Uncertain Monte Carlo

Christian Kamtchueng
CTK corp, London, e-mail: christian.kamtchueng@gmail.com[1]

Abstract
Recent troubles in the financial industry have led market participants to reconsider some influential pricing assumptions, such as the completeness of the market (mathematically, there is not only one risk neutral probability, as financially speaking all tradable assets can be replicated in different ways). Since the uncertainties in the market are real, practitioners have become more open to reconsider the basics. Indeed, some valuation methods initially rejected by the market now begin to be valuable. In this paper, we develop some numerical procedures that give us the ability to solve generically the valuation in an uncertain world.

Keywords
uncertain Monte Carlo, uncertain Volatility Model, uncertain tree, quantization, Markov Chain

1 Introduction

In these stressful times, practitioners have begun to realise how difficult it is to rely on market information and even more difficult to extrapolate it. For instance, market volatility is far from being perfectly implied. It is well known that the standard option pricing model of Black–Scholes (1973 [1]) and Merton (1976 [2]), which considers a flat volatility for the underlying, is inconsistent with observed market prices. In such model, the observed skew is not the only reason for this inconsistency, indeed it is also due to the bid-ask information linked to the asset liquidity which is not considered in the derivative valuations (although is taken into account via an "'add-on"' post-evaluation called hedging adjustment). The standard way to price exotics is based on the market's transformation into an ideal world called "'risk neutral world"'.

Avellaneda (1995 [4]) has developed the Uncertain Volatility Model (UVM), later Kamtchueng (2008 [14]) proposed an extension to the equity correlation (Uncertain Correlation Model). For many reasons that we will mention later, those models used to be put aside. However, the interest on this type of view has increased significantly in the past two years. Our scope is to provide a generic methodology (not product dependent) to evaluate the exotic in an uncertain world via a panel of alternative numerical methods. Our framework can be compared to recent works based on second order BSBEs, see for instance Touzi (2004, [9]; 2007 [12]; 2009 [13]), Guyon (2010 [15]). We do not go in this direction, rather, we provide a naive forward way to resolve our uncertain pricing problem, also a backward solution of the BSB via Markov Chain. Firstly we will describe the UVM theory. After presenting the disadvantages of the usual method, we will introduce the Uncertain Tree, then the Uncertain Monte Carlo and try to extend our approach to the multi-asset case. In Section 4, we present various numerical method and describe each of them precisely, in section 5, we are focus on the results for the single asset and multi-asset case.

Our main results are the following: by considering a Double Markov Chain we managed to reduce considerably the time computation of the uncertain price. Indeed the presented algorithm avoids any Monte Carlo of $O\left(N_S^{N_S}\right)$ order, or $O\left(N_S^{N_m}\right)$ with N the number of paths and N_m the number of mesh paths in order to compute the Gamma. Accuracy is correct and can be adjusted. This is a big achievement which in addition can be extended to the multi-factor or multi-asset framework. We managed to elaborate some efficient numerical algorithms to determine the uncertain price of multi-asset or multi-factor product via Markov Chain construction.

2 Notations

- N_S number of paths
- N_t number of steps
- N_m number of mesh's paths
- $N_{B,t}$ number of Buckets at time t
- N_B maximum of the $\left(N_{B,t}\right)_t$, $t \in \{1 \ldots N\}$
- $\hat{Mesh}(S_t)$ intermediate Mesh construct via the intermediate density at S_t
- $\hat{Mesh}_t^n = \hat{Mesh}(S_t^n)$ intermediate Mesh construct via the intermediate density at S_t^n
- $\hat{\Gamma}_t^n$ intermediate local gamma construct via the intermediate density at S_t^n
- $\hat{\Delta}_t^n$ intermediate local delta construct via the intermediate density at S_t^n
- $Weight^{\Gamma}$ intermediate local gamma construct via the intermediate density
- $Weight_{t,N_t}^{\Gamma,n}$ intermediate local gamma construct via the intermediate density at S_t^n for the maturity T: $\mathbb{E}_t^{\mathbb{Q}}\left[Weight_T^{\Gamma} | S_t = S_t^n\right]$
- $Weight_{n,k}^{\Gamma,*}$ intermediate local gamma construct via the density f^* at $\hat{y}_t^n$ to the bucket $\hat{y}_t^k$: $\mathbb{E}_t^{\mathbb{Q}^*}\left[Weight_{t+1}^{\Gamma} 1_{\{S_{t+1} \in B_{t+1}^n\}} | S_t \in B_t^k\right]$ with $* \in \{+,-\}$
- $\hat{f}$ intermediate density related to the mid level volatility
- f^+ the density related to the upper bound volatility
- f^- the density related to the lower bound volatility
- $\hat{S}$ path generated with the density $\hat{f}$
- $\hat{S}^+$ path generated with the density f^+
- $\hat{S}^-$ path generated with the density f^-
- $\hat{B}_t^k$ bucket is represented by $\hat{y}_t^k$ and construct via Monte Carlo simulation of S following the $\hat{f}$ density
- $B_t^{k,*}$ bucket is represented by $y_t^{k,*}$ and construct via Monte Carlo simulation of S following the f^* density with $* \in \{+,-\}$
- $\hat{y}_t^k$ called centroid can be seen as: $\mathbb{E}_t^{\mathbb{Q}}\left[S_t | S_t \in B_t^k\right]$
- $y_t^{k,*}$ called centroid can be seen as: $\mathbb{E}_t^{\mathbb{Q}^*}\left[S_t | S_t \in B_t^k\right]$ with $* \in \{+,-\}$
- $h^U(t,S_t)$ the uncertain price process
- $h(t,S_t)$ the uncertain price process defined by backward propagation.

[1]The opinions of this article are those of the author and do not reflect in any way the views or business of his employer.

- H is the payoff of the contingent claim
- $\Gamma_t, S_t := \frac{\partial^2 h^U(t, S_t)}{\partial s^2}$
- $\hat{h}(t,S_t)$ the price process computed with the intermediate density for the stock.
- $\hat{\Gamma}_t, S_t := \frac{\partial^2 \hat{h}(t, S_t)}{\partial s^2}$
- $\tilde{S}$ is the uncertain path generated with a uncertain vol funtion of $\hat{\Gamma}$
- $\check{h}(t,S_t)$ the price process computed backward via an importance sampling method.
- $\check{\Gamma}_t, S_t := \frac{\partial^2 \check{h}(t,S_t)}{\partial s^2}$
- $\check{S}$ is the uncertain path generated with a uncertain vol funtion of $\check{\Gamma}$
- $\bar{h}(t,S_t)$ the price process computed backward via an double Markov Chain.
- $\bar{\Gamma}_t, S_t := \frac{\partial^2 \bar{h}(t, S_t)}{\partial s^2}$
- $\bar{S}$ is the uncertain path generated with a uncertain vol funtion of $\bar{\Gamma}$

3 Uncertain volatility model

Introduced by Avellaneda, the Uncertain Volatility Model (see [4]) is a way to estimate the risk in the volatility (in an incomplete market, the vol is more than risky, it is uncertain). The UVM solution is the solution of the Black Scholes Barenblatt equation (see for instance [5] 2001). σ_s, the volatility of S_t, is bounded by σ^+ and σ^-:

$$\sigma^- \leq \sigma_S \leq \sigma^+ \tag{1}$$

BSB equation will be referred to as:

$$\begin{cases} \frac{\partial h}{\partial t} + \frac{1}{2}\sigma^{*2} S_t^2 \frac{\partial^2 h}{\partial s^2} & = 0 \\ h_T & = H(S_T) \end{cases} \tag{2}$$

(Notice that it is assumed that the interest rate is null.)

If one is short the option, σ^* is defined as described below.

$$\sigma^* := 1_{\{\Gamma_S > 0\}} \sigma^+ + 1_{\{\Gamma_S < 0\}} \sigma^- \tag{3}$$

$$\Gamma_S := \frac{\partial^2 h}{\partial s^2}$$

(Notice that the solution to the BSB equation is equivalent to the solution to the BS equation with the underlying's volatility equal to σ^* volatility. This can be interpreted as the worst case scenario in terms of hedging cost given the boundaries within the hypothesis of the uncertain parameter.)

The main disadvantage of this approach is PDE implementation, which given the fact that the dynamic is unknown, remains the only stable way to implement this model. We will try in this paper to described some alternative numerical methods.

The BSB equation is defined as follows in the multi-underlying case:

$$\begin{cases} \frac{\partial h}{\partial t} + max_{\{(\sigma_i^*)_{i=1\ldots N} | \forall 1 \leq i \leq N, \sigma_i^* \in [\sigma_i^-, \sigma_i^+]\}} \left\{ \frac{1}{2} \sum_{i,j=1}^N \sigma_i^* \sigma_j^* \rho_{i,j} S_t^i S_t^j \frac{\partial^2 h}{\partial s_i \partial s_j} \right\} & = 0 \\ h_T & = H\left((S_T^i)_{i=1\ldots N} \right) \end{cases} \tag{4}$$

In the rest of this paper, we will propose to solve the BSB equation via other numerical methods. So we put ourselves in the UVM context but this approach can be generalized to other parameters in an uncertainty world.

4 Uncertain tree

The binomial tree is a well-known numerical method - we propose in this section to extend it to take into account an uncertain world. In the context of the Uncertain Volatility Model (UVM), the dynamic of our underlying is determined by the local gamma of the option. In fact with this type of model, the Profit and Loss is our major focus and not the asset itself. We will build our tree by considering the following algorithm:

- $\hat{S}_0^{*,1} = s, ^* \in \{+,-\}$.
- $\hat{S}_t^{*,j}$ is the node j at time t, from $\hat{S}_t^{*,j}$ we compute 4 nodes $\hat{S}_{t+1}^{*,f^u(j)}$ and $\hat{S}_{t+1}^{*,f^d(j)}$ with the volatility $\sigma^*, ^* \in \{+,-\}$. $\hat{S}_{t+1}^{*,f^u(j)} = \hat{S}_t^{*,j}(1 + Up(\sigma^*))$ $\hat{S}_{t+1}^{*,f^d(j)} = \hat{S}_t^{*,j}(1 + Dwn(\sigma^*))$

In fact at time t, we do not care about the origin of the node $\hat{S}_t^{*,j}$, so we can identify it as $\hat{S}_t^j$. We use $f^u(n)$ and $f^d(n)$ as functions that determine the number of the nodes computed via the node n from the previous time (see Figure 4 in section UCM Graphs).

For the backward diffusion of the price process $\dot{h}(t,S_t)$, the algorithm is the following:

$$\forall n \leq NbNodes[N_t]$$

$$\dot{h}\left(T, \hat{S}_{N_t}^n\right) = H\left(\hat{S}_{N_t}^n\right)$$

$$\forall t < N^t$$

$$\forall n \leq NbNodes[t]$$

$$\dot{h}(t, \hat{S}_t^n) = \mathbb{E}_t^{\mathbb{Q}^*}\left[\dot{h}(t+1, \hat{S}_{t+1}^n)\right]$$

$$= \left[p^{up,*}\dot{h}\Big(t+1, \underbrace{\hat{S}_t^n(1 + Up(\sigma^*))}_{S_{t+1}^{*,\hat{f}^u(n)}}\Big) + (1 - p^{up,*})\dot{h}\Big(t+1, \underbrace{\hat{S}_t^n(1 + Dwn(\sigma^*))}_{S_{t+1}^{*,\hat{f}^d(n)}}\Big)\right]$$

with $\{*=+\}$ if $\Gamma(t,\hat{S}_t^n)>0$, and $\{*=-\}$ otherwise, and we defined the risk neutral probability:

$$p^{up,*} = \mathbb{Q}^*\left(\frac{\hat{S}_{t+1}^*}{\hat{S}_t} - 1 = Up(\sigma^*)\right).$$

Remark: By risk neutral probability, we mean the risk neutral probability associated to the corresponding volatility (σ^+ or σ^-).

```
Uncertain Tree
| Tree.build
|   nbNodes[0] = 1
|   S[0] = s
|   for t = 1, t ≤ N_t:
|   |
|   |   S^prev = S
|   |   S.resize(nbNodes[t])
|   |   for n = 1, n ≤ nbNodes[t−1]:
|   |   |   S[f^{+,u}(n)] = S^prev[n] (1 + Up(σ+))
|   |   |   S[f^{+,d}(n)] = S^prev[n] (1 + Dwn(σ+))
|   |   |   S[f^{−,u}(n)] = S^prev[n] (1 + Up(σ−))
|   |   |   S[f^{−,d}(n)] = S^prev[n] (1 + Dwn(σ−))
| Tree.price
|   for n = 1, n ≤ nbNodes[N^t]: ḣ^n_{Nt} = H(S[n])
|   for t = N_t−1, t ≥ 1:
|   |   for n = 1, n ≤ nbNodes[t]:
|   |   |   Γ = Tree.gamma(t,n)
|   |   |   if Γ > 0:
|   |   |     ḣ^n_t = [p^{up,+} ḣ^{f^{+,u}(n)}_{t+1} + p^{dwn,+} ḣ^{f^{+,d}(n)}_{t+1}]
|   |   |   else:
|   |   |     ḣ^n_t = [p^{up,−} ḣ^{f^{−,u}(n)}_{t+1} + p^{dwn,−} ḣ^{f^{−,d}(n)}_{t+1}]
```

One other way to use a tree, is to compute the Γ of the option via an intermediate density function $\hat{f}$ then simulate the underlying by Monte Carlo with the following ""density"" $\hat{f} := 1_{\{\hat{\Gamma}>0\}} f^{+} + 1_{\{\hat{\Gamma}<0\}} f^{-}$.

```
Uncertain Tree
| Tree.build
|   nbNodes[0] = 1
|   S[0] = s
|   for t = 1, t ≤ N_t:
|   | nbNodes[t] = nbNodes[t−1] + 1
|   | S^prev = S
|   | S.resize(nbNodes[t])
|   | for n = 1, n ≤ nbNodes[t−1]:
|   | | S[f^u(n)] = S^prev[n] (1 + Up(σ̂))
|   | | S[f^d(n)] = S^prev[n] (1 + Dwn(σ̂))
| MCTree.price
|   for n = 1, n ≤ nbNodes[N^t]: ĥ^n_Nt = H(S[n])
|   for t = N_t−1, t ≥ 1:
|   | for n = 1, n ≤ nbNodes[t]:
|   | | ĥ^n_t = [p^up ĥ^{f^u(n)}_{t+1} + p^dwn ĥ^{f^d(n)}_{t+1}]
|   for t = 1, t ≤ N_t:
|   | for n = 1, n ≤ N_S:
|   | | Γ = Tree.gamma(t, Ŝ^n_t)
|   | | compute Ŝ^n_{t+1} in function of Γ
```

5 Uncertain Monte Carlo approach

5.1 Summary results

We have presented two implementation of the UVM (PDE-Tree). In this section, we will first consider forward naive method based on Γ computed by Monte Carlo via a Mesh $O(N_S N_t \times \Gamma^{cost})$. Then, we will be focus on a backward propagation via a Markov Chain $O(N_S N_t + \Gamma^{cost})$ or via a Double Markov Chain. This last method is relevant, we compute two Markov Chain then we consider an adjustment of the MC. In Annexe, we compare the Monte Carlo method against the PDE. We consider the one underlying case then present results for the two underlyings case.

5.2 Introduction

The most problematic part of the "uncertain world" pricing is the fact that the dynamic of the underlying is unknown. However, as Kamtchueng explained in [14], the dynamic of the underlying is product-paths dependent and can be defined as below.

$$\frac{dS_t}{S_t} = \sigma^{*}_{t,S_t} dW^S_t$$

$$S_0 = s$$

$$\sigma^{*}_{t,S_t} := 1_{\{\Gamma_{t,S_t}>0\}} \sigma^{+}_S + 1_{\{\Gamma_{t,S_t}<0\}} \sigma^{-}_S$$

$$\Gamma_{t,S_t} := \frac{\partial^2 h^U(t, S_t)}{\partial s^2}$$

$h^U(t,S_t)$ is the price in the uncertain world a time t, it could be determinated via backward propagation and for δ_t (discretisation step) small enough could be approximated by $h(t,S_t)$. For a contingent claim, we can defined h as follows:

$$h(T, S_T) = H(S_T)$$

$$\forall t < T$$

$$h(t, S_t) = \mathbb{E}^{\mathbb{Q}^+}_t \left[h(t+1, S_{t+1})\right] 1_{\{\Gamma_{t,S_t}>0\}} + \mathbb{E}^{\mathbb{Q}^-}_t \left[h(t+1, S_{t+1})\right] 1_{\{\Gamma_{t,S_t}<0\}}$$

$$\Gamma_{t,S_t} := \frac{\partial^2 h(t, S_t)}{\partial s^2}$$

First we will introduce the Explosive Monte Carlo then we will try to reconstruct the idea of the Uncertain Tree in the Monte Carlo framework via a Markov Chain. We will consider a discretisation of the underlying space for each time step represented by the set of the $\left(\hat{y}^k_t\right)_{k\in\{1...N_{B,t}\}}$ with $N_{B,t}$ the number of buckets. The buckets are the interval discretisation of the underlying space (see Figure 6 and for instance [10] [7] and [8]).

5.3 Methodologies

In this section, we present various implementation of the Uncertain Monte Carlo. Basically we first introduce naive ways then the Markov Chain based.

- Explosive Monte Carlo: this approach is very straigh forward. As explained in section Pseudo Code, we generate the local gamma $\vec{\Gamma}$ by using a finite difference method which cost us in time computation $O\left(N_S N_t \times \left[\underbrace{3N_S N_t}_{\Gamma^{cost}}\right]\right)$. As we will see, we can reduce the Γ^{cost} by using a $\hat{Mesh}$: $O\left(N_S N_t \times \left[\underbrace{3N_M N_t}_{\Gamma^{cost}}\right]\right)$.

The Monte Carlo algorithm can be summurized as follows (the details of the pseudo code is in Section 10):

(a) compute $\vec{\Gamma}$ via Mesh using $\hat{S}$
(b) generate $\tilde{S}$ in function of $\vec{\Gamma}$

Remark: The Mesh can be seen as a Mini Monte Carlo.

- Intuitive Mesh Malliavin: this method only differs by the use of Malliavin Weight in order to compute the local gamma $\hat{\Gamma}$. By Malliavin calculus we manage to reduce the computation time $O\left(N_S N_t \times \left[\underbrace{N_M N_t}_{\Gamma^{cost}}\right]\right)$.

The Monte Carlo algorithm can be summurized as follows (the details of the pseudo code is in Section 10):

(a) compute $\hat{\Gamma}$ via Mesh using $\hat{S}$
(b) generate $\tilde{S}$ in function of $\hat{\Gamma}$

Remark: the Gamma Malliavin Weight in Black and Scholes is defined as:

$Weight^{\Gamma}_{BS} := \frac{W_T}{\sigma T S_0^2} + \frac{W_T^2 - T}{(\sigma T S_0)^2}$, We are interested by the local $Weight^{\Gamma}_{BS}$

- Integration Mesh Malliavin: We build numerically a Markov Chain via a Monte Carlo simulation of the path $\tilde{S}$ (generated via the intermediate density $\hat{f}$). By backward propagation we retrieve the uncertain price via an importance sampling method (we need to know analytically f^{+} and f^{-}), alternatively (or in addition) we can generate another Monte Carlo to perform the convergeance given the local Gamma $\check{\Gamma}$ computed for each centroide point.

This approach distinguishes the simulation and the gamma computation. Indeed our simulations will be used for the construction of the Markov Chain but also for the Malliavin weight $O\left(N_S N_t + \left[\underbrace{N_B N_t}_{\Gamma^{cost}}\right]\right)$.

The Monte Carlo algorithm can be summurized as follows (the details of the pseudo code is in Section 10):

(a) generate $\hat{S}$ via the intermediate density
(b) construct a Markov Chain via $\hat{S}$
(c) compute $\check{\Gamma}$ via backward propagation and importance sampling or via a Mesh for each centroid $\hat{y}^k$ at each time steps
(d) generate $\check{S}$ in function of $\check{\Gamma}$

- Double Markov Chain: the main idea is to be able to retrieve the following transition probabilities

$$\mathbb{Q}^*_t\left(y^{n,*}_{t+1} | y^{k,*}_t\right) := \mathbb{Q}\left(S^*_{t+1} \in B^j_{t+1} | S_t \in B^k_t, \overline{\Gamma}\left(S_t\right)\right) \qquad (5)$$
$$* \in \{+,-\}$$

this can be done numerically via

- an Explosive Monte Carlo: As we can see in Figure 5, it is a method very similar to our uncertain tree
- a Double Monte Carlo: generate two Monte Carlo simulation, one with f^+ and another with f^-

We compute backward the Gamma of each Markov Chain $\overline{\Gamma}^+$ and $\overline{\Gamma}^-$ then if the sign are commun we derive the uncertain price in favor of the Gamma sign Markov Chain otherwise we re-compute the Gamma's with a finner discretisation of the underlying space as described in the Pseudo Code Section 10.

This approach distinguishes the simulation and the gamma computation. Indeed our simulations will be used for the construction of the Markov Chain but also for the Malliavin Weight $O\left(2N_S N_t + \left[\underbrace{N_B N_t}_{\Gamma^{cost}}\right]\right)$.

The Monte Carlo algorithm can be summurized as follows (the details of the pseudo code is in Section 10) :

(a) generate $\hat{S}^+$ and $\hat{S}^-$ via respectively f^+ and f^-
(b) construct two Markov Chain via $\hat{S}^+$ and $\hat{S}^-$
(c) compute $\overline{\Gamma}^*$ via backward propagation and importance sampling or via a Mesh for both Markov Chain for each centroide $y^{k,*}$ at each time steps
(d) generate $\overline{S}$ in function of $\overline{\Gamma}$

Remark: the last step of the Markov Chain methods can be avoided by a propagation of the uncertain price adjusted by an Importance Sampling. Can be seen in Table 5 for the one underlying case.

5.4 Local gamma

In this section, we describe the proxies used for the local gamma.

Remark: we assume 3 time steps (t_0, t, T), and compute the local gamma in t.

$$S_t = s \qquad (6)$$
$$\vec{\Gamma}_t\left(S_t\right) = \frac{\hat{h}_t\left(S_t + \epsilon\right) + \hat{h}_t\left(S_t - \epsilon\right) - 2\hat{h}_t\left(S_t\right)}{\epsilon^2} \qquad (7)$$
$$\approx \frac{\partial^2 \hat{h}_t(s)}{\partial s^2} \qquad (8)$$
$$\neq \frac{\partial^2 \hat{h}^U{}_t(s)}{\partial s^2} \qquad (9)$$
$$\hat{\Gamma}_t\left(S_t\right) = \mathbb{E}^{\mathbb{Q}}_t\left[H\left(S_T\right) Weight^{\Gamma}_{t,T}\right] \qquad (10)$$
$$\approx \frac{\partial^2 \hat{h}_t(s)}{\partial s^2} \qquad (11)$$
$$\neq \frac{\partial^2 \hat{h}^U{}_t(s)}{\partial s^2} \qquad (12)$$
$$S_t \in B^k_t \qquad (13)$$
$$\check{\Gamma}_t\left(y^k_t\right) = \mathbb{E}^{\mathbb{Q}}_t\left[H\left(S_T\right) Weight^{\Gamma}_{t,T} \overbrace{\frac{f\left(S_T, \hat{y}^k_t, t\right)}{\hat{f}\left(S_T, \hat{y}^k_t, t\right)}}^{W^f(S_T, \hat{y}^k_t)}\right] \qquad (14)$$
$$\neq \frac{\partial^2 \hat{h}_t(s)}{\partial s^2} \qquad (15)$$
$$\neq \frac{\partial^2 \hat{h}^U{}_t(s)}{\partial s^2} \qquad (16)$$
$$\overline{\Gamma}_t\left(y^k_t\right) = \underbrace{\mathbb{E}^+_t\left[\Gamma^+\left(y^k_t\right)\right]}_{\overline{\Gamma}^+} 1_{\left\{\overline{\Gamma}^+ \geq 0, \overline{\Gamma}^- \geq 0\right\}} + \underbrace{\mathbb{E}^-_t\left[\Gamma^-\left(y^k_t\right)\right]}_{\overline{\Gamma}^-} 1_{\left\{\overline{\Gamma}^+ \leq 0, \overline{\Gamma}^- \leq 0\right\}} \qquad (17)$$
$$\approx \frac{\partial^2 h^U_t(s)}{\partial s^2} \qquad (18)$$
$$\Gamma^*(s) = \mathbb{E}^*_t\left[H\left(S_T\right) Weight^{\Gamma,*}_{t,T}\right] \qquad (19)$$
$$= \int_{\mathbb{R}} H\left(S_T\right) Weight^{\Gamma,*}_{t,T} f^*\left(S_T, s\right) dS_T \qquad (20)$$

Remark: our major assumption is that the Local Gamma sign stays unchanged within the bucket.

Remark: we have considered uncertain pricing problem which a finite set of predefined solutions. In the general case, we are exposed to the optimisation problem. This can been solved by using importance sampling and a classical optimizer.

6 Results

6.1 Option with one underlying

We have implemented the methods described in the previous section to the pricing of a Digital Option.

First of all, we present the classic pricing result using the Monte Carlo method and then the Markov Chain method. As shown in Appendix 8.1, the speed of convergence is relatively different with either approach.

As described in Section 5.2, after using the paths in order to build the Markov Chain, we compute the price at each time (each bucket) via backwards propagation of the price, and therefore the computation time is longer.

Figure 2 in Appendix 8.3 is based on Table 3 results. We can see that the double Markov Chain converges faster than the other two Monte Carlo (Naive and Intuitive

Mesh Malliavin) with less Monte Carlo Paths. Indeed MC Naive and Malliavin MC are using 5000 paths unlike the DMarkov MC with only 1000 paths to build and to adjust (via a Monte Carlo).

In Tables 5 and 6, we can see that 1000 paths are enough to build our Markov Chain. An additional Monte Carlo adjustment of less than 2000 paths ensures good convergence. Therefore 2000 paths are enough to price the option with closed form Gamma. In Table 4, with 2000 paths the convergence is less accurate, 0.912 against 0.905, and more time consuming 20 seconds against 8.47 seconds.

Concerning the Markov Chain method, it can be seen in Tables 6, 7, and 8 and Figure 1, that we do not need a huge number of paths builder, or a number of buckets, to produce an accurate price.

The Monte Carlo adjustment improves the accuracy of pricing as it is shown in Figure 2, Uncertain Pricing computation.

6.2 Option with two underlyings

As we mentioned in introduction, the PDE implementation is limited by the dimension of our pricing problem. If the Naive Method is working, the time cost is huge. We are in favor of the Markov Chain approach. We propose two algorithms:

- Mid Markov: it is an extension of Integration Mesh-Malliavin. We compute numerically our Markov Chain via a multi-dimensional Monte Carlo with paths generated via the joint intermediate density. We then re-run another Monte Carlo, using the Gamma Grid from the Intermediate Markov Chain.
- Multi Markov Chain: it is an extension of the Double Markov Chain. We compute the entire combination of the pair of Markov Chain ($\sigma_1^{*}, \sigma_2^{**}$) with ($*\in\{+,-\}$, $**\in\{+,-\}$). Then from the Gamma profile of the set of Markov Chains we made our decision when we re-run our Monte Carlo.

6.2.1 OutPerformance or Choice Option

We have for this payoff a close form for the uncertain price. Indeed given a positive correlation we have the BSB solution provide by the upper volatilities.

$$OutPerf\left(t,S_t^1,S_t^2\right)=\mathbb{E}_t\left[\left(S_T^1-S_T^2\right)^+\right] \tag{21}$$

the Option price of the 6M OutPerf is 1.2005. Our results are presented in the Table 10.

6.2.2 Spread of Call

As a linear product with a cross gamma null, we can determine the solution of the BSB equation associated with the Spread Of Call defined as follows.

$$SpreadCall\left(t,S_t^1,S_t^2\right)=\mathbb{E}_t\left[\left(\left(S_T^1-K_1\right)^+-\left(S_T^2-K_2\right)^+\right)\right] \tag{22}$$

by taking the upper bound of the first asset and the lower bound for the second asset, we have the closed solution of our uncertain pricing problem. the Option price of the 1Y SpreadCall, with strikes $K_1 = 10.5$ and $K_2 = 11$ is 1.1367.Our results are presented in the Table 11.

6.2.3 Digital Basket

$$DigitalBasket\left(t,S_t^1,S_t^2\right)=\mathbb{E}_t\left[1_{\{w_1S_T^1+w_2S_T^2\geq K\}}\right] \tag{23}$$

Our Results on the 1Y Digital Basket, with strikes $K_1 = 10.5$, are presented in the Table 12. In order to provide the closed Gamma results we have used a Moment Matching proxy.

6.2.4 Digital Geometric Average

$$DigitalGeometricAverage\left(t,S_t^1,S_t^2\right)=\mathbb{E}_t\left[1_{\left\{\sqrt{S_T^1S_T^2}\geq K\right\}}\right] \tag{24}$$

Our Results on the 1Y Digital Geometric Average, with strikes $K_1 = 10.5$, are presented in the Table 12.

6.2.5 Comments

The generalisation confirms the result of the previous section. We have implemented the different approaches for the 4 exotics presented above.

In Table 10, the results are expected and the uncertain price is equal to the upper bound volatility price 1.2005.

In Table 11, because of the linearity of the payoff the uncertain price is known and equal at 1.1367. The Uncertain Monte Carlo methods are very closed.

We note a better convergeance for the Multi Markov Chain method.

In Table 12, we try to compute the uncertain price of a Digital Basket. The uncertain price is lower than the actual BS closed formula price which is a proxy using the Moment Matching volatility of the Basket. We can also see how the Monte Carlo price σ' is closed to our uncertain price Monte Carlo, this can be explained by the fact that our uncertain price is lower than our Monte Carlo error.

In Table 13, the results are similar at the one from table 12.

After a quick study of the Gamma profile of the option 12 and 13, we opte for a finner discretisation of the underlying space.

Remark: The results in Table 12 present some interests in sense that we can separate the different source of noise. Indeed The closed Gamma is biaised by the fact that the gamma profile of the Moment Matching proxy and the Digital Basket are not the same. The Naive Monte Carlo is time consuming without a good convergeance rate.

In Tables 14-16, we study the Gamma profile convergeance by increasing the paths of the Mesh used to compute the Markov Chain and the Malliavin Weight with another discretisation. The results are satisfying in sense that in the Table 16 we are closer to the closed solution.

7 Conclusion

As we have described in this paper, there are various ways to solve our numerical pricing problem that are independent of the product. The "'uncertain tree'" approach is very similar to the BSB PDE therefore does not resolve the multi-dimensional issue. As demonstrated in the Results section, use of the Markov Chain provides a faster and more accurate way to determine the uncertain price.

In view of our main concerns, computation time and accuracy, one can consider a Monte Carlo adjustment to perform the convergence. Another advantage of the Markov Chain approach is that it takes into account the multi-factor multi-underlying pricing problem (see [7] for the higher dimension quantisation theory).

We shown some results for the multi-underlyings uncertain pricing problem. If the performance is reasonable for the two underlyings case for both method Mid Markov and Multi Markov, the rate of convergeance of the last method will explode with the number of underlyings. Indeed the Multi Markov algorithm has a number of Markov Chain needed which increases exponentially with the number of assets.

The precision is relative to the product Gamma profile therefore a study product dependant should be done before making any choice regarding the method.

This method can be applied to more recent issues such as the exposure, collateral modelling, and liquidity risk (it will be discussed in a subsequent publication).

Concerning the uncertain Monte Carlo, our work can be extended:

- Firstly, to other uncertain parameters, UCM [14] and other extensions. In practice, even for the uncertainty volatility model we need more than the sign of the Γ, in the general case the exact value of the Δ and Γ are needed.
- Secondly, in addition to and as a consequence of the above point, our sensitivities have to be accurate enough to make a good sensitivity decision.
- Thirdly, our framework can be added to recent works by Touzi et al (2004, [9]; 2007 [12]; 2009 [13]), and Guyon et al (2010 [15]).

Appendix

8 Numerical Application One Underlying

- $S_0 = 10$
- $K = 9$
- $T = 1$
- $\sigma^+ = 30\%$
- $\sigma^- = 10\%$
- $\sigma_{mid} = 20\%$
- $Digital^{BS}(S_0, T, K, \sigma^+) = 0.5797$
- $Digital^{BS}(S_0, T, K, \sigma^-) = 0.8422$
- $Digital^{BS}(S_0, T, K, \sigma_{mid}) = 0.6652$
- $Digital^{UVM}(S_0, T, K, \sigma^+, \sigma^-) = 0.9020$ computed by PDE, $\delta^x = 0.1$ and $\delta^t = 0.01$ computed in 1.1 seconds.

8.1 Pricing Digital Maturity 1Y, Strike 90%

Table 1: Digital Pricing by Monte Carlo, strike 90%

	Monte Carlo Pricing of Digital					
NbSteps	1	1	50	50	100	100
NbPaths	5000	10000	5000	10000	5000	10000
Price	0.5836	0.5695	0.5763	0.584	0.5772	0.579
Variance	0.0882	0.0882	0.0882	0.0883	0.0883	0.0882
Confidence	*0.5918*	*0.5753*	*0.5845*	*0.5898*	*0.5854*	*0.5848*
Interval	*0.5753*	*0.5636*	*0.568*	*0.5781*	*0.5689*	*0.5731*
Time	0.06	0.07	0.41	0.69	0.69	1.3

Table 2: Digital Pricing via Markov Chain, strike 90%

	Markov Chain					
NbSteps	1	1	50	50	150	150
NbPaths	5000	10000	5000	10000	5000	10000
Price	0.5856	0.5845	0.5724	0.5855	0.581	0.5725
NbBuckets	50	50	50	50	100	100
Time	0.2	0.28	2.16	2.97	6.59	8.44

8.2 Uncertain Pricing Digital Maturity 1Y, Strike 90%

In this section, we present the result concerning the following methods:

- MC Gamma is a Monte Carlo using a closed formulae for the gamma.
- Naive MC is described in Section 5.3
- Mesh Malliavin (intuitive) is described in Section 5.3
- DMarkov (Double Markov Chain) is described in Section 5.3
- DMarkov Gamma is DMarkov algorithm using a closed formulae for the gamma instead of Malliavin
- DMarkov MC is DMarkov with an additional Monte Carlo

Table 3: Uncertain Monte Carlo, 100 steps, 5000 paths and 1000 paths for the Mesh (1000 paths to build the Markov Chain)

Method	Naive MC	Mesh Malliavin	DMarkov MC	DMarkov Γ MC	DMarkov Γ	MC	PDE
Price	0.6178	0.8388	0.912	0.9472	0.8709	0.883	0.902
Variance	0.2361	0.1353	0.081	0.046601		0.104	
Time	1190.58	529.78	7.55	20.64	5.08	11.8	1.1
NbBuckets			100	100	100		

Table 4: Uncertain Digital Price, convergeance MC Gamma

	Monte Carlo Gamma						
NbSteps	100	100	100	100	100	100	100
NbPaths	500	1000	2000	3000	4000	5000	6000
Price	0.914	0.883	0.912	0.9033	0.9097	0.9028	0.90283
Variance	0.0802	0.1949	0.08	0.0875	0.0823	0.0879	0.0878
Confidence	*0.9388*	*0.903*	*0.924*	*0.913*	*0.918*	*0.9110*	*0.9103*
Interval	*0.8891*	*0.863*	*0.899*	*0.892*	*0.9*	*0.8945*	*0.8953*
Time	7.55	11.8	24.05	36.78	49.73	58.73	69.47

Table 5: Uncertain Digital Price, Double Markov Chain Gamma

	Double Markov Chain Gamma						
NbPathBuilder	500	1000	2000	3000	4000	5000	6000
100 Buckets	0.88337	0.87	0.8741	0.8735	0.8644	0.8655	0.8626
time	0.466	5.08	5.13	5.75	6	6.58	7.38
150 Buckets	0.8725	0.8762	0.8946	0.8798	0.8852	0.8776	0.8807
time	7.181	7.87	8.14	8.47	9.3	9.88	10.2
200 Buckets	0.8881	0.879	0.8873	0.8893	0.8864	0.888	0.8891
Time	10.34	10.59	10.78	11.31	11.94	12.05	13.22

Table 6: Uncertain Digital Price, Double Markov MC Gamma

	Double Markov Chain MC Gamma (PathBuilder 1000, nbBucket 150)						
NbPath MC	500	1000	2000	3000	4000	5000	6000
Price	0.892	0.912	0.905	0.8883	0.8987	0.896	0.8998
Variance	0.0979	0.081	0.0863	0.0994	0.0912	0.09334	0.0902
Confidence	*0.9194*	*0.91788*	*0.91788*	*0.89961*	*0.9081*	*0.90446*	*0.9074*
Interval	*0.8645*	*0.89211*	*0.8921*	*0.877*	*0.8893*	*0.88753*	*0.8922*
Time	7.52	8.47	8.47	8.97	9.47	9.55	9.89

Table 7: Uncertain Digital Price, Markov MC Gamma

	Markov Mid Vol Gamma MC (Path Builder 1000, nbBucket 150)						
NbPath MC	500	1000	2000	3000	4000	5000	6000
Price	0.898	0.889	0.89	0.897	0.905	0.899	0.8985
Variance	0.0932	0.09946	0.0906	0.0926	0.0861	0.09032	0.0913
Confidence	*0.9247*	*0.9085*	*0.9125*	*0.90789*	*0.914*	*0.9081*	*0.906*
Interval	*0.8712*	*0.8694*	*0.8862*	*0.8861*	*0.8959*	*0.8914*	*0.8908*
Time	4.83	5.11	5.1	5.55	6.3	6.11	6.63

Table 8: Uncertain Digital Price, Double Markov MC							
	Double Markov Chain						
NbPath MC	1000	1000	4000	4000	6000	8000	10000
Price	0.886	0.952	0.936	0.938	0.936	0.9493	0.932
Variance	0.102	0.0466	0.0607	0.0581	0.059	0.0481	0.0634
Confidence	*0.914*	*0.9653*	*0.9512*	*0.9457*	*0.944*	*0.9549*	*0.9383*
Interval	*0.8579*	*0.93862*	*0.9207*	*0.9307*	*0.9289*	*0.9438*	*0.9256*
Time	18.16	20.64	59.31	60.61	60.58	87.89	145.27
NbPathBuilder	500	1000	1000	4000	4000	6000	6000

Table 9: Uncertain Digital Price, Markov Mid MC							
	Markov σ_{mid} MC						
NbPath MC	500	1000	1000	6000	4000	4000	8000
Price	0.74	0.774	0.812	0.786	0.7758	0.798	0.7794
Variance	0.193493	0.1755	0.1533	0.1682	0.174	0.1613	0.1779
Confidence	*0.7785*	*0.7999*	*0.8362*	*0.7936*	*0.7887*	*0.8104*	*0.7898*
Interval	*0.70144*	*0.748*	*0.7877*	*0.7756*	*0.7629*	*0.7855*	*0.7689*
Time	10.61	10.42	33.03	52.45	36.59	34.63	86.59
NbPathBuilder	1000	1000	4000	8000	6000	4000	10000

8.3 Uncertain Pricing Computation time and accuracy

Figure 1: UMC Time Computation.

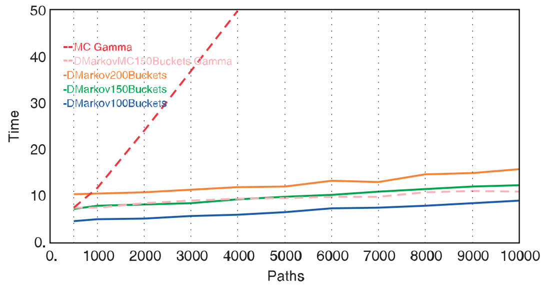
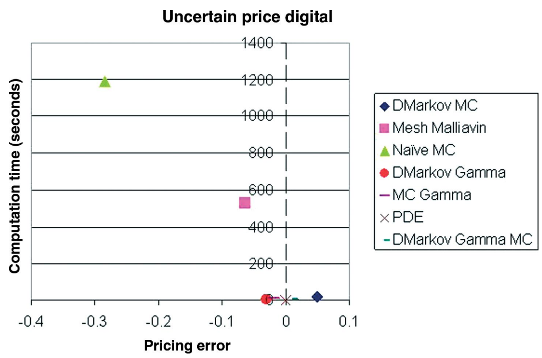


Figure 2: UMC Performance.



9 Numerical Application Two Underlyings

- $S_0^1 = S_0^2 = 10$
- $K_1 = 10.5$
- $K_2 = 11$
- $\sigma_1^+ = 40\%$
- $\sigma_1^- = 20\%$
- $\sigma_1^{mid} = 30\%$
- $\sigma_2^+ = 15\%$
- $\sigma_2^- = 5\%$
- $\sigma_2^{mid} = 10\%$
- $\rho = 0\%$

In this section, we present the result concerning the following methods:

- Naive MC is a Monte Carlo based on the intermediate law Gamma provided by a closed formulae
- Mid Markov (Integration Mesh Malliavin) is based on one Markov Chain built via the intermediate law see section 5.3.
- Mid Markov Γ is Mid Markov algorithm using a closed formulae for the gamma instead of Malliavin
- Multi Markov (Double Markov Chain) is based on a combination of Markov Chain which reflect the possible (σ_1^*, σ_2^*) solution of our BSB pricing problem.
- Multi Markov Γ is Multi Markov algorithm using a closed formulae for the gamma instead of Malliavin
- σ^+ MC is a Monte Carlo price with the f^+ density for the two underlyings using 20000 paths.
- σ^- MC is a Monte Carlo price with the f^- density for the two underlyings using 20000 paths.
- σ^+ BS is the closed form price with the f^+ density for the two underlyings.
- σ^- BS is the closed form price with the f^- density for the two underlyings.

Table 10: Uncertain Monte Carlo price for a OutPerformance Maturity 6 Months

Method	Naive MC	Mid Markov Γ	Multi Markov Γ	Mid Markov	Multi Markov	σ^+ MC	σ^- MC
Price	1.2057	1.1773	1.2022	1.1845	1.1946	1.1982	0.575
Variance	0.2361	3.9021	4.155	4.0937	4.0432	4.1421	0.8597
Time	28380	23.25	220.27	52.86	199.47	**BS**	**BS**
NbSteps	20	20	20	20	20	1.2005	0.581
NbPaths	20000	20000	20000	20000	20000		
NbMesh		5000	5000	5000	5000		
NbBuckets		20	20	20	20		

Table 11: Uncertain Monte Carlo price for a Spread of Call Maturity 1Year, strikes 10.5 and 11

Method	Naive MC	Mid Markov Γ	Multi Markov Γ	Mid Markov	Multi Markov	σ^+ MC	σ^- MC
Price	1.3859	1.2174	1.2294	1.1127	1.1329	1.15	0.5815
Variance	8.0022	8.1761	8.5354	8.1108	8.2057	8.147568	1.2702
Time	1180.5	42	230.75	56.59	205.98	**BS**	**BS**
NbSteps	20	20	20	20	20	1.1367	0.5848
NbPaths	20000	20000	20000	20000	20000		
NbMesh		5000	5000	5000	5000		
NbBuckets		30	30	30	30		

Table 12: Uncertain Monte Carlo price for a Digital Basket Maturity 1Y, strike 10.5

Method	Naive MC	Mid Markov Γ	Multi Markov Γ	Mid Markov	Multi Markov	σ⁺ MC	σ⁻ MC
Price	0.2015	0.3299	0.3346	0.3361	0.332	0.3302	0.2827
Variance	0.1606	0.221	0.2226	0.2231	0.2217	0.2286	0.2029
Time	1126.06	26.08	228.02	48.42	198.28	**BS**	**BS**
NbSteps	20	20	20	20	20	0.3698	0.3008
NbPaths	20000	20000	20000	20000	20000		
NbMesh		5000	5000	5000	5000		
NbBuckets		20	20	20	20		

Table 13: Uncertain Monte Carlo price for a Digital Geometric Average Maturity 1Y strike 10.5

Method	Naive MC	Mid Markov Γ	Multi Markov Γ	Mid Markov	Multi Markov	σ⁺ MC	σ⁻ MC
Price	0.2062	0.3435	0.3469	0.3522	0.3431	0.3452	0.2929
Variance	0.1636	0.2255	0.2265	0.2281	0.2254	0.2214	0.2032
Time	1126.81	25.69	225.41	45.23	211.3	**BS**	**BS**
NbSteps	20	20	20	20	20	0.3292	0.2821
NbPaths	20000	20000	20000	20000	20000		
NbMesh		5000	5000	5000	5000		
NbBuckets		20	20	20	20		

Table 14: Uncertain Monte Carlo with Malliavin Gamma price for a Basket Digital 1Year, strike 10.5

Method	Naive MC	Mid Markov			Multi Markov		
Price	0.3454	0.3644	0.3659	0.3628	0.369	0.357	0.3694
Variance	0.2261	0.2316	0.232	0.2311	0.2328	0.2295	0.2329
Time	1012.2	32.77	10.64	30.33	221.84	249.75	263.36
NbSteps	5	5	5	5	5	5	5
NbPaths	5000	10000	20000	20000	10000	20000	20000
NbMesh	5000	10000	10000	20000	10000	10000	20000
NbBuckets		30	30	30	30	30	30

Table 15: Uncertain Monte Carlo with Malliavin Gamma price for a Geometric Average Digital 1Year, strike 10.5

Method	Naive MC	Mid Markov			Multi Markov		
Price	0.3294	0.3224	0.324	0.3268	0.3366	0.3565	0.3549
Variance	0.2209	0.2184	0.219	0.22	0.2233	0.2294	0.2289
Time	918.27	27.53	30.2	10.23	231.63	241.34	234.03
NbSteps	5	5	5	5	5	5	5
NbPaths	5000	10000	20000	20000	10000	20000	20000
NbMesh	5000	10000	10000	20000	10000	10000	20000
NbBuckets		30	30	30	30	30	30

Table 16: Uncertain Monte Carlo with Malliavin Gamma price for a Spread of Call Option 1Year, strikes 10.5, 11

Method	Naive MC	Mid Markov			Multi Markov		
Price	0.3294	1.1459	1.0387	1.1161	1.2082	1.1534	1.1403
Variance	0.2209	8.1587	6.8075	6.9995	8.5192	8.3773	8.4039
Time	918.27	23.36	30.31	20.6	238.3	262.64	263.88
NbSteps	5	5	5	5	5	5	5
NbPaths	5000	10000	20000	20000	10000	20000	20000
NbMesh	5000	10000	10000	20000	10000	10000	20000
NbBuckets		30	30	30	30	30	30

Uncertain Monte Carlo : Explosive Monte Carlo

for t = 1, t ≤ N_t:

  for n = 1, n ≤ N_S:

    compute $\hat{Mesh}(S_t^n)$

    compute price $\hat{h}(S_t^n, t)$ via $\hat{Mesh}(S_t^n)$

    compute $\hat{Mesh}(S_t^n + \in)$

    compute price $\hat{h}(S_t^n + \in, t)$ via $\hat{Mesh}(S_t^n + \in)$

    compute $\hat{Mesh}(S_t^n - \in)$

    compute price $\hat{h}(S_t^n - \in, t)$ via $\hat{Mesh}(S_t^n - \in)$

    compute $\hat{\Gamma}_t^n := \dfrac{\hat{h}\left(S_t^n + \epsilon, t\right) + \hat{h}\left(S_t^n - \epsilon, t\right) + 2\hat{h}\left(S_t^n, t\right)}{\epsilon^2}$ *by definite difference*

    compute $\hat{S}_{t+1}^n$ in function of $\hat{\Gamma}_t^n$

Uncertain Monte Carlo : Intuitive Mesh-Malliavin

for t = 1, t ≤ N_t:

  for n = 1, t ≤ N_S:

    compute $\hat{Mesh}_t^n$

    compute $\hat{\Gamma}_t^n$ via Malliavin

    compute $\hat{S}_{t+1}^n$ in function of $\hat{\Gamma}_t^n$

10 Pseudo-Code

Uncertain Monte Carlo : Integration Mesh-Malliavin

simulation $\hat{S}_{t+1}^n$

Markov Chain build

for t = N_t, t ≥ 1:

  for k = 1, t ≤ $N_{B,t}$:

    compute $\breve{\Gamma}(t, \hat{y}_t^k)$

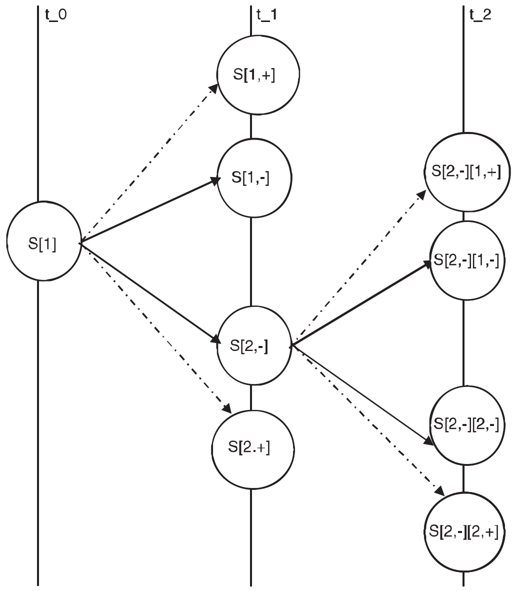
    compute $\breve{h}(t, \hat{y}_t^k)$ via importance sampling

Uncertain Monte Carlo : Double Markov Chain
  simulation $\hat{S}_t^{n,+}$ and $\hat{S}_t^{n,-}$
  Markov Chain build
  for t = N_t, $t \geq 1$:
    for k = 1, t ≤ $N_{B,t}$:
      compute $\bar{\Gamma}^{+}(t, \hat{y}_t^k)$
      compute $\bar{\Gamma}^{-}(t, \hat{y}_t^k)$
      if $\bar{\Gamma}^{+}(t, \hat{y}_t^k) > 0$ and $\bar{\Gamma}^{-}(t, \hat{y}_t^k) > 0$
        compute $\bar{h}^{+}(t, \hat{y}_t^k)$
      else if $\bar{\Gamma}^{+}(t, \hat{y}_t^k) < 0$ and $\bar{\Gamma}^{-}(t, \hat{y}_t^k) < 0$
        compute $\bar{h}^{-}(t, \hat{y}_t^k)$
      else
        affine bucket discretisation

11 UMC by Graphs

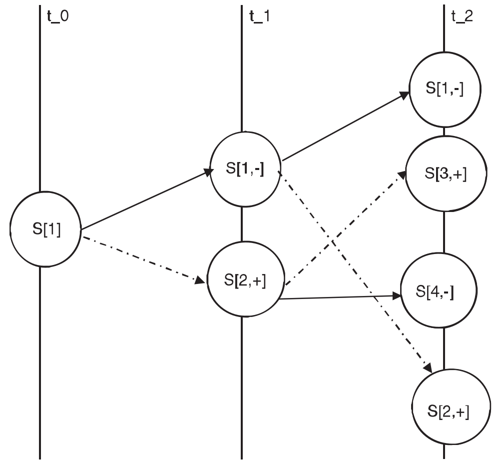
11.1 Uncertain Tree

Figure 3: UMC Tree: At each time steps, each node has two pair of sons computed via two densities f^+ and f^-



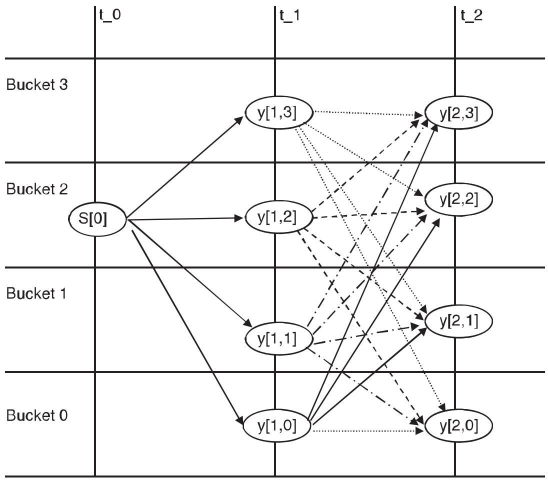
11.2 Explosive Monte Carlo

Figure 4: UMC Explo MC: similar to the tree approach, for each time step, we generate for one path two paths via two densities f^+ and f^-



11.3 Markov Chain

Figure 5: UMC Markov Chain : the transition probability matrix is computed forward numerically then the price and gamma backward



Christian Kamtchueng is the founder of CTK corp, a consultancy company that provides quantitative finance services. He is also a professor at one of the top French business schools and the associated founder of the MathSup French E-learning Company specializing in Mathematics.

REFERENCES

[1] F. Black and M. Scholes (1973) Pricing of options and corporate liabilities. *Journal of Political Economy* **81**(3): 637–654.
[2] R.C. Merton (1976) Theory of rational option pricing. *Bell Journal of Economics and Management Science* **4**(1): 141–183.
[3] R.C. Merton (1976) Options pricing when underlying stock returns are discontinuous. Massachussets Institute of Technology, Cambridge, MA.
[4] M. Avellaneda, A. Levy, A. Paras (1995) Pricing and hedging derivative securities in markets with uncertain volatility. *Applied Mathematical Finance* **2**(2): 73–88.
[5] T. Vargiolu (2001) Existence, uniqueness and smoothness for the Black–Scholes–Barenblatt equation. Technical report, Department of Pure and Applied Mathematics, University of Padova.
[6] E. Fournie, J.-M. Lasry, J. Lebuchoux, P.-L. Lions (2001) Applications of Malliavin calculus to Monte Carlo methods in finance II. *Finance and Stochastics* **5**: 201–236.
[7] V. Bally, G. Pages (2003) A quantization algorithm for solving discrete time multidimensional optimal stopping problems. *Bernoulli* **9**: 1003–1049.
[8] G. Pages, J. Printemps, H. Pham (2004) Optimal quantization methods and applications to numerical problems in finance. In *Handbook on Numerical Methods in Finance*. Birkhäuser: Boston, MA; pp. 253–297.
[9] B. Bouchard, N. Touzi (2004) Discrete-time approximation and Monte Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications* **111**: 175–206.
[10] V. Bally, G. Pages, J. Printemps (2005) A quantization method for pricing and hedging multidimensional American style options. *Mathematical Finance* **15**(1): 119–168.
[11] P. Glasserman, N. Chen (2006) Malliavin greeks without Malliavin calculus. *Stochastic Processes and their Applications* **117**: 1689–1723.
[12] P. Cheridito, M. Soner, N. Touzi, N. Victoir (2007) Second order backward stochastic differential equations and fully nonlinear parabolic PDEs. *Communications on Pure and Applied Mathematics* **60**(7): 1081–1110.
[13] A. Fahim, N. Touzi, X. Warin (2009) A probabilistic numerical method for fully nonlinear parabolic PDEs. Working Paper 19.
[14] C. Kamtchueng (2010) Uncertain correlation model. *Wilmott Journal* **2**(6): 317–349.
[15] J. Guyon, P. Henry-Labordere (2011) Uncertain volatility model: A Monte Carlo approach. *Journal of Computational Finance* **14**(3): 37–71.