Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Fisica

Particle methods in option pricing

Relatore:
Prof. Andrea Pascucci

Presentata da:
Vera Compiani

Anno Accademico 2016/2017

To my family

Abstract

Lo scopo di questa tesi è la calibrazione del modello di volatilità locale-stocastico (SLV) usando il metodo delle particelle. Il modello SLV riproduce il prezzo di un asset finanziario descritto da un processo stocastico. Il coefficiente di diffusione o volatilità del processo è costituito da una parte stocastica, la varianza, e da una parte locale chiamata funzione di leva che dipende dal processo stesso e che dà origine ad un'equazione differenziale alle derivate parziali (PDE) non lineare. La funzione di leva deve essere calibrata alla tipica curva che appare nella volatilità implicita dei dati di mercato, il volatility-smile. Per fa ciò si utilizza un metodo computazionale preso dalla fisica: il metodo delle particelle. Esso consiste nell'approssimare la distribuzione di probabilità del processo con una distribuzione empirica costituita da N particelle. Le N particelle consistono in N variabili aleatorie indipendenti e identicamente distribuite che seguono ciascuna l'equazione differenziale stocastica del prezzo con N moti Browniani indipendenti. La funzione di leva dipenderà così da una misura di probabilità casuale e la PDE non-lineare si ridurrà ad una PDE lineare con N gradi di libertà. Il risultato finale è una funzione di leva determinata dall'interazione tra tutte le particelle! La simulazione al computer viene eseguita tramite la tecnica di implementazione in parallelo che accelera i calcoli sfruttando l'architettura grafica della GPU.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The interconnection between physics, mathematics and computer science is now inevitable. In addition, every other scientific discipline can only progress with the use of these main branches of the science. Finance is one of these disciplines that requires the powerful concepts and tools of physics and mathematics. In particular the problem of option pricing. This thesis begins with the study of Bloomberg's articles [17, 18] and the thesis [19] on Stochastic Local Volatility model (SLV). The model describes the stochastic process of the spot price characterized by a stochastic-local volatility or diffusion coefficient. The volatility coefficient has a local part, the leverage function and a stochastic part, the variance, that follows a stochastic process with a Brownian motion correlated to the price process one. The model wants to replicate the implied volatility from the data market that forms a typical curve that is called volatility smile. To do this, the leverage function has to be calibrate to the implied volatility of the market. The problem arises in the partial differential equation (PDE) associated to the stochastic differential equation (SDE) of the SLV model: the Fokker-Planck PDE of the probability density of the process is nonlinear and then the leverage function depends on the probability density itself and it is difficult to compute. To overcome the problem, a physical computation method is taken: the particle method. Particle method borns in physics as deterministic method to simulate physical flows [7]. Its development was permitted by the supercomputer and the new arising of computational science. As stochastic method, it consists in the approximation of the probability distribution of the stochastic process with an empirical distribution of N indistinguible particles. The N particles are N random variables independent and identically distributed whose probability measure converge at the true probability thanks to the theorem of the propagation of chaos. The N particles have N independent Brownian motions and each particle follows the stochastic equation of the spot price. Consequently the Fokker-Planck PDE becomes an N-dimensional linear PDE and the leverage function depends on N interacting particles. Initially we give the basis to understand the main concepts of financial models and the mathematics behind them.

First of all, chapter 2 describes the deterministic particle method used in physics to simulate flows. The basic idea of particle method is that the flow is approximated by point particles whose positions are defined by the Dirac delta function and that follow deterministic equations of motion. We see the particle-particle method (PP) and particle-mesh (PM) method. The first method is discussed in the case of a discrete system of N real particles, the second one consist in putting the particles on a grid to limit the interaction among the nearest ones. An example of particle-mesh method is applied to a 1-dimensional plasma model to give a computational algorithm of the method.

Chapter 3 shows the main results in probability and stochastic theory applied to particle method as stochastic method. Because of in the stochastic framework particles are random variables with discrete distribution, we give the basic definitions of these notions and the fundamental theorems to calculate the expected value of a random variable. Then we give the definition of stochastic process, martingale and Brownian motion that characterize particles that follow stochastic differential equation. Furthermore, we see the main theorems useful to particle method: the Itô formula and the Feynman-Kač representation formula.

In chapter 4 and 5 we apply the stochastic theory to financial market giving the main concepts of option pricing in continuous time. We see the first model of option pricing: the Black-Scholes model (B-S). Other models are formulated to exceed B-S model and to better reproduce the data market: the local volatility model (LV) and the stochastic one (SV). An hybrid formulation of the two models is the stochastic local volatility model (SLV), the model that we want to calibrate with particle method. SLV model describes a stochastic process in 2-dimensions, the price process and the variance process, correlated by two Brownian motions. The stochastic part of the spot process has a local function, the leverage function that controls the weights of the volatility and that has to be calibrated to the implied volatility of the data market. Under assumptions, the SLV model has a unique solution and the transition density is the unique solution of a nonlinear Fokker-Planck PDE. Moreover, given a terminal condition, the option pricing PDE of the SLV model has a unique solution given by the Feynman-Kač formula. These solutions of the SLV model serve to calibrate the leverage function but there is a problem: the PDE is nonlinear with volatility coefficient that depends on stochastic term. To overcome the difficulty, particle method comes to solve it.

Chapter 6 presents a mathematical formulation of particle method as approximation of the probability distribution of a McKean SDE, exactly the equation's type of the SLV model. The approximation is given by a discrete empirical distribution of a fixed large number N of particles each of which follows the SDE of the spot price process. The problem switches from a nonlinear to a linear N-dimensional problem and the leverage function becomes a function that depends on the interaction between the particles. The initial system of N particles consists of N independent and identically distributed random variables with the same distribution and for N that tends to infinity, they are asymptotically independent with their empirical measure that converges toward the true measure. That property is called the propagation of chaos and the theorem of the propagation of chaos proves the convergence of particle method. We can now calibrate the leverage function with particle method: the leverage function is approximated by N interacting particles and their dynamics is discretized by the Euler finite difference scheme in two dimensions.

At last in chapter 7 we give an example of a numerical result of calibration of the leverage function with particle method using parallel computing techniques. The implementation of particles algorithm is done with the copute unified device architecture (CUDA) developed by NVIDIA. First steps is the generation of Brownian motions, second step is to compute in parallel the particles dynamics and then to compute the leverage function. As numerical example we give a set of implied volatilities from the data market that have to be interpolated to compute the local volatility of the LV model. Then the local volatility surface is used to compute the leverage function of the SLV model.

Chapter 2

Particle methods in physics

2.1 Introduction

Particle methods represent an adaptive, efficient, stable and accurate computational method to simulate flow phenomena. At first, they developed in physics for continuum flows such as in astrophysical systems, space and plasma physics, electrons and condensed matter, developing the Vortex Methods (VMs) and Smooth Particle Hydrodynamics (SPH). They are deterministic methods where the physical flow is simulated by N particles. The particles transport the property of the flow ρ and their trajectories and the evolution of the properties carried by the particles are written through Lagrangian formulation and follow ordinary differential equations:

$$\frac{d\mathbf{x}_i^N}{dt} = \mathbf{v}_i^N(t, \mathbf{x}_i^N) = \sum_{j=1}^{N} K(\mathbf{x}_i^N, \mathbf{x}_j^N; \rho_j^N, \rho_j^N) \tag{2.1}$$

$$\frac{d\rho_i^N}{dt} = \sum_{j=1}^{N} F(\mathbf{x}_i^N, \mathbf{x}_j^N; \rho_j^N, \rho_j^N) \tag{2.2}$$

where $(\mathbf{x}_i$,$\mathbf{v}_i)_{i=1}^N$ are the positions and velocities of the N particles, ρ_i are the particle properties such as density, velocity, vorticity and K,F represent the dynamics of the simulated flow.

2.1.1 Point particle approximation

We consider flow $\rho(, t, \mathbf{x}(t))$ such as the density is advected by the velocity vector field $\mathbf{v}$. iI cartesian coordinates the flow follows the advection equation:

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\mathbf{v}) = F(\rho, \nabla\rho, \cdots) \tag{2.3}$$

The right hand side F can take various forms involving derivatives of $\mathbf{v}$ and depends on the physics of the flow system that is being simulated. An example of F is the diffusion term $F(\rho, \nabla f, \Delta\Delta\Delta) = \nabla^2\rho$. The velocity vector field $\mathbf{v}$ can itself be a function of ρ, which leads to non linear transport equations. For simplicity, we consider the case of conservative model in which $F \equiv 0$ and the continuity equation becomes

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\mathbf{v}) = 0 \tag{2.4}$$

In the Lagrangian framework with $\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{dx}{dt}\frac{\partial}{\partial x}$ and assuming that the $\nabla \cdot \mathbf{v} = 0$,the flow equation (2.4) in the 1-dimensional case becomes

$$\frac{dx}{dt} = v(t) \tag{2.5}$$

$$\frac{D\rho}{Dt} = 0 \tag{2.6}$$

Particles method consists in sample the flow $\rho(t, x)$ by individual points or point particles $(x_i^N)_{i=1}^N$, whose locations can be defined with the Dirac δ-function:

$$\rho(t, x) \approx \rho^N(t, x) = \sum_{i=1}^{N} h\,\rho(x_i^N)\delta(x - x_i^N(t)) \tag{2.7}$$

$$\frac{dx_i^N}{dt} = v_i^N \tag{2.8}$$

We denote $\omega_i = \rho(x_i^N)\,h$ the volume of the particle in x_i^N or the weight of the particle. The system of N differential equation (2.7) is solved by a time discretization method sometimes called "*particle pusher*". At every time step dt the position and velocities of the N particles are updated, then the flow $\rho(t, x)$ is calculated from the approximation formula (2.7)

2.1.2 Smooth particle approximation

In the case of continuous systems, the point particles approximation (r2.7) needs to be enhanced in order to recover continuous fields from point samples. We consider the approach of regularizing their support, replacing δ by a smooth cut-off function or *kernel function* W_ϵ which has the same unity mass for every sample points and a small support ϵ:

$$\delta(x) \simeq W_\epsilon(x) = \frac{1}{\epsilon^d} W(\frac{x}{\epsilon}) \tag{2.9}$$

where d is the dimension of the computational space and $\epsilon << 1$ is the range of the cut-off. In most cases, one uses a function with radial symmetry. A typical and often used example is the Gaussian kernel $W(x) = \frac{1}{\sqrt{2\pi}^d} \exp(-\frac{|x|^2}{2})$. The particles approximation formula (2.7) with the regularizing kernel W, becomes a *blob* representation:

$$\rho(t,x) \approx \rho_\epsilon^N(t,x) = \sum_{i=}^{N} h\, \rho(x_i^N) W_\epsilon(x - x_i^N(t)) \tag{2.10}$$

2.2 Particle-Particle Method

Below we described a discrete system of N particle with an action at distance formulation of the force law. This formulation of the forces give an interaction between the N particles and leads to an exactly particle approximation called Particle-Particle Method (PP). We consider a correlated system where exist one-to-one correspondences between physical particles and the computer particles. The initial state is defined by the set of positions and velocities in the 1-dimensional case $(x_i, v_i)_{i=1}^N$ of the particles at the time $t = 0$. The force F_i of the i-particles is the sum of the forces due to the remaining $N - 1$ particles plus any external forces.

$$F(x_i) = \sum_{j \neq i} [F(x_i, x_j) + F^{ext}(x_i)] \tag{2.11}$$

$F_{ij} = F(x_i, x_j)$ is the force of the particle j on the particle i. The potential energy of the particle i, $\phi(x_i)$ is similarly describe by

$$\phi(x_i) = \sum_{j \neq i} [\phi(x_i, x_j) + \phi^{ext}(x_i)] \tag{2.12}$$

The relation between the force and the potential energy field is given by

$$F(x) = -\frac{d\phi(x)}{dx} \qquad \frac{d^2\phi(x)}{dx^2} = -\frac{\rho}{\epsilon_0} \tag{2.13}$$

where the potential field is related to the source of distribution ρ by the field equation that corresponding to the Poisson's equation. The action at distance and force at point formulations are equivalent. The physical system is completed by boundary conditions of the external forces and the volume of the space, the computational box, and by the equation of motion

$$\frac{dx_i}{dt} = v_i \tag{2.14}$$

$$\frac{dv_i}{dt} = \frac{F_i}{m_i} \tag{2.15}$$

At last, we consider the probability density function $\rho(t, x, v)$ in phase space Γ that follow the Liouville's conservation equation:

$$\frac{\partial \rho}{\partial t} + \sum_{i=1}^{N} v_i \frac{\partial \rho}{\partial x_i} + \sum_{i=1}^{N} \frac{F_i}{m_i} \frac{\partial \rho}{\partial v_i} = 0 \tag{2.16}$$

where $\rho(t, x_1, \cdots, x_N; v_1, \cdots, v_N) dx_1 \cdots dx_N dv_1 \cdots dv_N$ is the probability that the system is in the volume $[(x_1, x_1 + dx_1) \cdots]$ of the phase space Γ at time t. A single computer experiment correspond to ρ being a δ-function in phase space. The state of an N particle ensamble at time t can be specified by the exact one-particle distribution function:

$$F(t, x, v) = \sum_{i=1}^{N} \delta(x - x_i(t)) \delta(v - v_i(t)) \tag{2.17}$$

where $F(t, x, v)$ is the force computed on a single particle (x, v) given by the interaction of all the other $N - 1$ particles. Hence, in this case of a flow corresponding to an N-particles system $(x_i, v_i)_{i=1}^{N}$, is exactly approximated by the N particles of the Particle method $(x_i^N, v_i^N)_{i=1}^{N}$. Because of the interaction between the N particles by an action at a distance formulation of the force law, the approximation is called Particle-Particle method (PP). The simulation model of PP method is given by the following steps:

Algorithm of the PP method

1. Compute forces.

```
Clear force accumulators
for $i = 1, \cdots, N$ do
  $F_i := 0$
Accumulate forces
for $i = 1, \cdots, N-1$ do
for $j = 1, \cdots, N$ do
  Find force $F_{ij}$ of particle $j+1$ on particle $i$
    $F_i := F_i + F_{ij}$
    $F_j := F_j - F_{ij}$
```

2. Integrate equations of motion.
   for $i = 1, \cdots, N$ do

```
$v_i^{new} := v_i^{old} + \frac{F_i}{m_i} DT$
$x_i^{new} := x_i^{old} + v_i DT$
```

3. Update time counter.

```
t:=t+DT
```

where the state of the physical system at time t is given by the set of particle positions and velocities $(x_i(t), v_i(t))_{i=1}^{N}$ and at every time step DT the force interaction between the N particle is computed and then used it to move the particles with their equation of motion through a finite difference scheme. Though the PP model is trivial to programming, that scheme is used few time because of the computational costs, generally consisting of at least N^2 operations count.

2.3 Particle-Mesh Method

Besides the Particle-Particle model, there is a method that approximate the forces on a mesh, called Particle-Mesh model (PM), that decreases the computational operations. It is a faster method but generally less accurate then the PP method. Field quantities are represented approximately by values on a regular mesh points and differential operators are replace by finite-difference approximation on the mesh. Potential and forces are obtained by interpolating on the mesh-defined values while the mesh-defined densities are obtained by interpolating back onto the particles. We restrict our attention to a one dimensional equispaced regular grid with mesh-size h. We consider $(x_i^N, v_i^N)_{i=1}^N$ the N-particles positions and velocities and $(x_p^{mesh})_{p=1}^m$ the m-points of the mesh and $\rho(x)$ the density of the particles to interpolate on the mesh points. The step of PM method are the following:

1. **P → M** *Interpolation onto the mesh.*
   The first step is to interpolate the quantities $\rho(x_i^N)$ with $i = 1, \cdots, N$ from the scattered particle locations x_i^N onto the grid x_p^{mesh}. The interpolation is done through the interpolation function $W(x - x_i^N)$. This step is also called *charge assignment* to the grid: the grid nodes becomes particles x_p^{mesh} with $p = 1, \cdots, m$ with grid spacing h.

   $$\rho(x_p^{mesh}) = \sum_{i=1}^{N} q_i \, W(x_p^{mesh} - x_i^N) \qquad \text{with } p = 1, \cdots, m \tag{2.18}$$

   where x_i^N is the location of the particle i at the time t_n. We denote $\rho_p = \rho(x_p^{mesh})$. The properties of the interpolant are the moment properties:

   $$\int W(y)dy = 1 \; ; \qquad \int y^\alpha W(y)dy = 0 \; i1 \leq |\alpha| \leq m - 1 \tag{2.19}$$

   The charge assignment is done at every time step DT.

2. **M → M** *Compute the field equations onto the mesh.*
   We evaluate the operators onto the mesh through a finite difference scheme (FD) with grid spacing h.

   $$F_p = (\Delta^{h,FD} \rho)_p \tag{2.20}$$

3. **M → P** *Interpolation onto the particles.*
   We return onto the particles interpolating back from the mesh to the particles through the interpolation function W, but now the sum is over the m grid point:

   $$F(x_i^N) = \sum_{p=1}^{m} F_p \, W(x_i^N - x_p^{mesh}) \qquad \text{with } i = 1, \cdots, N \tag{2.21}$$

4. $t_n \rightarrow t_{n+1}$ *Time integration.*
   Move the particles at the time-step t_{n+1} solving the equations of motion

through a finite difference scheme such as the leapfrog method:

$$\begin{cases} \frac{dx_i^N}{dt} = v_i^N \\ \frac{dv_i^N}{dt} = F(x_i^N) \end{cases} \quad \rightarrow \quad \begin{cases} v_i^{n+\frac{1}{2}} = v_i^{n-\frac{1}{2}} + F(x_i^n)DT \\ x_i^{n+1} = x_i^n + v_i^{n+\frac{1}{2}}DT \end{cases} \tag{2.22}$$

where x_i^n is the position of the particle i at time t_n.

2.3.1 Particle mesh to one-dimensional plasma model

A very illustrative example of the Particle Mesh method is its application to simulation of electrostatic waves in a collisionless plasma. Now we have a continuous system that we approximate with N particles. The equations describing the idealized plasma in 1-dimensional model with total charge density ρ:

$$\rho(x) = q \int f(t,x,v)dv \tag{2.23}$$

are the Vlasov equation describing the electron distribution $f(t,x,v)$

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} + \frac{F}{m}\frac{\partial f}{\partial v} = 0 \tag{2.24}$$

and the electrostatic force-at-a-point F, the potential ϕ, electric field E given by:

$$F = qE = -q\frac{d\phi}{dx}\ , \qquad \frac{d^2\phi}{d^2x} = -\frac{\rho}{\epsilon_0} \tag{2.25}$$

were $q(= -e)$ is the electron charge.

Discretization of the model

To simulate the one dimensional plasma model we have to approximate the mathematical equations (2.24)-(2.25) with N particles with the Particle-Mesh (PM) method. Following the steps of the previous section, the first stage is to discretize the equations taking a sample of points $(x_i^N, v_i^N)_{i=1}^N$ and their orbits through the $x - v$ phase space are given by:

$$\frac{dx_i^N}{dt} = v_i^N \tag{2.26}$$

$$\frac{dv_i^N}{dt} = \frac{F(x_i^N)}{M} \tag{2.27}$$

where $M = N_s m_e$ and m_e is the electron mass and $N_s = \int_i f dx dv$. We construct a grid of $p = 1, \cdots, m$ points onto the the ragion of space (the range of x values) spanned by the simulation model. This region is called the computational box. Boundary conditions are given by the potentials $\phi(x)$ at the surfaces $x = 0$ and $x = L$ of the box and the charge density distribution within the box. We divided the box in regular intervals h. The points at which values are recorded are the mesh or grid-points x_p. Mesh points lie at the centers of cells of width h. If the origin is taken at mesh-point 0, the position of mesh-point p is at $x_p = ph$. An integral number of cell width h fit into the computational box length L. For periodic boundary conditions, the number of cells is equal to the number of grid points m. $L = mh$.

1. **P → M** *Charge assignment to the mesh*
   The operation to obtain the charge density at mesh-points $\rho(x_p)$ with $p = 1, \cdots, m$ from the distribution of the N particles, is called *charge assignment.* We used an interpolated function by the simplest scheme, the *nearest grid point scheme* (NGP), in which the charge density at mesh-point p is given by the total charge in the cell surrounding mesh-point p divided by the cell volume:

$$\rho_p = \rho(x_p) = \frac{1}{h} \sum_{\text{particles } i \text{ in cell } p} N_s q \qquad p = 1, \cdots, m \tag{2.28}$$

   The explicit form of the charge assignment function or interpolation function W is:

$$W(x) = \begin{cases} 1 & |x| < \frac{h}{2} \vee x = \frac{h}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

$$\rho(x_p) = \frac{qN_s}{h} \sum_{i=1}^{N} W(x_p - x_i^N) \qquad p = 1, \cdots, m \tag{2.30}$$

2. **M → M** *Solve the field equations on the mesh*
   To compute the field equations onto the grid-points, we used a finite differences scheme on the mesh at time t_n as follows:

$$\frac{d\phi}{dx}|_{x_p} \simeq \frac{\phi^n_{p-+} - \phi^n_{p-1}}{2h} \tag{2.31}$$

$$\frac{d^2\phi}{dx^2}|_{x_p} \simeq \frac{\phi^n_{p+1} - 2\phi^n_p + \phi^n_{p-1}}{h^2} \tag{2.32}$$

where n indicate the time t_n. The resulting finite-difference field equations are:

$$\frac{\phi_{p+1}^n - 2\phi_p^n + \phi_{p-1}^n}{h^2} = -\frac{\rho_p^n}{\epsilon_0} \tag{2.33}$$

$$E_p^n = \frac{\phi_{p-1}^n - \phi_{p+1}^n}{2h} \tag{2.34}$$

where we used the compact notations $\phi_p \equiv \phi(x_p)$.

3. $\mathbf{M} \rightarrow \mathbf{P}$ *Force interpolation on the particles*
   To interpolate the forces from the mesh to the particles we used the NGP interpolation function (2.29) in a similar way to charge assignment. The force field in a cell is given by the value at the mesh-point at the center of that cell. Thus the force on some particle i at x_i^N is:

$$F(x_i^N) = N_s q E(x_p) \qquad x_p - \frac{h}{2} < x_i^N \leq x_p + \frac{h}{2} \tag{2.35}$$

   This also may be rewritten in terms of the NGP charge assignment function:

$$F_i^N = F(x_i^N) = N_s q \sum_{p=0}^{m-1} W(x_i^N - x_p) E_p \tag{2.36}$$

   Note that the sum is over mesh-points, whereas in charge assignment the sum is over particles.

4. $t_n \rightarrow t_{n+1}$ *Move the particles*
   The final stage of discretization of the Vaslov equation (2.24)is to replace the continuous time variables in eq.(2.26) by a discrete set of time-levels separated in time by a small time interval, the time-step DT. One of the most used scheme is the finite-difference approximation leapfrog scheme. Positions and fields are defined at integral time-levels n and velocities are defined at half-integral time-levels $n \pm \frac{1}{2}$. The leapfrog finite-difference approximations of eq.(2.26) are:

$$\frac{v_i^{n+\frac{1}{2}} - v_i^{n-\frac{1}{2}}}{DT} = \frac{F(x_i^n)}{N_s m_e} \tag{2.37}$$

$$\frac{x_i^{n+1} - x_i^n}{DT} = v_i^{n+\frac{1}{2}} \tag{2.38}$$

Chapter 3

Probability theory and Stochastic processes

In the previous chapter particle method has applied in deterministic phenomena and the particles follow deterministic laws such as the Newton's laws. To the aim of the thesis, particle method has to be applied at *stochastic processes*, where the particles are *random variables* that follow discrete distribution. In the following sections we present the basic ideas of probability and stochastic processes with the introductions of the main formulas, Itô formula and Feynman-Kač formula, useful to the resolution of a financial model simulated by the particle method.

3.1 Probability theory

The main concepts in probability theory are those of random variable r.v, distribution of a random variable, expectation of a r.v and conditional expectation of a r.v.. The space considered below is the probability space $(\Omega, \mathcal{F}, P)$ with Ω the sample space, $\mathcal{F}$ the σ-algebra of the events $A \in \mathcal{F}$ and P the probability measure such that $P(\Omega) = 1$.

Definition 1. (Random variable) A *random variable r.v.* on the probability space $(\Omega, \mathcal{F}, P)$ is a measurable function X from Ω with values in $\mathbb{R}^d$:

$$X : \Omega \to \mathbb{R}^d \qquad \text{such that } (X \in H) \in \mathcal{F} \quad \forall H \in \mathscr{B} \tag{3.1}$$

$$(X \in H) := X^{-1}(H) = \{\omega \in \Omega | X(\omega) \in H\}, \quad \forall H \in \mathscr{B} \tag{3.2}$$

In other words, a random variable is a (measurable) *function on a probability space*, while a variable is a *number* on $\mathbb{R}$, and for every $H \in \mathscr{B}$, $(X \in$

$H) \in \mathcal{F}$ is an event. In this case we write $X \in m\mathcal{F}$ and we call that X is $\mathcal{F}$-measurable. At every random variable on the probability space $(\Omega, \mathcal{F}, P)$, is naturally associated the distribution defined by

Definition 2. (Distribution μ_X of X) A *distribution* is a measure of probability on $(\mathbb{R}^N, \mathscr{B})$. *The distribution* μ_X *of* X with notation $X \sim \mu_X$ is such that

$$\mu_X : \mathscr{B} \to [0,1] \qquad \text{such that} \tag{3.3}$$

$$\mu_X(H) := P(X \in H) = P(X^{-1}(H)), \qquad H \in \mathscr{B} \tag{3.4}$$

$P(X \in H)$ is the probability that the r.v X belongs to the Borel set H.

For example, the positions of a particle in $\mathbb{R}^d$ are not observable with precision because of the uncertainty principle. Therefore, $H \in \mathscr{B}$ is the event in which "the particle is in H" and $\mu_X(H)$ is the "probability that the particles is in H". The stochastic form of particle method use the discrete distribution: linear combinations of δ-Dirac.

Definition 3. (Discrete distribution) A *discrete distribution* is a distribution of the form

$$\mu_X(H) := \sum_{n=1}^{\infty} p_n \delta_{x_n}(H), \qquad H \in \mathscr{B} \tag{3.5}$$

where (x_n) is a succession of points in $\mathbb{R}^d$ and $(p_n)_{n\in\mathbb{N}}$ a succession of real number such that $\sum_{n=1}^{\infty} p_n = 1$ and $p_n \geq 0, \quad n \in \mathbb{N}$.

Naturally associated to the discrete distribution μ_X is a function $\bar{\mu}_X : \mathbb{R}^d \to [0,1]$ called *distribution function* of μ_X, (in the case of absolutely continuous distribution, it is the density function), where $\bar{\mu}_X(x) = \mu_X(\{x\}), \ x \in \mathbb{R}^d$. Explicitly

$$\bar{\mu}_X(x) = \begin{cases} p_n & \text{if } x = x_n \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

and

$$P(X = x_n) = \bar{\mu}_X(x_n) = p_n \tag{3.7}$$

The expected value of a random variable is the integral of the r.v respect with the probability measure.

Definition 4. (Expectation of X) The *expectation of a r.v X* with $X : \Omega \to \mathbb{R}^d$ is defined as the integral of X over the space $(\Omega, \mathcal{F}, P)$:

$$\mathbb{E}[X] := \int_\Omega X dP = \int_\Omega X(\omega) P(d\omega) \tag{3.8}$$

The following theorem gives the fundamental tool to calculated the expectation a random variable.

Theorem 1. ***Fundamental theorem***
Let be X a r.v on $(\Omega, \mathcal{F}, P)$ and distribution $X \sim \mu_X$, f a function $\mathscr{B}$-measurable, $f \in m\mathscr{B}$ such that

$$X : \Omega \to \mathbb{R}^d \qquad f : \mathbb{R}^d \to \mathbb{R}^N \tag{3.9}$$

Then $f \circ X \in L^1(\Omega, P)$ if and only if $f \in L^1(\mathbb{R}^d, \mu_X)$ and

$$\mathbb{E}[f(X)] = \int_{\mathbb{R}^d} f d\mu_X \tag{3.10}$$

if $\mu_x = \sum_{k=1}^{\infty} p_k \delta_{x_k}$ is a discrete distribution then

$$\mathbb{E}[f(X)] = \sum_{k=1}^{\infty} f(x_k) p_k \tag{3.11}$$

while if μ_X is absolutely continuous distribution with density function γ_X then

$$\mathbb{E}[f(X)] = \int_{\mathbb{R}^d} f(x)\gamma_X(x)dx \tag{3.12}$$

with $\gamma_X : \mathbb{R}^d \to [0, +\infty]$ such that it is $\mathscr{B}$-measurable, non-negative and $\int_{\mathbb{R}^d} \gamma_X(x)dx = 1$.

3.2 Stochastic processes

Given the basic definitions of probability theory, we continue with the introduction of stochastic processes that characterize continuous-time financial processes. In the next section we briefly see the main useful results of the stochastic theory such as the Itô's formula, the Feynman-Kac formula and a particular stochastic physical motion: the Brownian motion. Here follow the definition of discrete and continuous stochastic process.

Definition 5. (Discrete stochastic process) A discrete stochastic process in $\mathbb{R}^N$ is a collection $X = (X_n)_{n\in\mathbb{N}_0}$ of random variables define on a probability space $(\Omega, \mathcal{F}, P)$ with values in $\mathbb{R}^N$:

$$X_n : \Omega \to \mathbb{R}^N, \qquad n \in \mathbb{N}_0 \tag{3.13}$$

The collection of σ-algebras $(\mathcal{F}_n^X)_{n\in\mathbb{N}_0}$, defined by

$$\mathcal{F}_n^X = \sigma(X_k, 0 \leq k \leq n),$$

is called *natural filtration for X*. The **process X is adapted** to the filtration $(\mathcal{F}_n)$ if X_n is $\mathcal{F}_n$-measurable or equivalently $\mathcal{F}_n^X \subseteq \mathcal{F}_n$, for every $n \in \mathbb{N}_0$. We say that X is a *integrable process* if $X_n \in L^1(\Omega, P)$ for every $n \in \mathbb{N}_0$.

Definition 6. (Stochastic process) A measurable stochastic process in $\mathbb{R}^N$ is a collection $(X_t)_{t \in I}$ of random variables with values in $\mathbb{R}^N$ define on a probability space $(\Omega, \mathcal{F}, P)$ and Ia real interval of the form $[0, T]$ or $\mathbb{R}_{\geq 0}$, such the map

$$X : I \times \Omega \to \mathbb{R}^N, \qquad X(t, \omega) = X_t(\omega) \tag{3.14}$$

is a measurable with respect to the product σ-algebra $\mathscr{B} \otimes \mathcal{F}$. We say that X is integrable if $X_t \in L^1(\Omega, P)$ for every $t \in I$.

The concept of stochastic process extends that of deterministic function

$$f : I \to \mathbb{R}^N \tag{3.15}$$

F associates t to a variable the number $f(t) \in \mathbb{R}^N$; similarly X, the stochastic process associates t to the random variable $X_t \in \mathbb{R}^N$. Hence a stochastic process describes a random phenomenon that evolves in time.For example, X_t can be the price of a risky asset at time t or X_t in $\mathbb{R}^3$ the position of a particle in space at time t. The function $f : I \to \mathbb{R}^N$ can be think as a curve or trajectory in $\mathbb{R}^N$ such that

$$\gamma = \{f(t) | t \in I\} \tag{3.16}$$

the curve for the particle or

$$\gamma_\omega = \{X_t(\omega) | t \in I\}, \qquad \omega \in \Omega \tag{3.17}$$

the curve for the stochastic process. In this case a different path correspond to any $\omega \in \Omega$.

A fundamental notion to describe stochastic process is that of continuous-time *martingale*:

Definition 7. (Martingale) Let M be an integrable adapted stochastic process on the filtered probability space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t))$. We say that M is a:

i *martingale* with respect to if $(\mathcal{F}_t)$ and to the measure P if

$$M_s = \mathbb{E}[M_t | \mathcal{F}_s], \qquad \text{for every } 0 \leq s \leq t; \tag{3.18}$$

ii *super-martingale* if

$$M_s \geq \mathbb{E}[M_t|\mathcal{F}_s], \qquad \text{for every } 0 \leq s \leq t; \tag{3.19}$$

iii *sub-martingale* if

$$M_s \leq \mathbb{E}[M_t|\mathcal{F}_s], \qquad \text{for every } 0 \leq s \leq t; \tag{3.20}$$

The *mean of a martingale M is constant in time*:

$$\mathbb{E}[M_t| = \mathbb{E}[\mathbb{E}[M_t|\mathcal{F}_0]] = \mathbb{E}[M_0], \qquad t \geq 0 \tag{3.21}$$

3.3 Brownian motion

We introduce the definition of Brownian motion and its fundamental properties.

Definition 8. (Real Brownian motion) Let $(\Omega, \mathcal{F}, P)$ be a filtered probability space. A real Brownian motion is a stochastic process $W = (W_t)_{t\geq 0}$ in $\mathbb{R}$ such that
i) $W_0 = 0$ a.s;
ii)W is $\mathcal{F}_t$-adapted and continuous;
iii) for $t > s \geq 0$, the random variable $W_t - W_s$ ha normal distribution $\mathcal{N}_{0,t-s}$ and is independent of $\mathcal{F}_s$.

from the definition of Brownian motion we have that for every t, W_t follow the law

$$W_t \sim \mathcal{N}_{0,t} \tag{3.22}$$

since $W_t = W_t - W_0$ a.s.
Brownian motion was originally created as a probabilistic model for the motion of a particle and in finance it is used in the model of a risky asset. Furthermore, there exist a connection between Brownian motion and the heat equation in two variables:

$$L^* = \frac{1}{2}\partial_{xx} + \partial_t \qquad (t,x) \in \mathbb{R}^2 \tag{3.23}$$

Let be $W^{t,x}$ the stochastic process defined by $W_T^{t,x} = x + W_T - W_t$ for fixed $x \in \mathbb{R}$ and $t \geq 0$ with $T \geq t$, called Brownian motion starting at time t from x. It has the distribution $W_T^{t,x} \sim \mathcal{N}_{x,T-t}$ for $T \geq t$ and density the fundamental solution of the adjoin heat equation

$$\Gamma^*(t,x;T,y) = \frac{1}{\sqrt{2\pi(T-t)}} \exp\big(-\frac{(x-y)^2}{2(T-t)}\big) \tag{3.24}$$

the function $\Gamma^* = \Gamma^*(t,x;T,\cdot)$ is called transition density of the Brownian motion from the inital point (t,x) to the final time T and is the fundamental of L^*. Consequently for every $\phi \in C_b(\mathbb{R})$ the Cauchy problem

$$\begin{cases} L^*u(t,x) = 0 & (t,x) \in]0,T[\times\mathbb{R}, \\ u(T,x) = \phi(x) & x \in \mathbb{R} \end{cases} \tag{3.25}$$

has a classical solution

$$u(t,x) = \int_{\mathbb{R}} \Gamma^*(t,x;T,y)\phi(y)dy \qquad t < T,\ x \in \mathbb{R} \tag{3.26}$$

with the probabilistic representation

$$u(t,x) = \mathbb{E}[\phi(W_T^{t,x})], \qquad x \in \mathbb{R},\, t \in [0,T] \tag{3.27}$$

and the conditional expectation of a Brownian motion with Markov property:

$$\mathbb{E}[\phi(W_T)|\mathcal{F}_t] = u(t, W:t), \qquad T \geq t \tag{3.28}$$

where u is the solution in (3.26) of the Cauchy problem (3.25) and thanks to the probabilistic formula (3.27), a numerical resolution is easily to implement via Monte Carlo methods.

3.4 Itô's calculus

Let W be a real Brownian motion on filtered probability space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t))$

Definition 9. (Itô process) An Itô process is a stochastic process X of the form

$$X_t = X_0 + \int_0^t \mu_s ds + \int_0^t \sigma_s dW_s, \qquad t \in [0,T] \tag{3.29}$$

where X_0 is a $\mathcal{F}_0$-measurable random variable, $\mu \in \mathbb{L}^1_{loc}$ and $\mathbb{L}^2_{loc}$. in the "differential form"

$$dX_t = \mu_t dt + \sigma_t dW_t \tag{3.30}$$

The process μ and σ are called *drift* and *diffusion* coefficients respectively. μ gives the "direction" to the process X and σ is a local martingale giving a "stochastic contribution" to the evolution of X. dX_t is called *stochastic differential.*

Theorem 2. ***(Itô formula)*** *Let X be an Itô process $dX_t = \mu_t dt + \sigma_t dW_t$ and $f = f(t,x) \in C^{1,2}(\mathbb{R}^2)$. Then the stochastic process*

$$Y_t = f(t, X_t) \tag{3.31}$$

is an Itô process and we have

$$df(t,X_t) = \partial_t f(t,X_t)dt + \partial_x f(t,X_t)dX_t + \frac{1}{2}\partial_{xx} f(t,X_t)d\big\langle X_t \big\rangle \tag{3.32}$$

Since $d\big\langle X_t \big\rangle = \sigma_t^2 dt$, formula (5.23) can be written in the form

$$df(t,X_t) = \big(\partial_t + \mu_t \partial_x + \frac{1}{2}\sigma_t^2 \partial_{xx}\big) f(t,X_t)dt + \sigma_t \partial_x f(t,X_t)dW_t \tag{3.33}$$

A geometric Brownian motion is a solution of the stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{3.34}$$

where μ, $\sigma \in \mathbb{R}$. The process S can be written in the form $S_t = f(t, W_t)$ and applying the Itô formula we get:

$$dS_t = \big(\partial_t + \frac{1}{2}\partial_{xx}\big) f(t,W_t)dt + \partial_x f(t,W_t)dW_t \tag{3.35}$$

and imposing (3.34) we have that for $(t,x) \in \mathbb{R}_{>0} \times \mathbb{R}$:

$$\begin{cases} \partial_x f(t,x) = \sigma f(t,x) \\ f(t,x) + \frac{1}{2}\partial_{xx} f(t,x) = \mu f(t,x) \end{cases} \tag{3.36}$$

The solution of equation(3.34) is the following:

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \tag{3.37}$$

If $\sigma = 0$ the solution (3.37) is deterministic $S_t = S_0 e^{\mu t}$.

Theorem 3. ***(Multi-dimensional Itô formula)*** *Let X be an N-dimensional Itô process of the form*

$$dX_t^i = \mu_t^i dt + \sum_{j=1}^{d} \sigma_t^{ij} dW_t^j, \qquad i = 1, \cdots, N \tag{3.38}$$

and $f = f(t,x) \in C^{1,2}(\mathbb{R} \times \mathbb{R}^N)$. *Then*

$$df = \partial_t f dt + \nabla f \cdot dX_t + \frac{1}{2}\sum_{i,j=1}^{N} \partial_{x_i x_j} f d\big\langle X^i, X^j \big\rangle_t \tag{3.39}$$

with $f = f(t, X_t)$ *and* $\nabla f = (\partial_{x_1} f, \cdots, \partial_{x_N} f)$.

In compact form, if we put $\mathcal{C} = \sigma\sigma^*$ and $\big\langle X \big\rangle_t = \mathcal{C}_t dt$, $d\big\langle X^i, Y^j \big\rangle_t = dX_t^i \cdot dY_t^j$ with $X, Y \in \mathbb{R}^N$ two Itô processes and using the following rules

$$\begin{cases} dt \cdot dt = dt \cdot dW^i = dW^i \cdot dt = 0 \\ dW^i \cdot dW^j = \delta_{ij} dt \end{cases} \tag{3.40}$$

multidimensional Itô formula can be written as:

$$df = \big(\frac{1}{2}\sum_{i,j=1}^{N} \mathcal{C}_t^{ij} \partial_{x_i x_j} + \sum_{i=1}^{N} \mu_t^i \partial_{x_i} + \partial_t\big) f dt + \nabla f \cdot \sigma_t dW_t \tag{3.41}$$

3.5 Feynman-Kač representation formulas

There exist a deep connection between the stochastic differential equation SDEs and the parabolic partial differential equation PDEs where the trait of union is the Itô formula. Let be $Z \in \mathbb{R}^N$ and the drift b and the diffusion coefficient σ two measurable functions

$$b = b(t,x) : [0,T] \times \mathbb{R}^N \to \mathbb{R}^N, \qquad \sigma = \sigma(t,x) : [0,T] \times \mathbb{R}^N \to \mathbb{R}^{N \times d} \tag{3.42}$$

We consider a SDE in $\mathbb{R}^N$ with bounded continuous coefficients and suppose there exists a solution X to the SDE relative to a d-dimensional Brownian motion

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t \tag{3.43}$$

$\mathcal{S}_t$ is the bounded domain in $\mathbb{R}^N$ and we assume that:
i) the coefficients are measurable and have at most linear growth in x;
ii) for every $(t,x) \in \mathcal{S}_t$ there exists a solution $X^{t,x}$ of the SDE (3.43) such that $X_t^{t,x} = x$ relative to a d-dimensional Brownian motion W on the space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t))$.

applying the Itô formula for every $f \in \mathcal{C}_0^2(\mathbb{R}^N)$ we have

$$df(X_t) = \mathcal{A}_t f(X_t)dt + \nabla f(X_t) \cdot \sigma(t, X_t)dW_t \tag{3.44}$$

with $(c_{ij}) = \sigma\sigma^*$ and where

$$\mathcal{A}_t f(x) := \frac{1}{2} \sum_{i,j=1}^{N} c_{ij}(t,x)\partial_{x_i x_j} f(x) + \sum_{j=1}^{N} b_j(t,x)\partial_{x_j} f(x) \tag{3.45}$$

The operator $\mathcal{A}_t$ is called the *characteristic operator* of the SDE (3.43). The Feynman-Kač theorem gives a representation of the solution u of the Cauchy-Dirichlet problems relative to (3.45) in terms of expectation of $u(t, X_t)$. The Cauchy problem is the following:

$$\begin{cases} \mathcal{A}u - au + \partial_t u = f, & \text{in } \mathcal{S}_t :=]0, T[\times \mathbb{R}^N \\ u(T, \cdot) = \phi \end{cases} \tag{3.46}$$

where f, a, ϕ are given functions.

Theorem 4. ***(Feynman-Kač formula)*** *Let $u \in \mathcal{C}^2(\mathcal{S}_T) \cap \mathcal{C}(\bar{\mathcal{S}}_t)$ be a solution of the Cauchy problem (3.46) where $a \in \mathcal{C}(\mathcal{S}_T)$ is such that $a_0 = \inf a > -\infty$. Assume that i), ii) and at least one of the following conditions are in force:*
1) there exist two positive constants M,p, such that

$$|u(t,x)| + |f(t,x)| \le M(1 + |x|^p), \qquad (t,x) \in \mathcal{S}_T; \tag{3.47}$$

2) the matrix σ is bounded and there exist two positive constants M and α, with α small enough, such that

$$|u(t,x)| + |f(t,x)| \le M e^{\alpha|x|^2}, \qquad (t,x) \in \mathcal{S}_T; \tag{3.48}$$

Then for every $(t,x) \in \mathcal{S}_T$, we have the representation formula:

$$u(t,x) = \mathbb{E}\left[e^{-\int_t^T a(s,X_s)ds}\phi(X_T) - \int_t^T e^{-\int_t^s a(r,X_r)dr} f(s,X_s)ds\right] \tag{3.49}$$

where for the sake of simplicity $X = X^{t,x}$

The Feynman-Kač representation formula allows to connect the solution of the PDE of the Cauchy problem to the transition density of the SDE. Indeed in the case of the transition density of Brownian motion, if the operator $\mathcal{A} + \partial_t$ has a fundamental solution $\Gamma(t, x; T, y)$ then, for every $\phi \in \mathcal{C}_b(\mathbb{R}^N)$ the function

$$u(t,x) = \int_{\mathbb{R}^N} \phi(y)\Gamma(t,x;T,y)dy \tag{3.50}$$

is the classical solution of the Cauchy problem (3.46) with $a = f = 0$ and by the Feynman-Kač formula, we have that

$$\mathbb{E}[\phi(X_T^{t,x})] = \int_{\mathbb{R}^N} \phi(y)\Gamma(t,x;T,y)dy \tag{3.51}$$

by the arbitrariness of ϕ, for fixed $x \in \mathbb{R}^N$ and $t < T$, the function

$$y \mapsto \Gamma(t,x;T,y) \tag{3.52}$$

is the density of the random variable $X_T^{t,x}$. We say that Γ is the *transition density* of the SDE (3.43) and follow the deep connection between PDEs and SDEs:

Theorem 5. *If there exists the fundamental solution of the differential operator $\mathcal{A} + \partial_t$ with*

$$\mathcal{A} = \frac{1}{2}\sum_{i,j=1}^{N} c_{ij}\partial_{x_i x_j} + \sum_{j=1}^{N} b_j \partial_{x_j} \tag{3.53}$$

then it is equal to the transition density of the SDE

$$dX_t = b(t,X_t)dt + \sigma(t,X_t)dW_t \tag{3.54}$$

Theorem 6. ***(Fokker-Planck equation)**, Suppose that X_t follows the SDE (3.43). Given $X_0 = x$, the Fokker-Planck equation or Kolmogorov forward equation describes the evolution of the transition probability density $p(t,X_t)$ as*

$$\partial_t p = \frac{1}{2}\sum_{i,j=1}^{N} \partial_{x_i x_j}[c_{ij}p] - \sum_{j=1}^{N} \partial_{x_j}[b_j p] \tag{3.55}$$

with initial condition $p(0,X_0) = \prod_{i=1}^{N} \delta(X_0^i - x^i)$.

3.6 Change of measure

We consider a d-dimensional Brownian motion $(W_t)_{t\in[0,T]}$ on the space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t))$ and $\lambda \in L^2_{loc}$ a d-dimensional process such that

$$Z_t^\lambda = e^{-\int_0^t \lambda_s \cdot dW_s - \frac{1}{2}\int_0^t |\lambda_s|^2 ds} \tag{3.56}$$

Z_t^λ *is the exponential martingale associated to* λ and by the Itô formula we have $dZ_t^\lambda = -Z_t^\lambda \lambda_t \cdot dW_t$. We suppose that Z^λ is a martingale and we define the measure Q on $(\Omega, \mathcal{F})$ by:

$$\frac{dQ}{dP} = Z_T^\lambda \qquad \text{i.e} \tag{3.57}$$

The derive is to be interpreted as the stochastic integral $Q(F) = \int_F Z_T^\lambda dP, \quad F \in \mathcal{F}$. The following Girsanov's theorem shows that is possible to substitute arbitrarily the drift of an Itô process by modifying appropriately the probability measure and Brownian motion.

Theorem 7. ***(Girsanov's theorem)*** *Let* Z^λ *be the exponential martingale associated to the process* $\lambda \in L^2_{loc}$. *We assume that* Z^λ *is a P-martingale and we consider the measure Q defined by*

$$\frac{dQ}{dP} = Z_T^\lambda \tag{3.58}$$

Then the process

$$W_t^\lambda = W_t + \int_0^t \lambda_s ds, \qquad t \in [0, T] \tag{3.59}$$

is a Brownian motion on $(\Omega, \mathcal{F}, Q, (\mathcal{F}_t))$.

A change of measure from P to an equivalent measure Q, modifies the coefficient of an Itô process in the following way:

Theorem 8. ***(Change of drift)*** *Let Q be a probability measure equivalent to P. The Radon-Nikodyn derivative of Q with respect to P is an exponential martingale*

$$\frac{dQ}{dP}|_{\mathcal{F}_t^W} = Z_t^\lambda, \qquad dZ_t^\lambda = -Z_t^\lambda \lambda_t \cdot dW_t \tag{3.60}$$

with $\lambda \in L^2_{loc}$ *and the process* W^λ *define by*

$$dW_t = dW_t^\lambda - \lambda_t dt \tag{3.61}$$

is a Brownian motion on $(\Omega, \mathcal{F}, Q, (\mathcal{F}_t))$.

Hence, a change of measure only affects the drift coefficient of the process X while the volatility or diffusion coefficient does not vary. For example if X is an N-dimensional Itô process

$$dX_t = b_t dt + \sigma_t dW_t \tag{3.62}$$

The Q-dynamics of X is given by

$$dX_t = (b_t - \sigma_t \lambda_t) dt + \sigma_t dW_t^\lambda \tag{3.63}$$

Chapter 4

Option pricing models

4.1 An introduction to financial derivatives

We want to apply the stochastic theory to option pricing in financial market where the SLV model of the thesis is located. A particular financial market is that of *financial derivative.* Basics financial derivatives are the *options.* An options is a contracts between two parties, the option seller and the option buyer whereby the option buyer is granted the right (but not the obligation) to exercise the option at some moment in the future, secured by the option seller. Options depend on the prices of other basic securities or *underlying assets* such as stocks or bonds. Stock are shares in the net asset value not bearing fixed interest rate.

A bond is a certificate issued by a government or a public company promising to repay borrowed money at a fixed rate of interest at a specific time. The bond is assume to be a risk-free asset B_t whose final value is given by:

$$B_T = B_0 e^{rT} \tag{4.1}$$

with a fixed risk-free interest rate r and B_0 the initial value of the bond. So, to obtain the final wealth B at time T it is necessary to invest at time 0 the amount Be^{-rT} called the *discounted value* of B.

A class of options are *call options* and *put option.* Respectively, call option (put) grants its holder the right (not the obligation) to buy (to sell) the option's underlying asset at some moment in the future at a fixed predetermined price. The predetermined price is referred to as the *strike price*, and the future date is called the *expiration date.* The option gives the holder the right to do something but he is not obliged to exercise this right. The investor need to pay an option's price to a second party at the initial date when the contract is entered into. Another class of options are the American options that may be exercised at any time on or before the expiry date. Hence, options are characterized by the *underlying asset*, the *exercise price* K or strike-price and the *maturity* time T. Let be S_T the stock price at the terminal date T and it is not known at time 0. Hence S_T give rise to uncertainty. The payoff g at maturity T from a *European Call option* with strike K is given by the formula

$$g(S_T) = (S_T - K)^+ := \max\{S_T - K, 0\} \tag{4.2}$$

where at time T we have two possibilities

$$g(S_T) = \begin{cases} S_T - K & \text{if} \quad S_T > K \quad \text{(option is exercised)} \\ 0 & \text{if} \quad S_T \leq K \quad \text{(option is abandoned)} \end{cases} \tag{4.3}$$

If at the maturity date the stock price S_T is lower than the strike price K, the option is not profitable and the payoff is zero. On the other hand, if the stock price is greater than the strike price, the holder should exercise his right to buy the underlying stock at the strike price K and the profit is given by buying the underlying asset at price K and then selling at the market price S_T. So, at the spot market, the holder should realize a net profit $S_T - K$. Instead the *put option* gives the holder the right to sell the underlying asset by a certain date for a prespecific price. The payoff of a European put option is:

$$h(S_T) = (K - S_T)^+ := \max\{K - S_T, 0\} \tag{4.4}$$

or explicitly

$$h(S_T) = \begin{cases} 0 & \text{if} \quad S_T \geq K \quad \text{(option is abandoned)} \\ K - S_T & \text{if} \quad S_T < K \quad \text{(option is exercised)} \end{cases} \tag{4.5}$$

At the present time 0 the value of the underlying asset at the maturity time T is not known but the value of the option that depend on it has to be fixed in order to create the contract. This arise the so called *option pricing* problem: the determination of the fair price of the option, the *premium* that the buyer of the option has to pay at the initial time. A connected problem to the pricing is the *hedging* problem: a bank that sells a derivative has to find an investment strategy that can replicate the payoff at maturity in order to hedge the risk of a potentially unlimited loss.

Going on with the pricing problem, we have to illustrate the following definition considering two moments in time, the initial date 0 and the maturity date T in a two-state $[0,T]$ option model. We assume tath there exists a bond with risk-free rate r and initial value $B_0 = 1$ and there is a risky asset S whose final value depends on some random event that can assume only two possible states ω_1 and ω_2 (the probability of the rise of fall of the stock price), in which S_T takes values S^u and S^d such that:

$$S_T = \begin{cases} S^u & \text{if} \quad \omega = \omega_1 \\ S^d & \text{if} \quad \omega = \omega_2 \end{cases} \tag{4.6}$$

We determine the value S_0 assigning a probability to the events: $P(\omega_1)) = p$ and $P(\omega_2) = 1 - p$ with $p \in]0,1[$ such that an estimate of the final average value is:

$$S_T = pS^u + (1-p)S^d \tag{4.7}$$

Discounted that value at the present time we have the *risk-neutral price*:

$$\widehat{S}_0 = e^{-rT}(pS^u + (1-p)S^d) \tag{4.8}$$

If S_0 is observable by the market, imposing that $S_0 = \widehat{S}_0$, the risk-neutral pricing formula hold with respect some probability defined in terms of $q \in]0,1[$:

$$S_0 = e^{-rT}(qS^u + (1-q)S^d) \tag{4.9}$$

obtaining the *risk-neutral probability*:

$$q = \frac{e^{rT}S_0 - S^-}{S^+ - S^-}, \qquad 1 - q = \frac{S^+ - e^{rT}S_0}{S^+ - S^-} \tag{4.10}$$

It represents the unique probability to be assigned to the events ω_1, ω_2 so that S_0 is a *risk-neutral price.*

If we have two risky assets S and C depending on the same random event:

$$C_T = \begin{cases} C^u & \text{if} \quad \omega = \omega_1 \\ C^d & \text{if} \quad \omega = \omega_2 \end{cases} \tag{4.11}$$

For example C is a Call option whose underlying is the risky asset S. If S_0 is known by the market, we can find the risk-neutral probability q and then *the risk-neutral price* of $\widehat{C}_0$ of the derivative C under the probability q:

$$\widehat{C}_0 = e^{-rT}(qC^u + (1-q)C^d) \tag{4.12}$$

So we have not estimate in advance the probability of random events but the information are extrapolate from the market.

4.2 Model of financial markets

In continuous time, a market model is defined by a N-dimensional stochastic differential equation with a d-dimensional Brownian motion on the probability space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t))$

$$dX_t^i = b^i(t, X_t)dt + \sum_{j=1}^{d} \sigma_{i,j}(t, X_t)dW_t^j \qquad i = 1, \cdots, N \tag{4.13}$$

where X^i is called the risky asset that can be sold and bought in the market and P is the real probability measure under which we model our market. In our model there is another positive stochastic process B_t, the risk-free asset, that satisfies

$$dB_t = r_t B_t dt, \qquad B_0 = 1 \tag{4.14}$$

i.e

$$B_t = e^{\int_0^t t_s ds} \tag{4.15}$$

with $r - t$ the short term interest rate. We set

$$D_{tu} := B_t B_u^{-1} = e^{-\int_t^u r_s ds} \tag{4.16}$$

the discounted factor from time u to t and we denote $\tilde{Y_t} := D_{0t} Y_t$ the discounted value of any price process Y_t and follows that $d\tilde{Y_t} = D_{0t}(dY_t - r_t Y_t dt)$.

Let assume we have a portfolio π_t of an investor consisting of N risky assets $X = (X_t^i, \cdots, X_t^N)$ and the risk-free asset B_t. We use the notation $X^0 = B$ for the risky free asset. The portfolio at time t is composed by α_t^i assets X_t^i and α_t^0 units of the X_t^0. The $alpha_t^i$ is $\mathcal{F}_t$-measurable and the portfolio value is:

$$\pi_t = \sum_{i=0}^{N} \alpha_t^i X_t^i \tag{4.17}$$

Definition 10. (Self-financing portfolio) A portfolio π_t (4.17 is called self-financing if

$$d\pi_t = \sum_{i=0}^{N} \alpha_t^i dX_t^i \tag{4.18}$$

at every variation of the values of the assets that is a variation of the value of portfolio without a variation of α_t^i. i.e no cash is ever injected or removed from the portfolio between time t and $t + dt$. In term of discounted values we have

$$d\tilde{\pi}_t = \sum_{i=0}^{N} \alpha_t^i d\tilde{X}_t^i = \sum_{i=1}^{N} \alpha_t^i d\tilde{X}_t^i \tag{4.19}$$

from $d\tilde{Y}_t = D_{0t}(dY_t - r_t Y_t dt)$ and from the dynamics (4.13) we have that

$$d\tilde{\pi}_t = D_{0t}\alpha_t^i \cdot [(b(t, X_t) - r_t X_t)dt + \sigma(t, X_t)dW_t] \tag{4.20}$$

Definition 11. (Admissible portfolio) (α_t) with $t \in [0, T]$ defines an admissible portfolio if $\tilde{\pi}_t$ is bounded from below for all P-a.s, i.e there exists $M \in \mathbb{R}$ such that

$$P(\tilde{\pi}_t \geq M) = 1 \qquad \text{for every } t \in [0, T] \tag{4.21}$$

Now we introduce an important concept in financial derivatives: the arbitrage opportunities. An arbitrage is a self-financing strategy that is worth zero initially and yields a positive gain without any risk.

Definition 12. (Arbitrage) A self-financing admissible portfolio is called an arbitrage if corresponding value process π_t satisfies

$$\pi_t = 0, \quad \pi_T \geq 0 \quad \text{and} \quad P(\pi_T > 0) > 0 \tag{4.22}$$

Absence of arbitrage opportunities is a natural modeling assumption because of the as soon as they appear they tend to disappear in the market. The following proposition guarantee the condition for a absence of arbitrage in a market model

Proposition 1. *Suppose there exists a measure Q on $(\Omega, \mathcal{F})$ such that $Q \sim P$ and such that for all asset X^i the discounted process $(\tilde{X}_t^i)_{t\in[0,T]}$ with $i = 1, \cdots, N$, is a local martingale with respect to Q. Then the market $(X_t)_{t\in[0,T]}$ ha no arbitrage.*

the proposition (1) introduce the definition of equivalent local martingale measure that plays a central role in financial derivatives: any *equivalent martingale measure* is associated to a market price of risk that determines a risk-neutral price for derivatives to avoid the introduction of arbitrage opportunities.

Definition 13. (Equivalent local martingale measure) An equivalent local martingale measure (ELMM) Q is a probability measure on $(\Omega, \mathcal{F})$ such that

i Q is equivalent to P;

ii for all asset X^i, the discounted price process $(\tilde{X}_t^i)_{t\in[0,T]}$

$$\tilde{X}_t^i = e^{-\int_0^t r_s ds} X_t^i \qquad t \in [0,T] \tag{4.23}$$

is a local Q-martingale.

An ELMM is also called a *risk-neutral measure.*

Hence, from proposition (1) we have that if there exists an equivalent local martingale measure, then the market has no arbitrage opportunities.

We introduce in the portfolio an European option payoff F_T and maturity T at the price of z. The payoff is a function of the paths (x_t^i) with $t \in [0,T]$. The final value of the buyer's portfolio, discounted at time 0, is:

$$\tilde{\pi}_T^B = -D_{0t}z + \int_t^T \alpha_s \cdot d\tilde{X}_s + D_{0T}F_T \tag{4.24}$$

We define the buyer's super-replication price at time t as the greatest price z such that the value of the buyer's portfolio $\tilde{\pi}_T^B$ is P-a.s non negative.

Definition 14. (Buyer's price)
$\mathcal{B}_t(F_T) = \sup \Big[z \in \mathcal{F}_t |$ there exists an admissible portfolio α such that

$$\tilde{\pi}_T^B = -D_{0t}z + \int_t^T \alpha_s \cdot d\tilde{X}_s + D_{0T}F_T \geq 0, P - a.s. \Big] \tag{4.25}$$

Similarly the seller's super-replication prices the lowest price z such that:

Definition 15. (Seller's price)
$\mathcal{S}_t(F_T) = \inf \Big[z \in \mathcal{F}_t |$ there exists an admissible portfolio α such that

$$\tilde{\pi}_T^B = D_{0t}z + \int_t^T \alpha_s \cdot d\tilde{X}_s - D_{0T}F_T \geq 0, P - a.s. \Big] \tag{4.26}$$

Definition 16. (Attainable payoff) A payoff F_T is said to be attainable at time 0 if there exists an admissible portfolio α and a real number z such that

$$z + \int_0^T \alpha_s \cdot d\tilde{X}_s - D_{0T}F_T = 0 \tag{4.27}$$

P-a.s and $\int_0^t \alpha_s \cdot \tilde{X}_s$ is a true Q-martingale with $Q \sim P$.

Therefore a payoff is attainable if we can generate the wealth F_T at time T from an initial wealth z by trading only in the underlying assets in a self-financing way. The portfolio α and the real number z are unique because $\tilde{X}$ is a nontrivial Q-local martingale and consequently there is a unique fair price at $t = 0$ for an attainable payoff, and the price is z.

Definition 17. (Complete market) A market is said to be complete when every payoff is attainable at time 0.

Then follow the fundamental theorems of arbitrage pricing.

Theorem 9. *For a complete market, for any ELMM Q and any payoff F_T we have*

$$\mathcal{B}_t(F_T) = \mathcal{S}_t(F_T) = \mathbb{E}^Q[D_{tT}F_T|\mathcal{F}_t] \tag{4.28}$$

Theorem 10. *A market is complete if and only if there exists a unique ELMM.*

From these theorems follow that the price of an option is unique under the risk-neutral measure Q and theorem (9) gives the martingale pricing formula to avoid arbitrage opportunities.

From the self-financing equation of the portfolio (4.20) and defining the process

$$dY_t = -Y_t\lambda \cdot dW_t \tag{4.29}$$

where λ_t is a d-dimensional adapted process called *market price of risk*, such that

$$(\sigma(t, X_t)\lambda_t)^i = b^i(t, X_t) - r_t X_t^i \qquad \text{for every } i = 1, \cdots, N \tag{4.30}$$

the market model (4.13) is complete if there exists a unique $\mathcal{F}_t$-adapted vector $\lambda \in \mathbb{R}^d$ such that

The unique ELMM Q is given by

$$\frac{dQ}{dP}|_{\mathcal{F}_T} \equiv e^{-\int_0^T \lambda_s \cdot dW_s - \frac{1}{2}\int_0^T |\lambda_s|^2 ds} \tag{4.31}$$

The unique *arbitrage-free price* is

$$\mathcal{B}_t(F_T) = \mathcal{S}_t(F_T) = \mathbb{E}^Q[D_{tT}F_T|\mathcal{F}_t] \tag{4.32}$$

4.3 Black-Scholes model

The primary model to arbitrage pricing in continuous time is the Black-Scholes model. In this model the market consist of a non-risky asset, a bond B and a risky asset, a stock S. The bond price follows the equation:

$$dB_t = rB_t dt \tag{4.33}$$

where r is the locally risk-free interest rate assumed to be constant. Hence the bond follows a deterministic dynamics:

$$B_t = e^{rt} \qquad B_0 = 1 \tag{4.34}$$

The price of the risky asset is a geometric Brownian motion, satisfying the equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \qquad \mu \in \mathbb{R}, \sigma \in \mathbb{R}_{>0} \tag{4.35}$$

where μ is the average rate of return and σ is the volatility, W_t with $t \in [0, T]$ is a real Brownian motion on the probability space $(\Omega, \mathcal{F}, P, (\mathcal{F}_t))$. The explicit solution of SDE of the risky asset (4.35) is:

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \tag{4.36}$$

We consider the European Call option with strike K and maturity T with the payoff of the form:

$$F(S_T) = (S_T - K)^+ \tag{4.37}$$

In continuous time, the extension of the definition of a self-financial strategy is the follow:

Definition 18. A strategy or portfolio is a stochastic process (α_t, β_t) where $\alpha \in \mathbb{L}^2_{\text{loc}}$ and $\beta \in \mathbb{L}^1_{\text{loc}}$. The value of the portfolio (α, β) is the stochastic process defined by

$$V_t^{(\alpha,\beta)} = \alpha_t S_t + \beta_t B_t \tag{4.38}$$

α, β are to be interpreted as the amount of S and B held by the investor in the portfolio and they can take negative values. Furthermore, the investment strategy depends only on the amount of the information available at that moment, so α, β are progressively measurable.

Definition 19. A strategy (α_t, β_t) is self-financing if

$$dV_t^{(\alpha,\beta)} = \alpha_t dS_t + \beta_t dB_t \tag{4.39}$$

holds, that is

$$V_t^{(\alpha,\beta)} = V_0^{(\alpha,\beta)} + \int_0^t \alpha_s dS_s + \int_0^t \beta_s dB_s \tag{4.40}$$

From an intuitive point of view, the self-financing expresses the fact that the instantaneous variation of the value of the portfolio is caused uniquely by the changes of the prices of the assets and not by injecting fund from outside. Now we define the discounted prices

$$\tilde{S}_t = e^{-rt} S_t \qquad \tilde{V}_t = e^{-rt} V_t \tag{4.41}$$

and follow the proposition:

Proposition 2. *A strategy (α, β) is self-financing if and only if*

$$d\tilde{V}_t^{(\alpha,\beta)} = \alpha_t d\tilde{S}_t \tag{4.42}$$

holds, that is

$$\tilde{V}_t^{(\alpha,\beta)} = \tilde{V}_0^{(\alpha,\beta)} + \int_0^t \alpha_s d\tilde{S}_s \tag{4.43}$$

Thanks to this proposition, the value of a self-financing strategy (α, β) is determined uniquely by its initial value $\tilde{V}_0^{(\alpha,\beta)}$ and by the process α.

Definition 20. A strategy $\alpha_t, \beta_t)$ is Markovian if

$$\alpha_t = \alpha(t, S_t), \qquad \beta_t = \beta(t, S_t) \tag{4.44}$$

where α, β are function in $C^{1,2}([0, T[\times \mathbb{R}_{>0})$.

Therefore the value of a Markovian strategy (α, β) is a function of time and of the price of the underlying asset:

$$f(t, S_t) := V_t^{(\alpha,\beta)} = \alpha(t, S_t) S_t + \beta(t, S_t) e^{rt}, \qquad t \in [0, T[\tag{4.45}$$

with $f \in C^{1,2}([0, T[\times \mathbb{R}_{>0})$. The function f is uniquely determined by (α, β)

Theorem 11. *Suppose that (α, β) is a markovian strategy and set $f(t, S_t) = V_t^{(\alpha,\beta)}$. The following two conditions are equivalent:*

1. *(α, β) is self-financing;*

2. f is solution to the PDE

$$\frac{\sigma^2 s^2}{2}\partial_{ss} f(t,s) + rs\partial_s f(t,s) + \partial_t f(t,s) = rf(t,s) \tag{4.46}$$

with $(t,s) \in [0,T[\times\mathbb{R}_{>0}$ and we have that

$$\alpha(t,s) = \partial_s f(t,s) \tag{4.47}$$

equation (4.46) is called Black-Scholes partial differential equation.

The coefficient of the Black-Scholes PDE depend on the volatility σ of the risky asset and the risk-free rate r but they do not depend on the average rate of return μ as we expected because arbitrage pricing does not depend on the subjective estimate of the future value of the risky asset.

Definition 21. A strategy (α,β) is admissible if it is bounded from below, i.e. there exists a constant C such that

$$V_t^{(\alpha,\beta)} \geq C, \qquad t \in [0,T], a.s \tag{4.48}$$

We denote by $\mathcal{A}$ the family of Markovian, self-financing admissible strategies.

The financial interpretation is that investment strategies which request unlimited debt are not allowed.

Definition 22. A European derivative $F(S_T)$ is replicable if there exist an admissible portfolio $(\alpha,\beta) \in \mathbb{A}$ such that

$$V_T^{(\alpha,\beta)} = F(S_T) \qquad \text{in} \quad \mathbb{R}_{>0} \tag{4.49}$$

We say that (α,β) is a replicating portfolio for $F(S_T)$.

Now follows the central result in Black-Scholes theory:

Theorem 12. *The Black-Scholes market model is complete and arbitrage-free, this meaning that European derivative $F(S_T)$ is replicable in a unique way. Indeed there exists a unique strategy $h = (\alpha,\beta) \in \mathcal{A}$ replicating $F(S_T)$, that is given by*

$$\alpha_t = \partial_s f(t,S_t), \qquad \beta_t = e^{-rt}(f(t,S_t) - S_t\partial_s f(t,S_t)) \tag{4.50}$$

where f is lower bounded solution of the Cauchy problem

$$\begin{cases} \frac{\sigma^2 s^2}{2}\partial_{ss} f(t,s) + rs\partial_s f(t,s) + \partial_t f(t,s) = rf(t,s) & \text{in } [0,T[\times\mathbb{R}_{>0} \\ f(T,s) = F(s) & s \in \mathbb{R}_{>0} \end{cases} \tag{4.51}$$

By definition, $f(t,S_t) = V_t^{(\alpha,\beta)}$ is the arbitrage price of $F(S_T)$

The following corollary is the famous Black-Scholes Formula for the price of European Call and Put options:

Corollary 1. ***(Black-Scholes Formula)*** *Let us assume the Black-Scholes dynamics for the underlying asset*

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{4.52}$$

and let denote by r the short rate. Then, if K is the strike price and T is the maturity, the following formulas for the price of European Call and Put options hold:

$$c_t = S_t \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2) \tag{4.53}$$

$$p_t = K e^{-r(T-t)} \Phi(-d_2) - S_t \Phi(-d_1) \tag{4.54}$$

where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{y^2}{2}} dy \tag{4.55}$$

is the standard normal distribution function and

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}[log(\frac{S_t}{K} + (r + \frac{\sigma^2}{2})(T-t)] \tag{4.56}$$

$$d_2 = d_1 - \sigma\sqrt{T-t} = \frac{1}{\sigma\sqrt{T-t}}[log(\frac{S_t}{K} + (r - \frac{\sigma^2}{2})(T-t)] \tag{4.57}$$

Chapter 5

Stochastic Local Volatility model

5.1 Introduction

The main problem in Black-Scholes model is that the volatility parameter is assumed to be constant and deterministic for all strike and maturity. However from the empirical market option prices we can see the *volatility smile* that is not a constant: the implied volatility of at-the money (ATM) strike ($K = S_0$) is lower than that of in-the-money (ITM) and out-of-the-money (OTM) strikes. Hence different model were proposed to reproduce the volatility skew: the local volatility (LV) and the stochastic volatility (SV) models and an hybrid between the two: the stochastic local volatility (SLV) model.

5.2 Local volatility model

We introduce the Local volatility model in which the volatility term σ_{LV} depends on the time and on the underlying risky asset, the current spot S_t:

$$dS_t = (r_d - r_f)S_t dt + \sigma_{LV}(t, S_t)S_t dW_t \tag{5.1}$$

Dupire, Derman and Kani used the market option prices to derive the local volatility function rather than explicitly construct the function. From Feynman-Kač formula, the price of a European call option is:

$$C(K,T) = e^{-r_dT}\int_0^{+\infty}(S_T - K)^+ p(S_T, T|S_0)dS_T = e^{-r_dT}\int_K^{+\infty}(S_T - K)p(S_T, T|S_0)dS_T \tag{5.2}$$

where the transition probability density $p(S_T, T|S_0)$ of the volatility model follows the Fokker-Planck equation

$$\frac{\partial p}{\partial T} = -\frac{\partial}{\partial S_T}[(r_d - r_f)S_T p] + \frac{1}{2}\frac{\partial^2}{\partial S_T^2}[\sigma_{LV}(T, S_T)^2 S_T^2 p] \tag{5.3}$$

Differentiating (5.2) $C(K, T)$ with respect to K and to T and using (5.3), with the assumption that probability density rapidly decays to zero as $S_T \to +\infty$, we have:

$$\frac{\partial C}{\partial T} = -r_f C - (r_d - r_f)K\frac{\partial C}{\partial K} + \frac{1}{2}\sigma_{LV}(K, T)^2 K^2 \frac{\partial^2 C}{\partial K^2} \tag{5.4}$$

Hence we obtain the local volatility function $\sigma_{LV}(t, S)$, given market prices of the call options:

$$\sigma_{LV}(t, S) = \sqrt{\frac{\frac{\partial C}{\partial T} + (r_d - r_f)K\frac{\partial C}{\partial K} + r_f C}{\frac{K^2}{2}\frac{\partial^2 C}{\partial K^2}}}\Big|_{K=S, T=t} \tag{5.5}$$

Another useful formula to derive the local volatility $\sigma_{LV}(t, S)$ from the implied volatility $\sigma_{IV}(K, T|S_0)$ in the Black-Scholes formula is the following:

$$\sigma_{LV}(t, S) = \sqrt{\frac{\sigma_{IV}^2 + 2\sigma_{IV}T\frac{\partial \sigma_{IV}}{\partial T} + 2(r_d - r_f)\sigma_{IV}KT\frac{\partial \sigma_{IV}}{\partial K}}{(1 + d_1(S, K)K\sqrt{T}\frac{\partial \sigma_{IV}}{\partial K})^2 + \sigma_{IV}K^2T[\frac{\partial^2 \sigma_{IV}}{\partial K^2} - d_1(S, K)\sqrt{T}(\frac{\partial \sigma_{IV}}{\partial K})^2]}}\Big|_{K=S, T=t} \tag{5.6}$$

The option pricing PDE under local volatility model is

$$\frac{\partial u}{\partial \tau} = (r_d - r_f)S\frac{\partial u}{\partial S} + \frac{1}{2}\sigma_{IV}^2 S^2 \frac{\partial^2 u}{\partial S^2} - r_d u \tag{5.7}$$

with $\tau = T - t$. This local volatility model has problem in pricing barrier options that depend not only on the final state of the underlying price but also on the evolution of the risky asset. Hence the local volatility model could predicts a wrong volatility surface fro the future.

5.3 Stochastic volatility model

An alternative model to the local volatility is the stochastic volatility model (SV). We consider a particular SV model: the Heston stochastic model that assumes to follow the dynamics for the spot rate S_t and the stochastic variance V_t under the risk neutral measure

$$dS_t = (r_d - r_f)Stdt + \sqrt{V_t}S_t dW_t^1, \qquad S_0 = s \tag{5.8}$$

$$dV_t = k(\theta - V_t)dt + \lambda\sqrt{V_t}dW_t^2, \qquad V_0 = v \tag{5.9}$$

$$dW_t^1 \cdot dW_t^2 = \rho dt \tag{5.10}$$

$(k, \theta, \lambda, \rho)$ are the Heston parameters; high vol of vol λ leads to high stochastic volatility and law λ degenerates the SV model to a model with deterministic volatility. A negative correlation ρ leads to a left-skewness and a positive one to a right-skewness. The long-run variance θ determines the height of implied volatility smile and high mean reversion k increases the level of around at-the-money volatilities. The pricing PDE of the Heston stochastic volatility model with $\tau = T - t$ is:

$$\frac{\partial u}{\partial \tau} = (r_d - r_f)S\frac{\partial u}{\partial S} + \frac{1}{2}S^2V\frac{\partial^2 u}{\partial S^2} + \lambda\rho SV\frac{\partial^2 u}{\partial S\partial V} + k(\theta - V)\frac{\partial u}{\partial V} + \frac{1}{2}\lambda^2V\frac{\partial^2 u}{\partial V^2} - r_d u \tag{5.11}$$

5.4 Stochastic local volatility model

In the FX options market, where are traded options like exotic options and barrier options, it is necessary an hybrid model between the local volatility model and the stochastic one, called stochastic-local volatility model or SLV model. The SLV model considered, is not a linear average between the two previous models but is a non-linear and non-parametric combination of LV model and SV model. This model is a stochastic process in two dimensions $X_t = (S_t, V_t)$, the spot price S_t and the stochastic variance V_t, with a 2-dimensional Brownian motion $W_t = (W_t^1, W_t^2)$. The general dynamics of the hybrid SLV model are:

$$dS_t = \mu_1(S_t, t)dt + L(S_t, t)\sigma_1(S_t, V_t, t)dW_t^1 \tag{5.12}$$

$$dV_t = \mu_2(V_t, t)dt + \sigma_2(V_t, t)dW_t^2 \tag{5.13}$$

$$dW_t^1 \cdot dW_t^2 = \rho dt \tag{5.14}$$

The correlation between the two Brownian motion is $\rho \in]-1, 1[$. In the dynamic of the spot price, the weights of the volatility is controlled by the leverage function $L(S_t, t)$, determined by the information of the market. The leverage represents the weight of LV model and it can be numerically calibrated to the market data. Considering the Heston-like SLV model, the dynamic of the spot price S_t and the stochastic variance V_t under the risk neutral measure in (5.12) becomes:

$$dS_t = (r_d - r_f)S_t dt + L(S_t,t)\sqrt{V_t}S_t dW_t^1,\ S_0 = s \tag{5.15}$$

$$dV_t = k(\theta - V_t)dt + \lambda\sqrt{V_t}dW_t^2,\ V_0 = v \tag{5.16}$$

$$dW_t^1 \cdot dW_t^2 = \rho dt \tag{5.17}$$

With (k,θ,λ,ρ) the Heston parameters that have a term structures, r_d, r_f the domestic and foreign interest rate respectively that are locally deterministic and we define $r = r_d - r_f$.

Because the presence of the square root in the stochastic volatility process, $\sqrt{V_t}$, the diffusion term does not satisfy the Lipschitz condition to guarantee the existence and the unicity of the solution of (5.15) but, under the assumption:

$$\begin{aligned}&(i)\ \textit{The initial values } S_0, V_0 > 0. \textit{ The coefficients } k,\theta,\lambda > 0,\ -1 < \rho < 1,\\ &\textit{and } r_d, r_f \in \mathbb{R},\ \textit{and the leverage function } L \textit{ is positive and bounded.}\\ &(ii)\ \textit{The Feller condition } \tfrac{2k\theta}{\lambda^2} \geq 1, \textit{ which guarantees the positivity of the variance process.}\end{aligned} \tag{5.18}$$

Lemma 1. *(Feller condition)*
Given $\frac{2k\theta}{\lambda^2}\geq 1$ *and* $V_0 > 0$, *we have* $Prob[V_t > 0] = 1$ *for all* $t > 0$.

Proposition 3. *Under the assumptions (i), (ii), there exists a unique solution* (S_t, V_t) *to the SLV model (5.15), given* $V_t > 0$ *for all* $t > 0$. *Moreover, there exists a function* $p(S_t, V_t)$ *which is the transition density of the SLV model and is the unique solution of the Fokker-Planck equation*

$$\frac{\partial p}{\partial t} + \frac{\partial}{\partial S}[rSp] + \frac{\partial}{\partial V}[k(\theta - V)p] - \frac{1}{2}\frac{\partial^2}{\partial S^2}[S^2 V L(t,S)^2 p] - \frac{1}{2}\lambda^2\frac{\partial^2}{\partial V^2}[Vp] - \lambda\rho\frac{\partial^2}{\partial S \partial V}[SVL(t,S)p] = 0 \tag{5.19}$$

5.5 Option pricing PDE

Option pricing PDE of the SLV model (5.15) can be derived by a martingale approach. B_t^d is the domestic bond under the risk-neutral measure

$\mathbb{Q}$ following the dynamic $B_t^d = e^{\int_0^t r_d ds}$. The discounted option price is a martingale from the first fundamental theorem of the market:

$$\frac{u(S_t, V_t, t)}{B_t^d} = \mathbb{E}\big[\frac{u(S_T, V_T, T)}{B_T^d}|S_t, V_t\big] \tag{5.20}$$

Then, the value function of an option under the risk neutral measure is:

$$u(S_t, V_t, t) = \mathbb{E}\big[e^{-\int_t^T r_d ds} u(S_T, V_T, T)|S_t, V_t\big] \tag{5.21}$$

Defining the discounted option price $\bar{u}_t(S_t, V_t, t)$:

$$\bar{u}_t(S_t, V_t, t) = \frac{u(S_t, V_t, t)}{B_t^d} = e^{-\int_0^t r_d ds} u(S_t, V_t, t) \tag{5.22}$$

It is a martingale under the risk neutral measure $\mathbb{Q}$. From the Îto's Lemma on $\bar{u} = \bar{u}(S_t, V_t, t)$ with the Heston SLV model (5.15):

$$d\bar{u}_t = \frac{\partial \bar{u}}{\partial t}dt + \frac{\partial \bar{u}}{\partial S}dS_t + \frac{\partial \bar{u}}{\partial V}dV_t + \frac{1}{2}\frac{\partial^2 \bar{u}}{\partial S^2}(dS_t \cdot dS_t) + \frac{1}{2}\frac{\partial^2 \bar{u}}{\partial V^2}(dV_t \cdot dV_t) + \frac{\partial^2 \bar{u}}{\partial S \partial V}(dS_t \cdot dV_t) \tag{5.23}$$

$$\begin{aligned} dS_t \cdot dS_t &= L^2 S_t^2 V_t dt \\ dV_t \cdot dV_t &= \lambda^2 V_t dt \\ dS_t \cdot dV_t &= \lambda \rho L S_t V_t dt \end{aligned} \tag{5.24}$$

Inserting the SLV model (5.15) in the Itô'formula (5.23) using the correlation (5.24), we obtain that:

$$\begin{aligned} d\bar{u}_t =& \big[\frac{\partial \bar{u}}{\partial t} + rS\frac{\partial \bar{u}}{\partial S} + k(\theta - V)\frac{\partial \bar{u}}{\partial V} + L^2 S^2 V \frac{1}{2}\frac{\partial^2 \bar{u}}{\partial s^2} + \lambda \rho L S V \frac{\partial^2 \bar{u}}{\partial S \partial V} + \lambda^2 V \frac{1}{2}\frac{\partial^2 \bar{u}}{\partial V^2}\big]dt \\ &+ \frac{\partial \bar{u}}{\partial S} L\sqrt{V} S dW_t^1 + \frac{\partial \bar{u}}{\partial V}\lambda\sqrt{V} S dW_t^2 \end{aligned} \tag{5.25}$$

Because the discounted price is a martingale, the coefficient of the drift term has to be null, obtaining the discounted price PDE:

$$\frac{\partial \bar{u}}{\partial t} + rS\frac{\partial \bar{u}}{\partial S} + k(\theta - V)\frac{\partial \bar{u}}{\partial V} + L^2 S^2 V \frac{1}{2}\frac{\partial^2 \bar{u}}{\partial s^2} + \lambda \rho L S V \frac{\partial^2 \bar{u}}{\partial S \partial V} + \lambda^2 V \frac{1}{2}\frac{\partial^2 \bar{u}}{\partial V^2} = 0 \tag{5.26}$$

Deriving the partial derivative from the definition of the discounted price (5.22)$\frac{\partial \bar{u}}{\partial t} = e^{-\int_0^t r_d ds}\frac{\partial u}{\partial t} - r_d u e^{-\int_0^t r_d ds}$, and inserted into the discounted price PDE (5.26), we obtain the forward option price PDE under the domestic risk neutral measure $\mathbb{Q}$ of the SLV model:

$$\frac{\partial u}{\partial t} - r_d u + (r_d - r_f)S\frac{\partial u}{\partial S} + k(\theta - V)\frac{\partial u}{\partial V} + L^2 S^2 V \frac{1}{2}\frac{\partial^2 u}{\partial S^2} + \lambda\rho LSV\frac{\partial^2 u}{\partial S \partial V} + \lambda^2 V \frac{1}{2}\frac{\partial^2 u}{\partial V^2} = 0 \tag{5.27}$$

Replacing t by $\tau = T - t$, the PDE becomes a backward PDE and with proper initial and boundary condition, (5.27) can be solved with the Feynman-Kač theorem

Proposition 4. *Given a terminal condition $u(S_T, V_T, T) = g(S_T, V_T, T)$ at time T, there exists a unique solution $u(S_t, V_t, t)$ of the option pricing PDE (5.27) under the domestic risk neutral measure $\mathbb{Q}$ and the solution is given by*

$$u(s, v, t) = \mathbb{E}\left[e^{-\int_t^T r_d ds} g(S_T, V_T, T) | S_t = s, V_t = v\right] \tag{5.28}$$

5.6 Leverage surface calibration

The main problem of this thesis is the calibration of the leverage function $L(t, S_t)$ of the local stochastic volatility model LSV to the local one LV, such that it matches the vanilla market prices of European options. This problem is called *leverage surface calibration* of the LSV model to entire volatility surface of the vanilla market. The calibration is performed by resolution of the Fokker-Planck PDE for the LSV model using simultaneously the local volatility of the LV model. The first step is to match the market with a local volatility model. The second step is find a leverage surface such that the LSV has the same marginal distribution of the local one. This is allowed because of the link between the two transition probability density of the LSV model $p(t, S_t, V_t)$ and the transition probability density of the LV one $p(t, S_t)$, thanks to the mimicking theorem.

Proposition 5. *To mimic the LV model, the local volatility of the SLV model (5.15) must follow:*

$$\sigma(t, S)^2 = \mathbb{E}[L(t, S_t)^2 V_t | S_t = S] = L(t, S)^2 \mathbb{E}[V_t | S_t = S] \tag{5.29}$$

Furthermore, the probability distribution of the LV model is the same as the marginal distribution of the LSV model with the following relation between the transition probability densities:

$$p(t,S) = \int_{\mathbb{R}_+} p(t,S,V)dV \tag{5.30}$$

Now we can continue with the two following step to calibrate the volatility surface:

1. First step:
   We start from the local volatility model

   $$dS_t = rS_t dt + \sigma(t,S_t)S_t dW_t \tag{5.31}$$

   and we use the Dupire formula to extract the implied volatility surface from the available data and obtain the local volatility $\sigma_{\text{Dup}} = \sigma(t,S_t)$

2. Second step:
   Thanks to the previous proposition (5), we connect the marginal distribution of the LSV model to the LV model, giving the following local volatility part of the LSV model, the leverage function:

$$L(t,S) = \frac{\sigma(t,S)}{\sqrt{\mathbb{E}[V_t|S_t = S]}} = \sigma(t,S)\sqrt{\frac{\int_{\mathbb{R}_+} p(t,S,V)dV}{\int_{\mathbb{R}_+} Vp(t,S,V)dV}} \tag{5.32}$$

This ensure that the LSV is calibrated to the vanilla market. From (7.4), there are two degenerate cases: 1. if $L(t,S_t) = 1$ the LSV model becomes the pure Heston stochastic volatility model, 2. if $\lambda = 0$, the process for V_t becomes deterministic with $L = \sigma/\sqrt{V_t}$ and the LSV model degenerates to the pure local volatility model. Now, from the current calibrated leverage function (7.4) arises a consequent problem: replacing $L(t,S_t)$ in the Fokker-Planck PDE (6.2), we obtain a nonlinear PDE:

$$\frac{\partial p}{\partial t} - \frac{1}{2}\frac{\partial^2}{\partial S^2}\Big[S^2 V\sigma(t,S)^2 \frac{\int_{\mathbb{R}_+} p(t,S,V)dV}{\int_{\mathbb{R}_+} Vp(t,S,V)dV}p\Big] - \frac{1}{2}\lambda^2\frac{\partial^2}{\partial V^2}[Vp]$$
$$-\lambda\rho\frac{\partial^2}{\partial S\partial V}[SV\sigma(t,S)\sqrt{\frac{\int_{\mathbb{R}_+} p(t,S,V)dV}{\int_{\mathbb{R}_+} Vp(t,S,V)dV}}p] + \frac{\partial}{\partial S}[rSp] + \frac{\partial}{\partial V}[k(\theta - V)p] = 0 \tag{5.33}$$

The problem is an implementation problem of a nonlinear PDE with coefficients that depend on stochastic terms! In fact the calibrated leverage function contains the expectation of the stochastic process V_t: $\mathbb{E}[V_t|S_t = S]$. The first approach to the implementation of the calibrated function were performed by a Monte Carlo simulation and a finite difference scheme. Sequentially, the faster, robustly and secure method discovered to solve a nonlinear PDE was solved by the Particle Method.

Chapter 6

Particle method applied in finance

6.1 Introduction

In the first chapter we have seen the particle method applied in physics in the description of physical flows with *deterministic* equations of motion of the particles. In the financial application to the calibration of the LSV model, the particle method is *stochastic*: the "flow" of the price process is approximated by a discrete numbers of particles that are *random variables* follow a *discrete distribution*.

6.1.1 McKean SDE

The mathematical formalization of particle method was introduced to solve nonlinear stochastic differential equation such as McKean SDE and the calibration of the LSV model is exactly a McKean nonlinear SDE, since the volatility coefficient depends on the marginal law of the process. Hence, particle method seems to be the right one to solve the calibration of volatility surface. In order to proceed with the resolution, we start from the definition of the McKean nonlinear stochastic differential equation.

Definition 23. Let X_t a n-dimensional process, let $b(t, X_t, \mathbb{P}_t)$ the drift a n-dimensional vector and $\sigma(t, X_t, \mathbb{P}_t)$ the volatility $n \times d$ - dimensional matrix. b and σ depend on the current value X_t of the process and also on the probability distribution $\mathbb{P}_t$ of X_t. The McKean stochastic differential equation for the n-process X_t with a d-dimensional Brownian motion W_t is:

$$dX_t = b(t, X_t, \mathbb{P}_t)dt + \sigma(t, X_t, \mathbb{P}_t)dW_t, \qquad X_t \sim \mathbb{P}_t, \qquad X_0 \in \mathbb{R}^n \tag{6.1}$$

The probability density function $p(t,y)dy = \mathbb{P}_t(dy)$ of X_t is a solution to the Fokker-Planck PDE which the initial condition: $\lim_{t\to 0} p(t,x) = \delta(x - X_0)$.

$$\partial_t p(t,x) + \sum_{i=1}^{n} \partial_i (b^i(t,x,\mathbb{P}_t)p(t,x)) - \frac{1}{2} \sum_{i,j=1}^{n} \partial_{ij} \big(\sum_{k=1}^{d} \sigma_k^i(t,x,\mathbb{P}_t)\sigma_k^j(t,x,\mathbb{P}_t)p(t,x) \big) = 0 \tag{6.2}$$

The PDE (6.2) is nonlinear because of $b^i(t,x,\mathbb{P}_t$ and $\sigma_k^i(t,x,\mathbb{P}_t)$ depend on the unknown $p(t,x)$.

Example 1. *An example of (6.1) is the McKean-Vlasvov SDE in which the drift and the volatility are the mean value of some function $b^i(t,X_t,\cdot)$, $\sigma_j^i(t,X_t,\cdot)$, for $1 \le i \le n$ and $1 \le j \le d$, with respect to the distribution $\mathbb{P}_t$ of X_t. Assuming that $B(t,x,y)$ and $\Sigma(t,x,y)$ are Lipschitz-continuous in x and y we have*

$$b^i(t,x,\mathbb{P}_t) = \int B^i(t,x,y)\mathbb{P}_t(dy) = \mathbb{E}[b^i(t,x,X_t)] \tag{6.3}$$

$$\sigma_j^i(t,x,\mathbb{P}_t) = \int \Sigma_j^i(t,x,y)\mathbb{P}_t(dy) = \mathbb{E}[\sigma_j^i(t,x,X_t)] \tag{6.4}$$

6.1.2 Particle method

The fundamental concept to develop the particle method to simulate the stochastic McKean SDE (6.1) consists of approximate the law $\mathbb{P}_t$ with the empirical distribution of a fixed large number N of particles $(\xi_t^{i,N})_{i=1}^N$

$$\mathbb{P}_t \approx \mathbb{P}_t^N := \frac{1}{N} \sum_{i=1}^{N} \delta_{\xi_t^{i,N}} \tag{6.5}$$

where $(\xi_t^{i,N})_{i=1}^N$ are solutions to the $(\mathbb{R}^n)^N$-dimensional classical linear SDE:

$$d\xi_t^{i,N} = b(t,\xi_t^{i,N},\mathbb{P}_t^N)dt + \sigma(t,\xi_t^{i,N},\mathbb{P}_t^N)dW_t^i, \qquad \xi_0^{i,N} \sim \mathbb{P}_0 \qquad i = 1,\cdots,N \tag{6.6}$$

Each of the N particles $\xi_t^{i,N}$ can be seen as a n-dimensional Itô process following the stochastic equation (6.6) with N independent d-dimensional Brownian motions, $(W_t^i)_{i=1}^N$ and $\mathbb{P}_t^N$ a random measure on $\mathbb{R}^n$. We assume that the initial law $\mathbb{P}_0^N$ is a product distribution of the form $\mathbb{P}_0^N = (\mathbb{P}_0 \times \cdots \times$

$\mathbb{P}_0$) or in other words that the initial system $\xi_0^{i,N}$ of N particles consists of N independent and identically distributed random variables $\xi_0^{i,N}$ with common law $\mathbb{P}_0$. So, given the empirical distribution $\mathbb{P}_t^N$ of the $\xi_t^{i,N}$ particles, these $\xi_t^{i,N}$ evolve independently of each other.

Example 2. *In the case of the example (1), $\mathbb{P}_t$ is approximated by (6.5), and in the McKean SDE (6.1) we have*

$$d\xi_t^{i,N} = \Big(\int B(t, \xi_t^{i,N}, y)\mathbb{P}_t^N(y)\Big)dt + \Big(\int \Sigma(t, \xi_t^{i,N}, y)\mathbb{P}_t^N(y)\Big)dW_t^i \tag{6.7}$$

that is equivalent to

$$d\xi_t^{i,N} = \frac{1}{N}\sum_{j=1}^{N} B(t, \xi_t^{i,N}, \xi_t^{j,N})dt + \frac{1}{N}\sum_{j=1}^{N} \Sigma(t, \xi_t^{i,N}, \xi_t^{j,N})dW_t^i \tag{6.8}$$

We can interpret the result as the interaction between the particles $\xi_t^{i,N}$ with each other! In fact the drift and diffusion coefficients of $\xi_t^{i,N}$ depend on the position of the particle $\xi_t^{i,N}$ and also on the interaction with the other $N-1$ particles. This is the main difference with the Monte Carlo approach.Furthermore, we speak of particle method because of a system of *N interacting particles.* Furthermore, in analogy with statistical physics, the paths of price process can be viewed as indistinguible interacting particles and then we fall into the case of bosonic particles. Moreover, because of at $t = 0$, the $\xi_0^{i,N}$ are *independent particles*, for $N \to \infty$, for any fixed $t > 0$, the $\xi_t^{i,N}$ are *asymptotically independent* and their empirical measure $\mathbb{P}_t^N$ converges in distribution toward the true measure $\mathbb{P}_t$. This property is called the *propagation of chaos.* It means that for all functions $f \in C_b(\mathbb{R}^n)$

$$\frac{1}{N}\sum_{i=1}^{N} f(\xi_t^{i,N}) \xrightarrow[N\to\infty]{L^1} \int_{\mathbb{R}^n} f(x)p(t,x)dx \tag{6.9}$$

where $p(t,\cdot)$ is the fundamental solution to the nonlinear Fokker-Planck PDE (6.2). Hence the particle method is convergent and in the large N limit, the $(\mathbb{R}^n)^N$-dimensional linear Fokker-Planck PDE approximates the nonlinear n-dimensional Fokker-Planck PDE (6.2). This approximation is called in statistical physics: *mean field approximation.*

6.2 Convergence of the particle method

The convergence of the particles approximation is due to the property of the *propagation of chaos.* We consider the 1-dimensional case $d = n = 1$ of the McKean-Vlasov SDE (6.1) and that the drift and the volatility does not depend on time t.

Definition 24 (**Empirical measure**). Let $\xi_1, \cdots, \xi_N$ be i.i.d. random variables with law μ. The empirical measure associated to the configuration $(\xi_1, \cdots, \xi_N)$ is:

$$\widehat{\mu}^N := \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i} \qquad (\xi_1, \cdots, \xi_N) \sim \mu \tag{6.10}$$

This is a random probability measure with expectation $\mathbb{E}_\mu[\widehat{\mu}^N] = \mu$, meaning that for all event A, $\mathbb{E}_\mu[\widehat{\mu}^N(A)] = \mu(A)$.

Definition 25 (**μ-chaotic distribution**). Let $\{\mu^N\}_{N\in\mathbb{N}}$ be a sequence of symmetric probabilities on $(\mathbb{R}^n)^N$. Let μ be a probability measure on $\mathbb{R}^n$. We say that μ^N is μ-chaotic if for each integer $k \geq 1$ and for all test functions $\phi_1, \cdots, \phi_k \in \mathcal{C}_b(\mathbb{R}^n)$(i.e for all bounded continuous functions), we have:

$$\int \phi_1(x_1)\cdots\phi_k(x_k)d\mu^N(x_1, \cdots, x_N) \xrightarrow[N\to\infty]{} \int \phi_1 d\mu \cdots \int \phi_k d\mu \tag{6.11}$$

This mean that k particles within N are asymptotically independent and identically distributed as $N \to \infty$. We say that μ^N is chaotic if there exist μ such that μ^N is μ-chaotic.

Definition 26 (**Propagation of chaos**). Let us consider an N-dimensional SDE flow that associates to an initial probability measure μ_0^N a probability μ_t^N at time t. We say that this flow propagates the chaos if, for initial chaotic measure μ_0^N and any $t > 0$, μ_t^N is chaotic.

Theorem 13. *Let $\{\mu^N\}_{N\in\mathbb{N}}$ be a sequence of symmetric probabilities on $(\mathbb{R}^n)^N$, and μ be a probability measure on $\mathbb{R}^N$. The following properties are equivalent:*

1. *$\{\mu^N\}_{N\in\mathbb{N}}$ is μ-chaotic*

2. *For all test function ϕ_1, $\phi_2 \in \mathcal{C}_b(\mathbb{R}^n)$:*

$$\int \phi_1(x_1)\phi_2(x_2)d\mu^N(x_1, \cdots, x_N) \xrightarrow[N\to\infty]{} \int \phi_1 d\mu \int \phi_2 d\mu \tag{6.12}$$

3. *Let $\xi_1, \cdots, \xi_N$ be a r.v. with law $(\xi_1, \cdots, \xi_N) \sim \mu^N$. Then, for all $\phi \in \mathcal{C}_b(\mathbb{R}^n)$:*

$$\frac{1}{N}\sum_{i=1}^{N} \phi(\xi_i) \xrightarrow[N\to\infty]{L^1} \int \phi d\mu \tag{6.13}$$

4. Let $\hat{\mu}^N$ be the empirical measure associated to μ^N, then:

$$\mathbb{E}_{\mu^N}\big[\big|\int \phi d\hat{\mu}^N - \int \phi d\mu\big|\big] \xrightarrow[N\to\infty]{} 0 \tag{6.14}$$

for all $\phi \in \mathcal{C}_b(\mathbb{R}^n)$.

Theorem 14. *The propagation of chaos holds for the McKean-Vlasov SDE (1).*

Theorem (14) proves the convergence of the particle method: for all t and all $\phi \in \mathcal{C}_b(\mathbb{R}^n)$

$$\frac{1}{N}\sum_{i=1}^{N}\phi(\xi_t^{i,N}) \xrightarrow[N\to\infty]{L^1} \int \phi(x)p(t,x)dx \tag{6.15}$$

is equivalent to the propagation of chaos for the SDE $(\xi_t^{1,N}, \cdots, \xi_t^{N,N})$.

6.3 Calibration of the LSV model with particle method

With the particles approximation, we can formulate the calibration problem of the local stochastic volatility model to market smiles using particle method. Different methods has been used to solve this problem, such as finite difference schemes for one factor stochastic volatility models and Monte Carlo methods for multi factor models, but for this specific problem particle method is the best one because of the presence of the McKean SDE as equations of the prices process and therefore the dependence of the volatility term on the probability distribution of the process.

We calibrating the stochastic local volatility model in the case of deterministic interest rate r. The LSVM is defined by the following SDE for the spot process:

$$dS_t = rS_t dt + S_t L(t, S_t) V_t dW_t \tag{6.16}$$

where V_t is a stochastic process and $L(t, S_t)$ is the local volatility function or leverage function, introduced in order to exactly calibrate the volatility to the whole implied volatility surface or market smiles.

As we have seen in the chapter (5) of the construction of the leverage function, the first step of calibration is to recover the implied volatility from the local volatility model using the Dupire formula obtaining the Dupire local volatility $\sigma(t, S_t)$.
The second step connects the marginal distribution of the LSV model to the LV one, obtaining the exactly calibration of the leverage function to market smiles by the formula:

$$\sigma(t, S)^2 = L(t, S)^2 \mathbb{E}_{\mathbb{P}}[V_t | S_t = S] \tag{6.17}$$

where the Dupire local volatility that is inferred from the vanilla market smile is:

$$\sigma(t, S)^2 = \frac{\partial_t C(t, S)}{\frac{1}{2} S^2 \partial_S^2 C(t, S)} \tag{6.18}$$

where $C(t, S)$ is the market price of a call option. Replacing (6.17) in the SDE (6.16), we obtain:

$$dS_t = rS_t dt + S_t \frac{\sigma(t, S_t)}{\sqrt{\mathbb{E}_{\mathbb{P}}[V_t | S_t]}} V_t dW_t \tag{6.19}$$

In this form is clear the dependence of the leverage function on the joint probability density function $p(t,S,V)$ of (S_t,V_t):

$$L(t,S,p)=\sigma(t,S)\sqrt{\frac{\int p(t,S,V)dV}{\int Vp(t,S,V)dV}} \tag{6.20}$$

In fact the local volatility function depends on the joint distribution of (S_t,V_t) through the conditional expectation $\mathbb{E}_{\mathbb{P}}[V_t^2|S_t]$ and (6.19) is clearly an example of McKean SDE. Hence, the calibrated LSVM is describing by the following McKean SDE:

$$dS_t=rS_tdt+S_tL(t,S_t,\mathbb{P}_t)V_tdW_t \tag{6.21}$$

where $\mathbb{P}_t$ denotes the distribution of (S_t,V_t) under $\mathbb{P}$ and the local volatility is:

$$L(t,S_t,\mathbb{P}_t)=\frac{\sigma(t,S_t)}{\sqrt{\mathbb{E}_{\mathbb{P}}[V_t|S_t]}} \tag{6.22}$$

The particle method applied to simulate the McKean SDE consists of replacing the unknown marginal distribution $\mathbb{P}_t$ by its empirical distribution $\mathbb{P}_t^N$ of N interacting particles $S_t^{i,N}$ following the dynamics:

$$dS_t^{i,N}=rS_t^{i,N}dt+S_t^{i,N}L(t,S_t^{i,N},\mathbb{P}_t^N)V_t^{i,N}dW_t^i \tag{6.23}$$

and the conditional expectation become:

$$\mathbb{E}_{\mathbb{P}_t^N}[V_t|S_t=S]=\frac{\int Vp_N(t,S,V)dV}{\int p_N(t,S,V)dV}=\frac{\sum_{i=1}^N V_t^{i,N}\delta(S_t^{i,N}-S)}{\sum_{i=1}^N\delta(S_t^{i,N}-S)} \tag{6.24}$$

Instead of the Dirac function we use a regularizing kernel $W_{t,h}(x)=\frac{1}{h}W(\frac{x}{h})$, where W is a fixed, symmetric kernel with a bandwidth h that tends to zero. Examples of the kernel are the exponential kernel $W(x)=\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$ and the quartic kernel $W(x)=\frac{15}{16}(1-x^2)^2 1_{|x|\geq 1}$. Then, the steps to follow to calibrate the leverage function using particles are:

1. Approximation of the leverage function $L^N(t,S)$ with N particles:

$$L^N(t,S)=\sigma(t,S)\sqrt{\frac{\sum_{j=1}^N W_{t,h}(S_t^{j,N}-S)}{\sum_{j=1}^N V_t^{j,N}W_{t,h}(S_t^{j,N}-S)}} \tag{6.25}$$

where $\sigma(t,S)$ the Dupire local volatility inferred from the implied volatility of the market smile.

2. The N particles $(S_t^{i,N}, V_t^{i,N})_{i=1}^N$ following the stochastic dynamics:

$$dS_t^{i,N} = rS_t^{i,N}dt + S_t^{i,N}L^N(t, S_t^{i,N})V_t^{i,N}dW_t^i \qquad i = 1, \cdots, N \tag{6.26}$$
$$dV_t^{i,N} = k(\theta - V_t^{i,N})dt + \lambda V_t^{i,N}(\rho dW_t^{i,1} + \sqrt{1-\rho^2}dW_t^{i,2}) \tag{6.27}$$

where $(k, \theta, \lambda, \rho)$ are the Heston parameters calibrated to the market and $W_t^{i,1}$, $W_t^{i,2}$ are two independent Brownian motion of the i-particle.

To implement the particles method's algorithm we have to discretize the time interval $[0, T]$ in m time-step of width DT and simulate from t_{k-1} to t_k the N particles processes $(S_t^{i,N}, V_t^{i,N})$ for $i = 1, \cdots, N$ following the LSV dynamics (6.26), using a discretization scheme such as an Euler scheme. Then we can compute the leverage function $L^N(t_k, S)$ at time t_k using (7.7) on the i-particle for $i = 1, \cdots, N$. Notice that Hence, in the leverage function calculated on the i-particle $L^N(t, S_t^{i,N})$, there is the interaction between all the $N-1$ particles at every time-step. So, step 1 is more difficult to parallelize with a parallel program using GPU, while step 2 is easily to perform in parallel on a GPU.

6.4 Particle mesh method

The particle method described is the basic particle-particle method (PP) as seen in chapter one in physics and in which the particles interact with each other following the dynamics of the LSV model. The second particle method seen in physics is the particle mesh method (PM) that sets the particles on a grid and at every time step they are remeshed onto the grid. Remeshing the particles on grid reduces the time computation respect with the particle-particle algorithm because of the interactions do not happen among all the particles but only between something particles. Particles methods more advanced involve the combination of the previous ones, generating hybrid particle-particle-particle-Method (PPPM).

From the previous description of the particle method to calibrate the SLV model, we add a 2-dimensional grid of size h and we choose an interpolating function $W_{t,h}(x)$ in which the interaction happens only between the closer particles from that fixed one. The leverage function (7.7) is interpolated onto the grid-points $(x_p)_{p=1}^m$ and then we interpolate back onto the particles $(S_t^{i,N})_{i=1}^N$. Finally, we move the particles with the discretization scheme of their dynamics (6.26). Hence we add to the PP algorithm the following steps:

1. **P → M** *Interpolation onto the mesh*
   We interpolate the approximated local volatility $L^N(t,S)$ from the N particles onto a grid of m points using an interpolating kernel $W(S_t^{i,N} - S_t^p)$ where S_t^p is the grid-point and $S_t^{i,N}$ is the position of the particle i at the time t.

   $$L^N(t,S_t^p) = \sum_{i=1}^{N} L^N(t,S_t^{i,N}) W_{t,h}(S_t^p - S_t^{i,N}) \tag{6.28}$$

   In this step we sum over the particles and the interpolation is done at every time step.

2. **M → P** *Interpolation onto the particles*
   We interpolating back from the mesh to the particles through a different interpolation kernel $\zeta_{t,h}$ and now we sum over the m-grid points:

   $$L^N(t,S_t^{i,N}) = \sum_{p=1}^{m} L(t,S_t^p)\zeta_{t,h}(S_t^{i,N} - S_t^p) \tag{6.29}$$

Finally we reconnect to the PP algorithm and perform the time integration from t_n to t_{n+1} moving the particles using an Euler discretization scheme to their stochastic dynamics (6.26). In the next chapter we see the implementation of the algorithm using CUDA C/C++, a parallel computation by NVIDIA corporation.

Chapter 7

Parallel Program in CUDA C

7.1 An introduction to parallel programming

We have to implement the particle method's algorithm to compute the calibration of the LSV model. The problem can be simulate using basic programming languages such as C/C++ or it can be used a powerful program from the graphic architecture: the parallel computation. Parallel computing techniques accelerate the calculation in the computer simulation. To gain fast computation speed we use the graphics processing unit (GPU). Such implementation of parallel techniques is through compute unified device architecture (CUDA)with an extension in C/C++ developed by NVIDIA. The calibration of SLV with particle method is partially parallelized due to fact that in the leverage function the particles interact between each other at every time-step. Instead, it is easier to compute in parallel each trajectory of the particles since they are independent and identically distributed random variables that evolve independently of each other. We first introduce some basic ideas of NVIDIA CUDA parallel programming and then apply them in the calibration with particle method.

The CUDA programming system connect two heterogeneous computing driver: the *Host* and the *Device*. The Host refers to the CPU and its memory, the Device refers to the GPU (graphics processing unit) and its memory. The aim of the parallel computing is to delegate laborious calculations from CPU to GPU driver which it has got hundred of core that can work simultaneously. So, a simple processing flow is to copy the input data from CPU memory to GPU memory, load the GPU code and execute it and finally come back coping the results from GPU memory to CPU memory. Consequently, in the source code there are the device and the host code. In a classic C code, the new syntactic elements to add the device code is the function '*_global_void* *device_function(void){}*'. The keyword is '*_global_*' that indicates a function that runs on the device and is called from the host code. In the main C function, the new elements that call from host code to device code are the triple angle brackets *kernelCUDA* $<<< 1, 1, >>>$ (*argument*). A simple example where the function execute on the device does nothing is in table 7.1.

```
_global_void device_function (void) {
}
int main(void) {
            device_function<<< 1,1 >>>();
            printf("Void function" \n");
            return 0;
}
```

Table 7.1: Void function in CUDA C

The powerful of the parallel programming is that CUDA has a hierarchy of threads structured on two level:

1. **grid**: an orderly grid of blocks;

2. **block**: an orderly collection of threads;

A block may contain at most 1024 threads. Grid and blocks may have 1, 2 or 3-dimension. Threads are uniquely identified by the coordinates:

1. *blockIdx*: index of block into the grid;

2. *threadIdx*: index of thread into the blcok;

The dimensions of grid and block are specified by the variables *blockDim* (dimension of block, measured in threads) and *gridDim* (dimension of grid, measured in blocks). To understand the usefulness of the parallel calculation we continue with a simple function that performs the vector addition.

```
#define N 1024          //vector size
#define M 32            //dimension of block = threads per block

_global_void add (int *a, int *b, int *c) {
            int index = threadIdx.x + blockIdx.x * blockDim.x;
             if(index< N)
                   c[index] = a[index] + b[index];
}
int main(void) {
             int *a, *b, *c;              //host copies of a,b,c
             int *d_a, *d_b, *d_c;        //device copies of a,b,c
             int size = N * sizeof(int);

            //Allocate space for device copies of a,b,c
            cudaMalloc((void **)&d_a, size);
            cudaMalloc((void **)&d_b, size);
            cudaMalloc((void **)&d_c, size);

             //Allocate space for host copies a, b,c and set up input values
             a = (int *)malloc(size); random_ints(a, N);
             b = (int *)malloc(size); random_ints(b, N);
             c = (int *)malloc(size);

             //Copy input to device
             cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
             cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);

             //Launch add() kernel on GPU
             add<<<N/M,M>>>(d_a, d_b, d_c);

             //Copy result back to host
             cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);

            //Cleanup
             free(a); free(b); free(c);
             cudaFree(d_a; cudaFree(d_b); cudaFree(d_c);
             return 0;
}
```

Table 7.2: Vectors sum in CUDA C with dimGrid=N and dimBlock=M

In table (7.2), the *global* function is executed on the device and it has been called from the host. The recursive steps to include CUDA C program in the *main* C function are the following:

1. Allocate space in the Device for the copies of the vectors a,b,c with the CUDA function *cudaMalloc.*

2. Allocate space for host copies of a,b,c and set up the input values of the vectors.

3. Copy the input to Device with the function *cudaMemcpy* from the Host to the Device.

4. Launch and execute N times in parallel the *add()* function on GPU using the triple angle brackets *add* $>>> N/M$. Each parallel launch of *add()* is called *block*. N/M = dimension of the grid = number of blocks. The index of each block is given by *blockIdx.x* and in the Device each block can execute in parallell. A block can be split into M parallel *threads* with index *threadIdx.x.* Parallel threads are launched in main function with the brackets : $M >>> ()$. Hence, with N block each of which has M trheads, the unique index is given by *index=threadIdx.x* + *blockIdx.x* $*$ *blockDim.x*, where *blockDim.x* is the dimension of a block.

5. Copy the result back to the Host using the function *cudaMemcpy* from the Device to the Host.

6. Cleanup the memory of the Device with the function *cudaFree.*

An important feauture is that between threads within a block, data is shared by the function _*shared*_ but data is not visible to threads in other blocks. Then, with function _*syncthreads()* all threads within a block are synchronized that guarantees all the data is available.

7.2 Exemple of a C vs CUDA C code

A brief but clear example of a program that executes the sum of vectors and shows the main difference between C and CUDA program is the matrix sum algorithm below.

Matrix sum in C	*Matrix sum in CUDA C*
#**define** N 1000	
#**define** M 1000	
int main (){	**void** gpuMatrixAdd
double A[N][M], B[N][M], C[N][M];	(**double** *A, **double** *B, **double** *C) {
for($i = 0$; $i <$ N; $i++$)	*//index is a unique identifier of each GPU thread*
for($j = 0$; $j <$ M; $j++$)	**int** *index* $= \cdots$
C[i][j] = A[i][j] + B[i][j];	C[*index*] = A[*index*] + B[*index*];
}	}

In the first C code, at every step of the *for* cycle, the CPU memory is occupied one time at each step. In the parallel program the GPU executes simultaneously N processes in the same time.

7.3 Calibration with particle method in CUDA C/C++

Now we use parallel programming techniques to the calibration of the stochastic local volatility model with particle method. Recalling the SLV model, the spot process S_t and the stochastic variance V_t, follow the dynamics

$$dS_t = rS_t dt + S_t^i L^N(t, S_t)\sqrt{V_t^i} dW_t^1, \quad S_0 = s \tag{7.1}$$

$$dV_t = k(\theta - V_t)dt + \lambda\sqrt{V_t}\rho dW_t^2, \quad V_0 = v \tag{7.2}$$

$$dW_t^1 \cdot dW_t^2 = \rho dt \tag{7.3}$$

with $r = r_d - r_f$ the deterministic interest rate and leverage function that depends on the conditional expectation of the variance

$$L(t, S) = \frac{\sigma(t, S)}{\sqrt{\mathbb{E}[V_t|S_t = S}} \tag{7.4}$$

where $\sigma(t, S)$ is the Dupire local volatility inferred from the market smile of the implied volatility $\sigma_{IV}(T, K)$ with maturity T and strike K. We approximate the law $\mathbb{P}_t$ on which the leverage function depends, with th empirical distribution of N particles $(S_t^{i,N}, V_t^{i,N})_{i=1}^N$ that are the solution of the two $\mathbb{R}^N$-dimensional linear SDEs

$$
\begin{aligned}
dS_t^{i,N} &= rS_t^{i,N}dt + S_t^{i,N}L^N(t, S_t^{i,N})\sqrt{V_t^{i,N}}dW_t^{i,1} && (7.5)\\
dV_t^{i,N} &= k(\theta - V_t^{i,N})dt + \lambda\sqrt{V_t^{i,N}}\ (\ \rho dW_t^{i,1} + \sqrt{1-\rho^2}dW_t^{i,2}\ ) && (7.6)
\end{aligned}
$$

and the conditional expectation of the leverage function on the i-particle is approximated by

$$
L^{i,N}(t, S_t^{i,N}) = \sigma(t, S_t^{i,N})\sqrt{\frac{\sum_{j\neq i,j=1}^{N} W_{t,h}(S_t^{j,N} - S_t^{i,N})}{\sum_{k\neq i,k=1}^{N} V_t^{k,N}W_{t,h}(S_t^{k,N} - S_t^{i,N})}} \qquad (7.7)
$$

We can see clearly the interaction between the $N-1$ particle with the i particle in the local volatility (7.7). We discretize the N particle dynamics (7.5)under the Euler discretization scheme with m time-steps and from the time t_{k-1} to the time t_k we have

$$
\begin{aligned}
S_{k+1}^{i,N} &= S_k^{i,N} + rS_k^{i,N}DT + S_k^{i,N}L_k(S_k^{i,N})V_k^{i,N}\sqrt{DT}Z_k^{1,i} && (7.8)\\
V_{k+1}^{i,N} &= V_k^{i,N} + k(\theta - (V_k^{i,N})^+)DT + \lambda\sqrt{(V_k^{i,N})^+}\ [\rho\sqrt{DT}Z_k^{1,i} + \sqrt{1-\rho^2}\sqrt{DT}Z_k^{2,i}] && (7.9)\\
V_{k+i^i} &= (V_{k+1})^+ && (7.10)
\end{aligned}
$$

Where the time interval is $DT = t_{k+1} - t_k = \frac{T}{m}$ and the variance process is discretized by a full truncation scheme that produce the smallest bias for preserving the positivity of the variance process. The $2\times N$ Brownian motions $(W_t^{i,1}, W_t^{i,2})_{i=1}^N$ are simulated by $\Delta W_k^{i,1} = \sqrt{DT}\cdot Z^{i,1}$ and $\Delta W_k^{i,2} = \sqrt{DT}\cdot Z^{i,2}$ with $(Z^{i,1}, Z^{i,2}) \sim \mathcal{N}_{0,1}$, i.e. $Z^{i,1}$ and $Z^{i,2}$ are two standard normal random numbers for the particle i at time t_k. The particles dynamics (7.8) is easily to compute in parallel because of the evolution of the N independent and identically distributed particles, while the leverage function is more difficult to parallelize because of the interaction between all the particles at every time-step DT. The power of the parallel computing is that the simulation of the N particles dynamics is performed simultaneously on GPU. Furthermore, the generation of the $2\times N$ standard normal random numbers $(Z^{i,1}, Z^{i,2})_{i=1}^N$ are implemented in parallel. The normally distributed random numbers are generated by the device API functions including the file *curand_ kernel.h* in the program of CUDA C/C++.

```
_global_void setup_ kernel (curandState *state, unsigned long long Seed) {
        int id = threadIdx.x + blockIdx.x * blockDim.x;
                curand_init(Seed, id, 0, &state[id]);
}

_global_void generate_ normal_ kernel
(curandState *globalState, double *DevResult) {
        int id = threadIdx.x + blockIdx.x * blockDim.x;
                curandState localState = globalState[id];
                double random = curand_normal_double(&localState);
                globalState[id] = localState;
                DevResult[id] = random;
}
```

Table 7.3: Device functions to generate normally distributed random numbers

1. First step is the generation of a sequence of pseudorandom numbers with the device function: _device_void *curand_ init*(unsigned long long seed, unsigned long long sequence, unsigned long long offset, curandState *state). This function sets up an initial state using a given seed, sequence number and offset within the sequence. Different seeds produce different sequences.

2. Second step is the generation of a single normally distributed numbers with double precision with mean 0 and standard deviation using the device API function: _device_void *double_ curand_ normal* (curandState *state), with input the previously set up state.

Table (7.3) shows the algorithm to generate of normally distributed random numbers in CUDA C/C++ code to compute the Brownian motion on the device of the GPU. The first device function *setup_ kernel* gives the global state that is used in the second device function *generate_ normal_ kernel* to generate normally distributed random numbers allocated in DevResult. Each thread gets a different sequence numbers and no offset. For efficiency, globalState is copied to local memory and then normally random number are generated in localState and copied back to global memory.

Now we have to run in parallel the particle dynamics (7.8), we choice to occupied the device memory with a 1-dimensional grid and with a single particle per thread with n_{thread} the total thread and grid size N. Hence there are N/n_{thread} total blocks. At every time-step the device memory is deleted and the particles can move interacting each other. The difficulty to efficiently occupied the GPU cores are the interaction between particles at every time-step in the leverage function and it can't be simulated simultaneously in parallel. While to simulate the N independently particles process, in the main C function the sequence to develop are the following:

1. Set up the initial value at time t_0 of the vectors $(S_0^i, V_0^i)_{i=1}^N$ and $(L_0^i)_{i=1}^N$ that represent the initial particles position and the initial value of the leverage function equal to the Dupire local volatility at time 0.
2. Begin the time cycle.
3. Allocate space for the vectors of normally random numbers $(Z^{i,1}, Z^{i,2})_{i=1}^N$ at time t_k on the host and allocate space for the copies of the vectors $(S_{dev}^i, V_{dev}^i)_{i=1}^N$ and $(Z_{dev}^{i,1}, Z_{dev}^{i,2})_{i=1}^N$ on the device. In addition, allocate space for the two vector states $(State_{dev}^{i,1}, State_{dev}^{i,2})_{i=1}^N$ on the device to the generation of the normally random numbers and the leverage function on the device $(L_{dev}^i)_{i=1}^N$,
4. Copy the host vectors in the device vectors: $S^i, V^i, L^i \rightarrow S_{dev}^i, V_{dev}^i, L_{dev}^i$.
5. Set up the pseudo random generation states launching N time in parallel the *setup_kernel* for (Z_{dev}^1, Z_{dev}^2) with two different seed:
   setup_kernel $<<<\ N/n_{thread}\ ,\ n_{thread}\ >>>$ $(State_{dev}^1,$ seed1)
   setup_kernel $<<<\ N/n_{thread}\ ,\ n_{thread}\ >>>$ $(State_{dev}^2,$ seed2)
6. Generate and use normal pseudo-random states launching on device the global function:
   generate_normal_kernel $<<<\ N/n_{thread}\ ,n_{thread}\ >>>\ (State_{dev}^1,\ Z_{dev}^1)$;
   generate_normal_kernel $<<<\ N/n_{thread}\ ,\ n_{thread}\ >>>\ (State_{dev}^2,\ Z_{dev}^2)$;
7. simulate the N particles processes (7.8) in parallel by the launch on GPU of the device function:
   SimulateProcess $<<<\ N/n_{thread}\ ,\ n_{thread}\ >>>\ (S_{dev},\ V_{dev},\ Z_{dev}^1,\ Z_{dev}^2,\ L_{dev})$;
8. Copy device memory to host: $S_{dev}^i, V_{dev}^i, L_{dev}^i \rightarrow S^i, V^i, L^i$.
9. Compute the leverage function (7.7) with the interacting particles with values the vectors $(S^i, V^i)_{i=1}^N$.
10. Cleanup the memory.
11. Set the t_{k+1} time.

12. Iterate from step three up to the maturity T.

Table 7.4: Device global kernel for the particle process

```
_global_void SimulateProcess
(double *d_S, double *d_V, double *d_Z1, double *d_Z2, double d_*L) {
        int id = threadIdx.x + blockIdx.x * blockDim.x;
                d_S[id] += r*d_S[id]*DT + d_L[id]*d_S[id]*sqrt(d_V[id]*DT)*d_Z1[id];
                d_V[id] += K*(θ-fabsf(d_V[id]))*DT + λ*sqrt(fabsf)(d_V[id])*
                        (ρ*sqrtT*d_Z1[id]+sqrtT2*d_Z2);
                d_V[id] = fabsf(d_V[id]);
}
```

Eventually, in the same calibration's code, it is possible to pricing the option calculating the expected value of the payoff on the final value of the path of the particle at the maturity time T. Table (7.4) contains the CUDA kernel function to compute on the device the particle process (S_k^i, V_k^i) at time t_k. The device global kernel that computes on GPU the particle process (S_k, V_k) at time t_k has as input: the device's vectors (d_)-(S_{k+1}, V_{k-1}) at the previous time t_{k-1}, the normally distributed random numbers calculated on the device d_Z1 and d_Z2 and the leverage function on the device d_L. Every thread corresponds at one particle process.

7.4 Numerical results of the calibration

We calibrate the leverage function of SLV model to market data on EUR/USD example. The conventions in the foreign option market is that implied volatilities are quoted against the corresponding delta value of vanilla call $\Delta_c = \frac{\partial V_{call}}{\partial S}$ or put $\Delta_c = \frac{\partial V_{put}}{\partial S}$ options. The delta measures the change of option value as the spot rate changes. Typical delta values are $10 - \Delta$ and $25 - \Delta$ for both the two type of options. For the convention, the implied volatility is given to the at-the-money volatility σ_{ATM}, $10 - \Delta$, $25 - \Delta$ butterfly (BF) and $10 - \Delta$, $25 - \Delta$ risk reversal (R)R. The σ_{ATM} is defined to be $0 - \Delta$ straddle, where the strike is chosen such that a call and put have the same delta value with different signs. BF, RR and volatilities of calls and puts for different delta values σ_{Δ_c} and σ_{Δ_p} are given by

$$BF = \frac{\sigma_{\Delta_c} + \sigma_{\Delta_p}}{2} - \sigma_{ATM} \tag{7.11}$$

$$RR = \sigma_{\Delta_c} - \sigma_{\Delta_p} \tag{7.12}$$

$$\sigma_{\Delta_c} = \sigma_{ATM} + BF + \frac{RR}{2} \tag{7.13}$$

$$\sigma_{\Delta_p} = \sigma_{ATM} + BF - \frac{RR}{2} \tag{7.14}$$

The traded market prices are listed in the table (7.4) while in table (7.4) there are the value of the implied volatility at-the-money σ_{ATM}, (where the strike price is equal to the spot price $K = S_0$) and in the columns on the side there are the variation of the volatility in % at 5 different strike prices, $10 - \Delta$, $25 - \Delta$ BF and $25 - \Delta$, $10 - \Delta$ RR respectively.

Table 7.5: EUR/USD parameter settings

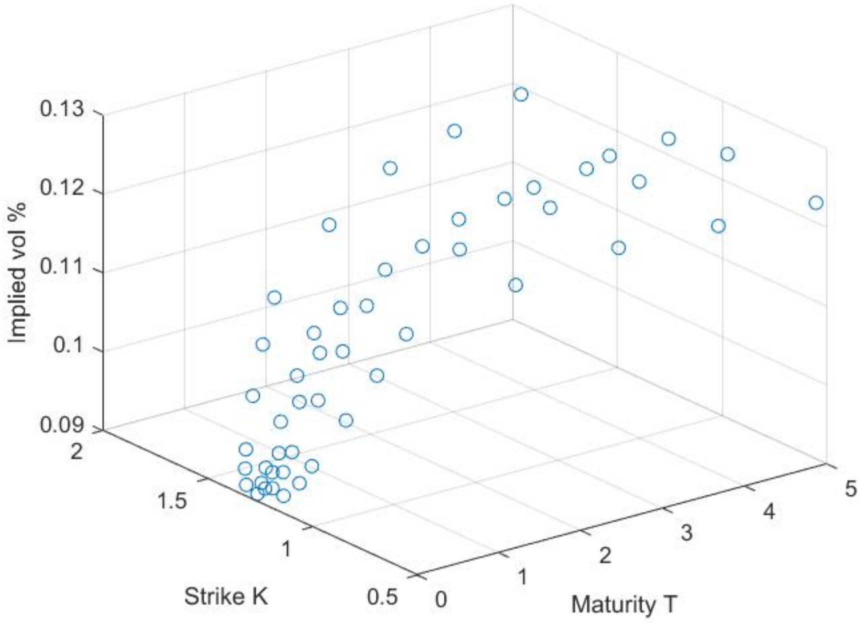
Domestic currency	USD
Foreign currency	EUR
Date	23 August 2012
Spot	1.257 USD per EUR

Table 7.6: EUR/USD market data (in %)

Maturity	Domestic yield	Foreign yield	$10-\Delta$ BF	$25-\Delta$ BF	σ_{ATM}	$25-\Delta$ RR	$10-\Delta$ RR
1m	0.41	0.04	0.51	0.17	9.15	-0.68	-1.22
2m	0.51	0.11	0.70	0.22	9.33	-1.18	-2.12
3m	0.66	0.23	0.94	0.28	9.55	-1.50	-2.73
6m	0.95	0.47	1.25	0.36	10.13	-1.92	-3.55
9m	1.19	1.62	1.42	0.42	10.68	-2.10	-3.94
1y	1.16	0.64	1.62	0.47	11.18	-2.25	-4.22
2y	0.60	0.03	1.52	0.44	11.68	-2.32	-4.40
3y	0.72	0.03	1.28	0.37	12.00	-2.30	-4.37
4y	0.72	0.03	1.19	0.36	12.10	-2.38	-4.50
5y	0.72	0.03	1.21	0.38	12.20	-2.43	-4.60

The domestic yield r_d and the foreign yield r_f vary at different maturities. The market data of the table (7.4) are represented in figure (7.4) in terms of 5 different strikes K for each of the 10 maturities T. The strikes is obtain from the 10% or 25% of the variation from the σ_{ATM} divided by the square of the maturity.

Figure 7.1: Scatter EUR/USD market implied volatility surface $\sigma_{IV}(T, K)$



From the scatter surface of the data market, the fitting of the implied volatility surface $\sigma_{IV}(T,K)$ is done with MATLAB. The points are fitted by a polynomial surface of degreee 2 in T and 2 degree in K with a 95% of confidence for the coefficients. The resulting plot is in the figure (7.4).

Figure 7.2: Fit of the EUR/USD market implied volatility surface $\sigma_{IV}(T,K)$

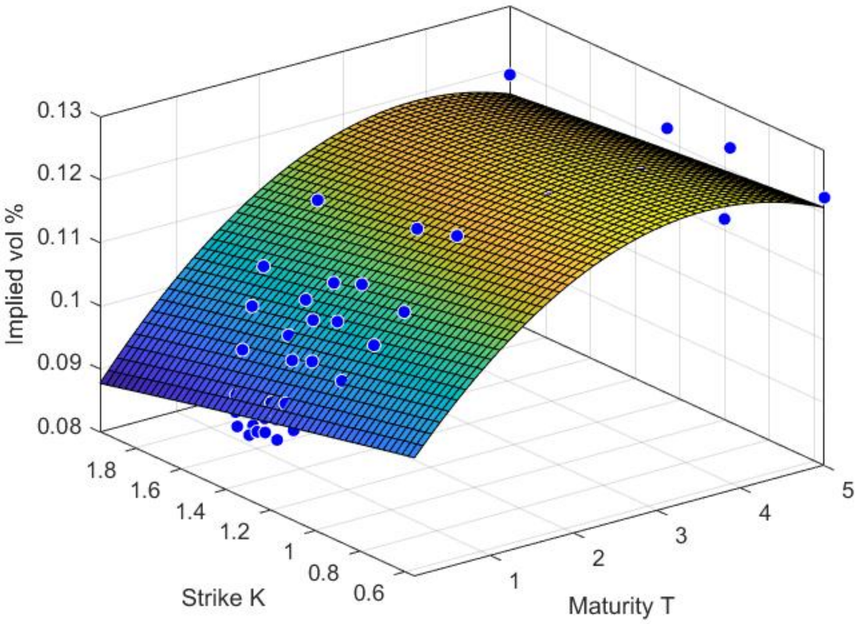


Figure (7.4) shows that the volatility is left-skewed with higher volatilities for in-the-money ITM strikes ($K < S_0$) than out-of-the-money OTM strikes ($K > S_0$). In order to represent the left-skewness, a negative ρ is used in the SLV model. Then, from the implied volatility surface $\sigma_{IV}(T,K)$ we compute the Dupire local volatility $\sigma_{LV}(t,S)$ with the formula

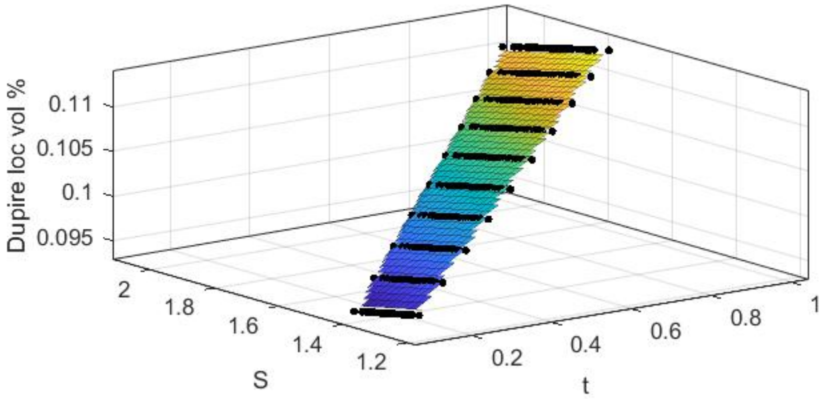
$$\sigma_{LV}(t,S) = \sqrt{\frac{\sigma_{IV}^2 + 2\sigma_{IV}T\frac{\partial\sigma_{IV}}{\partial T} + 2(r_d - r_f)\sigma_{IV}KT\frac{\partial\sigma_{IV}}{\partial K}}{(1 + d_1(S,K)K\sqrt{T}\frac{\partial\sigma_{IV}}{\partial K})^2 + \sigma_{IV}K^2T[\frac{\partial^2\sigma_{IV}}{\partial K^2} - d_1(S,K)\sqrt{T}(\frac{\partial\sigma_{IV}}{\partial K})^2]}}\Big|_{K=S,\,T=t} \tag{7.15}$$

The partial derivatives are computed on the polynomial equation fitting the implied volatility surface $\sigma_{IN}(T,K)$ then setting $K = S$ and $T = t$. $d_1(S,K)$ is defined by

$$d_1(S,K) = [log(S/K) + (r + \sigma^2/2)]T/\sigma\sqrt{T} \tag{7.16}$$

where $r = r_d - r_f$ and σ is the Black-Scholes volatility. The Dupire local volatility scatter surface obtained is in the figure (7.4).

Figure 7.3: Dupire local volatility $\sigma(t, S)$



From figure (7.4), it is evident that only Dupire local model do not reproduces the curvature of the implied volatility surface.

We continue with the calibration of the SLV model setting as initial conditions

$$S_0 = 1.257 \tag{7.17}$$

$$V_0 = 0.008 \tag{7.18}$$

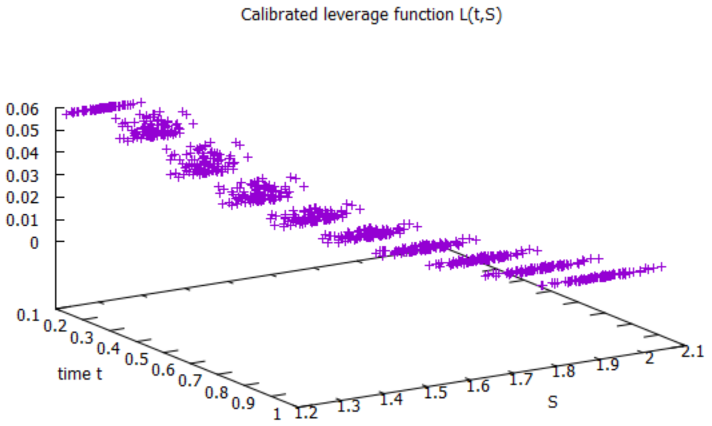
The Heston parameters considered are calibrated to market implied volatility data at the maturity of 3 months $T = 0.3$. The lack of the Heston parameters calibration in the leverage function program, affects the scale of the results.

Table 7.7: Heston parameters in the SLV model

Period	k	θ	λ	ρ
3m − 6m	0.816	0.039	0.430	−0.474

With these parameters, the calibrated leverage function is presented in figure (7.4).

Figure 7.4: Leverage function $L(t, S)$



The surface of the leverage function is realized from 1 month to 1 year because of the used of the calibrated Heston parameters of the period of 3 months and 6 months. For a complete calibration of the local volatility, it is necessary the calibration of the Heston parameters to the market data. The input value of the spot prices S_t are the positions of the particles at every time step. The value of the leverage function is calculated in % as the data available and reproduce the skewness of the implied volatility from the market. Hence, only the Dupire local volatility of a stochastic model is not enough to reproduce the volatility smile of the implied data market and with the adding of a stochastic part, the model become a stochastic-local volatility model that reproduces the skewness of the implied volatilty after a calibration of the local part, the leverage function.

7.5 Conclusion

In this thesis we saw the calibration of stochastic local volatility model with particle method. We first dealt deterministic particle method applied in physics and then its stochastic application in option pricing. We introduced the Black-Scholes model, the local and the stochastic models from which the SLV model originates. The SLV model have to be calibrate in order to replicate the implied volatility surface from the market. The main difficulty arises from the computation of the Fokker-Planck equation, the solution of the McKean stochastic differential equation of the model. It is a nonlinear equation in which the leverage function depends on the probability distribution of the process. The stochastic formulation of particle method is applied to solve SDE of Mckean type. Hence it is an efficient approximation method to solve the calibration's problem. In particular, the probability distribution of the price process is approximated by N identically and independent particles with the same empirical distribution that converges at the true distribution thanks to the theorem of the propagation of chaos. Therefore particle method is a robustly method that converge for $N \to \infty$ and allows to calibrate the leverage function to the implied volatility surface from the data market. Lastly, particle method enables an interconnection between different research areas from the theory of stochastic process to the simulation of physical flows and financial modeling for option pricing.

Bibliography

[1] Bergomi L. [2016]. Stochastic local volatility modeling, Chapman & Hall/CRC financial mathematics series.

[2] Carmona R.A, Del Moral P., Hu P, Oudjane N. [2010]. Numerical methods in finance, Springer.

[3] Flannery B. P., Teukolsky S.A & Co. [1992]. Numerical recipes in C, The art of scientific computing, Second ed., Cambridge University Press.

[4] Glasserman P. [2004]. Monte Carlo methods in financial engineering, Springer.

[5] Guyon J., Henry-Labordère P. [2013]. Nonlinear option pricing, Chapman & Hall/CRC, Financial Mathematics Series.

[6] Guyon J., Henry-Labordère P. [2013]. The smile calibration problem solved, Bloomberg quant seminar.

[7] Hockney R.W, Eastwood J.W. [1988]. Computer simulation using particles, Adam Hilger.

[8] In'Hout K.J., Foulon S. [2010]. ADI finite differences schemes for option pricing in the Heston model with correlation. International journal of numerical analysis and modeling, Vol. 7, N. 2, pp. 303-320.

[9] Koumoutsakos P., Cottet G.H, Rossinelli D. [2008]. Flow simulations using particles Bridging computer graphics and CFD, Class notes. http://www.cse-lab.ethz/teaching/classes/mulsup.html.

[10] Henry-Labordère P. [2009]. Calibration of local stochastic volatility model to market smiles: a Monte-Carlo approach, Risk magazine.

[11] Hull J. C. [2009]. Options, futures and other derivatives, Seventh ed., Pearson Prentice Hall.

[12] Musiela M., Rutkowski M. [2000]. Martingale methods in financial modelling, Second ed., Stochastic modelling and applied probability, Springer.

[13] Pascucci A. [2011]. PDE and martingale methods in option pricing, Bocconi University Press, Bocconi & Springer Series.

[14] Pascucci A. [2016-2017]. Lezioni di probabilità e statistica, Corso di laurea triennale in matematica, a.a.2016-2017.

[15] Quarteroni A., Saleri F. [2006]. Introduzione al calcolo scientifico Esercizi e problemi risolti con MATLAB, Terza ed., Springer.

[16] Quarteroni A., Saleri F. [2008]. Calcolo scientifico Esercizi e problemi risloti con MATLAB e Octave, Quarta ed., Springer.

[17] Tataru G., Fisher T. [2010]. Stochastic local volatility, Quantitative development group, Bloomberg.

[18] Tataru G., Fisher T., Yu J. [2012]. The Bloomberg Stochastic local volatility model for FX exotics, Quantitative finance development Bloomberg L.P.

[19] Tian Y. [2013]. The hybrid stochastic-local volatility model with applications in pricing FX options, Thesis, School of Mathematical sciences Monash University, Australia.

[20] Zeller C. [2011] Cuda c/c++ basics supercomputing 2011 tutorial, NVIDIA Corporation.

[21] NVIDIA, CUDA Toolkit Documentation -v8.0.61 [2017]. http://docs.nvidia.com/cuda/index.html