STOCHASTIC DIFFERENTIAL EQUATIONS, MONTE-CARLO SIMULATIONS, AND THE STATISTICS OF RANDOMLY ADVECTED TRIANGLES

HONORS THESIS
SUBMITTED TO THE DEPARTMENT OF MATHEMATICS
OF THE COLLEGE OF STATEN ISLAND
CITY UNIVERSITY OF NEW YORK
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF SCIENCE IN MATHEMATICS
WITH HONORS

Bassem Nawar

May 2016

I certify that I have read this honors thesis and that, in my opinion, it is fully adequate in scope and quality as an honors thesis for the degree of Bachelor of Science in Mathematics.

(Tobias Schäfer) Principal Adviser

I certify that I have read this honors thesis and that, in my opinion, it is fully adequate in scope and quality as an honors thesis for the degree of Bachelor of Science in Mathematics.

(Andrew Poje)

Approved for the Department of Mathematics.

Contents

List of Figures

Chapter 1

Background

1.1 Stochastic Differential Equations

What is a Stochastic Differential Equation? A stochastic differential equation (SDE) is a differential equation in which one or more of the terms are represented by a stochastic process, resulting in a solution which is also a stochastic process. SDEs are used to model various phenomena such as randomly varying stock prices or physical systems subject to thermal fluctuations. In general, a stochastic differential equation can be written in the form

$$dX = \mu(t, X(t)dt + \sigma(t, X(t))dW_t \tag{1.1}$$

Usually, our goal is to find a stochastic process of $X(t)$ satisfying the above equation. The coefficients μ and σ are functions that can depend on the time t and the stochastic process X itself. dW_t is the Brownian increment and we will discuss its role later in detail. For now, one might think of this increment as the term that adds the random fluctuations to the system. Finding a solution to (1.1) is often a complicated task, involving not only analytical, but often numerical techniques as well. When moving from deterministic to stochastic differential equations, it is helpful to introduce the concept of stochastic differential equation with an simple example, for instance a population growth model. Let $x = x(t) \equiv x_t$ denote the population at time t, and assume a constant (proportional) population growth rate, so that the change in the population at t is given by the deterministic differential equation:

$$dx_t = Kx_t dt\,, \qquad x(0) = x_0\,, \tag{1.2}$$

where K is some positive constant. Now suppose that due to some inherent randomness we can no longer assume that the intial condition x_0 to be determinstic constant. Then we may assume x_0 to be random variable $X_0(\omega)$ and to model the population growth by the differential equation:

$$dX_t(\omega) = KX_t(\omega)dt, \qquad x(0) = X_0(\omega) \tag{1.3}$$

The solution to this equation is $X_t(\omega) = X_0(\omega)e^{Kt}$. Note that $X_t(\omega)$ is a random variable, and in this case its randomness comes from the initial condition $X_0(\omega)$. Now suppose that even K is not known for certain, but that our knowledge of K is perturbed by some randomness, which we will model as the increment of a stochastic process so that

$$dX_t(\omega) = (Kdt + dW_t(\omega))X_t(\omega), \qquad x(0) = X_0(\omega) \tag{1.4}$$

The equation above is an example of stochastic differential equation, more generally a SDE is written as

$$dX_t(\omega) = f_t(X_t(\omega))dt + \sigma_t(X_t(\omega))dW_t(\omega) \tag{1.5}$$

where the function f corresponding to the determinstic part of the SDE is called the drift (the subscript t indicates that it may depend on the time t). The function σ_t is called the diffusion coefficient. $dW_t(\omega)$ is generally referred to underlying diffusion process which is in our case Brownian motion (also called Wiener process) denoted by $W(t)$ and we will give a short introduction to Brownian motion in the following section.

1.2 Brownian Motion

Standard Brownian Motion:

Definition: A Wiener process $W(t)$ is a stochastic process with the following properties:

1. $W(0) = 0$,
2. $W_t \sim N(0, t)$,
3. $W_t - W_s \sim N(0, t - s)$

Here, the symbol $N(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2. Brownian motion can be constructed as a limit of random walks.

Using the three properties listed above, one can show that, for each $t > 0$ the random variable defined by $W(t) = W(t) - W(0)$ is the increment in time interval $[0, t]$: This time increment is normally distributed with zero mean, variance t and a time-dependent density

$$f(t, x) = \frac{1}{\sqrt{2\pi t}} \mathrm{e}^{\frac{-x^2}{2t}} \tag{1.6}$$

Brownian Motion with constant drift:

Definition: A Brownian Motion with a constant drift μ is the stochastic process $X(t)$ which is the solution of an SDE (with diffusion coefficient σ) given by

$$dX(t) = \mu dt + \sigma dW(t)$$

with initial value $X(0) = x_0$. By direct integration one can see that

$$X(t) = x_o + \mu t + \sigma W(t) \tag{1.7}$$

and hence $X(t)$ is normally distributed, with mean $x_0 + \mu t$ and variance $\sigma^2 t$. Its density function is then given by

$$f(t, x) = \frac{1}{\sigma\sqrt{2\pi t}}\, \mathrm{e}^{-(x-x_0-\mu t)^2/(2\sigma^2 t)} \tag{1.8}$$

1.3 Ito's Lemma

A main tool for manipulating (and sometimes even solving) stochastic differential equations is Ito's Lemma. Remember that, for a function $f(x, y)$, the differential df is defined by an expansion which is correct to first order by

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy\,. \tag{1.9}$$

However, what if we have a function f which depends not only on a real variable t, but also on a stochastic process such as Brownian motion we need to take higher order terms into account as well. Suppose that f is given by $f = f(t, W_t)$, where W_t

denotes Brownian motion so we can write for the first-order expansion

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial W_t} dW_t \,. \tag{1.10}$$

If we expand df using Taylor's formula including higher-order terms, we obtain directly:

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial W_t} dW_t + \frac{1}{2}\frac{\partial^2 f}{\partial t^2}(dt)^2 + \frac{\partial^2 f}{\partial t \partial W_t} dt dW_t + \frac{1}{2}\frac{\partial^2 f}{\partial (W_t)^2}(dW_t)^2 + R \tag{1.11}$$

where the remainder R collects the higher-order terms. We discard all terms involving dt to a power higher than 1. Note that the term $dtdW_t$ has magnitude $(dt)^{\frac{3}{2}}$. this leaves the following expression for df:

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial W_t} dW_t + \frac{1}{2}\frac{\partial^2 f}{\partial W_t^2}(dW_t)^2 \tag{1.12}$$

We next use the fact that $(dW_t)^2 = dt$ and write

$$df = \left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial W_t^2} \right) dt + \frac{\partial f}{\partial W_t} dW_t \,. \tag{1.13}$$

This equation is called Ito's Lemma, and gives us the correct expression for caluclating differentials of composite function which depend on Brownian processes.

1.4 Examples of the application of Ito's Lemma

.

We can consider Ito's Lemma as a generalization of the determinstic chain rule. In the following we discuss several examples and applications of Ito's Lemma. Consider the special case where $X_t = W_t$ and $Y_t = f(W_t)$ then we have $\mu_t = 0$ and $\sigma_t = 1$ and so

$$dY_t = f'(W_t)dW_t + \frac{1}{2}f''(W_t)dt$$

Another example is the application to $f(x) = e^x$. Then, if we calculate the first and second derivative, we get $f(x) = f'(x) = f''(x)$. Let's assume constant volatility $\sigma_t = \sigma$ and constant drift $\mu_t = \mu$ and define

$$Y_t = f(X_t) = e^{X_t} = e^{\sigma W_t + \mu t} \tag{1.14}$$

This process is called *exponential Brownian motion*, and, by applying Ito's Lemma, we find that

$$\begin{aligned} dY_t &= f'(X_t)dX_t + \frac{1}{2}\sigma^2 f''(X_t)dt \\ &= Y_t\left(\sigma dW_t + \left(\mu + \frac{1}{2}\sigma^2\right)dt\right). \end{aligned}$$

In particular we see that for $\mu = -\frac{\sigma}{2}$, hence for

$$Y_t = Y_0\, \mathrm{e}^{\sigma W_t - \frac{\sigma^2 t}{2}}$$

we have $dY_t = \sigma Y_t dW_t$. This example is of relevance in the context of the Black-Scholes model which we discuss briefly below.

1.5 The Black-Scholes model

The Black Scholes model is used to price European options (which assumes that they must be held to expiration) and related custom derivatives. It takes into account that you have the option of investing in an asset earning the risk-free interest rate. The model acknowledges that the option price is purely a function of the volatility of the stock's price (the higher the volatility the higher the premium on the option).

The Black Scholes Model: The price of a European option is given by

$$C = SN(d_1) - K\mathrm{e}^{-rt}N(d_2) \tag{1.15}$$

where

$$d_1 = \frac{\ln\frac{S}{K} + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}, \qquad d_2 = \frac{\ln\frac{S}{K} + (r - \frac{\sigma^2}{2})t}{\sigma\sqrt{t}} \tag{1.16}$$

Here C is the theoretical call value, S the current stock price, N stands for the cumulative standard normal probability distribution. Moreover, the maturity is given by t until expiration, K is the option strike price and r the risk free interest rate. $\sigma =$ is the stock volatility. Most commonly, all the coefficients are given in the unit of years, sometimes people use days as the unit for time.

An example: Consider an option which has 20 days to expiration. The strike price is 105 and the price of stock is 100 and the stock has an daily volatility of 0.02. Assume an interest rate of 0.01. Then, we find

$$d_1 = \frac{ln\frac{100}{105} + \left(\frac{.01}{365} + \frac{.02^2}{2}\right)20}{.02\sqrt{20}} = -.49464$$

and, by similar calculation or use the following formula $d_2 = d_1 - \sigma\sqrt{t}$ we find that

$$d_2 = -0.49464 - \left(-0.02\sqrt{20}\right) = -0.58409$$

Therefore, as a result for the price of the European call option, we obtain

$$C = 100N\left(-0.49464\right) - 105e^{-0.01\left(\frac{20}{365}\right)}N\left(-0.58409\right) = 1.70$$

Below we show a MATLAB code to solve the Black Scholes Model:

```
function [vCall, Vput , d1 ,d2, Nd1, Nd2]= bs[ K, S0, sig, r,T]

d1 =(log(S0/K)+(r+sig^2/2)*T)/(sig*sqrt(T));
d2 =(log(S0/K)+(r-sig^2/2)*T)/(sig*sqrt(T));
Nd1 = my_normcdf(d1);
Nd2 = my_normcdf(d2);
vCall = S0*Nd1 - K*exp(-r*T)*Nd2;
vPut = my_normcdf(-d2)*K*exp(-r*T)-my_normcdf(-d1)*S0;
end

function y = my_normcdf(x)
  y = 0.5*erfc(-x/sqrt(2))$;
end
```

1.6 Examples involving Brownian motion, Ito's Lemma and Black Scholes

We conclude this introductory section by giving two more examples in the context of stochastic differential equations and their analytical description:

1. Example: Find dX_t where $X_t = X_0 e^{\sigma t W_t}$
   Solution: We consider $Y_t = X_0 e^X$ where $X = \sigma t W_t$. The stochastic product rule yields now $dX = \sigma t dW_t + \sigma W_t dt$ where $\sigma_t = \sigma t$ and $\mu_t = \sigma W_t$. Now we consider $f(x) = X_0\,\mathrm{e}^x$ so $f'(x) = X_0 e^x$ and $f''(x) = X_0 e^x$. With these preparations, we are ready to apply Ito's Lemma and we get

   $$dX_t = (\sigma t)(X_0 e^x) dW_t + (\sigma W_t)^2 (X_0 e^x) dt + \frac{1}{2}(\sigma t)^2 (X_0 e^x) dt$$

   Substituting back the original expression for X_t we obtain

   $$dX_t = X_t \left(\sigma t dW_t + \left(\sigma^2 W_t^2 + \frac{\sigma^2 t^2}{2} \right) dt \right)$$

2. Consider the stochastic differential equation given by

   $$dX_t = \sigma dW_t + \mu dt\,, \qquad X_0 = a > 0$$

   where σ and μ are constants.

   (a) Write down the solution X_t of the SDE.
   Solution:

   $$X_t = \sigma W_t + \mu t + a$$

   (b) Find mean and expectation value of X_t

Solution:

$\mathbf{E}(X_t) = a + \mu t, \qquad \mathbf{V}(X_t) = \sigma^2 t$

(c) Write down the probability density distribution function $p(x,t)$ of X_t

Solution:

$$p(x,t) = \frac{1}{\sqrt{2\pi\sigma^2 t}}\,\mathrm{e}^{\frac{-(x-a-\mu t)^2}{(2\sigma^2 t)}}$$

Chapter 2

Monte-Carlo Simulations

2.1 The basic idea of a Monte-Carlo simulation

One of the most common ways to estimate risk is the use of a Monte Carlo Simulation (MCS). For example, to calculate the value at risk of portfolio, we can run a Monte Carlo simulation that attempts to predict the worst likely loss for a portfolio given a confidence interval over a specified time horizon (which is the length of time over which an investment is made or held before it is liquidated). In this chapter, we will review a basic example of a Monte-Carlo simulation applied to a stock price. In order to do so, we need a model to specify the behavior of the stock price, and we could use one of the most common models in finance, which is the previously mentioned exponential Brownian motion commonly used in the Black-Scholes option pricing models. While MCS can refer to a universe of different approaches, we will start with the most basic one: a Monte-Carlo simulation for Brownian motion.

One of the most interesting features of a Monte Carlo simulation is the fact that one can attempt to predict the future many times over. At the end of the simulation, thousands or even millions of random trials produce a distribution of outcomes that can be analyzed with a variety of statistical tools.

2.2 A Monte-Carlo simulation for Brownian motion

In order to show how to generate random trials for a simple Monte-Carlo simulation, we present a code to that can be used to simulate Brownian motion:

```
function [t,W] = brownianMotion(nSteps,mpath)

t = zeros(1,nSteps+1);
W = zeros(mpath, nSteps+1);

tEnd = 2;
dt = tEnd/nSteps

W(1)=0;
t(1)=0;

for j=1:nSteps

    t(j+1) = t(j) +dt;
    W(:,j+1)= W(:,j) + t(j+1) *(sqrt(dt)*randn(mpath,1))

end
end
```

The purpose of this program is to create different realizations of the basic stochastic process $X_t = W_t$ satisfying the stochastic differential equation given by

$$dX_t = dW_t, \qquad X_0 = 0 \tag{2.1}$$

which is simple, drift-free Brownian motion. By calling the above function from the command window we can decide on the number of paths we would like to use for our simulation, for example

```
[t,W] = brownianMotion(100,100);
plot(t,W)
```

The above command will plot a total of 100 different realizations of the random paths for the case of Brownian motion and the number of realizations can be increased by changing the argument when calling the function. Figure 2.1 shows an example of the simulation result for 100 and for 1000 sample paths. By applying Monte Carlo by taking the end points of the random trials and we can create histogram and compare this to the theoretical probability distribution of Brownian motion. The simulation produced a distribution of hypothetical future outcomes. We keep in mind that Brownian Motion model assumes normality: the random numbers are normally distributed with expected return mean and standard deviation. In fact with more trials, the agreement of the probability distribution obtained by the Monte-Carlo simulation and to the theoretical distribution is expected to improve. For comparison, we use the following MATLAB codes:

```
[t,W]=brownianMotion(10000,10000);
dx=0.2;
ps=hist(W(:,end),x);
ps= ps/sum(ps)*1/dx;
plot(x,ps,x,1/sqrt(4*pi)*exp(-x.^2/4))
```

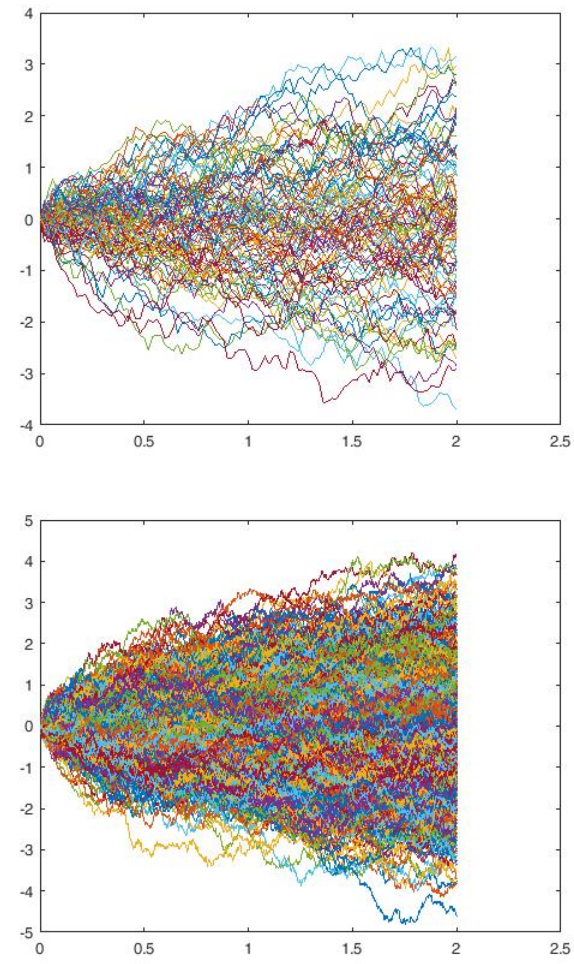

Figure 2.1: An example plot of 100 sample paths of simple Brownian motion (left) and 1000 sample paths (right) generated using a Monte-Carlo simulations.

Here, we made use of the fact that we simulate in the Monte-Carlo simulation up to a time $t = 2$. At this point in time, the analytical prediction for the probability density can be obtained from the fact that $W_t \sim N(0, t)$, hence we find

$$p(x, t = 2) = \frac{1}{\sqrt{4\pi}}\, \mathrm{e}^{-x.^2/4} \tag{2.2}$$

which we can use to compare to the histogram generated by the paths. Figure 2.2 shows the comparison for 100 and 1000 paths. Clearly, the agreement improves with larger n if n counts the number of sample paths.

Finally, a Monte Carlo Simulation applies a selected model to a large set of random trials in an attempt to produce a reasonable set of possible future outcomes.

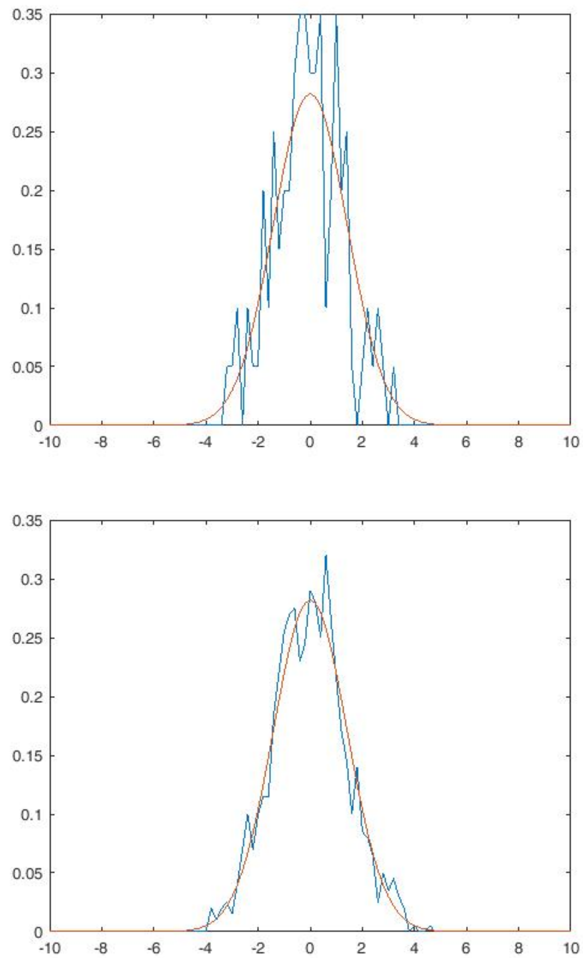

Figure 2.2: Comparison of the analytical probability density and the histogram obtained from the Monte-Carlo simulation paths.

Chapter 3

Application: Randomly Advected Triangles

3.1 Geometry of Lagrangian triangles

In the following we discuss an application of stochastic processes to understand the random advection of triangles. This work is based on the paper "Mechanisms driving shape distortion in two-dimensional flow" [2]:

In order to illustrate the physical processes governing the development of material areas in complex flow, we study the shape dynamics of three points Lagrangian clusters in an experimental quasi two dimensional flow. By comparing our measurements with simulations of triangles evolving purely diffusely, we show that the path taken by the mean triangle shape through a suitably defined phase space is indicative of the underlying flow dynamics.

In order to describe the shapes and sizes of Lagrangian triangles, we label their side lengths as A_1, A_2, A_3 with $A_1 \geq A_2 \geq A_3$ and their internal angles as $\theta_1, \theta_2, \theta_3$ with $\theta_1 \geq \theta_2 \geq \theta_3$. We characterize the size of a triangle with radius of gyration, given by $R_g^2 = \frac{1}{3}\,(A_1^2 + A_2^2 + A_3^2)$ and wee are interested in the triangle shape. The shape has two degrees of freedoms, and so we require two independent parameters to characterize it. One common choice is to define the vectors $\rho_1 \equiv \frac{(r_2 - r_1)}{\sqrt{2}}$ and $\rho_2 \equiv \frac{(2r_3 - r_2 - r_1)}{\sqrt{6}}$ where r_n is the position of the n-th triangle vertex. with these vectors, the parameters $\chi \equiv (\frac{1}{2})\,\arctan[2\rho_1\dot{\rho}_2/(\rho_2^2 - \rho_1^2)]$ and $\omega \equiv 2|\rho_1 \times \rho_2|/(\rho_1^2 + \rho_2^2)$ can then be defined to characterize the triangle shapes . Although χ and ω are independent quantities that are only functions of the triangle shape, they do not have a clear geometric interpretations . We therefore instead describe the triangle shape using set of parameters ,where the largest angle is θ, and ω to be the ratio of the smallest side to the intermediate side which gives a measure of the closeness of the nearest two vertices. we define $\theta \in [\frac{\pi}{3}, \pi]$ and $\omega \in [0, 1]$, we plot the joint probability density function of θ and ω.

3.2 Finding θ and ω using Monte-Carlo simulations

We can use the Monte-Carlo simulations in order to explore the probability density for the two random variables θ and ω. As a first step, we draw the edges of the random triangles using uniformly distributed random numbers. In MATLAB, this can be obtained using the function call

```
R = rand(6,n)
```

which produces a matrix with n columns(corresponding to the number of triangles) and 6 rows (we need two coordinates for each of the three points). Then, we can use the formulas of the previous section in order to create a histogram in the (θ, ω)-space. Below there is the corresponding MATLAB code:

```
function[bigTheta,Omega,R,cTheta,cOmega,c]= triangleLength(n,nBins)

R=rand(6,n); bigTheta=zeros(1,n); Omega=zeros(1,n);

for i=1:n
    j = 1;

    v1 = [R(j+2,i)-R(j,i), R(j+3,i)-R(j+1,i)];
    v2 = [R(j+4,i)-R(j+2,i),R(j+5,i)-R(j+3,i)];
    v3 = [R(j+4,i)-R(j,i), R(j+5,i)-R(j+1,i)];

    length1 = sqrt(v1(1)^2 + v1(2)^2);
    length2 = sqrt(v2(1)^2 + v2(2)^2);
    length3 = sqrt(v3(1)^2 + v3(2)^2);

    theta1 = acos(((v1(1)*v3(1))+(v1(2)*v3(2)))/(length1*length3));
    theta2 = acos(-((v1(1)*v2(1))+(v1(2)*v2(2)))/(length1*length2));
    theta3 = acos(((v2(1)*v3(1))+(v2(2)*v3(2)))/(length2*length3));

    A= [theta1, theta2, theta3]; bigTheta(i)=(max(A));
    C=[length1,length2,length3]; L=sort(C);  Omega(i)=(L(1)/L(2));
end

cTheta = linspace(pi/3,pi,nBins+1); cOmega = linspace(0,1,nBins+1);
dTheta = cTheta(2)-cTheta(1); dOmega = cOmega(2)-cOmega(1);
cTheta = cTheta(1:end-1); cOmega = cOmega(1:end-1);
c = zeros(nBins); v = [bigTheta;Omega];
```

```
for j=1:n
    x = bigTheta(j); y = Omega(j);
    i = find((cTheta<x)\& (x<cTheta+dTheta));
    k = find((cOmega<y)\& (y<cOmega+dOmega));
    c(i,k) = c(i,k)+1;
end
end
```

We can call the above function from the command window by simply typing

```
[bigTheta,Omega,R,cTheta,cOmega,c] = triangleLength(1e7,128);
```

and display the result in order to we get this Figure 3.1 which is very similar to one of the graphs (in Fig.2) of the referenced paper [2].

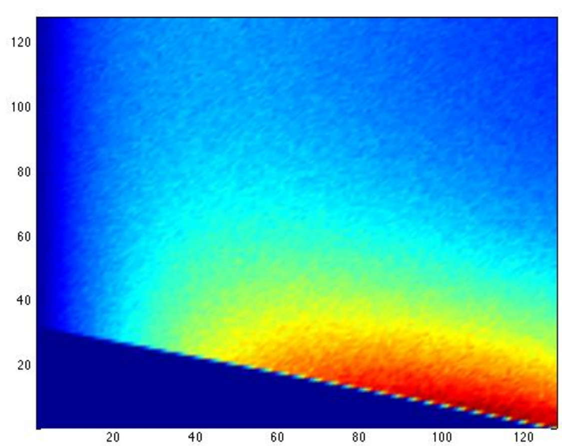

Figure 3.1: Two-dimensional histogram in (θ, ω)-space obtained from Monte-Carlo simulations of 10^7 triangles.

3.3 Advected Lagrangian triangles

It is fairly simple to add both a random and a deterministic flow to the triangles - as a simple example we consider the shear flow for the deterministic part of the flow given by

$$\dot{x} = y, \qquad \dot{y} = 0\,. \tag{3.1}$$

For the random part, we use Brownian motion. The MATLAB program needs to be modified: We need to generate random numbers at each step and, at the same time, monitor the evolution of the random field in (θ, ω)-space. Moreover, there are two main differences to the program before:

1. We start with an ensemble of triangles that have their vertices at $(0, 0)$, $(1, 0)$, and $(0, 1)$ instead of uniformly distributed vertices. In MATLAB, we achieve this by setting

   ```
   v = [0,0,1,0,0,1]'; R = repmat(v,[1,n]);
   ```

2. We follow the mean of the distribution rather than the distribution itself. This is done for computational purposes. Moreover, it is simpler to visualize the movement of the mean in the (θ, ω)-plane.

Below is the MATLAB code for this case:

```
function [t,bigThetaMean,OmegaMean]= triaPathShear(n,alpha)

v = [0,0,1,0,0,1]'; R = repmat(v,[1,n]);

nSteps = 100; dt = 0.01; t = zeros(1,nSteps+1);
bigThetaMean = zeros(1,nSteps+1); OmegaMean = zeros(1,nSteps+1);
```

```
bigTheta=zeros(1,n); Omega=zeros(1,n);

for k=1:nSteps+1
for i=1:n
    v1=[R(3,i)-R(1,i) , R(4,i)-R(2,i)];
        v2=[R(5,i)-R(1,i) , R(6,i)-R(2,i)];
        v3=[R(5,i)-R(3,i) , R(6,i)-R(4,i)];

        length1= sqrt(v1(1)^2 + v1(2)^2);
        length2= sqrt(v2(1)^2 + v2(2)^2);
        length3= sqrt(v3(1)^2 + v3(2)^2);

        theta1= acos(-((v1(1)*v3(1))+(v1(2)*v3(2)))/(length1*length3));
        theta2= acos(((v1(1)*v2(1))+(v1(2)*v2(2)))/(length1*length2));
        theta3= acos(((v2(1)*v3(1))+(v2(2)*v3(2)))/(length2*length3));

        A= [theta1, theta2, theta3]; bigTheta(i)=max(A);
        C=[length1,length2,length3]; L=sort(C); Omega(i)=L(1)/L(2);
end
    t(k) = (k-1)*dt;
    bigThetaMean(k) = mean(bigTheta); OmegaMean(k) = mean(Omega);

    R = R + randn(6,n)*sqrt(dt);
    Rdot = [R(2,:);0*R(2,:);R(4,:);0*R(4,:);R(6,:);0*R(6,:)];
    R = R + alpha*Rdot*dt;
end
end
```

By calling the above function from the command window:

```
[t,bigThetaMean,OmegaMean]= triaPathShear(100,0.5);
 plot(bigThetaMean,OmegaMean);
```

we obtained the following graph for the movement of the mean of the estimated probability distribution. From the graph, we recognize that the mean is approaching

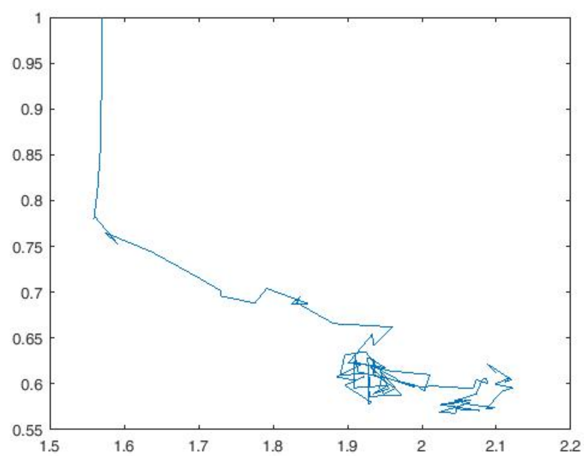

Figure 3.2: Dynamics of the mean of the distribution for a combination of Brownian motion and shear flow.

the area of the maximum of the probability distribution obtained from the random triangles in the first simulation (see Figure 3.1). This confirms as well one of the results of the paper by Quitry et al. [2].

3.4 Conclusion

In this work we introduced the concept of stochastic differential equations (SDEs), showed how to solve particular SDEs using stochastic calculus and discussed the concept of Monte-Carlo simulations for the numerical solution of SDEs in the context of Brownian motion and for an application concerning the Lagrangian advection of triangles following the work of Quitry et al. [2]. Here, we combined Brownian motion with the shear flow and followed the evolution of the mean in the (θ, ω)-plane. We did not have time to study other deterministic motions (for example deterministic dynamics given by the harmonic oscillator through the equation $\dot{x} = y,\ \dot{y} = -x$, but our codes are written in a way that this motion (and many others) can implemented without major effort for further studies.

Bibliography

[1] M. W. Baxter and A. J. O. Rennie, *Financial Calculus: An introduction to derivative pricing*, Cambridge University Press, New York, 1996.

[2] A. de Chaumont Quitry, D. H. Kelley and N. T. Ouellette, *Mechanisms driving shape distortion in two-dimensional flow*, EPL, 94 (2011) 64006.