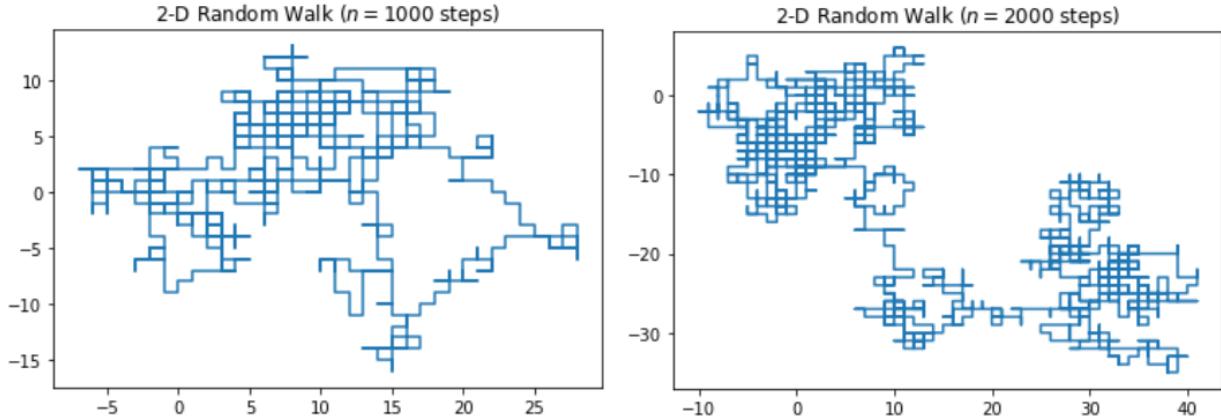
# filling the coordinates with random variables
direction= ["N", "S", "E", "W"]
for i in range(1,n):
    step = random.choice(direction)
    if step == "E":
        x[i] = x[i-1]+1
        y[i] = y[i-1]
    elif step == "W":
        x[i] = x[i-1]-1
        y[i] = y[i-1]
    elif step == "N":
        x[i] = x[i-1]
        y[i] = y[i-1]+1
    else:
        x[i] = x[i-1]
        y[i] = y[i-1]-1

# plotting stuff:
pylab.title("2-D Random Walk ($n = " + str(n) + "$ steps)")
pylab.plot(x, y)
pylab.savefig("rand_walk"+str(n)+".png", bbox_inches="tight", dpi=600)
pylab.show()

```

In the code above we assign a variable “direction” to four directions of movements i.e. North, South, West, East. Then, we randomly assign the direction of movement to the “step” variable with the help of “random. choice” function. The x and y coordinates of the particles are updated in accordance with the chosen direction, causing the particle to move randomly.

Result:



The output above visualizes the movement of a point (or particle) over a 2-D plane in a random manner. According to the chosen random direction, the particle can move in four directions (North, South, East and, West) over the course of 1000 steps. When choosing North, the x-coordinate increases by 1, when choosing South, the x-coordinate decreases by 1, when choosing East, the y-coordinate increases by 1, and when choosing West, the y-coordinate decreases by 1. Hence, the particle finishes its random walk.

4.1.2 Obtaining diffusion equations from random walks

We will now show how the diffusion equation results from two-dimensional random walks.

Let $f(x, y, t) = \text{density of particles at position } (x, y) \text{ in } \mathbb{R}^2 \text{ at time } t$. Suppose that each particle has a 50% chance of moving right or left, and 50% chance of moving up and down at each time interval. Then

$$\begin{aligned} f(x, y, t + \Delta_t) &= \frac{1}{4}f(x - \Delta_x, y - \Delta_y, t) + \frac{1}{4}f(x - \Delta_x, y + \Delta_y, t) \\ &\quad + \frac{1}{4}f(x + \Delta_x, y - \Delta_y, t) + \frac{1}{4}f(x + \Delta_x, y + \Delta_y, t) \end{aligned}$$

Using the Taylor polynomial with Lagrange remainder,

$$\begin{aligned} & f(x - \Delta_x, y - \Delta_y, t) \\ & \approx f(x, y) - f_x(x, y)\Delta_x - f_y(x, y)\Delta_y + \frac{f_{xx}(x, y)}{2}\Delta_x^2 + f_{xy}(x, y)\Delta_x\Delta_y + \frac{f_{yy}(x, y)}{2}\Delta_y^2 \end{aligned}$$

$$\begin{aligned} & f(x - \Delta_x, y + \Delta_y, t) \\ & \approx f(x, y) - f_x(x, y)\Delta_x + f_y(x, y)\Delta_y + \frac{f_{xx}(x, y)}{2}\Delta_x^2 - f_{xy}(x, y)\Delta_x\Delta_y + \frac{f_{yy}(x, y)}{2}\Delta_y^2 \end{aligned}$$

$$\begin{aligned} & f(x + \Delta_x, y - \Delta_y, t) \\ & \approx f(x, y) + f_x(x, y)\Delta_x - f_y(x, y)\Delta_y + \frac{f_{xx}(x, y)}{2}\Delta_x^2 - f_{xy}(x, y)\Delta_x\Delta_y + \frac{f_{yy}(x, y)}{2}\Delta_y^2 \end{aligned}$$

$$\begin{aligned} & f(x + \Delta_x, y + \Delta_y, t) \\ & \approx f(x, y) + f_x(x, y)\Delta_x + f_y(x, y)\Delta_y + \frac{f_{xx}(x, y)}{2}\Delta_x^2 + f_{xy}(x, y)\Delta_x\Delta_y + \frac{f_{yy}(x, y)}{2}\Delta_y^2 \end{aligned}$$

Here \approx means $=$ up to $\mathcal{O}((\Delta_x^2 + \Delta_y^2)^{3/2})$. So, we have

$$\begin{aligned} & \Rightarrow f(x, y, t + \Delta_t) = \frac{1}{4} \left(4f(x, y, t) + 4 \cdot \frac{f_{xx}(x, y)}{2}\Delta_x^2 + 4 \cdot \frac{f_{yy}(x, y)}{2}\Delta_y^2 \right) + \mathcal{O}((\Delta_x^2 + \Delta_y^2)^{3/2}) \\ & \Rightarrow f(x, y, t + \Delta_t) \approx f(x, y, t) + \frac{f_{xx}(x, y)}{2} \cdot \Delta_x^2 + \frac{f_{yy}(x, y)}{2} \cdot \Delta_y^2 \\ & \Rightarrow f(x, y, t + \Delta_t) - f(x, y, t) \approx \frac{f_{xx}(x, y)}{2} \cdot \Delta_x^2 + \frac{f_{yy}(x, y)}{2} \cdot \Delta_y^2 \\ & \Rightarrow \frac{f(x, y, t + \Delta_t) - f(x, y, t)}{\Delta_t} \approx \frac{\Delta_x^2}{2\Delta_t} \cdot f_{xx}(x, y) + \frac{\Delta_y^2}{2\Delta_t} \cdot f_{yy}(x, y) \\ & \Rightarrow \frac{\partial f(x, y, t)}{\partial t} \approx \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_x^2}{2\Delta_t} \right) \frac{\partial^2 f(x, y, t)}{\partial^2 x} + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_y^2}{2\Delta_t} \right) \frac{\partial^2 f(x, y, t)}{\partial^2 y} \end{aligned}$$

Assuming $\Delta_x \approx \Delta_y$, and $\left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_x^2}{2\Delta_t}\right) = \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_y^2}{2\Delta_t}\right) = D$ exists as constant, then

$$\frac{\partial f(x, y, t)}{\partial t} = D \cdot \frac{\partial^2 f(x, y, t)}{\partial^2 x} + D \cdot \frac{\partial^2 f(x, y, t)}{\partial^2 y} \quad (4.1)$$

This is the two-dimensional diffusion equation.

4.2 Two Dimensional Random Walks in Polar Coordinates

4.2.1 Python code to simulate random walks:

Here I provided the Python code to simulate 2 dimensional random walks with choosing a direction with random angle at each step. We assign the starting point to be $(0, 0)$, and then it choose a direction with random angle $\theta \in [0, 2\pi]$ at each step. Using this formula below to replicate the moving pattern.

$$F(t + \Delta_t) = F(t) + N(0, dt^2, t, t + dt),$$

where $N(a, b, t_1, t_2)$ is a normally distributed random variable with mean a and variance b . The parameters t_1 and t_2 make explicit the statistical independence of N on different time intervals; that is, if $[t_1, t_2)$ and $[t_3, t_4)$ are disjoint intervals, then $N(a, b; t_1, t_2)$ and $N(a, b; t_3, t_4)$ are independent.

Python Code::

```
"""
Created on Fri Jun 24 11:13:45 2022
@author: Hoang Long Nguyen
"""

from scipy.stats import norm
delta = 1
dt = 0.1
x = 0.0
n = 20
```

```
for k in range(n):
    x = x + norm.rvs(scale=delta**2*dt)

from math import sqrt
from scipy.stats import norm
import numpy as np

def brownian(x0, n, dt, delta, out=None):
    x0 = np.asarray(x0)

    # For each element of x0, generate a sample of n numbers from
    # a normal distribution.
    r = norm.rvs(size=x0.shape + (n,), scale=delta*sqrt(dt))
    if out is None:
        out = np.empty(r.shape)
        np.cumsum(r, axis=-1, out=out)
        out += np.expand_dims(x0, axis=-1)

    return out

import numpy
from pylab import plot, show, grid, xlabel, ylabel
import numpy
from pylab import plot, show, grid, axis, xlabel, ylabel, title

delta = 0.1
T = 10.0
N = 500
dt = T/N
x = numpy.empty((2,N+1))
x[:, 0] = 0.0

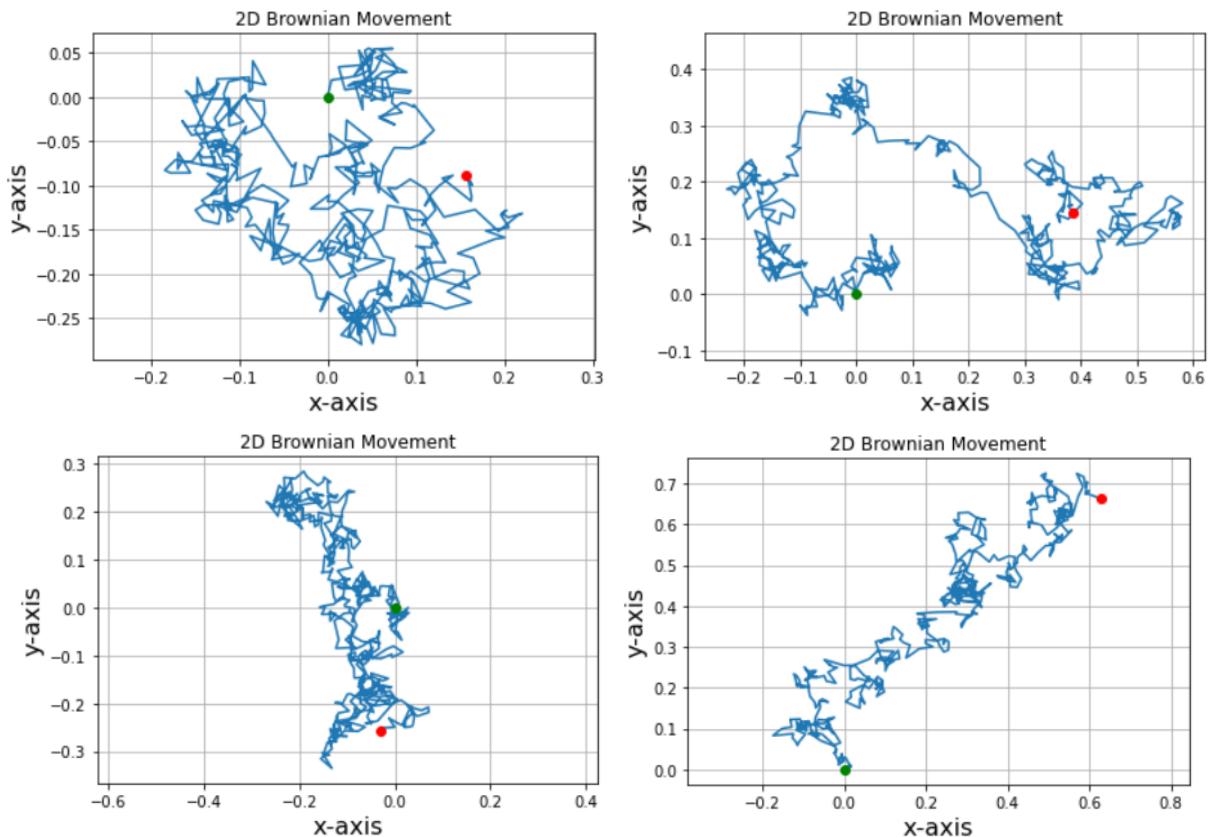
brownian(x[:,0], N, dt, delta, out=x[:,1:])
```

```
# plotting random created 2-D Brownian motion
plot(x[0],x[1])
plot(x[0,0],x[1,0], 'go')
plot(x[0,-1], x[1,-1], 'ro')

title('2D Brownian Movement')
xlabel('x-axis', fontsize=16)
ylabel('y-axis', fontsize=16)
axis('equal')
grid(True)
show()
```

Result:

I provide here some of random results generated by the code above. The output above shows the Brownian movement over a 2-D plane in a random manner. At each step, the point (or particle) can choose a random angle for next directions.



4.2.2 Obtaining diffusion equations from random walks

In time Δt , the particles move Δs and choose a direction with random angle $\theta \in [0, 2\pi]$ at each step. Let $f(x, y, t)$ be the distribution of particles at position (x, y) at time t . The function $f(x, y, t)$ is assumed to be a C^4 function, then $f(x, y, t + \Delta t) = \text{average of } f(x, y, t) \text{ that are distance } \Delta s \text{ from } (x, y)$.

$$f(x, y, t + \Delta t) = \frac{1}{2\pi} \int_0^{2\pi} f(x + \Delta s \cos \theta, y + \Delta s \sin \theta, t) d\theta$$

Using Taylor polynomial with Lagrange remainder,

$$\begin{aligned} f(x + \Delta x \cos \theta, y + \Delta y \sin \theta, t) &= f(x, y, t) + f_x(x, y, t) \cdot \Delta_s \cos \theta + f_y(x, y, t) \cdot \Delta_s \sin \theta \\ &\quad + \frac{f_{xx}(x, y, t)}{2} \cdot \Delta_s^2 \cos^2 \theta + f_{xy}(x, y, t) \cdot \Delta_s^2 \cos \theta \sin \theta + \frac{f_{yy}(x, y, t)}{2} \cdot \Delta_s^2 \sin^2 \theta \end{aligned}$$

$$\begin{aligned} \Rightarrow f(x, y, t + \Delta t) &= \frac{1}{2\pi} \int_0^{2\pi} \left(f(x, y, t) + f_x(x, y, t) \cdot \Delta_s \cos \theta + f_y(x, y, t) \cdot \Delta_s \sin \theta \right. \\ &\quad \left. + \frac{f_{xx}(x, y, t)}{2} \cdot \Delta_s^2 \cos^2 \theta + f_{xy}(x, y, t) \cdot \Delta_s^2 \cos \theta \sin \theta + \frac{f_{yy}(x, y, t)}{2} \cdot \Delta_s^2 \sin^2 \theta \right) d\theta \end{aligned}$$

We integrate each component

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi} f(x, y, t) d\theta &= \frac{1}{2\pi} f(x, y, t) \cdot \theta \Big|_0^{2\pi} \\ &= \frac{1}{2\pi} f(x, y, t) \cdot (2\pi - 0) \\ &= f(x, y, t) \end{aligned}$$

$$\begin{aligned}
\frac{1}{2\pi} \int_0^{2\pi} (f_x(x, y, t) \cdot \Delta_s \cos \theta) d\theta &= \frac{1}{2\pi} f_x(x, y, t) \cdot \Delta_s \cdot \sin \theta \Big|_0^{2\pi} \\
&= \frac{1}{2\pi} f_x(x, y, t) \cdot \Delta_s \cdot (\sin(2\pi) - \sin(0)) \\
&= \frac{1}{2\pi} f_x(x, y, t) \cdot \Delta_s \cdot 0 \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\frac{1}{2\pi} \int_0^{2\pi} (f_x(x, y, t) \cdot \Delta_s \sin \theta) d\theta &= \frac{1}{2\pi} f_x(x, y, t) \cdot \Delta_s \cdot (-\cos \theta) \Big|_0^{2\pi} \\
&= \frac{1}{2\pi} f_x(x, y, t) \cdot \Delta_s \cdot (-\cos(2\pi) + \cos(0)) \\
&= \frac{1}{2\pi} f_x(x, y, t) \cdot \Delta_s \cdot 0 \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\frac{1}{2\pi} \int_0^{2\pi} \left(\frac{f_{xx}(x, y, t)}{2} \cdot \Delta_s^2 \cos^2 \theta \right) d\theta &= \frac{f_{xx}(x, y, t) \cdot \Delta_s^2}{4\pi} \cdot \left(\frac{\theta}{2} + \frac{1}{4} \sin(2\theta) \right) \Big|_0^{2\pi} \\
&= \frac{f_{xx}(x, y, t) \cdot \Delta_s^2}{4\pi} \cdot \left(\frac{2\pi}{2} + \frac{1}{4} \sin(4\pi) - 0 - \frac{1}{4} \sin(0) \right) \\
&= \frac{f_{xx}(x, y, t) \cdot \Delta_s^2}{4\pi} \cdot \pi \\
&= \frac{f_{xx}(x, y, t) \cdot \Delta_s^2}{4}
\end{aligned}$$

$$\begin{aligned}
\frac{1}{2\pi} \int_0^{2\pi} (f_{xy}(x, y, t) \cdot \Delta_s^2 \cos \theta \sin \theta) d\theta &= \frac{1}{2\pi} f_{xy}(x, y, t) \cdot \Delta_s^2 \cdot \left(\frac{\sin^2 \theta}{2} \right) \Big|_0^{2\pi} \\
&= \frac{1}{2\pi} f_{xy}(x, y, t) \cdot \Delta_s^2 \cdot \left(\frac{\sin^2(2\pi)}{2} - \frac{\sin^2(0)}{2} \right) \\
&= \frac{1}{2\pi} f_{xy}(x, y, t) \cdot \Delta_s^2 \cdot 0 \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\frac{1}{2\pi} \int_0^{2\pi} \left(\frac{f_{yy}(x, y, t)}{2} \cdot \Delta_s^2 \sin^2 \theta \right) d\theta &= \frac{f_{yy}(x, y, t) \cdot \Delta_s^2}{4\pi} \cdot \left(\frac{\theta}{2} - \frac{1}{4} \sin(2\theta) \right) \Big|_0^{2\pi} \\
&= \frac{f_{yy}(x, y, t) \cdot \Delta_s^2}{4\pi} \cdot \left(\frac{2\pi}{2} - \frac{1}{4} \sin(4\pi) - 0 + \frac{1}{4} \sin(0) \right) \\
&= \frac{f_{yy}(x, y, t) \cdot \Delta_s^2}{4\pi} \cdot \pi \\
&= \frac{f_{yy}(x, y, t) \cdot \Delta_s^2}{4} \\
\Rightarrow f(x, y, t + \Delta_t) &= f(x, y, t) + \frac{f_{xx}(x, y, t) \Delta_s^2}{4} + \frac{f_{yy}(x, y, t) \Delta_s^2}{4} \\
\Rightarrow f(x, y, t + \Delta_t) - f(x, y, t) &= \frac{f_{xx}(x, y, t)}{4} \Delta_s^2 + \frac{f_{yy}(x, y, t)}{4} \Delta_s^2 \\
\Rightarrow \frac{f(x, y, t + \Delta_t) - f(x, y, t)}{\Delta_t} &= \frac{f_{xx}(x, y, t)}{4\Delta_t} \Delta_s^2 + \frac{f_{yy}(x, y, t)}{4\Delta_t} \Delta_s^2 \\
\Rightarrow \frac{\partial f(x, y, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t} \right) \left(\frac{\partial^2 f(x, y, t)}{\partial^2 x} + \frac{\partial^2 f(x, y, t)}{\partial^2 y} \right),
\end{aligned}$$

Assuming $\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t} = D$ exists as constant, then

$$\frac{\partial f(x, y, t)}{\partial t} = D \left(\frac{\partial^2 f(x, y, t)}{\partial^2 x} + \frac{\partial^2 f(x, y, t)}{\partial^2 y} \right)$$

which is the 2-dimensional diffusion equation found at (4.1).

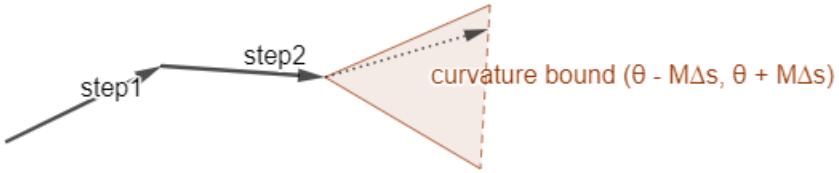
4.3 Two Dimensional Random Smooth Walks with Curvature Bounds

In what follows, we examine random particle paths where the paths are constrained by a curvature bound. In time Δ_t , the particles move Δ_s and choose a direction with random angle restricted

by a curvature bound at each step. If we assume that the curvature k satisfies $|k| \leq M$, then

$$\begin{aligned} -M &\leq k(s) \leq M \\ \Rightarrow -M &\leq \theta'(s) \leq M \\ \Rightarrow -M &\leq \frac{\Delta_\theta}{\Delta_s} \leq M \quad \left(\text{Since } \theta'(s) \approx \frac{\Delta_\theta}{\Delta_s} \right) \\ \Rightarrow -M\Delta_s &\leq \Delta_\theta \leq M\Delta_s \end{aligned}$$

Let $f(x, y, \theta, t)$ denote the density of particles at position (x, y) in \mathbb{R}^2 moving in direction θ at time t .



Imagine that at n^{th} step, the particle moved Δ_s with an angle θ , so in the next step, the particle can move toward any direction with new angle $\varphi \in (\theta - M\Delta_s, \theta + M\Delta_s)$.

Let $f(x, y, \theta, t + \Delta_t) = \text{average of } f(x, y, t) \text{ that are distance } \Delta_s \text{ from } (x, y) \text{ such that the angles are in the correct range}$. We have that a particle at time $t + \Delta_t$ with position $z = x + iy$ and the direction angle θ can come from a point at $z - \Delta_s e^{i\varphi}$ with direction angle φ as long as $\varphi \in [\theta - M\Delta_s, \theta + M\Delta_s]$.

Averaging over all possibilities, we have

$$f(z, \theta, t + \Delta_t) = \frac{1}{2M\Delta_s} \int_{\theta - M\Delta_s}^{\theta + M\Delta_s} f(z - \Delta_s e^{i\varphi}, \varphi, t) d\varphi$$

We will use Taylor polynomial with Lagrange remainder in complex coordinates where $z = x + iy$.

$$\frac{\partial}{\partial z} = \frac{1}{2} \cdot \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right), \quad \frac{\partial}{\partial \bar{z}} = \frac{1}{2} \cdot \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right)$$

$$f(z+h) = f(z) + h f_z(z) + \bar{h} f_{\bar{z}}(z) + \frac{h^2}{2} f_{zz}(z) + h \bar{h} f_{z\bar{z}}(z) + \frac{\bar{h}^2}{2} f_{\bar{z}\bar{z}}(z) + \mathcal{O}(|h|^3)$$

$$\begin{aligned} & \Rightarrow f(z - \Delta_s e^{i\varphi}, \varphi, t) = \\ & f(z, \varphi, t) - \Delta_s e^{i\varphi} \cdot f_z(z, \varphi, t) - \overline{\Delta_s e^{i\varphi}} \cdot f_{\bar{z}}(z, \varphi, t) + \frac{(\Delta_s e^{i\varphi})^2}{2} \cdot f_{zz}(z, \varphi, t) \\ & + \Delta_s e^{i\varphi} \cdot \overline{\Delta_s e^{i\varphi}} \cdot f_{z\bar{z}}(z, \varphi, t) + \frac{(\overline{\Delta_s e^{i\varphi}})^2}{2} \cdot f_{\bar{z}\bar{z}}(z, \varphi, t) + \mathcal{O}(|\Delta_s|^3) \end{aligned}$$

According to Euler's formula,

$$e^{i\varphi} = \cos \varphi + i \sin \varphi \Rightarrow \overline{e^{i\varphi}} = \cos \varphi - i \sin \varphi \Rightarrow \overline{e^{i\varphi}} = e^{-i\varphi}$$

$$\begin{aligned} & \Rightarrow f(z - \Delta_s \cdot e^{i\varphi}, \varphi, t) = f(z, \varphi, t) - \Delta_s e^{i\varphi} \cdot f_z(z, \varphi, t) - \Delta_s e^{-i\varphi} \cdot f_{\bar{z}}(z, \varphi, t) \\ & + \frac{\Delta_s^2 e^{2i\varphi}}{2} \cdot f_{zz}(z, \varphi, t) + \Delta_s^2 \cdot f_{z\bar{z}}(z, \varphi, t) + \frac{\Delta_s^2 e^{-2i\varphi}}{2} \cdot f_{\bar{z}\bar{z}}(z, \varphi, t) + \mathcal{O}(|\Delta_s e^{i\varphi}|^3) \end{aligned}$$

then,

$$\begin{aligned} f(z, \theta, t + \Delta_t) &= \frac{1}{2M\Delta_s} \int_{\theta}^{\theta + M\Delta_s} \left(f(z, \varphi, t) - \Delta_s e^{i\varphi} \cdot f_z(z, \varphi, t) - \Delta_s e^{-i\varphi} \cdot f_{\bar{z}}(z, \varphi, t) \right. \\ &+ \left. \frac{\Delta_s^2 e^{2i\varphi}}{2} \cdot f_{zz}(z, \varphi, t) + \Delta_s^2 \cdot f_{z\bar{z}}(z, \varphi, t) + \frac{\Delta_s^2 e^{-2i\varphi}}{2} \cdot f_{\bar{z}\bar{z}}(z, \varphi, t) \right) d\varphi \end{aligned}$$

We use the Fourier Series formula for each item in this function.

Let $f(z, \theta, t) = \sum_{p \in \mathbb{R}} a_p(z, t) \cdot e^{ip\theta}$. In the following, we will make use of the function

$$g(u) = \begin{cases} \left(\frac{\sin u}{u} \right) & \text{if } u \neq 0 \\ 1 & \text{if } u = 0 \end{cases}$$

$$\begin{aligned}
& \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} f(z, \varphi, t) d\varphi \\
&= \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \sum_{k \in Z} a_k(z, t) e^{ik\varphi} d\varphi \\
&= \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \left(a_0(z, t) + \sum_{k \in Z, k \neq 0} a_k(z, t) e^{ik\varphi} \right) d\varphi \\
&= \frac{1}{2M\Delta_s} \left(a_0(z, t) \cdot \varphi + \sum_{k \in Z, k \neq 0} a_k(z, t) \cdot \frac{-ie^{ik\varphi}}{k} \right) \Big|_{\theta-M\Delta_s}^{\theta+M\Delta_s} \\
&= \frac{1}{2M\Delta_s} \left(a_0(z, t) \cdot 2M\Delta_s + \sum_{k \in Z, k \neq 0} a_k(z, t) \cdot \frac{ie^{ik(\theta-M\Delta_s)} - ie^{ik(\theta+M\Delta_s)}}{k} \right) \\
&= a_0(z, t) + \sum_{k \in Z, k \neq 0} a_k(z, t) \cdot \frac{ie^{ik(\theta-M\Delta_s)} - ie^{ik(\theta+M\Delta_s)}}{2M\Delta_s \cdot k} \\
&= a_0(z, t) + \sum_{k \in Z, k \neq 0} \frac{a_k(z, t)}{2M\Delta_s} \cdot \left(\frac{i}{k \cdot e^{ikM\Delta_s}} - \frac{ie^{ikM\Delta_s}}{k} \right) \cdot e^{ik\theta} \\
&= a_0(z, t) + \sum_{k \in Z, k \neq 0} \frac{a_k(z, t)}{2} \cdot \frac{2 \sin(kM\Delta_s)}{kM\Delta_s} \cdot e^{ik\theta} \\
&= a_0(z, t) + \sum_{k \in Z, k \neq 0} a_k(z, t) \cdot g(kM\Delta_s) \cdot e^{ik\theta}
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \Delta_s e^{i\varphi} \cdot f_z(z, \varphi, t) d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} e^{i\varphi} \cdot f_z(z, \varphi, t) d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} e^{i\varphi} \cdot \left(\sum_{k \in Z} a_{kz}(z, t) \cdot e^{ik\varphi} \right) d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \sum_{k \in Z} a_{kz}(z, t) \cdot e^{i(k+1)\varphi} d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \left(a_{-1z}(z, t) + \sum_{k \in Z, k \neq -1} a_{kz}(z, t) \cdot e^{i(k+1)\varphi} \right) d\varphi \\
&= \frac{1}{2M} \left(a_{-1z}(z, t) \cdot \varphi + \sum_{k \in Z, k \neq -1} a_{kz}(z, t) \cdot \frac{-ie^{i(k+1)\varphi}}{k+1} \right) \Big|_{\theta-M\Delta_s}^{\theta+M\Delta_s} \\
&= \frac{1}{2M} \left(a_{-1z}(z, t) \cdot 2M\Delta_s + \sum_{k \in Z, k \neq -1} a_{kz}(z, t) \cdot \frac{ie^{i(k+1)(\theta-M\Delta_s)} - ie^{i(k+1)(\theta+M\Delta_s)}}{k+1} \right) \\
&= a_{-1z}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq -1} a_{kz}(z, t) \cdot \frac{ie^{i(k+1)(\theta-M\Delta_s)} - ie^{i(k+1)(\theta+M\Delta_s)}}{2M(k+1)} \\
&= a_{-1z}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq -1} \frac{a_{kz}(z, t)}{2M} \cdot \left(\frac{i}{(k+1) \cdot e^{i(k+1)M\Delta_s}} - \frac{ie^{i(k+1)M\Delta_s}}{k+1} \right) \cdot e^{i(k+1)\theta} \\
&= a_{-1z}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq -1} \frac{a_{kz}(z, t) \cdot \Delta_s}{2} \cdot \frac{2 \sin((k+1)M\Delta_s)}{(k+1)M\Delta_s} \cdot e^{i(k+1)\theta} \\
&= a_{-1z}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq -1} a_{kz}(z, t) \cdot \Delta_s \cdot g((k+1)M\Delta_s) \cdot e^{i(k+1)\theta}
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \Delta_s e^{-i\varphi} \cdot f_{\bar{z}}(z, \varphi, t) d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} e^{-i\varphi} \cdot f_{\bar{z}}(z, \varphi, t) d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} e^{-i\varphi} \cdot \left(\sum_{k \in Z} a_{k\bar{z}}(z, t) \cdot e^{ik\varphi} \right) d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \sum_{k \in Z} a_{k\bar{z}}(z, t) \cdot e^{i(k-1)\varphi} d\varphi \\
&= \frac{1}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \left(a_{1\bar{z}}(z, t) + \sum_{k \in Z, k \neq 1} a_{k\bar{z}}(z, t) \cdot e^{i(k-1)\varphi} \right) d\varphi \\
&= \frac{1}{2M} \left(a_{1\bar{z}}(z, t) \cdot \varphi + \sum_{k \in Z, k \neq 1} a_{k\bar{z}}(z, t) \cdot \frac{-ie^{i(k-1)\varphi}}{k-1} \right) \Big|_{\theta-M\Delta_s}^{\theta+M\Delta_s} \\
&= \frac{1}{2M} \left(a_{1\bar{z}}(z, t) \cdot 2M\Delta_s + \sum_{k \in Z, k \neq 1} a_{k\bar{z}}(z, t) \cdot \frac{ie^{i(k-1)(\theta-M\Delta_s)} - ie^{i(k-1)(\theta+M\Delta_s)}}{k-1} \right) \\
&= a_{1\bar{z}}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq 1} a_{k\bar{z}}(z, t) \cdot \frac{ie^{i(k-1)(\theta-M\Delta_s)} - ie^{i(k-1)(\theta+M\Delta_s)}}{2M(k-1)} \\
&= a_{1\bar{z}}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq 1} \frac{a_{k\bar{z}}(z, t)}{2M} \cdot \left(\frac{i}{(k-1) \cdot e^{i(k-1)M\Delta_s}} - \frac{ie^{i(k-1)M\Delta_s}}{k-1} \right) \cdot e^{i(k-1)\theta} \\
&= a_{1\bar{z}}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq 1} \frac{a_{k\bar{z}}(z, t) \cdot \Delta_s}{2} \cdot \frac{2 \sin((k-1)M\Delta_s)}{(k-1)M\Delta_s} \cdot e^{i(k-1)\theta} \\
&= a_{1\bar{z}}(z, t) \cdot \Delta_s + \sum_{k \in Z, k \neq 1} a_{k\bar{z}}(z, t) \cdot \Delta_s \cdot g((k-1)M\Delta_s) \cdot e^{i(k-1)\theta}
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} \frac{\Delta_s^2 e^{2i\varphi}}{2} \cdot f_{zz}(z, \varphi, t) d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} e^{2i\varphi} \cdot f_{zz}(z, \varphi, t) d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} e^{2i\varphi} \cdot \left(\sum_{k \in Z} a_{kzz}(z, t) \cdot e^{ik\varphi} \right) d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} \sum_{k \in Z} a_{kzz}(z, t) \cdot e^{i(k+2)\varphi} d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} \left(a_{-2zz}(z, t) + \sum_{k \in Z, k \neq -2} a_{kzz}(z, t) \cdot e^{i(k+2)\varphi} \right) d\varphi \\
&= \frac{\Delta_s}{4M} \left(a_{-2zz}(z, t) \cdot \varphi + \sum_{k \in Z, k \neq -2} a_{kzz}(z, t) \cdot \frac{-ie^{i(k+2)\varphi}}{k+2} \right) \Big|_{\theta-M\Delta_s}^{\theta+M\Delta_s} \\
&= \frac{\Delta_s}{4M} \left(a_{-2zz}(z, t) \cdot 2M\Delta_s + \sum_{k \in Z, k \neq -2} a_{kzz}(z, t) \cdot \frac{ie^{i(k+2)(\theta-M\Delta_s)} - ie^{i(k+2)(\theta+M\Delta_s)}}{k+2} \right) \\
&= a_{-2zz}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq -2} \Delta_s a_{kzz}(z, t) \cdot \frac{ie^{i(k+2)(\theta-M\Delta_s)} - ie^{i(k+2)(\theta+M\Delta_s)}}{4M(k+2)} \\
&= a_{-2zz}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq -2} \frac{\Delta_s a_{kzz}(z, t)}{4M} \cdot \left(\frac{i}{(k+2) \cdot e^{i(k+2)M\Delta_s}} - \frac{ie^{i(k+2)M\Delta_s}}{k+2} \right) \cdot e^{i(k+2)\theta} \\
&= a_{-2zz}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq -2} \frac{a_{kzz}(z, t) \cdot \Delta_s^2}{4} \cdot \frac{2 \sin((k+2)M\Delta_s)}{(k+2)M\Delta_s} e^{i(k+2)\theta} \\
&= a_{-2zz}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq -2} \frac{a_{kzz}(z, t) \cdot \Delta_s^2}{2} \cdot g((k+2)M\Delta_s) \cdot e^{i(k+2)\theta}
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \Delta_s^2 \cdot f_{z\bar{z}}(z, \varphi, t) d\varphi \\
&= \frac{\Delta_s}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} f_{z\bar{z}}(z, \varphi, t) d\varphi \\
&= \frac{\Delta_s}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \sum_{k \in Z} a_{k_{z\bar{z}}}(z, t) e^{ik\varphi} d\varphi \\
&= \frac{\Delta_s}{2M} \int_{\theta-M\Delta_s}^{\theta+M\Delta_s} \left(a_{0_{z\bar{z}}}(z, t) + \sum_{k \in Z, k \neq 0} a_{k_{z\bar{z}}}(z, t) e^{ik\varphi} \right) d\varphi \\
&= \frac{\Delta_s}{2M} \left(a_{0_{z\bar{z}}}(z, t) \cdot \varphi + \sum_{k \in Z, k \neq 0} a_{k_{z\bar{z}}}(z, t) \cdot \frac{-ie^{ik\varphi}}{k} \right) \Big|_{\theta-M\Delta_s}^{\theta+M\Delta_s} \\
&= \frac{\Delta_s}{2M} \left(a_{0_{z\bar{z}}}(z, t) \cdot 2M\Delta_s + \sum_{k \in Z, k \neq 0} a_{k_{z\bar{z}}}(z, t) \cdot \frac{ie^{ik(\theta-M\Delta_s)} - ie^{ik(\theta+M\Delta_s)}}{k} \right) \\
&= a_{0_{z\bar{z}}}(z, t) \cdot \Delta_s^2 + \sum_{k \in Z, k \neq 0} a_{k_{z\bar{z}}}(z, t) \cdot \Delta_s \cdot \frac{ie^{ik(\theta-M\Delta_s)} - ie^{ik(\theta+M\Delta_s)}}{2M \cdot k} \\
&= a_{0_{z\bar{z}}}(z, t) \cdot \Delta_s^2 + \sum_{k \in Z, k \neq 0} \frac{a_{k_{z\bar{z}}}(z, t) \cdot \Delta_s}{2M} \cdot \left(\frac{i}{k \cdot e^{ikM\Delta_s}} - \frac{ie^{ikM\Delta_s}}{k} \right) \cdot e^{ik\theta} \\
&= a_{0_{z\bar{z}}}(z, t) \cdot \Delta_s^2 + \sum_{k \in Z, k \neq 0} \frac{a_{k_{z\bar{z}}}(z, t) \cdot \Delta_s^2}{2} \cdot \frac{2 \sin(kM\Delta_s)}{kM\Delta_s} \cdot e^{ik\theta} \\
&= a_{0_{z\bar{z}}}(z, t) \cdot \Delta_s^2 + \sum_{k \in Z, k \neq 0} a_{k_{z\bar{z}}}(z, t) \cdot \Delta_s^2 \cdot g(kM\Delta_s) \cdot e^{ik\theta}
\end{aligned}$$

$$\begin{aligned}
& \frac{1}{2M\Delta_s} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} \frac{\Delta_s^2 e^{-2i\varphi}}{2} \cdot f_{z\bar{z}}(z, \varphi, t) d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} e^{-2i\varphi} \cdot f_{z\bar{z}}(z, \varphi, t) d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} e^{-2i\varphi} \cdot \left(\sum_{k \in Z} a_{k\bar{z}\bar{z}}(z, t) \cdot e^{ik\varphi} \right) d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} \sum_{k \in Z} a_{k\bar{z}\bar{z}}(z, t) \cdot e^{i(k-2)\varphi} d\varphi \\
&= \frac{\Delta_s}{4M} \int_{\theta-M\Delta_s}^{\theta+\Delta_s} \left(a_{2\bar{z}\bar{z}}(z, t) + \sum_{k \in Z, k \neq 2} a_{k\bar{z}\bar{z}}(z, t) \cdot e^{i(k-2)\varphi} \right) d\varphi \\
&= \frac{\Delta_s}{4M} \left(a_{2\bar{z}\bar{z}}(z, t) \cdot \varphi + \sum_{k \in Z, k \neq 2} a_{k\bar{z}\bar{z}}(z, t) \cdot \frac{-ie^{i(k-2)\varphi}}{k-2} \right) \Big|_{\theta-M\Delta_s}^{\theta+M\Delta_s} \\
&= \frac{\Delta_s}{4M} \left(a_{2\bar{z}\bar{z}}(z, t) \cdot 2M\Delta_s + \sum_{k \in Z, k \neq 2} a_{k\bar{z}\bar{z}}(z, t) \cdot \frac{ie^{i(k-2)(\theta-M\Delta_s)} - ie^{i(k-2)(\theta+M\Delta_s)}}{k-2} \right) \\
&= a_{2\bar{z}\bar{z}}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq 2} \Delta_s a_{k\bar{z}\bar{z}}(z, t) \cdot \frac{ie^{i(k-2)(\theta-M\Delta_s)} - ie^{i(k-2)(\theta+M\Delta_s)}}{4M(k-2)} \\
&= a_{2\bar{z}\bar{z}}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq 2} \frac{\Delta_s a_{k\bar{z}\bar{z}}(z, t)}{4M} \cdot \left(\frac{i}{(k-2) \cdot e^{i(k-2)M\Delta_s}} - \frac{ie^{i(k-2)M\Delta_s}}{k-2} \right) \cdot e^{i(k-2)\theta} \\
&= a_{2\bar{z}\bar{z}}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq 2} \frac{a_{k\bar{z}\bar{z}}(z, t) \cdot \Delta_s^2}{4} \cdot \frac{2 \sin((k-2)M\Delta_s)}{(k-2)M\Delta_s} e^{i(k-2)\theta} \\
&= a_{2\bar{z}\bar{z}}(z, t) \cdot \frac{\Delta_s^2}{2} + \sum_{k \in Z, k \neq 2} \frac{a_{k\bar{z}\bar{z}}(z, t) \cdot \Delta_s^2}{2} \cdot g((k-2)M\Delta_s) \cdot e^{i(k-2)\theta}
\end{aligned}$$

Thus, letting

$$f(z, \theta, t + \Delta_t) = \sum_{k \in Z} a_k(z, t + \Delta_t) e^{ik\theta},$$

We have

$$\begin{aligned}
\sum_{k \in Z} a_k(z, t + \Delta_t) e^{ik\theta} &= \sum_{k \in Z} a_k(z, t) \cdot e^{ik\theta} \cdot g(kM\Delta_s) - \sum_{k \in Z} a_{k_z}(z, t) \cdot e^{i(k+1)\theta} \cdot \Delta_s \cdot g((k+1)M\Delta_s) \\
&\quad - \sum_{k \in Z} a_{k_{\bar{z}}}(z, t) \cdot e^{i(k-1)\theta} \cdot \Delta_s \cdot g((k-1)M\Delta_s) + \sum_{k \in Z} a_{k_{zz}}(z, t) \cdot e^{i(k+2)\theta} \cdot \frac{\Delta_s^2}{2} \cdot g((k+2)M\Delta_s) \\
&\quad + \sum_{k \in Z} a_{k_{z\bar{z}}}(z, t) \cdot e^{ik\theta} \cdot \Delta_s^2 \cdot g(kM\Delta_s) + \sum_{k \in Z} a_{k_{\bar{z}\bar{z}}}(z, t) \cdot e^{i(k-2)\theta} \cdot \frac{\Delta_s^2}{2} \cdot g((k-2)M\Delta_s) \quad (4.2)
\end{aligned}$$

We have list of equations, $\forall p \in \mathbb{Z}$:

$$\begin{aligned}
a_p(z, t + \Delta_t) &= a_p(z, t) \cdot g(pM\Delta_s) - a_{p-1_z}(z, t) \cdot \Delta_s \cdot g(pM\Delta_s) \\
&\quad - a_{p+1_{\bar{z}}}(z, t) \cdot \Delta_s \cdot g(pM\Delta_s) + a_{p-2_{zz}}(z, t) \cdot \frac{\Delta_s^2}{2} \cdot g(pM\Delta_s) \\
&\quad + a_{p_{z\bar{z}}}(z, t) \cdot \Delta_s^2 \cdot g(pM\Delta_s) + a_{p+2_{\bar{z}\bar{z}}}(z, t) \cdot \frac{\Delta_s^2}{2} \cdot g(pM\Delta_s) \quad (4.3)
\end{aligned}$$

$$\Rightarrow a_p(z, t + \Delta_t) - a_p(z, t) =$$

$$\begin{aligned}
&a_p(z, t) \cdot (g(pM\Delta_s) - 1) - a_{p-1_z}(z, t) \cdot \Delta_s \cdot g(pM\Delta_s) \\
&- a_{p+1_{\bar{z}}}(z, t) \cdot \Delta_s \cdot g(pM\Delta_s) + a_{p-2_{zz}}(z, t) \cdot \frac{\Delta_s^2}{2} \cdot g(pM\Delta_s) \\
&+ a_{p_{z\bar{z}}}(z, t) \cdot \Delta_s^2 \cdot g(pM\Delta_s) + a_{p+2_{\bar{z}\bar{z}}}(z, t) \cdot \frac{\Delta_s^2}{2} \cdot g(pM\Delta_s) \quad (4.4)
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \frac{a_p(z, t + \Delta_t) - a_p(z, t)}{\Delta_t} &= \\
&a_p(z, t) \cdot \frac{g(pM\Delta_s) - 1}{\Delta_t} - a_{p-1_z}(z, t) \cdot \frac{\Delta_s \cdot g(pM\Delta_s)}{\Delta_t} \\
&- a_{p+1_{\bar{z}}}(z, t) \cdot \frac{\Delta_s \cdot g(pM\Delta_s)}{\Delta_t} + a_{p-2_{zz}}(z, t) \cdot \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{2\Delta_t} \\
&+ a_{p_{z\bar{z}}}(z, t) \cdot \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{\Delta_t} + a_{p+2_{\bar{z}\bar{z}}}(z, t) \cdot \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{2\Delta_t} \quad (4.5)
\end{aligned}$$

For every $p \in \mathbb{Z}$, we have:

$$\begin{aligned}\frac{\partial a_p(z, t)}{\partial z} &= \frac{1}{2} \cdot \left(\frac{\partial a_p(z, t)}{\partial x} - i \frac{\partial a_p(z, t)}{\partial y} \right) \\ \frac{\partial a_p(z, t)}{\partial \bar{z}} &= \frac{1}{2} \cdot \left(\frac{\partial a_p(z, t)}{\partial x} + i \frac{\partial a_p(z, t)}{\partial y} \right) \\ \Rightarrow a_{p_{z\bar{z}}} &= \frac{\partial^2 a_p(z, t)}{\partial z \partial \bar{z}} = \frac{1}{4} \cdot \left(\frac{\partial a_p(z, t)}{\partial x} - i \frac{\partial a_p(z, t)}{\partial y} \right) \cdot \left(\frac{\partial a_p(z, t)}{\partial x} + i \frac{\partial a_p(z, t)}{\partial y} \right) \\ &= \frac{1}{4} \cdot \left(\frac{\partial^2 a_p(z, t)}{\partial x^2} + \frac{\partial^2 a_p(z, t)}{\partial y^2} \right)\end{aligned}$$

Situation 1: Assume $f(z, \varphi, t) = a_0(z, t)$, so that we have $a_p = 0 \forall p \neq 0$. This is the situation where all directions are distributed uniformly, for all time.

From equation (5), we have:

$$\begin{aligned}a_0(z, t + \Delta_t) &= a_0(z, t) + a_{0_{z\bar{z}}} \cdot \Delta_s^2 \\ \Rightarrow a_0(z, t + \Delta_t) - a_0(z, t) &= a_{0_{z\bar{z}}} \cdot \Delta_s^2 \\ \Rightarrow \frac{a_0(z, t + \Delta_t) - a_0(z, t)}{\Delta_t} &= \frac{\Delta_s^2}{\Delta_t} \cdot a_{0_{z\bar{z}}} \\ \Rightarrow \frac{\partial a_0(z, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{\Delta_t} \right) \cdot \frac{1}{4} \cdot \left(\frac{\partial^2 a_0(z, t)}{\partial x^2} + \frac{\partial^2 a_0(z, t)}{\partial y^2} \right) \\ \Rightarrow \frac{\partial a_0(z, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t} \right) \cdot \left(\frac{\partial^2 a_0(z, t)}{\partial x^2} + \frac{\partial^2 a_0(z, t)}{\partial y^2} \right)\end{aligned}$$

Assume the diffusivity $D = \lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t}$ exists, then

$$\frac{\partial a_0(z, t)}{\partial t} = D \cdot \left(\frac{\partial^2 a_0(z, t)}{\partial x^2} + \frac{\partial^2 a_0(z, t)}{\partial y^2} \right)$$

Moreover, from equation (4), we also get:

$$a_0(z, t + \Delta_t) =$$

$$\begin{aligned} & a_0(z, t) - a_{0z}(z, t) \cdot e^{i\theta} \cdot \Delta_s \cdot g(M\Delta_s) - a_{0\bar{z}}(z, t) \cdot e^{-i\theta} \cdot \Delta_s \cdot g(-M\Delta_s) \\ & + a_{0zz}(z, t) \cdot e^{2i\theta} \cdot \frac{\Delta_s^2}{2} \cdot g(2M\Delta_s) + a_{0z\bar{z}} \cdot \Delta_s^2 + a_{0\bar{z}\bar{z}}(z, t) \cdot e^{-2i\theta} \cdot \frac{\Delta_s^2}{2} \cdot g(-2M\Delta_s) \end{aligned}$$

$$\Rightarrow a_0(z, t + \Delta_t) =$$

$$\begin{aligned} & a_0(z, t) - a_{0z}(z, t) \cdot e^{i\theta} \cdot \Delta_s \cdot g(M\Delta_s) - a_{0\bar{z}}(z, t) \cdot e^{-i\theta} \cdot \Delta_s \cdot g(M\Delta_s) \\ & + a_{0zz}(z, t) \cdot e^{2i\theta} \cdot \frac{\Delta_s^2}{2} \cdot g(2M\Delta_s) + a_{0z\bar{z}} \cdot \Delta_s^2 + a_{0\bar{z}\bar{z}}(z, t) \cdot e^{-2i\theta} \cdot \frac{\Delta_s^2}{2} \cdot g(2M\Delta_s) \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{a_0(z, t + \Delta_t) - a_0(z, t)}{\Delta_t} &= -\frac{\Delta_s}{\Delta_t} \cdot (a_{0z}(z, t) \cdot e^{i\theta} + a_{0\bar{z}}(z, t) \cdot e^{-i\theta}) \cdot g(M\Delta_s) \\ & + \frac{\Delta_s^2}{2\Delta_t} (a_{0zz}(z, t) \cdot e^{2i\theta} \cdot g(2M\Delta_s) + 2a_{0z\bar{z}} + a_{0\bar{z}\bar{z}}(z, t) \cdot e^{-2i\theta} \cdot g(2M\Delta_s)) \end{aligned}$$

$$\begin{aligned} \Rightarrow \frac{\partial a_0(z, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} -\frac{\Delta_s \cdot g(M\Delta_s)}{\Delta_t} \right) \cdot (a_{0z}(z, t) \cdot e^{i\theta} + a_{0\bar{z}}(z, t) \cdot e^{-i\theta}) \\ & + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{2\Delta_t} \right) (a_{0zz}(z, t) \cdot e^{2i\theta} \cdot g(2M\Delta_s) + 2a_{0z\bar{z}} + a_{0\bar{z}\bar{z}}(z, t) \cdot e^{-2i\theta} \cdot g(2M\Delta_s)) \end{aligned}$$

We see this only happen if $\frac{\partial a_0}{\partial t} = 0$, or in other words, $a_0(z, t)$ is constant in \mathbb{Z} .

Situation 2: Assume $f(z, \varphi, t) = a_p(z, t) \cdot e^{ip\varphi}$, $\forall p \neq 0, p \in \mathbb{Z}$, then we also have $a_q(z, t) = 0, \forall q \neq p$. Since $p \neq 0$, then $pM\Delta_s \neq 0$, we have

$$g(pM\Delta_s) = \frac{\sin(pM\Delta_s)}{pM\Delta_s}$$

According to Taylor series,

$$\sin(pM\Delta_s) = pM\Delta_s - \frac{(pM\Delta_s)^3}{3!} + \frac{(pM\Delta_s)^5}{5!} - \frac{(pM\Delta_s)^7}{7!} + \dots$$

$$\Rightarrow \frac{\sin(pM\Delta_s)}{pM\Delta_s} = 1 - \frac{(pM\Delta_s)^2}{3!} + \frac{(pM\Delta_s)^4}{5!} - \frac{(pM\Delta_s)^6}{7!} + \dots$$

$$\begin{aligned} \Rightarrow \frac{\sin(pM\Delta_s)}{pM\Delta_s} &= 1 - \frac{(pM\Delta_s)^2}{3!} + \mathcal{O}((pM\Delta_s)^4) \\ &\approx 1 - \frac{(pM\Delta_s)^2}{3!} \end{aligned}$$

from equation (5), we get

$$\begin{aligned} a_p(z, t + \Delta_t) &= a_p(z, t) \cdot g(pM\Delta_s) + a_{pz\bar{z}}(z, t) \cdot \Delta_s^2 \cdot g(pM\Delta_s) \\ \Rightarrow a_p(z, t + \Delta_t) - a_p(z, t) &= a_p(z, t) \cdot (g(pM\Delta_s) - 1) + a_{pz\bar{z}}(z, t) \cdot \Delta_s^2 \cdot g(pM\Delta_s) \\ \Rightarrow \frac{a_p(z, t + \Delta_t) - a_p(z, t)}{\Delta_t} &= \frac{(g(pM\Delta_s) - 1)}{\Delta_t} \cdot a_p(z, t) + \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{\Delta_t} \cdot a_{pz\bar{z}}(z, t) \\ \Rightarrow \frac{\partial a_p(z, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{(g(pM\Delta_s) - 1)}{\Delta_t} \right) \cdot a_p(z, t) + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{\Delta_t} \right) \cdot a_{pz\bar{z}}(z, t) \\ \Rightarrow \frac{\partial a_p(z, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{(g(pM\Delta_s) - 1)}{\Delta_t} \right) \cdot a_p(z, t) \\ &\quad + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{\Delta_t} \right) \cdot \frac{1}{4} \cdot \left(\frac{\partial^2 a_p(z, t)}{\partial x^2} + \frac{\partial^2 a_p(z, t)}{\partial y^2} \right) \\ \Rightarrow \frac{\partial a_p(z, t)}{\partial t} &= \left(\lim_{\Delta_t \rightarrow 0} \frac{(g(pM\Delta_s) - 1)}{\Delta_t} \right) \cdot a_p(z, t) \\ &\quad + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2 \cdot g(pM\Delta_s)}{4\Delta_t} \right) \cdot \left(\frac{\partial^2 a_p(z, t)}{\partial x^2} + \frac{\partial^2 a_p(z, t)}{\partial y^2} \right) \end{aligned}$$

$$\Rightarrow \frac{\partial a_p(z, t)}{\partial t} = \left(\lim_{\Delta_t \rightarrow 0} \frac{\left(1 - \frac{(pM\Delta_s)^2}{3!} - 1\right)}{\Delta_t} \right) \cdot a_p(z, t) + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2 \cdot \left(1 - \frac{(pM\Delta_s)^2}{3!}\right)}{4\Delta_t} \right) \cdot \left(\frac{\partial^2 a_p(z, t)}{\partial x^2} + \frac{\partial^2 a_p(z, t)}{\partial y^2} \right)$$

$$\Rightarrow \frac{\partial a_p(z, t)}{\partial t} \approx \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t} \right) \cdot \left(-\frac{2p^2 M^2}{3} \right) \cdot a_p(z, t) + \left(\lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t} \right) \cdot \left(\frac{\partial^2 a_p(z, t)}{\partial x^2} + \frac{\partial^2 a_p(z, t)}{\partial y^2} \right)$$

Consider C as constant, then $C = \frac{-2p^2 M^2}{3}$ as constant.

Assume the diffusivity $D = \lim_{\Delta_t \rightarrow 0} \frac{\Delta_s^2}{4\Delta_t}$ exists, then

$$\frac{\partial a_p(z, t)}{\partial t} = D \cdot C \cdot a_p(z, t) + D \cdot \left(\frac{\partial^2 a_p(z, t)}{\partial x^2} + \frac{\partial^2 a_p(z, t)}{\partial y^2} \right)$$

Also from equation (4), we have:

$$\begin{aligned} a_p(z, t + \Delta_t) e^{ip\theta} &= a_p(z, t) \cdot e^{ip\theta} \cdot g(pM\Delta_s) - a_{p_z}(z, t) \cdot e^{i(p+1)\theta} \cdot \Delta_s \cdot g((p+1)M\Delta_s) \\ &\quad - a_{p_{\bar{z}}}(z, t) \cdot e^{i(p-1)\theta} \cdot \Delta_s \cdot g((p-1)M\Delta_s) + a_{p_{zz}}(z, t) \cdot e^{i(p+2)\theta} \cdot \frac{\Delta_s^2}{2} \cdot g((p+2)M\Delta_s) \\ &\quad + a_{p_{z\bar{z}}}(z, t) \cdot e^{ip\theta} \cdot \Delta_s^2 \cdot g(pM\Delta_s) + a_{p_{\bar{z}\bar{z}}}(z, t) \cdot e^{i(p-2)\theta} \cdot \frac{\Delta_s^2}{2} \cdot g((p-2)M\Delta_s) \end{aligned}$$

Again, this only happens if a_p is a constant.

CHAPTER 5. RANDOM CURVES ENDPOINTS DISTRIBUTION HISTOGRAM

In this section, we will numerically model the random walks through curvature-constrained curves. We will plot end points of each curve and investigate the distribution of them. We create n random curves using Python, but we only record the end points. We examine 2 situations. The first situation ($randoma = 0$) corresponds to starting each curve with initial velocity in the specific x-direction. The second situation ($randoma = 2\pi$ times random floating point number between 0 and 1) corresponds to starting each curve with a random direction.

Python code for Histogram distribution of curves' end points:

```
"""
Created on Fri Jan 14 09:13:42 2022
@author: Hoang Long Nguyen
"""

import matplotlib.pyplot as plt
import numpy as np
from scipy import integrate as IN
import random as ra
import itertools

textinterval = input('Enter an interval b [-b, b]:')
nu = input('Enter number points:')
num = 5000
last_posx = []
last_posy = []
n = eval(nu)
h = 0.05
N = int(2*np.pi/h)

def K(x,y,s):
    jm = max([j for j in range(n) if x[j]<s])
    return np.exp(-((x[jm]-s)**2/(2*h**2)))
    #return np.exp(-((x[jm]-s)**2/(2*(h*N)**2)))
```

```

    return (s-x[jm])*(y[jm+1]-y[jm])/(x[jm+1]-x[jm]) + y[jm]

def TRAP(x,y,s):
    difx = 1/(n-1)*2*np.pi
    jm = max([j for j in range(n) if x[j]<s])
    sum1 = sum([y[j] for j in range(jm+1)])
    return (sum1- 1/2*(y[0]+y[jm]))*difx+ 1/2*(y[jm]+ K(x,y,s))*(s - x[jm])
def theta(x,y,s,randoma):
    return TRAP(x,y,s) + randoma
for _ in itertools.repeat(None, num):

    a = np.random.uniform(-eval(textinterval), eval(textinterval), n)
    # print(a)
    randoma = 2*np.pi*ra.random()
    #randoma=0

    x = []
    y = []
    sum1 = 0

    for i in range(n):
        x.append(i/(n-1)*2*np.pi)
        y.append(a[i])

    def f(u):
        return np.cos(theta(x,y,u,randoma))

    def f2(v):
        return np.sin(theta(x,y,v,randoma))

    def integrateF(u):
        return IN.quad(f,0,u)[0]

    def integrateF2(u):
        return IN.quad(f2,0,u)[0]

    t = [e*h for e in range(1,N)]

```

```

x1 = [f(z) for z in t]
y1 = [f2(z) for z in t]
vecx = [(x1[j]+x1[j+1])*h/2 for j in range(len(x1)-1)]
vecy = [(y1[j]+y1[j+1])*h/2 for j in range(len(y1)-1)]
intx = [sum(vecx[:j]) for j in range(1,len(vecx)-1)]
inty = [sum(vecy[:j]) for j in range(1,len(vecy)-1)]

last_posx.append(intx[-1])
last_posy.append(inty[-1])

fig, ax = plt.subplots(figsize =(10, 7))
MM = max(last_posx)
mm = min(last_posx)
numb = 35
bbins = [mm+i*(MM-mm)/numb for i in range(numb)]
ax.hist(last_posx, bins =bbins)

MMy = max(last_posy)
mmy = min(last_posy)

fig = plt.figure()
ax = fig.add_subplot(projection='3d')
hist, xedges, yedges = np.histogram2d(last_posx, last_posy, bins=20,
range=[[mm,MM], [mmy, MMy]])
# Construct arrays for the anchor positions of the 16 bars.
xpos, ypos = np.meshgrid(xedges[:-1], yedges[:-1], indexing="ij")
xpos = xpos.ravel()
ypos = ypos.ravel()
zpos = 0

dx = dy = 0.5 * np.ones_like(zpos)
dz = hist.ravel()

```

```

ax.bar3d(xpos, ypos, zpos, dx, dy, dz, zsort='average')

# Plotting stuff:
plt.show()

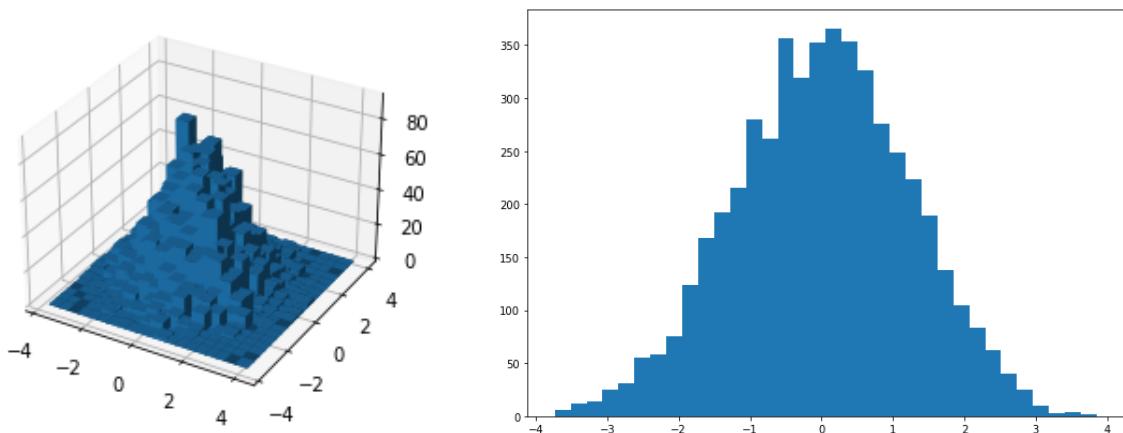
```

Result:

Situation 1: randoma = 0

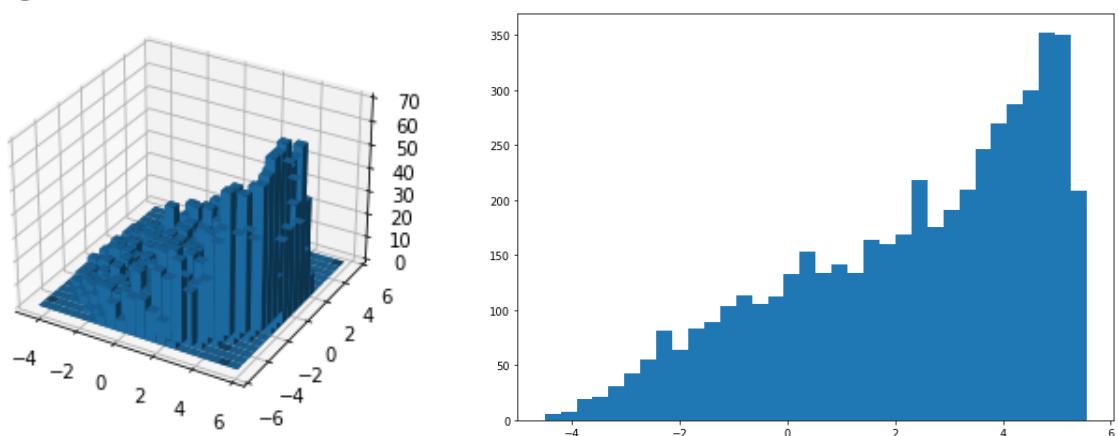
$N = 100$ and the interval $[-20, 20]$

3-D Histogram of Curves' End Points Distribution



$N = 100$ and the interval $[-6, 6]$

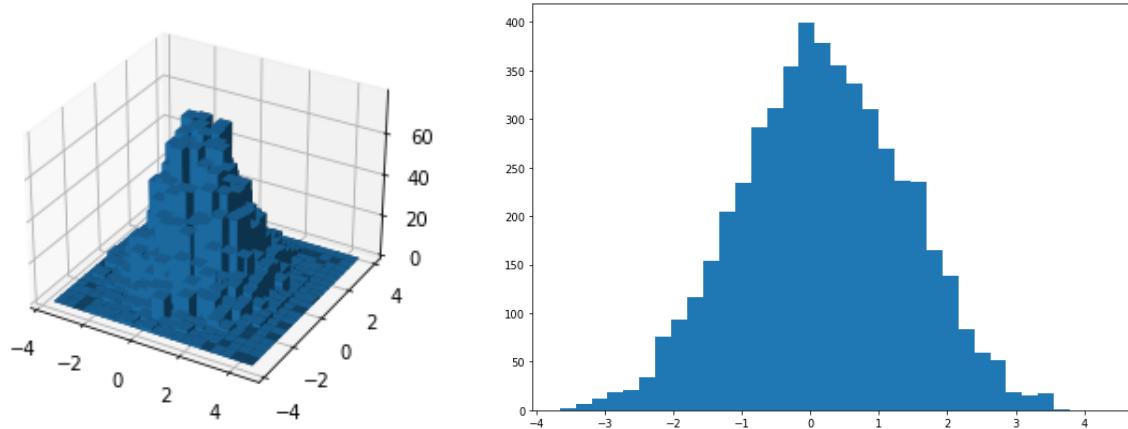
3-D Histogram of Curves' End Points Distribution



Situation 2: $\text{randoma} = 2\pi \cdot \text{ra.random}()$

$N = 100$ and the interval $[-20, 20]$

3-D Histogram of Curves' End Points Distribution



$N = 100$ and the interval $[-6, 6]$

3-D Histogram of Curves' End Points Distribution

