M2 IF Apprentissage projet composant

Spécifications composant 2

Boucle de Monte Carlo

|  |
| --- |
| **Groupe 3** |
| Lauren BARTHELEMY |
| Maxime BETTY |
| Long DO |
| Thardsajini SATKUNARAJAH |

|  |  |  |  |
| --- | --- | --- | --- |
| Version doc | Date | Auteur(s) | Modifications |
| 1.0 | 27/01/2015 | Jose Luu | Version initiale |
| 1.1 | 27/01/2015 | Jose Luu | Modification pour exemple |
| 1.2 | 23/02/2015 | Groupe 3 | Création des spécifications du composant « Monte Carlo » |
| 1.3 | 07/04/2015 | Groupe 3 | Mise à jour des fonctions et erreurs |
| 1.4 | 12/04/2015 | Groupe 3 | Mise à jour des erreurs |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Ce document a pour but de rassembler les spécifications concernant le composant 2 « Boucle Monte Carlo » du projet de classe.

1. Fonctionnement du composant

Les fonctions du module Monte-Carlo sont la gestion des itérations et du séquençage. L’objectif du composant est de calculer l’espérance du PayOff à partir de la valeur finale des N itérations. Ce composant est en interactivité avec d’autres composants. Voici un schéma explicatif du mode de fonctionnement du composant demandé :

 Composant 1 : XLL Interface

Composant 6 : Constructeur de chemin

Composant 3 : GNA Gaussien

Composant 7 : Echantillonnage

1

2

3

4

5

6

Composant 2 : Monte Carlo

*Calcul de l’espérance du Payoff*

7

1 – Réception de l’appel doMonteCarlo() du composant 1

2 – Appel getPath() du composant 6

3 – Retour de getPath() : vecteurs de valeurs du composant 6

4 – Appel de pricePath() du composant 7

5 – Retour de pricePath() du composant 7

6 – Le composant 2 calcul l’espérance du Payoff

7 – Retour de doMonteCarlo() au composant 1

Dans la suite du document, chacune des étapes du schéma sont explicitées afin de mieux comprendre le fonctionnement du composant et son mode de réalisation.

1. Les étapes
   1. Réception du composant 1

Notre module reçoit du composant “XLL interface” un certain nombre de paramètres :

* le nombre d’itérations
* la valeur du Strike
* le numéro du PayOff correspondant au type du PayOff (Américain, Européen …)
* La maturité fixée à 2 ans avec une période journalière soit 252 jours par an et 504 jours au total

double doMonteCarlo(string typePayoff, double maturity, double strike, int nbTrials) ;

* 1. Récupération des n valeurs

Pour récupérer n valeurs de PayOff, nous allons itérer n fois le processus suivant :

***2.2.1 Appel du composant 6***

Notre module appelle le constructeur du chemin (Composant 6) pour obtenir le vecteur des 504 valeurs aléatoires gaussiennes normalisées. En amont, C6 a reçu ces informations du composant 3 « *GNA gaussien* ».

Double\* getPath() ;

***2.2.2 Appel du composant 7***

Notre module envoie le numéro du PayOff, le vecteur des valeurs, le strike et la maturité au composant 7 « *Echantillonnage* ». En retour, il nous fournit la valeur du PayOff. Nous la stockons.

double pricePath(string typePayoff, double path[], double strike, double maturity) ;

* 1. Calcul de l’espérance

A partir des N PayOff reçus, nous calculons l’espérance du PayOff dans la méthode doMonteCarlo().

* 1. Envoi au module 1

Le résultat est renvoyé au module 1 « *XLL Interface* ».

Retour de la fonction doMonteCarlo(string typePayoff, double maturity, double strike, int nbTrials) : double

1. Description des erreurs

Pour le composant, chacune des erreurs est à gérer avec un code erreur définit comme suit (des modifications peuvent être apportées plus tard) :

* 1. Etape 2.1

|  |  |
| --- | --- |
| Nom Fonction : *doMonteCarlo (string typePayoff, double maturity, double strike,  int nbTrials)* | |
| *Erreur :* données négatives | *Negative value* |
| *Erreur :* données manquantes | *Missed data* |

* 1. Etape 2.2

***4.3.1 Etape 2.2.1***

|  |  |
| --- | --- |
| Nom Fonction : *getPath()* | |
| *Erreur :* données négatives | *Negative value* |
| *Erreur :* données manquantes | *Missed data* |

*4.3.2 Etape 2.2.2*

|  |  |
| --- | --- |
| Nom Fonction : *pricePath(string typePayoff, double[] path, double strike,  double maturity)* | |
| *Erreur :* données négatives | *Negative value* |
| *Erreur :* données manquantes | *Missed data* |
| *Erreur :* valeur supérieure à 1 000 000 | *Value greater than 1 000 000* |