



Ecole polytechnique  
Promotion X2012  
EL KHADIR Bachir

# **RAPPORT DE STAGE DE RECHERCHE**

## **Modèle de Hull–White à deux facteurs**

---

---

### **CONFIDENTIEL**

Département de Mathématiques Appliquées  
MAP595 : Mathématiques financières  
Directeur de stage : M. Peter TANKOV  
Maître de stage : M. John Paul BARJAKTAREVIC  
Dates du stage : du 16 Mars au 31 Août  
Nom et adresse de l'organisme : J.P. Morgan 25 Bank Street, Canary Wharf, London E14 5JP England

# TABLE DES MATIÈRES

<b>1</b>	<b>Préliminaires sur les taux d'intérêts</b>	<b>7</b>
1.1	Définition . . . . .	7
1.2	Mesures équivalentes . . . . .	7
1.3	Produits financier d'intérêts . . . . .	10
<b>2</b>	<b>Les différents modèles des taux d'intérêts</b>	<b>11</b>
2.1	Le modèle à deux facteurs . . . . .	11
2.1.1	Motivation . . . . .	11
2.1.2	Limite des modèle à un seul facteur . . . . .	12
2.2	Approximation de la solution par un arbre binomial . . . . .	13
2.3	Améliorations . . . . .	14
2.3.1	Formule exacte . . . . .	16
2.3.2	Ajout d'un shift déterministe . . . . .	17
2.3.3	Optimiser la taille des slices . . . . .	18
2.3.4	Paramètres dépendant du temps . . . . .	18
<b>3</b>	<b>Résultats</b>	<b>19</b>
<b>4</b>	<b>Application : calibration et pricing</b>	<b>21</b>
4.1	Calibration du drift . . . . .	21
4.2	Méthode de calibration . . . . .	22
<b>5</b>	<b>Conclusion</b>	<b>25</b>
<b>A</b>	<b>Code de simulation en Python</b>	<b>26</b>
<b>B</b>	<b>Formule exacte des obligations zéro coupon pour des paramètres dépendant du temps</b>	<b>30</b>

# TABLE DES FIGURES

2.1	Libor . . . . .	12
2.2	Tableau de corrélation . . . . .	12
2.3	Arbre construit de façon naïve . . . . .	15
2.4	Arbre optimisé . . . . .	15
2.5	$t = 1$ . . . . .	16
2.6	$t = 10$ . . . . .	16
2.7	$t = 20$ . . . . .	16
2.8	$t = 30$ . . . . .	16
2.9	En utilisant l'arbre pour le calcul des $P(t, T)$ . . . . .	17
2.10	En utilisant la formule exacte . . . . .	17
3.1	Slice 3D . . . . .	19
3.2	Erreur bf vs cf Slice 3D . . . . .	20
4.1	Calibration des caps . . . . .	23
4.2	Surface théorique en bleu - . . . . .	24

# REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à mon professeur, Mr Stefano De Marco de l'Ecole Polytechnique qui m'a beaucoup aidé dans ma recherche de stage. Son écoute et ses conseils m'ont permis de cibler mes candidatures et de trouver ce stage me correspondait totalement.

Je tiens à remercier vivement mon maitre de stage, Mr John Paul Barjaktarevic, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions avec son aide précieuse dans les moments les plus délicats.

Je remercie également toute l'équipe "rates exotic" pour leur accueil, leur esprit d'équipe.

Enfin, je tiens à remercier toutes les personnes qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage.

## RÉSUMÉ

Ce rapport présente quelques outils utilisés lors du pricing de dérivés de taux. Il est particulièrement centré sur le modèle de Hull White qui est conduit par deux facteurs (appelé aussi G2++). J'adopterai le plan suivant dans la rédaction de ce rapport :

- La première partie introduit le marché de taux par un certain nombre de définitions qui lui sont propres, ainsi que les produits financiers principaux utilisés pour la calibration des modèles.
- Dans la deuxième partie je présenterai quelques modèles existant, et je discuterai de leurs limites et de leurs implémentation pratique.
- La troisième partie présente les résultats concrets que j'ai obtenus dans le cas du modèle gaussien à deux facteurs, ainsi que la partie calibratoir de modèle.

## ABSTRACT

This report presents some tools used in the pricing for interest rates derivatives. It's focused on two factors Hull-White model (called G2++).

- The first part introduces some definitions relative to the interest rates market, as well as some derivative used later for the calibration process.
- In the second part I will present some existing models, their limitations and possible implementation.
- The last part presents the results I obtained using the G2++ model and the calibration process.

# INTRODUCTION

J'ai fait mon stage a JP Morgan equipe exotic rates. La plus grosse partie du travail consistait à trouver le bon modèle mathématique pour pricer un produit et le calibrer au marche.

L'intérêt que porte les banques aux produits exotiques a considérablement augmenté aucours des dernières années. Plusieurs modèles ont été développé pour refeléter au mieux le comportement des marchés financiers.

Le département au sein duquel j'ai effectué mon stage s'occupait des produits basé sur les taux d'intêret. Dans ce domaine, une propriété appréciable dans un modèle est le fait qu'il possèd un nombre de paramètres suffisant pour être calibré parfaitement aux prix observable dans la réalité, sans pour autant "overfitter" l'échantillon disponible. En effet, dernièrement, surtout depuis la crise, le marché des taux a connu des changement radicaux, la possiblité d'observer des taux négatifs en est un exemple.

Un produit en particulier n'a cessé de gagner en notoriété : les cancelables spread options. Ma mission de stage était de comprendre les modèles existant et leurs implémentations qui permettent de pricer (entre autres) ce type de produits, comprendre leurs limites, et essayer de trouver des améliorations.

Ce stage est à forte composante informatique, une attention particulière a été accordée aux détails d'implémentation et optimisation du code.

# CHAPITRE 1

## PRÉLIMINAIRES SUR LES TAUX D'INTERÊTS

### 1.1 DÉFINITION

**Definition 1.** On dénote le prix de zéro coupon forwardé  $P(T, S)$  le montant qu'il faut investir à dans un instrument risque-neutre au temps  $T$  pour obtenir une unité de monnaie au temps  $S$ .

**Definition 2.** On définit  $f(t, T)$  le taux d'intérêt instantané forward à la date  $t$  pour une maturité  $T$  la quantité

$$f(t, T) := -\frac{\delta}{\delta T} \log P(T, S)$$

**Definition 3.** Le taux instantané est défini par  $r_t = \lim_{T \rightarrow t} f(t, T)$

Le taux d'actualisation (stochastique) est :  $D_t^T := e^{-\int_t^T r_s ds}$

**Definition 4.** Le taux d'interet forward  $F(t; T, S)$  est la prévision à l'instant  $t$  du taux entre deux période  $T$  et  $S$ . Ce taux est la quantité  $L$  connu à l'instant qui annule la valeur de ce contrat à l'instant  $t$  :

- Recevoir l'intérêt  $L$  sur un 1 euro entre  $T$  et  $S$
- Payer le taux variable  $F(T, S)$  sur un 1 euro entre  $T$  et  $S$

le taux  $r_t$  n'est pas un produit échangé sur le marché que l'on peut mettre en portefeuille. On ne peut donc pas construire de couverture d'un produit donné de la même manière que dans un modèle d'action, et ce malgré la similitude des modèles mathématiques.

### 1.2 MESURES ÉQUIVALENTES

Nous nous plaçons dans le cadre d'une économie à temps continu, qui admet une espace de probabilité  $(\Omega, \mathcal{F}, \mathbb{P})$ , avec  $K + 1$  actifs tradables, que nous appellerons actifs de base, dont le prix est donné par  $(S_t = (S_t^0, \dots, S_t^K))_{t \geq 0}$ . Dans toute la suite nous confondons l'actif et son prix.  $S^0$  étant l'actif sans risque, qui évolue donc au temps sans risque

$$dS_t^0 = r_t S_t^0 dt$$

ie

$$S_t^0 = e^{-\int_0^t r_s ds}$$

Par définition, nous connaissons le prix des actifs  $K + 1$ , dans la prochaine section nous détaillerons la procédure de pricing de produits plus compliqués.

A travers les  $K + 1$  actifs de base, nous construisons des produits plus complexes. Le prix de ce dernier est donc intimement lié à la possibilité de trouver une stratégie auto-financée qui le réplique. Commençons d'abord par définir ce qu'est une stratégie auto financée.

**Definition 5.** — Une stratégie est un processus  $(\Phi_t = (\Phi_t^0, \dots, \Phi_t^K))_t$  localement borné et adapté à la filtration  $\mathcal{F}$ .

- La valeur associée à cette stratégie est donnée par  $V_t(\Phi) = \langle \Phi_t, S_t \rangle$ .
- Une stratégie est auto financée si  $dV_t = \Phi_t dS_t$

Une hypothèse souvent utilisée dans le cadre de la finance de marché est l'absence d'arbitrage. Une opportunité d'arbitrage est la possibilité d'investir 0 aujourd'hui, et recevoir, avec probabilité non nulle, un montant positif dans le futur. En d'autres termes, l'absence d'arbitrage signifie que si  $\Phi$  est une stratégie auto financée telle que  $V_0(\Phi) = 0$ , alors  $\mathbb{P}(V_t(\Phi) > 0) = 0$ . Ceci nous permettra de valoriser des produits complexes en répliquant leur payoff par une combinaison linéaire de produits simple dont le prix est connu.

Une deuxième hypothèse que nous admettrons dans la suite est la complétude du marché : Tout les produits utilisés seront considérés disponibles à tout moment et en quantité abondante (liquide), ie à chaque instant  $t$ , pour tout payoff  $H$ , il existe une stratégie autofinancée associée  $\Phi$  qui vérifie  $V_t(\Phi) = H$ . Nous ne traiterons pas le cas des produits illiquides. Ceci est justifié, le marché des taux étant l'un des plus gros en volume dans le monde.

Nous pouvons montrer(\*) que ces hypothèses sont équivalentes à l'existence d'une mesure de probabilité risque neutre  $Q$  unique sous laquelle le prix actualisé de tous les produits tradables sont des martingales. ie si on note  $H_t$  le prix à l'instant  $t$  d'un produit financier, alors

$$H_t = E^Q \left[ \frac{H_s}{S_t^0} | \mathcal{F}_t \right] = V_t(\Phi)$$

En particulier, le prix d'un zéro coupon qui paye 1 à l'instant  $T$  est donné par

$$P(t, T) := E^Q \left[ e^{-\int_t^T r} \right]$$

Nous pouvons interpréter le ratio  $\frac{H_s}{S_s^0}$  comme étant le nombre de  $H$  par unité de discount factor stochastique  $S^0$ . Le discount factor est appelé dans ce cas numéraire. Nous verrons maintenant que nous pouvons choisir un autre numéraire plus adapté au produit que nous voulons pricer. En effet le changement de numéraire préserve la propriété d'autofinancement d'un portefeuille.

**Definition 6.** Un numéraire est tout actif financier ne payant pas de dividendes

**Definition 7.** Mesure de probabilité équivalente.

Supposons qu'il existe un numéraire  $(M_t)_{t \geq 0}$  et une mesure martingale équivalente  $Q^M$  telle que le prix de chaque actif actualisé par le processus  $M$  soit une  $Q^M$ -martingale.

$$\frac{S_t^i}{M_t} = E^{Q^M} \left[ \frac{S_T^i}{M_T} | \mathcal{F}_t \right]$$

Soit  $(N_t)_{t \geq 0}$  un numéraire.

Alors il existe une mesure de probabilité  $Q^N$  telle que le prix de chaque actif actualisé par le processus  $N$  soit une  $Q^N$ -martingale, i.e.

$$\frac{S_t^i}{N_t} = E^{Q^N} \left[ \frac{S_T^i}{N_T} | \mathcal{F}_t \right]$$

où  $Q^N$  est définie par :

$$E^{Q^N} [H] = E^{Q^M} \left[ \frac{M_T / N_T}{M_0 / N_0} H \right]$$



Exemple : Mesure forward neutre

Le bond zéron coupon dont la maturité concide avec la date du payment d'un produit financier peut servir de numéraire. Nous appellerons la mesure de probabilité associé  $Q_T$ .

Dans ce cas  $P(T, T) = 1$ , et par conséquent il suffit de calculer l'espérance du payoff (divisé par 1) sous  $Q_T$ . Si nous notons le payoff de ce produit  $H$ , alors son prix à l'instant 0 est donné par

$$P(t, T) E^{Q_T} [H | F_t]$$

Pour que cela nous soit utile, il faut que la dynamique de  $H$  soit connue sous  $Q_T$ . Ceci est vérifié pour les contrats payant un taux d'intérêt sur un nominal fixe. En effet  $(F(t; S, T))_t$  est une martingale

$$E^{Q_T} [F(t; S, T) | F_u] = F(u; S, T)$$

**Preuve :** Si nous disposons de  $\frac{P(t, S)}{(1 + (T - S)F(t; S, T))}$  au temps  $t$ , nous pouvons acheter  $\frac{1}{(1 + (T - S)F(t; S, T))}$  unités de l'obligation  $P(t, S)$ , nous obtenons  $\frac{1}{(1 + (T - S)F(t; S, T))}$  au temps  $S$ , cette somme là est, par définition de  $F$ , équivalente à l'obtention de 1 à l'instant  $T$ , qui exactement le payoff de l'obligation  $P(t, T)$ . Par principe de **non arbitrage**, ces deux investissement doivent avoir le même coût, ie :  $\frac{1}{T - S} \left( \frac{P(t, S)}{P(t, T)} - 1 \right)$  La preuve en découle.

### 1.3 PRODUITS FINANCIER D'INTERÊTS

Le développement de la section précédente nous sera utile pour pricer les dériver des taux. Considérons le cas particulier d'un call européen à maturité  $T$ , strike  $K$ , dont le sous-jacent est bond zéro coupon qui expire à l'instant  $S$ . Le payoff d'un tel contrat est connu :  $(P(T, S) - K)^+$ . Son prix à un instant antérieur  $t$  est

$$ZBC(t, T, S, K) := E^Q \left[ e^{-\int_t^T r_s ds} (P(T, S) - K)^+ | F_t \right]$$

Il est plus pratique de considérer la forward mesure, sous laquelle le prix du call s'écrit

$$ZBC(t, T, S, K) = P(t, T) E^{Q_T} [(P(T, S) - K)^+ | F_t]$$

De même, pour un put

$$ZBP(t, T, S, K) = P(t, T) E^{Q_T} [(K - P(T, S))^+ | F_t]$$

Cette écriture nous rappelle la formule de blackscholes pour les options sur les actions.

**Definition 8.** Swap : Un swap est un contract entre deux parties qui s'engagent à échanger des flux financiers pendant une durée et à une fréquence déterminées. la plupart du temps, ces flux sont déterminé comme étant l'intérêt à un taux fixe  $K$  contre un taux variable (LIBOR  $L(T_i)_i$  par exemplesur un notionnel  $N$ .

$$N \sum D_t^{T_i} \tau_i (L(T_{i-1}) - K)$$

**Definition 9.** Caplet/floorlet : Un caplet/floorlet peut être vu comme un call/put européenne sur un Son payoff est le suivant

$$\tau(L(T, S) - K)^+$$

$$Cpl(t, T, S, \tau, X) = E^Q \left[ e^{-\int_t^S r_s ds} \tau(L(T, S) - K)^+ | F_t \right] \quad (1.1)$$

$$= E^Q \left[ e^{-\int_t^S r_s ds} P(t, T) \tau(L(T, S) - K)^+ | F_t \right] \quad (1.2)$$

$$= E^Q \left[ e^{-\int_t^S r_s ds} (1 - (1 + X\tau)P(t, T))^+ | F_t \right] \quad (1.3)$$

$$= (1 + X\tau) E^Q \left[ e^{-\int_t^S r_s ds} \left( \frac{1}{1 + X\tau} - P(t, T) \right)^+ | F_t \right] \quad (1.4)$$

$$= (1 + X\tau) ZBP(t, T, S, \frac{1}{1 + X\tau}) \quad (1.5)$$

**Definition 10.** Cap/floor : Un cap/floorlet peut être vu comme une somme de caplets

$$N \sum D_t^{T_i} \tau_i (L(T_{i-1}) - K)^+$$

La forme des payoff indique que le cap permet de protéger son détenteur d'une hausse des taux Libor, et symétriquement que le floor protège d'une éventuelle baisse de ces taux.

## CHAPITRE 2

# LES DIFFÉRENTS MODÈLES DES TAUX D'INTERÊTS

Nous avons vu dans la partie précédente que pour trouver le prix des dérivés de taux et pouvoir calculer l'espérance, il faut donner la dynamique qui régit le sous-jacent, dans notre cas  $r_t$ .

Les professionnels des marchés de taux ont pris l'habitude d'évaluer les options vanille avec la formule de Black & Scholes. Ils font l'hypothèse implicite que les taux Libor  $L(t, T)$  forward sont log normaux dans la mesure forward associée à leur échéance :

$$dL_t(T, S) = \sigma L_t(T, S) dW_t$$

Ce modèle spécifie la dynamique de chaque taux Libor sous sa mesure forward associée. Cependant, il est indispensable d'un point de vue pratique de pouvoir exprimer les taux Libor forward sous une même mesure.

Diffuser des taux Libor de différentes maturités sous une même mesure forward peut introduire des distorsions lors du traitement numérique. En effet, les équations montrent que plus la maturité de la mesure utilisée pour diffuser un taux Libor forward est éloignée de  $T_i$ , plus le terme de tendance est important, et par conséquent plus l'erreur numérique due à une approximation pour calculer cette tendance risque d'être importante.

L'approche historique, décrit la dynamique du taux d'intérêt instantané comme étant conduite par un driver à une seule dimension. Ceci est pratique dans le sens où les prix de zéro coupon et le taux sont directement disponibles dans le modèle. De plus, certains produits financiers dépendent directement la courbe de rendement. Nous verrons que la donnée du taux instantané  $r_t$  permet de caractériser complètement courbe de taux.

Il est important que la dynamique de  $r_t$  soit à la fois riche pour pouvoir décrire la courbe de rendement observée dans le marché, et suffisamment simple pour que le temps nécessaire pour le calcul ne soit trop long. Le modèle de Hull White a été introduit en 1990.

$$dr_t = (\theta(t) - \beta(t)r_t)dt + \sigma(t)dW_t$$

Un des atouts majeure de ce modèle est la possibilité de simuler la dynamique de  $r_t$  par un arbre. Ceci étant essentiel pour pricer des produits qui dépendent de toute une partie de la courbe de taux, et non seulement d'un point précis. Les bermuda options en sont un exemple.

## 2.1 LE MODÈLE À DEUX FACTEURS

### 2.1.1 MOTIVATION

Considérons un produit  $E$  dont le payoff dépend du spread entre un taux d'intérêt cumulé entre 0 et  $T_1$  pour le premier et 0 et  $T_2$  pour le second.  $E$  dépend donc de la distribution jointe des deux taux.

Dans la réalité, on observe sur les marchés que les taux à différentes maturités ne sont pas corrélés. Si on regarde le taux 2Y et 10Y

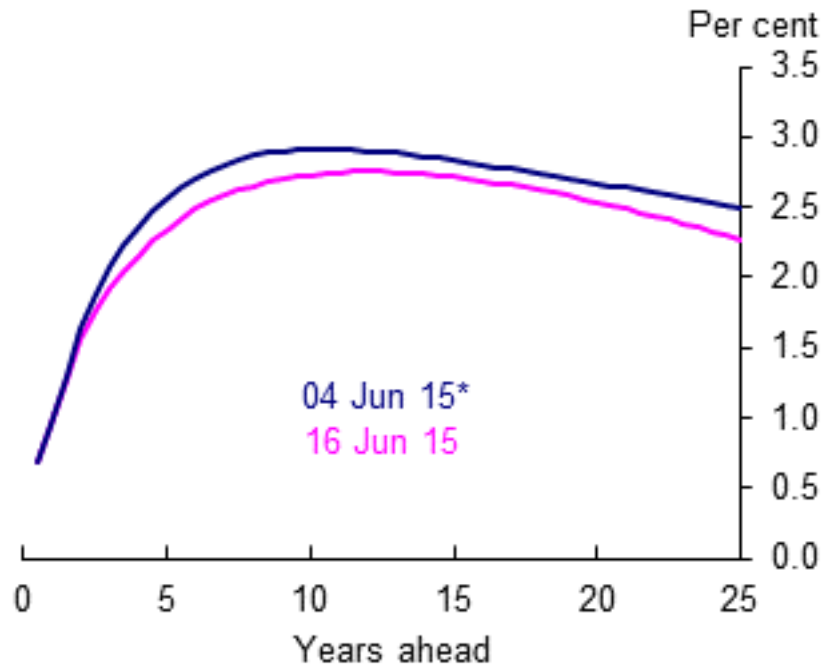


FIGURE 2.1 – Libor

La figure suivante, tiré cours [2] reproduit une matrice de corrélation par terme de variations quotidiennes de taux zéro-coupon. Il apparaît très clairement que des taux de maturités proches, comme le taux de maturité 3 ans et celui de maturité 4 ans, sont très corrélés, tandis que des taux de maturité éloignées (par exemple le taux 1 mois et le taux 10 ans) le sont très peu :

	1M	3M	6M	1A	2A	3A	4A	5A	7A	10A
1M	1									
3M	0.999	1								
6M	0.908	0.914	1							
1A	0.546	0.539	0.672	1						
2A	0.235	0.224	0.31	0.88	1					
3A	0.246	0.239	0.384	0.808	0.929	1				
4A	0.209	0.202	0.337	0.742	0.881	0.981	1			
5A	0.163	0.154	0.255	0.7	0.859	0.936	0.981	1		
7A	0.107	0.097	0.182	0.617	0.792	0.867	0.927	0.97	1	
10A	0.073	0.063	0.134	0.549	0.735	0.811	0.871	0.917	0.966	1

FIGURE 2.2 – Tableau de corrélation

### 2.1.2 LIMITE DES MODÈLE À UN SEUL FACTEUR

Montrons dans un premier temps pourquoi un modèle à un seul facteur n'est pas suffisant pour pricer ces produits qui dépendent non seulement de la distribution de chaque courbe de taux, mais aussi de leur corrélation.

Rappelons la dynamique de  $r_t$  dans le cadre de Hull White :

$$r_t = k(\theta - r_t)dt + \sigma dW_t$$

La formule analytique de l'obligation zéro coupon est donc

$$P(t, T) = A(t, T) \exp(-B(t, T)r_t)$$

En particulier le taux d'intérêt cumulé est une donné par une transformation affine du taux instantané :

$$R(t, T) = \frac{\ln P(t, T)}{T - t} =: a(t, T) + b(t, T)r_t$$

Le payoff du produit  $E$  est donc fonction de la distribution jointe de  $R(0, T_1)$  et  $R(0, T_2)$ . Sauf que :

$$\text{Cor}(R(0, T_1), R(0, T_2)) = 1$$

On en déduit qu'un choc à  $r_t$  agit de la même manière sur toutes les courbes.

Un modèle à un seul facteur ne capture pas ce comportement. Essayons de pallier à ce problème en rajoutons un facteur à ce modèle.

Dans cette section nous considérons un modèle où le taux d'intérêt instantané est donné par une somme de deux facteurs gaussiens centrés et corrélés. Dans ce modèle doit sa popularité au fait que le prix des obligations zéro coupon admet une formule exact, ainsi que le prix des caps et des floors.

$$\begin{aligned} dx &= -\beta^x x(t)dt + \sigma^x dW_t^x \\ dy &= -\beta^y y(t)dt + \sigma^y dW_t^y \\ dr &= x + y \end{aligned}$$

## 2.2 APPROXIMATION DE LA SOLUTION PAR UN ARBRE BINOMIAL

L'équation (\*) s'intègre simplement en :

$$\begin{aligned} x(t) &= x(s)e^{-\beta^x(t-s)} + \sigma^x \int_s^t e^{-\beta^x(t-u)} dW_u^x \\ y(t) &= y(s)e^{-\beta^y(t-s)} + \sigma^y \int_s^t e^{-\beta^y(t-u)} dW_u^y \end{aligned}$$

### CONSTRUCTION

Cette méthode a été d'abord suggéré par Hull-White (1994)

On discrétise l'intervalle  $[0, T]$  avec les temps  $T_i = i\Delta t$ , où  $\Delta t = \frac{T}{N}$ . Nous donnons une approximation de la dynamique processus  $x$  et  $y$ , par une suite de variable de discretisé  $((\tilde{x}_i, \tilde{y}_i) \approx (x(i\Delta t), y(i\Delta t)))_i$ . Pour cela nous calculerons les deux premier moment de  $(x, y)$ .

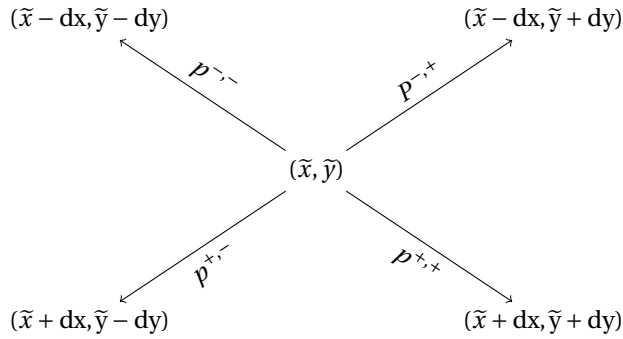
Nous faisons le calcul pour  $x$ , la même formule s'appliquera à  $y$ .

$$\begin{aligned} E(x(t + \Delta t)|F_t) &= x(t)e^{-a\Delta t} \\ V(x(t + \Delta t)|F_t) &= \frac{\sigma^x{}^2}{2a}(1 - e^{-2a\Delta t}) \\ \text{Cov}\{x(t + \Delta t), y(t + \Delta t)|F_t\} &= \frac{\sigma^x \sigma^y \rho}{a + b}(1 - e^{-(a+b)\Delta t}) \end{aligned}$$

Pour que le  $((\tilde{x}_i, \tilde{y}_i)$  et  $(x(i\Delta t), y(i\Delta t)))_i$  aient les même moment, la loi de  $((\tilde{x}_i, \tilde{y}_i)$  est donnée par :

$$P(\tilde{x}_{i+1} = \tilde{x}_i + a dx, \tilde{y}_{i+1} = \tilde{y}_i + b dy | \tilde{x}_i, \tilde{y}_i) = p^{a,b}(\tilde{x}_i, \tilde{y}_i)$$

où



- $a, b \in \{-1, +1\}$
- $p$  est donnée par :

$$p^{a,b}(x, y) = \frac{1 + a\rho}{4} - b \frac{\beta^y \sigma^x y + a \sigma^x \sigma^y x}{4 \sigma^x \sigma^y} \sqrt{\Delta t}$$

Le schéma suivant résume les transitions du processus  $(\tilde{x}, \tilde{y})$  ;

Nous appelons *slice* l'ensemble des noeuds qui sont équi distant de la racine. Une slice représente la distribution du processus  $(\tilde{x}_i, \tilde{y}_i)$  à un instant donné.

Le pricing se fait en deux temps :

- On diffuse le processus  $(\tilde{x}, \tilde{y})$  dans l'arbre en prenant soin de calculer la probabilité de transition d'un état à un autre
- On “drawback” dans l'arbre en partant de la date à laquelle on fait le payoff, en \*\*\*discountant\*\*\*

#### PETITE DISCUSSION SUR LA COURBE D'ACTUALISATION VS LA COURBE DE DIFFUSION

Avant la crise de 2008, il était d'usage courant que les banques considèrent le taux Libor comme reflétant la réalité du marché de crédit inter-bancaire. Le taux est publié quotidiennement par taux réel auquel les banques sont prête à se prêter de l'argent est appelé taux OIS (Overnight Index Swap). Pendant la crise, le spread entre LIBOR et OIS était si grand qu'il devenait impossible à ignorer. Depuis tous les modèles de taux intègrent deux courbe, une pour la diffusion (LIBOR par exemple) et une autre pour l'actualisation (OIS). Dans le développement de cet article, nous ignorons cette différence.

## 2.3 AMÉLIORATIONS

Si nous implémentons l'arbre de façon naïve (voir ?? ), le nombre de noeuds augmente de façon exponentielle en fonction du nombre de pas de temps. En pratique ceci est problématique et conduit vite à une saturation de mémoire. Dans l'exemple simplifié ci-dessus nous traçons l'arbre de diffusion du premier facteur  $(x(t))$ . A chaque pas de temps le nombre de noeuds double, ie pour  $n$  pas de temps, nous nous retrouvons avec  $2^n$  noeuds pour un facteur, ou  $4^n$  pour deux.

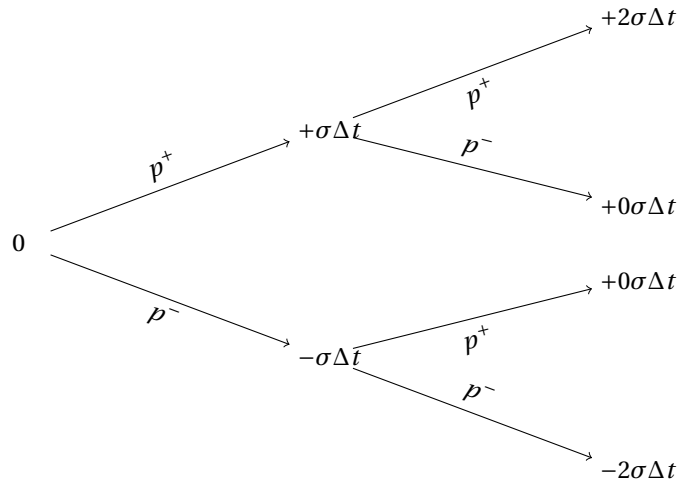


FIGURE 2.3 – Arbre construit de façon naïve

Remarquer que  $(x, y)_{i\tilde{t}}$  est un processus markovien homogène à valeurs discrètes nous permet d'optimiser la simulation de l'arbre. En effet, il nous suffit de calculer la table de transition une fois au début du programme et de la réutiliser pour avancer/reculer dans le temps. (voir ??)

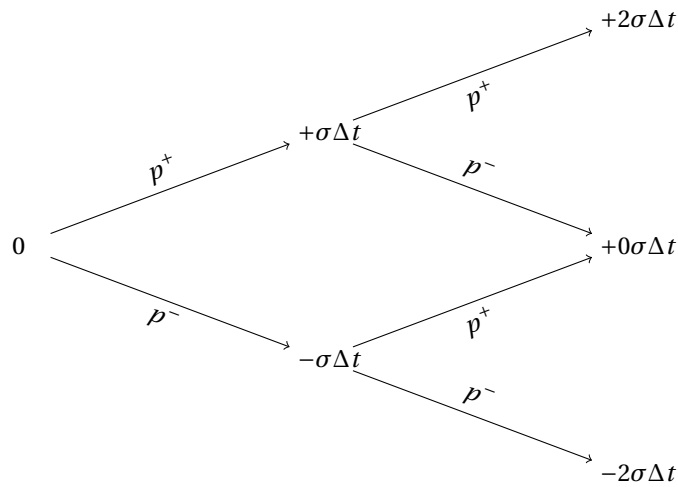


FIGURE 2.4 – Arbre optimisé

Voir l'annexe pour l'implémentation en python. Ci-dessous une visualisation graphique de plusieurs slices à différents instants, nous rappelons qu'une slice est l'ensemble des valeurs possibles d'une fonction aléatoire, le prix d'une obligation zéro coupon en loccurence, à un instant précis.

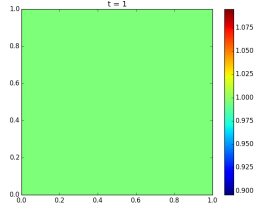


FIGURE 2.5 –  $t = 1$

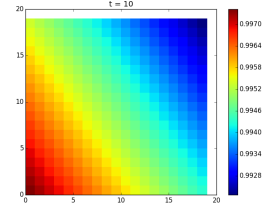


FIGURE 2.6 –  $t = 10$

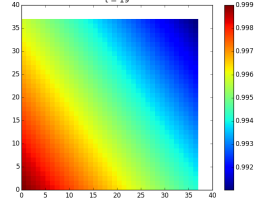


FIGURE 2.7 –  $t = 20$

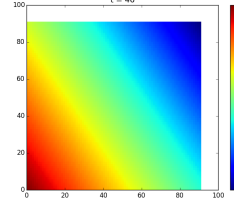


FIGURE 2.8 –  $t = 30$

Une autre amélioration possible est de trouver une formule analytique pour certains produits. En effet, si nous reprenons l'exemple d'un cancellable spread option 2Y10Y dont la maturité est dans 5 ans, nous devrions normalement construire l'arbre jusqu'en 2030 pour avoir le taux 10Y en 2020. Nous pouvons éviter cela en fournissant directement une formule exacte pour les zéro coupons. C'est ce que nous faisons dans la partie suivante.

### 2.3.1 FORMULE EXACTE

Cette partie est fortement inspirée de [1].

On rappelle l'expression du prix de l'obligation zéro coupon sous la mesure risque neutre  $Q$

$$P(t, T) = E^Q \left[ e^{-\int_t^T r_u du} \right]$$

Notons  $I(t, T) := \int_t^T x(u) + y(u) du$ , et montrons que conditionnellement à l'information accumulée jusqu'au temps  $t$ , que c'est une variable normale d'espérance  $M(t, T)$  et de variance  $V(t, T)$  où :

$$M(\beta^y, t, T) := \frac{1 - e^{-\beta^y(T-t)}}{\beta^y} \quad (2.1)$$

$$V(\beta^y, \sigma^x, \beta^y, \sigma^y, t, T) := \frac{\sigma^x \sigma^y}{\beta^y \beta^y} \left[ T - t + \frac{e^{-\beta^y(T-t)} - 1}{\beta^y} + \frac{e^{-\beta^y(T-t)} - 1}{\beta^y} - \frac{e^{-(\beta^y + \beta^y)(T-t)} - 1}{\beta^y + \beta^y} \right] \quad (2.2)$$

$$V(t, T) := V(\beta^y, \sigma^x, \beta^y, \sigma^x, t, T) + V(\beta^y, \sigma^y, \beta^y, \sigma^y, t, T) + 2\rho V(\beta^y, \sigma^x, \beta^y, \sigma^y, t, T) \quad (2.3)$$

$$M(t, T) := M(\beta^y, t, T)x(t) + M(\beta^y, t, T)y(t) \quad (2.4)$$

*Démonstration.* Par simple calcul [1] :

$$\int_t^T x(u) du = \frac{1 - e^{-\beta^y(T-t)}}{\beta^y} + \frac{\sigma^x}{\beta^y} \int_t^T [1 - e^{-\beta^y(T-t)}] dW_u^x$$

$$E^Q \left[ \int_t^T x(u) du \right] = M(\beta^y, t, T)x(t)$$

$$Var \left( \int_t^T x(u) du \right) = \left( \frac{\sigma^x}{\beta^y} \right)^2 E^Q \left[ \int_t^T [1 - e^{-\beta^y(T-t)}]^2 du \right] = V(\beta^y, \sigma^x, \beta^y, \sigma^x, t, T)$$

(isométrie d'Itô)

□



Nous avons donc

$$P(t, T) = \exp \left( - \int_t^T r_u du + \frac{1}{2} \text{Var} \left( - \int_t^T r_u du \right) \right) \quad (2.5)$$

$$= \exp \left( -M(\beta^y, t, T)x(t) - M(\beta^y, t, T)y(t) + \frac{1}{2} V(t, T) \right) \quad (2.6)$$

Nous utiliserons cette formule directement dans le pricer, ce qui nous évitera de construire l'arbre jusqu'à la date de maturité du dernier zéro coupon.

Pour mesurer l'impact de ce changement sur le temps d'exécution du programme, nous avons utilisé le logiciel **valegrind**. Ce logiciel permet le profilage du code, ie il permet de contrôler :

- la liste des fonctions appelées et le temps passé dans chacune d'elles
- l'utilisation processeur
- l'utilisation mémoire

Ci-dessous deux graphes produits par le logiciel qui illustre le nombre de fois que la fonction responsable de l'évolution de l'arbre `Hyb4_UpdateStatePricesOnly` est appelée :

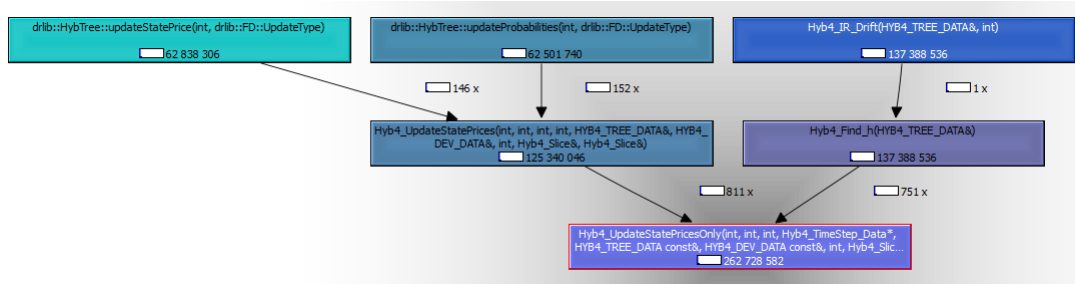


FIGURE 2.9 – En utilisant l'arbre pour le calcul des  $P(t, T)$

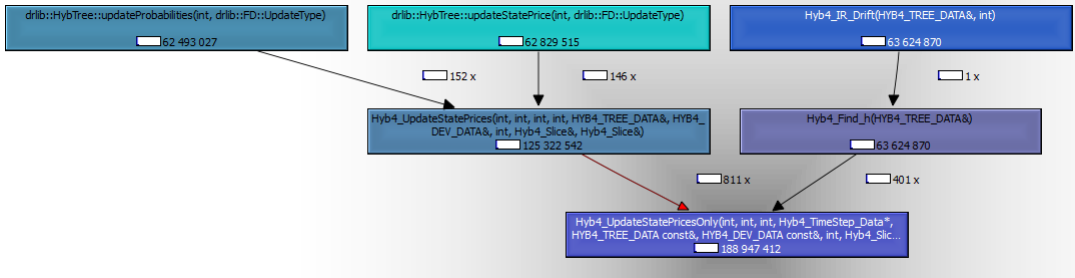


FIGURE 2.10 – En utilisant la formule exacte

Le nombre d'appels à cette fonction est environ réduit de moitié. Cette dernière étant très gourmande en ressource, le temps total requis par le programme est réduit de 30%.

### 2.3.2 AJOUT D'UN SHIFT DÉTERMINISTE

Le modèle, tel que développé jusqu'à présent, présente un inconvénient majeur : à tout instant,  $r_t$  est symétriquement distribué autour de 0. Ceci ne correspond pas à la réalité, puisque les taux négatifs ne sont observés dans les marchés que dans de très rares circonstances ( en Europe après crise inter-bancaire de 2009, Au Japon après des années de déflation ). Une autre raison est que le modèle ne permet pas de retrouver les prix des obligations zéro coupon.

Pour pallier à ce problème, nous rajoutons une fonction  $\phi$  au taux  $r_t$ . La fonction déterministe  $\phi(t)$  permet de fitter exactement la courbe de taux observée. Dans la partie “Calibraion nous” nous verrons comment calculer cette fonction à partir des prix de bonds zéro coupon.

En prenant en compte ce changement, le prix de l’obligation zéro coupon devient :

$$P(t, T) = E^Q [e^{-r_t + \phi(t)}]$$

Nous prendrons soin de modifier l’étape de “draw back” dans l’abre en changeant le facteur d’actualisation.

Le processus est markovien

### 2.3.3 OPTIMISER LA TAILLE DES SLICES

La slice que nous avons construite est rectangulaire. La plage que nous autorisons à  $(x, y)$  doit logiquement dépendre de leur corrélation. Ainsi pour simuler deux processus parfaitement corrélés, il suffit de simuler l’un deux, la slice dans ce cas est simplement linéaire. Dans le cas ou ils sont indépendants, il est nécessaire de simuler les deux, et la slice sera parfaitement rectangulaire. Dans le cas général nous utilisons la méthode suivante

Nous décomposons la source de stochasticité  $dW_t$  en deux composantes indépendantes :

$$dW_t := (dW_t^x, dW_t^y)^T = A(t)dZ_t$$

où

$$A(t) := \begin{pmatrix} \sigma^x & 0 \\ \rho\sigma^y & \sqrt{1-\rho^2}\sigma^y \end{pmatrix}$$

$Z = (Z^1, Z^2)$  un mouvement brownien 2D à composantes indépendantes.

A l’instant  $t_n$ ,

$$x(t_{n+1}) := x_i(t_n)(1 - \beta^y \Delta) + \dots Z_1$$

$$y(t_{n+1}) := y_i(t_n)(1 - \beta^y \Delta) + \dots Z_1 + \dots Z_2$$

Les limites de l’abre sont définie comme étant les bords de l’ellipsoid décrit par l’équation :

$$Z_1^2 + Z_2^2 \leq n_{\sigma^x}^2$$

Où  $n_{\sigma^x}$  est sans unité et dénote le degré de déviation de la moyenne qu’on autorise.

### 2.3.4 PARAMÈTRES DÉPENDANT DU TEMPS

Dans la partie précédente, les paramètres  $(\sigma^x, \sigma^y, \beta^y, \beta^y, \rho)$  étaient constantes. Pour plus de souplesse, nous considérons des variables qui dépendent du temps, ou plus précisément constantes par morceaux.

Le calcul du prix des obligations zéro coupon est fourni en annexe.

Deux problèmes cependant avec cette méthode :

- Le calcul est beaucoup plus long
- possibilité de overfitter les données historique du marché, ce qui affecte négativement le pouvoir prédictive du modèle

## CHAPITRE 3

# RÉSULTATS

Une fois toutes ces modifications prises en compte, une slice générée par l'arbre ressemble au schéma ci-dessous.  $(i, j)$

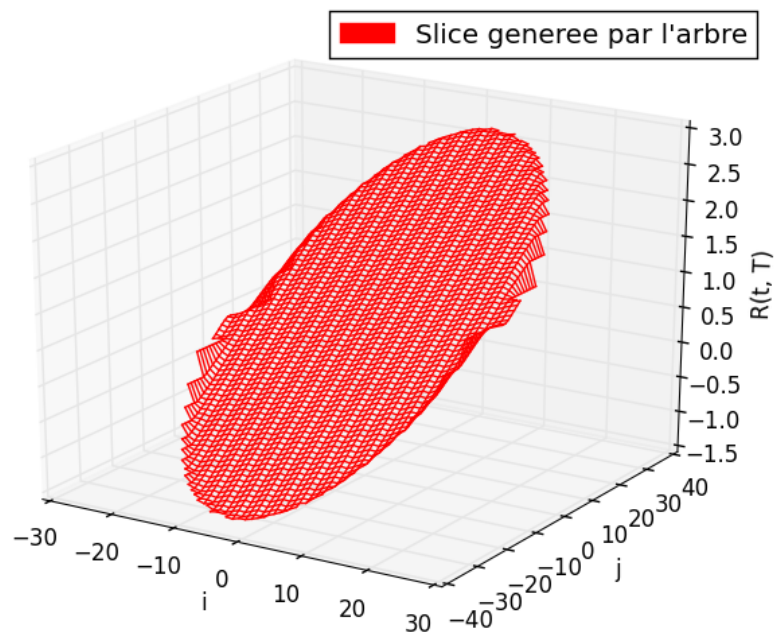


FIGURE 3.1 – Slice 3D

## PERFORMANCE

L'arbre ainsi construit est puissant dans le sens où il permet de pricer quasiment tous les instruments dont on a besoin. Cependant ce modèle souffre de quelques défaut observé en pratique.

- De par sa construction, l'arbre est bornée. Ce dernier ignore donc le comportement des driver loin

de leurs moyenne. Si ceci n'est pas un problème dans le cadre de variable normale (poids centré), ceci peut engendrer des erreurs non négligeables quand les variables ont des distribution à queues épaisses.

- Discrétisation des processus
- probabilités négatives

Dans le cas des obligations zéro coupon, nous disposons d'une formule exacte pour calculer les prix et les comparer à celles produites par l'arbre. La figure ci-dessus illustre la différence entre les prix théorique et les prix calculés l'arbre :

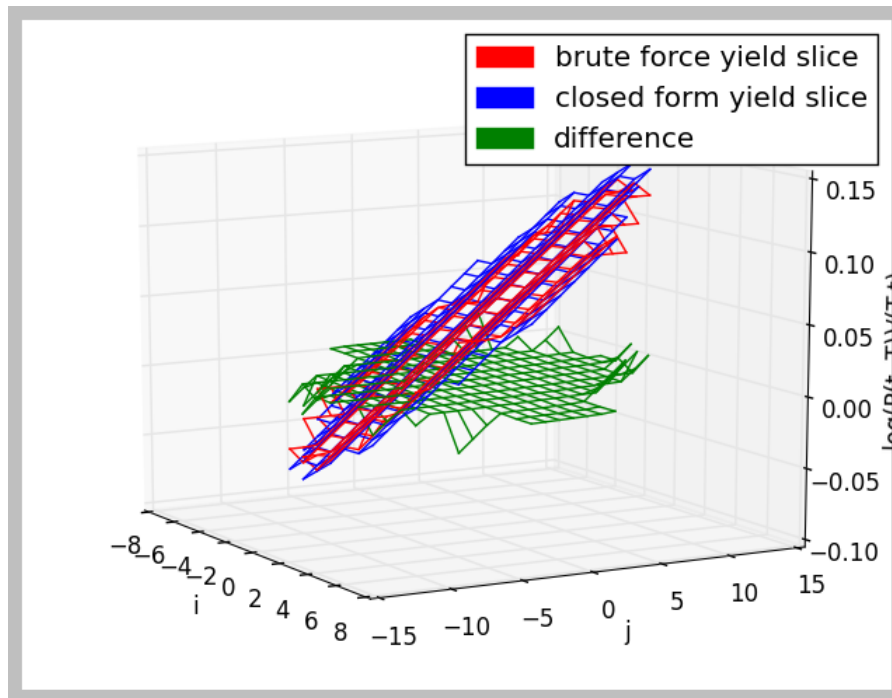


FIGURE 3.2 – Erreur bf vs cf Slice 3D

## CHAPITRE 4

# APPLICATION : CALIBRATION ET PRICING

Notre modèle possède à un certains nombres de paramètres libre que nous devons fixer. Pour celà, nous choisissons des actifs tradables dans le marché, dont le prix est donc connus, que nous appellerons benchmark. Nous essayerons ensuite de trouver les paramètres qui reproduisent le mieux ces prix là. Cette procédure est appelé calibration. Une question naturelle qui se pose est de savoir quels actifs choisir pour la calibration. Il existe plusieurs réponses possibles, en pratique on essaye de trouver un produit à la fois simple et liquide.

Dans notre cas il est indispensable que le modèle puissent retrouver les prix des bond zéro coupons. Cela se fait en calibrant  $\phi$

Nous essayerons en plus de retrouver les prix de caplets.

### 4.1 CALIBRATION DU DRIFT

Le modèle gaussien à deux facteurs est calibré sur la courbe  $P^M(0, T)$ ,  $T > 0$  de prix d'obligations zéro-coupon observés sur le marché si et seulement si  $\phi$  est définie par :

$$P^M(0, T) := e^{\int_0^T \phi(s) ds + Mx(T) + My(T) + \frac{1}{2}V}$$

Ce qui est équivalent à

$$\int_t^T \phi(s) ds := \frac{P^M(0, T)}{P^M(0, t)} e^{-\frac{1}{2}(V(0, T) - V(0, t))}$$

Cependant, l'arbre ainsi simulé ne redonnera par exactement les prix des obligations zéro-coupon  $P^M(0, T_i)$ . En effet, dans un arbre le taux simulé est considéré constant sur la période  $[T_i, T_{i+1}]$ , donc tout se passe comme si l'arbre simulait en fait un taux zéro-coupon  $R$ . En d'autres termes, le prix d'une obligation zéro-coupon de maturité  $T_1$  vaut  $P(0, T_1) = e^{-R(0, T_1)\Delta t}$  mais ce prix calculé directement sur l'arbre s'écrira  $e^{-r_0\Delta t}$ .

On propose donc ici une méthode alternative pour calibrer la fonction  $\Phi$  de façon à ce que l'arbre reproduise les prix des obligations zéro coupon. Il est intéressant de noter que nous n'avons pas besoin de toute la fonction  $\Phi$ , mais juste de sa somme entre les instant  $t_i$  et  $t_{i+1}$ . Nous noterons cette quantité  $h_i$

Soit  $\Pi_j$  state price (arrow debrew) (actif qui paye 1 si le noeud  $(t_n, j)$  est atteint) Le facteur d'actualisation que l'arbre utilise peut s'écrire :

$$D_j(h_n) := e^{-r(t_n)(t_{n+1}-t_n)}$$

Nous retrouvons les  $h_n$  récursivement en utilisant les deux équations suivantes :

$$\Pi_j(t_{n+1}) = \sum_{j' \text{ noeud}} \Pi_{j'}(t_n) p_{j',j}(t_n) D_{j'}(h_n) \quad (4.1)$$

$$\sum_{j \text{ noeud}} \Pi_j D_j(h_n) = P(0, t_{n+1}) \quad (4.2)$$

$p_{j,j'}(t_n)$  est la probabilité que le processus  $(\tilde{x}, \tilde{y})$  passe du noeud  $j$  au noeud  $j'$  à l'instant  $t_n$ . Cette fonction a été calculé dans la partie "construction de l'arbre".

En effectuant un développement de Taylor de la fonction  $D_j$ , les équations précédente deviennent polynomiales en  $h_n$ , et leur résolution est aisée.

## 4.2 MÉTHODE DE CALIBRATION

Les traders préfèrent bien avoir un contrôle fin sur le modèle qui reflète leur sensations sur le marché. Pour cela, le modèle doit être paramétrable, et les paramètres doivent avoir un sens/être compris par les traders.

Le modèle G2++ est maintenant bien compris, et le rôle de chacun de ses paramètres  $\theta = (\sigma^x, \sigma^y, \beta^x, \beta^y)$

La méthode de calibration par brute force repose sur les étapes suivantes :

- On s'autorise un intervalle pour les paramètres
- On utilise une grille (définie par le pas) pour définir les valeurs autorisées pour chaque paramètre. (Tradeoff entre pas petite grande précision et temps de calculs)
- On calcule le prix des caplets associés par la formule théorique donnée ci-dessous
- On choisit les paramètres qui reflètent le mieux les prix du marché. La plupart du temps on minimise l'erreur  $L_2$  entre les prix empirique et les prédictions du modèle.

### Prix d'un caplet

Nous rappelons l'expression du caplet

$$CPL(t, T, S, \tau, X) = (1 + X\tau) ZBP(t, T, S, \frac{1}{1 + X\tau})$$

ainsi que celle du put  $ZBP$  sous la mesure  $Q_T$

$$ZBP(t, T, \tau, K) = E^{Q_T} [(K - P(t, S))^+]$$

Maintenant que nous connaissons la dynamique de  $P(t, T)$ , nous pouvons fournir une formule explicite pour le prix des caplets.

$P(t, T)$  admet une distribution normale sous  $Q_T$  conditionnellement à  $F_t$ , dont on peut (voir [1]) calculer l'espérance et la variance :

Le prix théorique est donc donné par :

$$ZBC(t, T, S, K) = -P(t, T)N(d_1) + P(t, T)KN(d_2)$$

où

$$\begin{aligned} d_{1/2} &:= \frac{\ln \frac{KP(t,T)}{P(t,S)}}{\Sigma} + / - \frac{1}{2} \Sigma \\ \Sigma^2 &:= \Sigma^{x,x} + \Sigma^{y,y} + 2\rho \Sigma^{x,y} \\ \Sigma^{x,y} &:= \sigma^x M^x(t, T) M^y(t, T) \frac{1 - e^{(\alpha+\beta)(T-t)}}{\alpha + \beta} \end{aligned}$$

Malheureusement, les quotes de caplets, contrairement au caps, ne sont pas directement disponibles sur les marchés. Mais nous pouvons induire les prix de caplets à partir du prix de certains caps.

Nous utiliserons la table de données suivantes, fournie par [3] :

L'optimisation numérique donne :

$$(\beta^x, \beta^y, \sigma^x, \sigma^y, \rho) = (0.62, 0.025, 0.0069, 0.0081, 0.96)$$

Le coefficient de corrélation  $\rho$  est proche de 1, ce qui était attendu puisque le pricing de caplets (et donc de caps) ne prend pas en compte la corrélation des taux cumulées entre deux dates différentes.

Index	Maturity	Price
0	1	0.972411
1	2	-1.334463
2	3	0.614967
3	4	0.240523
4	5	0.877871
5	7	-0.665534
6	10	-0.151002
7	15	0.050241
8	20	0.133609

TABLE 4.1 – Prix marché de caps

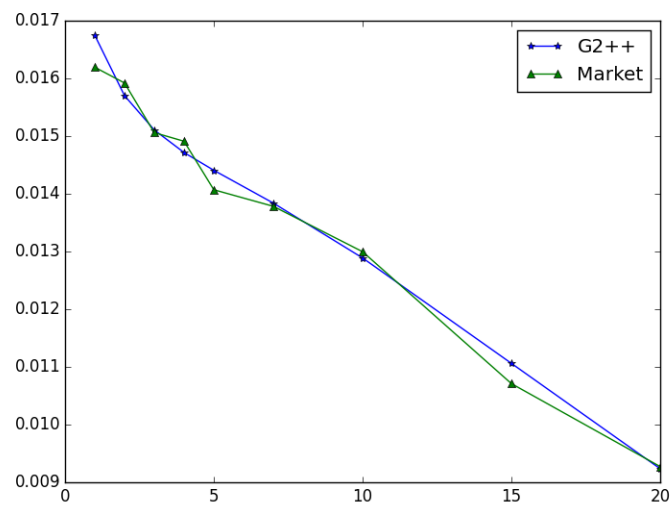


FIGURE 4.1 – Calibration des caps

Nous calibrons maintenant les swaptions,  $\rho > 0$

Nous comparons le résultat pour le modèle à un facteur (ie on force  $y$  à être identiquement nul) et deux facteurs :

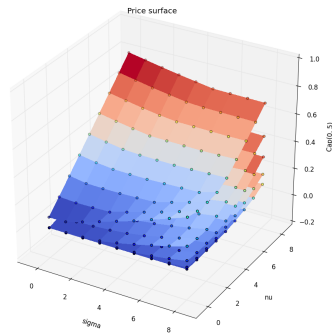


FIGURE 4.2 – Surface théorique en bleu -

**Un mot sur le multi threading** La méthode de calibration proposée ci-dessus à l'avantage d'être facilement parallélisable. En effet, les calculs aux différents points de la grille sont indépendants. Ceci permet de profiter de gpu



## CHAPITRE 5

# CONCLUSION

Une partie très importante dans la valorisation des produits financiers est dédiée à l'étude et l'implémentation des modèles de taux. Nous étions intéressés par la méthode de calibration et de modélisation du modèle Gaussien G2++ à deux facteurs. A cause des limites des modèles à un seul facteur, nous avons estimé que le modèle à deux facteurs peut améliorer la capacité de modélisation des modèles de taux. Les résultats a posteriori confirment notre intuition.

# ANNEXE A

## CODE DE SIMULATION EN PYTHON

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from multiprocessing import Pool
import multiprocessing
import scipy.stats as ss

directions = [ (1, 1), (1, -1), (-1, 1), (-1, -1) ]
a = 0
class TreePricer:
    """ Classe responsable de construire l'arbre de simulation.
    Garde en memoire les parametre du model"""

    def __init__(self, sigma=.02, nu=.03, rho=0.5, a=0.02, b=0.01, draw=False):
        self.sigma = sigma
        self.nu = nu
        self.rho = rho
        self.a = a
        self.b = b
        self.Dt = 1. / 365
        self.sqrtDt = np.sqrt(self.Dt)
        self.dx = self.sigma * self.sqrtDt
        self.dy = self.nu * self.sqrtDt
        self.draw = draw

    # Calculer la probabilite de transition en partant de (x, y)
    def calc_proba(self, x, y):
        return [ (1+j*self.rho) / 4 + i * (self.b*self.sigma*y + j * self.a*self.nu*x) * self.sqrtDt / (4*
                for (i, j) in [ (-1, 1), (1, -1), (-1, -1), (1, 1) ])

    # Renvoie les 4 noeuds fils dans l'arbre binomial
    def calc_node(self, x, y):
        return [ ((x+i)*self.dx, (y+j)*self.dy) for (i, j) in directions ]

    # Retour un pas de temps en arriere en actualisant
    def discount(self, payoff, proba_table, t):
        disc_payoff = np.zeros_like(payoff)
        n = len(payoff)
```

```

    for i in range(1, n-1):
        for j in range(1, n-1):
            neighbours = self.calc_node(i, j)
            for (c, (dir_x, dir_y)) in enumerate(directions):
                p = proba_table[i-dir_x][j-dir_y][c]
                disc_factor = np.exp(-sum(neighbours[c])*self.Dt)

                disc_payoff[i][j] += p * disc_factor * payoff[i-dir_x][j-dir_y]
    return disc_payoff[1:-1, 1:-1]

# en partant du payoff a l'instant n, renvoie la distribution des prix a l'instant t0
def price(self, t0, payoff):
    n = len(payoff)
    proba_table = np.array([[self.calc_proba(i*self.dx, j*self.dy) for j in range(-n/2, n/2)] for i in
    draw_payoff(payoff, n/2)
    for t in range(n/2-t0):
        payoff = self.discount(payoff, proba_table, t)
        if self.draw and (n/2-t) % (n/10) == 1:
            print t, n/2 -t
            draw_payoff(payoff, n/2-t)
    return payoff

# Calcule de l'obligation zero coupon
def Mx(self, t, T):
    return (1 - np.exp(-self.a * (T-t)))/self.a

def My(self, t, T):
    return (1 - np.exp(-self.b * (T-t)))/self.b

def V(self, t, T):
    a = self.a
    b = self.b
    V1 = (self.sigma / a)**2 * (T-t + 2*(np.exp(-a*(T-t))-1)/a - (np.exp(-2*a*(T-t))-1)/(2*a) )
    V2 = (self.nu / b)**2 * (T-t + 2*(np.exp(-b*(T-t))-1)/b - (np.exp(-2*b*(T-t))-1)/(2*b) )
    V3 = 2*self.rho*(self.sigma *self.nu) / (a*b) * (T-t + (np.exp(-a*(T-t))-1)/a + (np.exp(-b*(T-t))
    return V1 + V2 + V3

def P(self, t, T, x, y):
    return np.exp( -self.Mx(t, T) * x- self.My(t, T) * y + 0.5 * self.V(t, T))

# Price les caplet en utilisant l'arbre
def tree_caplet(self, t, T, K):
    nsteps = T
    X = np.arange(-nsteps, nsteps+1) * self.sigma
    Y = np.arange(-nsteps, nsteps+1) * self.nu
    X, Y = np.meshgrid(X, Y)
    R = X + Y
    payoff = np.maximum(R - K, 0) * (T - t)
    return self.price(t, payoff)

# Price les caplets en utilisant la closed form
def caplet_sigma(self, t, S, T):
    Mx = self.Mx(T, S)
    My = self.My(T, S)

```

```

a = self.a
b = self.b

Sigma = self.sigma **2 * Mx * Mx * (1-np.exp(-2*a*(T-t))) / (2*a) \
        + self.nu **2 * My * My * (1-np.exp(-2*b*(T-t))) / (2*b) \
        + 2 * self.rho * self.sigma * self.nu * Mx * My * (1-np.exp(-(a+b)*(T-t))) / (a+b)
return Sigma

# Prix d'un call sur taux
def zero_bond_call(self, t, T, S, x, y, K):
    bondT = self.P(t, T, x, y)
    bondS = self.P(t, S, x, y)
    Sigma = self.caplet_sigma(t, T, S)
    d1 = np.log( (K * bondT) / bondS) / Sigma - 0.5 * Sigma
    d2 = d1 + Sigma
    phi = ss.norm.cdf
    return -bondS * phi(d1) + K*bondT*phi(d2)

# Prix d'un put sur taux
def zero_bond_put(self, t, T, S, x, y, K):
    bondT = self.P(t, T, x, y)
    bondS = self.P(t, S, x, y)
    Sigma = self.caplet_sigma(t, T, S)
    d1 = np.log( (K * bondT) / bondS) / Sigma - 0.5 * Sigma
    d2 = d1 + Sigma
    phi = ss.norm.cdf
    return -bondS * phi(d1) + K*bondT*phi(d2)

# Prix d'un caplet(formule exact)
def cf_caplet(self, t, T, S, x, y, K):
    return (1 + K * (S-T) ) * self.zero_bond_put(t, T, S, x, y, 1/(1+K*(S-T)))

# Exemple d'utilisation

# Plot de la surface payoff en 2D
def draw_payoff(payload, t):
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.set_title('t = %d' % t)
    mesh = ax.pcolormesh(payload);
    fig.colorbar(mesh);

def draw_payoff_3d(H):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    X,Y = np.mgrid[:len(H), :len(H[0])]
    ax.plot_surface(X, Y, H, rstride=1, cstride=1, cmap=plt.cm.coolwarm,
                    linewidth=0, antialiased=False)
    ax.scatter(X.ravel(), Y.ravel(), H.ravel(), c=H.ravel())
    plt.show()

def price_caplet(t, T, K, a, b, sigma, nu, rho):
    pricer = TreePricer(a=a, b=b, sigma=sigma, nu=nu, rho=rho)
    return pricer.tree_caplet(t, T, K)[T-t:T+t+1, T-t:T+t+1]

```

```

def price_caplet_cf(t, T, S, K, a, b, sigma, nu, rho):
    pricer = TreePricer(a=a, b=b, sigma=sigma, nu=nu, rho=rho)
    return pricer.cf_caplet(t, T, S, K, 0, K)

def draw_surface():
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.set_xlabel('T')
    ax.set_ylabel('K')
    ax.set_zlabel('Caplet')
    ax.set_title('Price surface')

    H = np.array([[ price_caplet_cf(t=0, T=5, S=5+i, K=0.001*j, sigma=0.01, nu=0.02, rho=-0.5) for j in range(100) ] for i in range(100)])
    X,Y = np.mgrid[:len(H), :len(H[0])]
    ax.plot_surface(X, Y, H, rstride=1, cstride=1, cmap=plt.cm.coolwarm,
                    linewidth=0, antialiased=False)
    ax.scatter(X.ravel(), Y.ravel(), H.ravel(), c=H.ravel())
    plt.show()

def tree_zero_coupon():
    payoff = np.ones((99, 99))
    pricer = TreePricer(draw=True)
    pricer.price(0, payoff)
    plt.show()

#tree_zero_coupon()
#draw_surface()

```

## ANNEXE B

# FORMULE EXACTE DES OBLIGATIONS ZÉRO COUPON POUR DES PARAMÈTRES DÉPENDANT DU TEMPS

Dans cette partie nous détaillerons le calcul de prix d'obligation zéro coupon dans le modèle de Hull-White à deux facteurs dans le cas où les paramètres sont dépendants du temps, ou plus précisément, constants par morceaux.

Le modèle est toujours markovien, c'est à dire que nous pouvons toujours écrire  $P(t, T)$  comme une fonction déterministe de  $(x(t), y(t))$  :

$$P(t, T) := e^{\int_t^T \phi_s ds - M_x(t, T)x(t) - M_y(t, T)y(t) + \frac{1}{2}V(t, T)}$$

Nous calculerons ici  $M_x$ ,  $M_y$  et  $V$

### RAPPEL DU MODÈLE

$$dx_t = -\beta_x x_t dt + \sigma_x dW_t^1$$

$$dy_t = -\beta_y y_t dt + \sigma_y dW_t^2$$

$$r_t = \phi(t) + x_t + y_t$$

$$\rho_t = \langle dW_t^1, dW_t^2 \rangle$$

### CAS PARTICULIER : MODÈLE À UN SEUL FACTEUR

Nous commencerons par le cas particulier où  $y(t) = 0$ , l'équation différentielle vérifiée par  $x(t)$  s'intègre facilement en :

$$x(t) = \sum_{t_i < t} \sigma_i \int_{t_i}^{t \wedge t_{i+1}} e^{-a(t \wedge t_{i+1} - s)} dW_1(s)$$

Nous devons maintenant intégrer la fonction  $x$  entre  $t_0$  et  $t_f$  en la décomposant en somme d'intégrales entre les instants  $t_i$  et  $t_{i+1}$  où tous les paramètres sont constants et l'intégrale se calcule facilement.

$$\begin{aligned}
\int_{t_0}^{t_f} x(t) dt &= \sum_i \int_{t_i}^{t_{i+1}} x(t) dt \\
&= \sum_i \frac{1 - e^{-\beta_i(t_{i+1}-t_i)}}{\beta_i} x(t_i) + \frac{\sigma_i}{\beta_i} \int_{t_i}^{t_{i+1}} (1 - e^{-\beta_i(t_{i+1}-u)}) dW_u \\
&= \sum_i \frac{1 - e^{-\beta_i(t_{i+1}-t_i)}}{\beta_i} e^{-\int_{t_0}^{t_i} \beta} x(t_0) \\
&\quad + \sum_i \frac{1 - e^{-\beta_i(t_{i+1}-t_i)}}{\beta_i} \int_{t_0}^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u + \frac{\sigma_i}{\beta_i} \int_{t_i}^{t_{i+1}} (1 - e^{-\beta_i(t_{i+1}-u)}) dW_u \\
&=: M(t_0, t_f) x(t_0) + v(t_0, t_f) \sim \mathcal{N}(M(t_0, t_f), V(t_0, t_f))
\end{aligned}$$

où nous avons noté :  $V(t_0, t_f) := \text{Var}(\int_{t_0}^{t_f} x) = \text{Var}(v(t_0, t_f))$   
d'où

$$P(t, T) = E \left[ \exp \left\{ - \int_t^T h(u) + E(- \int_t^T x) - \frac{1}{2} \text{Var}(- \int_t^T x) \right\} \right]$$

Simplifions l'écriture de  $V(t_0, t_f)$

$$\begin{aligned}
v(t_0, t_f) &:= \sum_{i, t_0 \leq t_i \leq t_{i+1} \leq t_f} \frac{1 - e^{-\beta_i(t_{i+1}-t_i)}}{\beta_i} \int_{t_0}^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u + \frac{\sigma_i}{\beta_i} \int_{t_i}^{t_{i+1}} (1 - e^{-\beta_i(t_{i+1}-u)}) dW_u \\
&= \sum_i \frac{1 - e^{-\int_{t_i}^{t_{i+1}} \beta}}{\beta_i} \int_{t_0}^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u + \frac{\sigma_i}{\beta_i} \int_{t_i}^{t_{i+1}} 1 - e^{-\int_u^{t_{i+1}} \beta} dW_u \\
&= \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u + \sum_i \int_{t_0}^{t_i} \frac{\sigma_u}{\beta_i} e^{-\int_u^{t_i} \beta} dW_u - \int_{t_0}^{t_i} \frac{\sigma_u}{\beta_i} e^{-\int_u^{t_{i+1}} \beta} dW_u - \frac{\sigma_i}{\beta_i} \int_{t_i}^{t_{i+1}} e^{-\int_u^{t_{i+1}} \beta} dW_u \\
&= \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u + \sum_i \frac{\int_{t_0}^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u}{\beta_i} - \frac{\int_{t_0}^{t_{i+1}} \sigma_u e^{-\int_u^{t_{i+1}} \beta} dW_u}{\beta_i} \\
&= \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u + \sum_i \frac{K_i - K_{i+1}}{\beta_i} \quad \text{with } K_i = \int_{t_0}^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u \\
&= \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u + \sum_i \left( \frac{1}{\beta_i} - \frac{1}{\beta_{i+1}} \right) K_i \\
&= \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u + \sum_{i=1..n} c_i K_i \quad \text{avec } c_i = \frac{1}{\beta_i} - \frac{1}{\beta_{i-1}} \text{ and } \beta_n = \infty
\end{aligned}$$

Nous sommes intéressés par la variance de cette quantité là :

$$\begin{aligned}
\text{Var}(t_0, t_f) &:= \text{var}(v(t_0, t_f) | F_{t_0}) \\
&= \left\langle \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u, \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u \right\rangle + 2 \sum_i c_i \left\langle K_i, \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u \right\rangle + \sum_{i,j} c_i c_j \left\langle K_i, K_j \right\rangle \\
&= \omega + 2 \sum_i c_i \alpha_i + \sum_{i,j} c_i c_j \gamma_{ij} \\
&= \omega + 2 \sum_i c_i \alpha_i + 2 \sum_{i < j} c_i c_j e^{-\int_{t_i}^{t_j} \beta} \gamma_i + \sum_i c_i^2 \gamma_i \\
&= \omega + 2 \sum_i c_i \alpha_i + 2 \sum_{i=1..n} \gamma_i \frac{c_i}{I_i} \left( \sum_{j=i+1..n} I_j c_j \right) + \sum_i c_i^2 \gamma_i \\
&= \omega + 2 \sum_i c_i \alpha_i + 2 \sum_{i=1..n} \gamma_i \frac{c_i}{I_i} (S_n - S_i) + \sum_i c_i^2 \gamma_i
\end{aligned}$$

Où

$$\begin{aligned} I_i &:= e^{-\int_0^{t_i} \beta} \\ S_i &:= \sum_{j=0 \dots i} I_j c_j \\ \omega &:= \sum_i \left(\frac{\sigma_i}{\beta_i}\right)^2 (t_{i+1} - t_i) \end{aligned}$$

Et les suite  $\alpha_i$  et  $\gamma_i$  sont définies par récurrence :

$$\begin{aligned} \alpha_{i+1} &:= e^{-\beta_i(t_{i+1}-t_i)} \alpha_i + \left(\frac{\sigma_i}{\beta_i}\right)^2 (1 - e^{-\beta_i(t_{i+1}-t_i)}) \\ \gamma_{i+1} &:= e^{-2\beta_i(t_{i+1}-t_i)} \gamma_i + \frac{\sigma_i^2}{2\beta_i} (1 - e^{-2\beta_i(t_{i+1}-t_i)}) \end{aligned}$$

Dans la section suivante on détaille le calcul de  $\alpha$ ,  $\gamma$  et  $\omega$   
 Cette formule permet de calculer  $V$  en temps linéaire (ie  $O(t_f - t_0)$ )

## CALCULATIONS

Pour  $i < j$

$$\begin{aligned} \gamma_{i,j} &:= \langle K_i, K_j \rangle \\ &= \langle \int_t^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u, \int_t^{t_j} \sigma_u e^{-\int_u^{t_j} \beta} dW_u \rangle \\ &= e^{-\int_{t_i}^{t_j} \beta} \int_t^{t_i} (\sigma_u e^{-\int_u^{t_i} \beta})^2 du \\ &= e^{-\int_{t_i}^{t_j} \beta} \int_t^{t_i} \sigma_u^2 e^{-2\int_u^{t_i} \beta} du \\ &= e^{-\int_{t_i}^{t_j} \beta} \gamma_{i,i} \end{aligned}$$

$$\begin{aligned} \gamma_{i+1} &:= \gamma_{i+1,i+1} \\ &= \int_t^{t_{i+1}} \sigma_u^2 e^{-2\int_u^{t_{i+1}} \beta} du \\ &= e^{-2\beta_i(t_{i+1}-t_i)} \gamma_i + \int_{t_i}^{t_{i+1}} \sigma_i^2 e^{-2\beta_i(t_{i+1}-u)} du \\ &= e^{-2\beta_i(t_{i+1}-t_i)} \gamma_i + \frac{\sigma_i^2}{2\beta_i} (1 - e^{-2\beta_i(t_{i+1}-t_i)}) \end{aligned}$$

$$\begin{aligned} \alpha_i &:= \langle K_i, \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u \rangle \\ &= \langle \int_t^{t_i} \sigma_u e^{-\int_u^{t_i} \beta} dW_u, \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u \rangle \\ &= \int_t^{t_i} \frac{\sigma_u^2}{\beta_u} e^{-\int_u^{t_i} \beta} du \end{aligned}$$



$$\begin{aligned}
\alpha_{i+1} &= e^{-\beta_i(t_{i+1}-t_i)} \alpha_i + \int_{t_i}^{t_{i+1}} \frac{\sigma_u^2}{\beta_u} e^{-\beta_i(t_{i+1}-u)} du \\
&= e^{-\beta_i(t_{i+1}-t_i)} \alpha_i + \left(\frac{\sigma_i}{\beta_i}\right)^2 (1 - e^{-\beta_i(t_{i+1}-t_i)})
\end{aligned}$$

$$\begin{aligned}
\omega &:= \left\langle \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u, \int_{t_0}^{t_f} \frac{\sigma_u}{\beta_u} dW_u \right\rangle \\
&= \int_{t_0}^{t_f} \left(\frac{\sigma_u}{\beta_u}\right)^2 du \\
&= \sum_i \left(\frac{\sigma_i}{\beta_i}\right)^2 (t_{i+1} - t_i)
\end{aligned}$$

## LE MODÈLE À DEUX FACTEURS

Nous revenons au modèle original à deux facteurs. Les paramètres relatifs au facteur  $x$  (resp.  $y$ ) seront notés par un  $x$  (resp.  $y$ ) en exposant.

L'espérance étant linéaire, et la variance quadratique,  $M(t_0, t_f)$  est remplacée par  $M^x x + M^y y$ , et  $V(t_0, t_f)$  par  $V^{xx} + V^{yy} + 2V^{xy}$ , de sorte que :

$$\begin{aligned}
 P(t_0, T_f) &= \exp\left(-\int_{t_0}^{t_f} \Phi - M^x(t_0, t_f)x(t_0) - M^y(t_0, t_f)y(t_0) + \frac{V(t_0, t_f)}{2}\right) \\
 V^{xy} &:= \left\langle \int_{t_0}^{t_f} \frac{\sigma_u^x}{\beta_u^x} dW_u^x + \sum_{i=1..n} c_i^x K_i^x, \int_{t_0}^{t_f} \frac{\sigma_u^y}{\beta_u^y} dW_u^y + \sum_{i=1..n} c_i^y K_i^y \right\rangle \\
 &= \int_{t_0}^{t_f} \frac{\sigma_u^x \sigma_u^y}{\beta_u^x \beta_u^y} \rho_u du + \sum_{ij} c_i^x c_j^y \langle K_i^x, K_j^y \rangle + \sum_i c_i^x \langle K_i^x \int_{t_0}^{t_f} \frac{\sigma_u^y}{\beta_u^y} dW_u^y \rangle + c_i^y \langle K_i^y \int_{t_0}^{t_f} \frac{\sigma_u^x}{\beta_u^x} dW_u^x \rangle \\
 &= \omega^{x,y} + \sum_{ij} c_i^x c_j^y \gamma_{ij}^{xy} + \sum_i c_i^x \alpha_i^x + c_i^y \alpha_i^y
 \end{aligned}$$

avec comme pour le cas à un seul facteur :

$$\begin{aligned}
 \alpha_{i+1}^x &= e^{-\beta_i^x(t_{i+1}-t_i)} \alpha_i^x + \rho_i \frac{\sigma_i^x \sigma_i^y}{\beta_i^x \beta_i^y} (1 - e^{-\beta_i^x(t_{i+1}-t_i)}) \\
 \gamma_{i+1} &= e^{-(\beta_i^x + \beta_i^y)(t_{i+1}-t_i)} \gamma_i + \rho_i \frac{\sigma_i^x \sigma_i^y}{\beta_i^x + \beta_i^y} (1 - e^{-(\beta_i^x + \beta_i^y)(t_{i+1}-t_i)}) \\
 \omega &:= \sum_i \rho_i \frac{\sigma_i^x \sigma_i^y}{\beta_i^x \beta_i^y} (t_{i+1} - t_i)
 \end{aligned}$$

# BIBLIOGRAPHIE

- [1] Brigo D. and F. Mercurio. Interest rate models - theory and practice. 2006.
- [2] Edgar G. Johnson and Alfred O. Nier. Modèles stochastiques de taux d'intérêts. 2011.