



ENSIMAG

PEPS : Multimonde21

Présentation
Version Beta

IF - Groupe 9

Ibakuyumcu Arnaud
Lhuillery Paul
Rattanatray Ayutaya
Ruimy Benjamin
Voong Kwan

25/01/2017

Table des matières

I	Mathématiques	2
1	Modélisation du produit multi-économies	2
1.1	Notations	2
1.2	Modèle de Black&Scholes	2
1.2.1	Sous probabilité historique	2
1.2.2	Reconsidération des actifs en euros	3
1.3	Prix sous probabilité risque neutre	3
1.3.1	Dynamique sous Q^d	4
1.3.2	Portefeuille de couverture et prix	4
1.3.3	Calcul des deltas	4
1.4	Choix du modèle des taux sans risques	5
1.4.1	Courbe des taux plate	5
1.4.2	Modèle GAMTAUX	6
2	Calibration du modèle	7
2.1	Courbe des taux d'intérêt sans risque	7
2.2	Paramètres de la dynamique B&S	7
2.2.1	Matrice de volatilité	7
2.2.2	Drift	8
2.2.3	Trend	8
3	Méthode de Monte Carlo	8
3.0.1	Estimation de l'espérance	8
3.0.2	Minimisation de la variance	8
II	Informatiques	9
4	Librairie de Pricing	9
4.1	Simulation du modèle	9
4.2	Courbe des taux	9
4.3	Méthode de Monte Carlo	10
4.4	Parser et calculs statistiques	10
4.4.1	Construction du fichier CSV	10
4.4.2	Améliorations	10
4.4.3	Implémentation	11
4.4.4	ParseCSV	11
4.4.5	StatFactory	11
4.5	Parallélisation du programme	11
5	Interface utilisateur	11
III	Annexe	12
6	Test	12

Première partie

Mathématiques

1 Modélisation du produit multi-économies

Le produit `multimonde21` met en jeu des sous-jacents répartis sur plusieurs économies (et devises). Contrairement à la couverture en delta classique sur une seule économie, le portefeuille de réplication induit un achat de titres étrangers et par cette occasion une exposition au risque de change. La modélisation doit donc faire intervenir de nouveaux actifs risqués : les changes.

1.1 Notations

On pose H le pay off du produit multimonde. On posera pour la suite les indices $(S_k(t))_{k=1,\dots,6}$ du panier du produit et leur économie respective dans l'ordre suivant :

1. Eurostock50 (EUR)
2. Ftse (GBP)
3. P500 (USD)
4. Hangseng (HKD)
5. Nikkei (JPY)
6. SPASX200 (AUD)

On notera aussi $(r_k(t))_{k=1,\dots,6}$ les taux sans risque de chaque économie et $(X_k(t))_{k=1,\dots,6}$ les changes associés pour revenir à la monnaie domestique : l'euro.¹

Le pay off faisant intervenir seulement les performances de ces indices on peut noter $H = u(S_1, \dots, S_6)$.

1.2 Modèle de Black&Scholes

On utilisera la modélisation classique de Black& Scholes pour écrire la dynamique de prix des différents actifs.

1.2.1 Sous probabilité historique

On modélise le marché sous l'espace de probabilité historique filtré $(\Omega, (\mathcal{F}_t)_t, \mathbb{P})$ de la façon suivante :

★ Actif risqué domestique :

$$dS_1(t) = S_1(t) (\mu^1(t)dt + \hat{\sigma}^1(t).dW(t))$$

★ Taux de change :

$$(\forall k \in \{2, \dots, 6\}) dX_k(t) = X_k(t) (\mu^{X_k}(t)dt + \hat{\sigma}^{X_k}(t).dW(t))$$

★ Actifs risqués étrangers :

$$(\forall k \in \{2, \dots, 6\}) dS_k(t) = S_k(t) (\mu^k(t)dt + \hat{\sigma}^k(t).dW(t))$$

1. Bien évidemment $X_1(t) = 1$ et ne sera pas utilisé puisque S_1 est un actif domestique

★ Actifs non risqués :

$$(\forall k \in \{1, \dots, 6\}) dR_k(t) = R_k(t)r^k(t)dt$$

Où $\hat{\sigma}^k$ la k -ieme ligne et $\hat{\sigma}^{X_k}$ la $(k+5)$ -ieme ligne de σ et $W(t)$ est un MB à 11 dimensions.

Par la suite on supposera l'ensemble des tendances μ et la matrice de volatilité σ déterministes et constants au cours du temps.

1.2.2 Reconsidération des actifs en euros

On se place pour la suite dans l'économie domestique, pour cela il faut que chacun des actifs considérés soit exprimé en euro : on use donc des taux de changes pour convertir les actifs risqués.

La dynamique sous probabilité historique devient alors la suivante après application de la formule d'Itô :²

$$★ dS_1(t) = S_1(t) (\mu^1 dt + \hat{\sigma}^1 . dW(t))$$

$$★ dR_1(t) = R_1(t)r^1(t)dt$$

$$★ (\forall k \in \{2, \dots, 6\}) d(X_k S_k)(t) = X_k(t)S_k(t) [(\mu^k + \mu^{X_k} + \hat{\sigma}^k . \hat{\sigma}^{X_k}) dt + (\hat{\sigma}^k + \hat{\sigma}^{X_k}) . dW(t)]$$

$$★ (\forall k \in \{2, \dots, 6\}) d(X_k R_k)(t) = X_k(t)R_k(t) [(\mu^{X_k} + r^k(t)) dt + \hat{\sigma}^{X_k}(t) . dW(t)]$$

On pose alors le vecteur des trends et la matrice des volatilités dans l'univers euro :

$$\tilde{\mu}(t) = \begin{bmatrix} \mu_1 \\ (\mu_2 + \mu^{X_2} + \hat{\sigma}^2 . \hat{\sigma}^{X_2}) \\ \vdots \\ (\mu_6 + \mu^{X_6} + \hat{\sigma}^6 . \hat{\sigma}^{X_6}) \\ (\mu^{X_2} + r^2(t)) \\ \vdots \\ (\mu^{X_6} + r^6(t)) \end{bmatrix} \text{ et } \tilde{\sigma} = \begin{bmatrix} (\hat{\sigma}^1)^T \\ (\hat{\sigma}^2 + \hat{\sigma}^{X_2})^T \\ \vdots \\ (\hat{\sigma}^6 + \hat{\sigma}^{X_6})^T \\ (\hat{\sigma}^{X_2})^T \\ \vdots \\ (\hat{\sigma}^{X_6})^T \end{bmatrix}$$

1.3 Prix sous probabilité risque neutre

On suppose que la matrice de volatilité $\tilde{\sigma}$ est inversible et que $\tilde{\sigma} . \tilde{\sigma}^T$ est minoré au sens des matrices symétriques.

Sous ces hyposthèses, le marché est complet et il existe une probabilité risque neutre domestique Q^d sous laquelle les actifs domestiques actualisés au taux sans risque domestique sont des martingales.

La prime de risque associé est $\lambda(t) = \tilde{\sigma}^{-1} . (r^1(t) . \mathbf{1}_{11} - \tilde{\mu}(t))$.

2. L'ensemble des actifs sont dans l'économie domestique désormais

1.3.1 Dynamique sous Q^d

On note $W^d(t) = W(t) + \int_0^t \lambda(t)dt$ le mouvement brownien associé au nouvel espace risque neutre.

Les dynamiques des actifs sous cette probabilités sont alors :

$$\star dS_1(t) = S_1(t) (r^1(t)dt + \hat{\sigma}^1.dW^d(t))$$

$$\star dR_1(t) = R_1(t)r^1(t)dt$$

$$\star (\forall k \in \{2, \dots, 6\}) d(X_k S_k)(t) = X_k(t)S_k(t) [r^1(t)dt + (\hat{\sigma}^k + \hat{\sigma}^{X_k}).dW^d(t)]$$

$$\star (\forall k \in \{2, \dots, 6\}) d(X_k R_k)(t) = X_k(t)R_k(t) [r^1(t)dt + \hat{\sigma}^{X_k}.dW^d(t)]$$

1.3.2 Portefeuille de couverture et prix

Le portefeuille de couverture V est alors constitué des actifs suivants en t :

- $\Delta_1(t)$ actifs S_1
- $\Delta_k(t)$ actifs $X_k S_k$ pour $k \in \{2, \dots, 6\}$
- $\Delta_{k+6}(t)$ actifs $X_k R_k$ pour $k \in \{1, \dots, 6\}$

En supposant ce portefeuille de couverture autofinancé et admissible on a alors que le prix du portefeuille actualisé au taux sans risque domestique \tilde{V} est une martingale sous Q^d .

Dès lors on a l'égalité suivante pour $t < T$: $\tilde{V}(t) = E^{Q^d} [\tilde{V}(T) | \mathcal{F}_t]$.

Si on impose $V(T) = H$ alors par absence d'opportunité d'arbitrage, le prix du produit est le prix du portefeuille de couverture $V(t)$ et on a l'égalité suivante :

$$Prix(t) = V(t) = E^{Q^d} \left[\frac{R_1(t)}{R_1(T)} H \mid \mathcal{F}_t \right]$$

1.3.3 Calcul des deltas

Pour intégrer les sous-jacents du portefeuille de couverture on peut utiliser une nouvelle fonction $g(s, x, y)$ tel que :

$$g(S_1, (X_2 S_2, \dots, X_6 S_6), (X_2 R_2, \dots, X_6 R_6)) = u \left(S_1, \left(\frac{X_2 S_2}{X_2 R_2} * R_2 \right), \dots, \left(\frac{X_6 S_6}{X_6 R_6} * R_6 \right) \right) = H$$

Dans ce cas, le prix en t vaut $prix(t) = \phi(S_1(t), ((X_2 S_2)(t), \dots, (X_6 S_6)(t)), ((X_2 R_2)(t), \dots, (X_6 R_6)(t)))$ et on obtient les égalités suivantes :

- $\Delta_1(t) = \frac{\partial \phi}{\partial s}(t)$
- $(\forall k \in \{2, \dots, 6\}) \Delta_k(t) = \frac{\partial \phi}{\partial x_{k-1}}(t)$
- $(\forall k \in \{2, \dots, 6\}) \Delta_{k+6}(t) = \frac{\partial \phi}{\partial y_{k-1}}(t)$
- $\Delta_7(t) = \frac{1}{R_1(t)} \left(V(t) - \left(\Delta_1 S_1(t) + \sum_{k=2}^6 \Delta_k(t) (X_k S_k)(t) + \Delta_{k+6}(t) (X_k R_k)(t) \right) \right)$

On montre qu'avec cette stratégie, le portefeuille est autofinancé, admissible et réplique le pay off H . C'est donc un portefeuille de couverture du produit **multimonde21**.

1.4 Choix du modèle des taux sans risques

Dans notre modèle, on utilisera en premier lieu des modèles de courbe de taux déterministes. On rappelle que le prix $B(t, T)$ en t d'un ZC d'échéance T s'écrit :

$$B(t, T) = e^{-\int_t^T r_s ds} \text{ ou } B(t, T) = e^{-R(t, T) \cdot (T-t)}$$

Où r_t est le taux instantané et $R(t, T)$ le taux moyen entre t et T .

En univers déterministe, on a par absence d'opportunité d'arbitrage la relation suivante :

$$(\forall t < T) \quad B(0, T) = B(0, t)B(t, T)$$

On en déduit en particulier la relation suivante :

$$(\forall t < T) \quad R(t, T) = \frac{R(0, T)T - R(0, t)t}{T-t}$$

Ainsi, la plupart des modèles de courbes donnent une formule pour $R(0, t)$.

1.4.1 Courbe des taux plate

Dans ce modèle, on suppose la courbe des taux constante, c'est à dire que :

$$(\forall t < T) \text{ on a } r_t = r \text{ et } R(t, T) = r$$

C'est le modèle le plus simple et le seul paramètre à estimer est r .

Dans la version Beta c'est ce modèle qui sera utilisé pour chacune des devises.

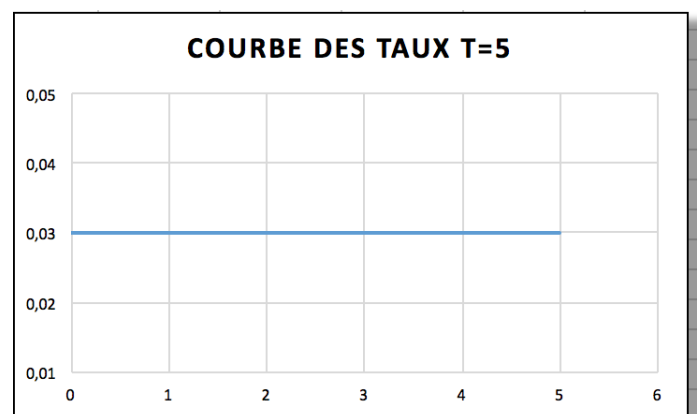


Figure 1 : Courbe des taux plate avec $r = 0.03$

1.4.2 Modèle GAMTAUX

Le modèle de GAMTAUX a été établie par Nicole El Karoui et le modèle de taux moyen s'écrit de la manière suivante :

$$R(0, t) = r_{inf} - S \cdot G_1(t) + \gamma \cdot G_2(t)$$

$$\text{avec } G_1(t) = \frac{(1 - e^{-at})}{at} \text{ et } G_2(t) = \frac{(1 - e^{-at})^2}{4at}$$

Où r_{inf} est le taux court, S le spread avec le taux long, γ un paramètre de courbure.

Le paramètre a a été fixé à 0.4 après de nombreux tests statistiques, les autres paramètres doivent être estimés.

Ci-dessous sont présentés les courbes $R(0, T)$ sous le modèle de GAMTAUX en fonction des différents paramètres.

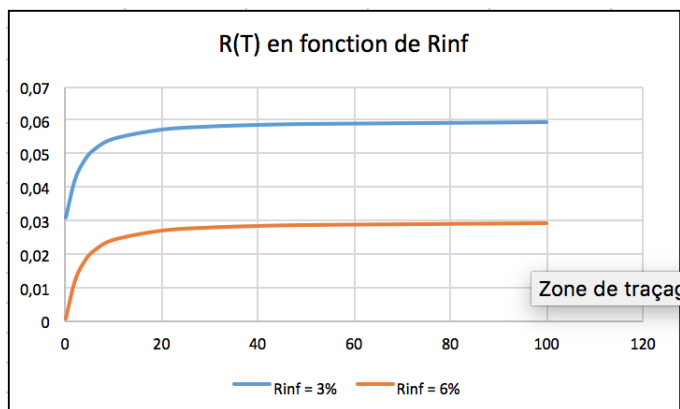


Figure 2 : Courbe en fonction de r_{inf}

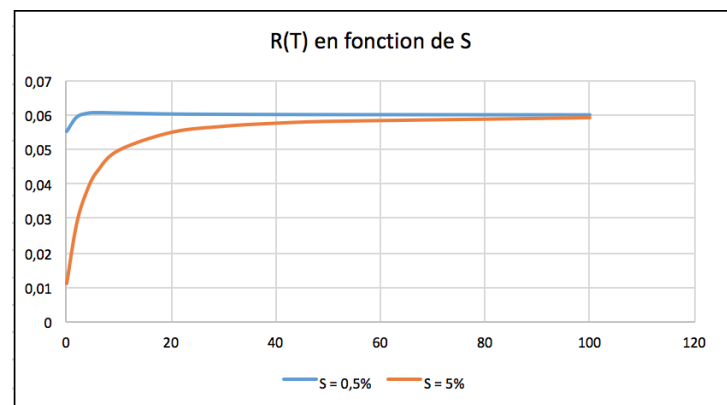


Figure 3 : Courbe en fonction de S

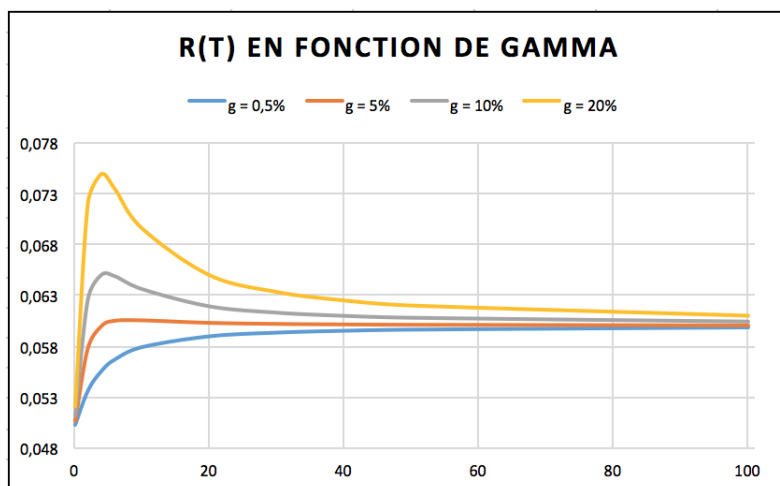


Figure 4 : Courbe en fonction de γ

2 Calibration du modèle

On part du modèle précédent avec paramètres constants au cours du temps.

2.1 Courbe des taux d'intérêt sans risque

Les paramètres de la courbe des taux seront estimés grâce au marché d'obligations de chaque économie.

On rappelle que le prix théorique d'une obligation de maturité T est (avec taux continu) :

$$Ob_{theo} = \sum_{i=1}^N F_{t_i} e^{-R(0,t_i)t_i} \text{ où } F_{t_i} \text{ correspond au flux versé en } t_i$$

Ainsi, le choix des paramètres se fait de manière à minimiser l'écart entre monde théorique et les prix marchés :

$$\text{Min} \left(\sum_{i=1}^M (Ob_{marche}^i - Ob_{theo}^i)^2 \right)$$

Par manque de temps, le paramètre de la courbe de taux plate de la version Beta sera fixé de notre propre gré à 0.03

2.2 Paramètres de la dynamique B&S

On rappelle que tout actif de dynamique $\frac{dS_i(t)}{S_i(t)} = \mu^i dt + \hat{\sigma}^i . dW(t)$, où W est un MB à n dimension sous probabilité historique, s'intègre en :

$$S_i(t+1) = S_i(t) \exp \left(\mu^i - \frac{(\sigma^i)^2}{2} + \hat{\sigma}^i . \varepsilon(t) \right)$$

Où $\varepsilon(t) \stackrel{iid}{\sim} \mathcal{N}(0, \mathbf{1}_n)$ et $(\sigma^i)^2 = \sum_{k=1}^n (\hat{\sigma}_k^i)^2$

Maintenant, en prenant $\mathcal{R}_i(t) = \log \left(\frac{S_i(t+1)}{S_i(t)} \right)$ les log-rendements de l'actif entre t et $t+1$ on obtient

$$\mathcal{R}_i(t) = \text{drift}^i + \hat{\sigma}^i . \varepsilon(t) \text{ où } \text{drift}^i = \mu^i - \frac{(\sigma^i)^2}{2}$$

2.2.1 Matrice de volatilité

En utilisant la covariance on obtient :

$$\text{CoVar}[\mathcal{R}_i; \mathcal{R}_j] = (\sigma . \sigma^T)_{ij}$$

Ainsi, en utilisant la matrice de variance-covariance empirique (débiaisé) sur un échantillon de m log-rentabilités des actifs on obtient A_m un estimateur sans biais de $\sigma . \sigma^T$.

En prenant la matrice de cholesky³ de A_m on obtient $\text{cholesky}(A_m) = \bar{\sigma}_m$ un estimateur sans biais de σ la matrice de vol.

3. On aurait pu prendre la matrice racine aussi, mais la matrice de cholesky a la particularité d'être triangulaire rendant les calculs plus rapide par la suite

2.2.2 Drift

En utilisant l'espérance on obtient :

$$E[\mathcal{R}_i] = drift^i$$

Ainsi, en utilisant la moyenne empirique sur un échantillon de m log-rentabilités de l'actif i on obtient M_m^i un estimateur sans biais à variance minimale du $drift^i$.

2.2.3 Trend

Connaissant le drift de l'actif i et la matrice de volatilité, sur un échantillon de m log-rentabilités, on obtient l'estimateur $\bar{\mu}_m^i = M_m^i - \frac{\bar{\sigma}_m}{2}$ sans biais du trend μ^i .

3 Méthode de Monte Carlo

3.0.1 Estimation de l'espérance

Le but est d'estimer le prix d'un produit financier. Ce prix en t s'écrit toujours sous la forme

$$E^* \left[e^{-\int_t^T r_s ds} H \mid \mathcal{F}_t \right] \text{ avec } H \text{ le pay-Off}$$

Où r_s est le taux sans risques instantané et E^* l'espérance sous probabilité risque neutre.

Par la loi forte des grands nombres, on obtient pour $(X_i)_i$ suite de VA iid :

$$\frac{1}{n} \sum_{i=1}^n X_i \rightarrow E[X_1] \text{ dans } \mathcal{L}^1$$

Ainsi, une estimation du prix classique par méthode de Monte Carlo est de simuler n trajectoire $P_i = \left(e^{-\int_t^T r_s ds} H \right)$ et d'en faire la moyenne.

De plus, par les théorème central-limite on obtient :

$$\frac{\sqrt{n}}{\sigma_n} \left(\frac{1}{n} \sum_{i=1}^n X_i - E[X_1] \right) \xrightarrow{\mathcal{L}} \mathcal{N}(0, 1)$$

avec $\sigma_n^2 = \frac{1}{n} \sum_{i=1}^n X_i^2 - \left(\frac{1}{n} \sum_{i=1}^n X_i \right)^2$.

On peut donc construire un intervalle de confiance à $\alpha\%$ pour le prix :

$$I_{C_n} = \left[\frac{1}{n} \sum_{i=1}^n X_i \pm u_\alpha * \frac{\sigma_n}{\sqrt{n}} \right] \text{ où } u_\alpha \text{ est l'}\alpha \text{ quantile de } \mathcal{N}(0, 1)$$

Ceci nous permet donc d'avoir un intervalle de confiance sur le prix affiché.

C'est cette méthode qui sera utilisé pour la version Beta du pricer `multimonde21`.

3.0.2 Minimisation de la variance

Bien évidemment le but est de minimiser la largeur de cet intervalle de confiance. Pour cela il existe plusieurs manières, la première étant d'augmenter le nombre n de simulations mais cela augmente le temps de calcul du prix.

Une autre manière est d'utiliser des méthodes de réduction de variance comme user de variable antithétique, de variable de contrôle, ou bien d'une fonction d'importance.

Ces méthodes seront présentées et implémentées dans la prochaine version du pricer.

Deuxième partie

Informatiques

4 Librairie de Pricing

Nous avons décidé de faire une librairie de pricing la plus générique possible de façon à pouvoir l'améliorer sans devoir changer tout le code.

Il est possible de pricer tous les produits désirés dès lors qu'une fonction de `payOff` est implémentée (dans le fichier `PayOffFunction.cpp`) et qu'une classe dérivant de la classe `ProductGen` est créée. Le lecteur pourra apprécier le diagramme de classe de la librairie dans le fichier `classDiagramPeps.jpg`.

4.1 Simulation du modèle

Le but est de simuler une trajectoire des sous-jacents selon le modèle de Black & Scholes à actifs corrélés.

Soit d actifs $((S_t^i)_{i=1,\dots,d})$ de matrice de volatilité σ et de tendance μ déterministes constantes. Soit $(W_t)_t$ un MB à d dimensions et $0 = t_0 < t_1 < \dots < t_n = T$ une discrétisation jusqu'à maturité. Le modèle de Black & Scholes nous donne alors la dynamique des sous-jacents :

$$dS_t^i = S_t^i (\mu^i dt + \hat{\sigma}^i \cdot dW_t)$$

Posons maintenant la matrice de corrélation $\Gamma = (\rho_{i,j})_{i,j \in \{1,\dots,d\}}$ des sous-jacents, on a :

$$(\forall i, j \in \{1, \dots, d\}) \rho_{i,j} = \frac{(\sigma \cdot \sigma^T)_{i,j}}{\sigma^i * \sigma^j} \text{ où } \sigma^i = \sqrt{(\sigma^i)^2}$$

On montre alors que le modèle se réécrit :

$$dS_t^i = S_t^i (\mu^i dt + \sigma^i * dB_t^i)$$

Où (B^1, \dots, B^d) sont d MB unidimensionnels et vérifiant :

$$\text{CoVar} [B_t^i, B_t^j] = \langle B^i, B^j \rangle_t = \rho_{i,j} t$$

Posons désormais $L = \text{cholesky}(\Gamma)$ on montre que $B \stackrel{Loi}{=} LW$

Le modèle s'écrit finalement sous forme intégrée et en probabilité risque neutre :

$$S_{t_{k+1}}^i = S_{t_k} \cdot \exp \left(\int_{t_k}^{t_{k+1}} r_s ds - \frac{(\sigma^i)^2}{2} (t_{k+1} - t_k) + \sigma^i \sqrt{(t_{k+1} - t_k)} L_i G_{k+1} \right)$$

Où L_i est la i -ième ligne de L et $(G_k)_{k \geq 1}$ suite iid de vecteur gaussien $\sim \mathcal{N}(0, \mathbf{1}_d)$

C'est cette forme qui sera implémentée dans la classe `BlackScholesModel` héritant de `ModelGen`.

4.2 Courbe des taux

Lors de la simulation des sous-jacents sous probabilité risque neutre, intervient le calcul de $\int_{t_k}^{t_{k+1}} r_s ds$.

Par soucis de rendre une librairie de calcul générique et donc ré-utilisable, nous avons créé une classe générique `RateModelGen` représentant un modèle de taux et ses classes héritières doivent implémenter la méthode `GetIntegrale(t, T)` qui permet le calcul de l'intégrale plus haut.

Cette classe fait partie des membres de chaque objet `ModelGen` et en particulier dans `BlackScholesModel`. Ainsi, il est facile de changer de modèle de courbe de taux sans changer l'ensemble du modèle.

4.3 Méthode de Monte Carlo

Cette partie sera complétée lors de l'implémentation d'une méthode à minimisation de variance. En effet la méthode de base n'a rien de compliquer à implémenter.

4.4 Parser et calculs statistiques

Les calculs statistiques sont effectués à partir d'une classe statistique nommée StatsFactory. Les paramètres sont estimés à partir des données historiques se trouvant dans un fichier csv. Un parser de csv a été implémenté dans une classe ParseCSV. Cette section permet de comprendre à la fois l'implémentation, la façon dont le fichier csv a été construit mais aussi les améliorations que nous comptons réaliser pour la version suivante.

4.4.1 Construction du fichier CSV

Le fichier CSV a été construit de façon rigoureuse. Il contient l'ensemble des données nécessaires afin d'obtenir l'estimation des paramètres.

La première ligne définit le nom des colonnes. Elle est ignorée lors de la lecture du fichier de façon automatique.

Les colonnes ont un ordre très précis. La première colonne permet d'indiquer la date de la ligne courante. Pour le moment, elle est ignorée et sert seulement à donner du sens au fichier lorsqu'il est ouvert indépendamment du parser.

Les autres colonnes permettent d'obtenir à chaque date (une date par WorkingDay), les valeurs des indices en devise étrangère et les taux de changes devise étrangère / EUR.

Ces données ont pour le moment été récupérées manuellement.

L'ordre de ces colonnes est le suivant :

- La date
- Le cours de l'EuroStock 50
- Le cours du FTSE 100
- Le cours du S&P 500
- Le cours du Nikkei
- Le cours du S&P ASX 200
- Le taux GBP/EUR
- Le taux USD/EUR
- Le taux HKD/EUR
- Le taux JPY/EUR
- Le taux AUS/EUR

Pour le moment, ce fichier contient les dates du 12/01/2017 au 13/06/2016.

4.4.2 Améliorations

L'amélioration qui sera implémentée incessamment sous peu consistera à la récupération automatique des données manquantes à partir du jour courant.

Il faudra être très vigilant quant à l'absence de donnée de sorte à ce que lorsqu'une donnée est manquante, la donnée du jour précédent est utilisée.

4.4.3 Implémentation

4.4.4 ParseCSV

La classe parseCSV possède un constructeur prenant en entrée un chemin vers un fichier csv. Pour le moment, l'ensemble des données du fichier csv sont récupérées et permettent de remplir l'attribut inputDate de type PnlMat*.

Amélioration : Une amélioration qui sera très bientôt implémentée sera de faire en sorte que seulement les 6 derniers mois soient parsés à partir du jour demander par l'utilisateur.

Il conviendra de lire la colonne de date et de récupérer l'indice à partir duquel nous souhaitons récupérer les données.

4.4.5 StatFactory

La classe StatFactory permet, à partir d'une matrice contenant les différentes quotations journalières, de déduire les choses suivantes :

- La matrice de logRendement
- Le vecteur de moyenne des logRendement
- La covariation de ces logRendement
- Le vecteur de volatilité de ces logRendement
- La matrice de corrélation de ces logRendement

Ces matrices ne sont pas toutes utilisées pour l'estimation des paramètres, ils sont retravaillés pour le produit multimonde, néanmoins, nous souhaitons faire une classe assez générale.

En effet, selon le modèle, les modifications sur ces données ne sont pas forcément les mêmes.

4.5 Parallélisation du programme

Une autre méthode pour augmenter la précision du pricer est d'augmenter le nombre de simulation en parallélisant la somme et donc les simulations. Cette méthode permet donc de garder le même temps de calcul⁴ tout en réduisant la largeur de l'intervalle de confiance du prix.

Le programme sera parallélisé dans les versions ultérieures

5 Interface utilisateur

Le lecteur est prié d'aller voir le pdf `Manuel_Interface.pdf`

4. Si on multiplie le nombre de tirage par m le nombre de coeurs de la machine

Troisième partie

Annexe

6 Test

Nous avons effectués plusieurs tests pour valider notre programme.

Test du pay off

`test_payOffMultimonde.te` : Permet de tester le pay off du multimonde en créant une trajectoire où le payoff est connu.

Test de la simulation

`test_blackScholes.te` : Permet de faire une simulation avec le modèle de Black&Scholes. On remarque d'ailleurs dans ce test la tendance du Nikkei a chuter vers 0. Ceci est dû à la période de calibration.

Test du pricing

`test_pricerCall-MCBS.te` : Grâce à la genericité de notre librairie, on a pu créer un produit Call facilement et comme nous connaissons le prix exact d'un Call (formule fermée) on a pu valider notre pricer en $t = 0$ ou $t > 0$

Test des deltas

`test_DeltaCall.te` On utilise les formules fermées comme précédemment

Test du multimonde

`test_pricer_multimonde.te` : Permet de pricer le multimonde en $t = 0$. On obtient un prix de 90.57 euros environ.

`test_HedgingMultimonde.te` : Permet de tester la couverture du produit sur un marché simulé (le test du modèle utilisé sera fait avec un back test mis en place pour la prochaine version).



Figure 5 : PnL du produit