Spécifications composant 2
Boucle de Monte Carlo

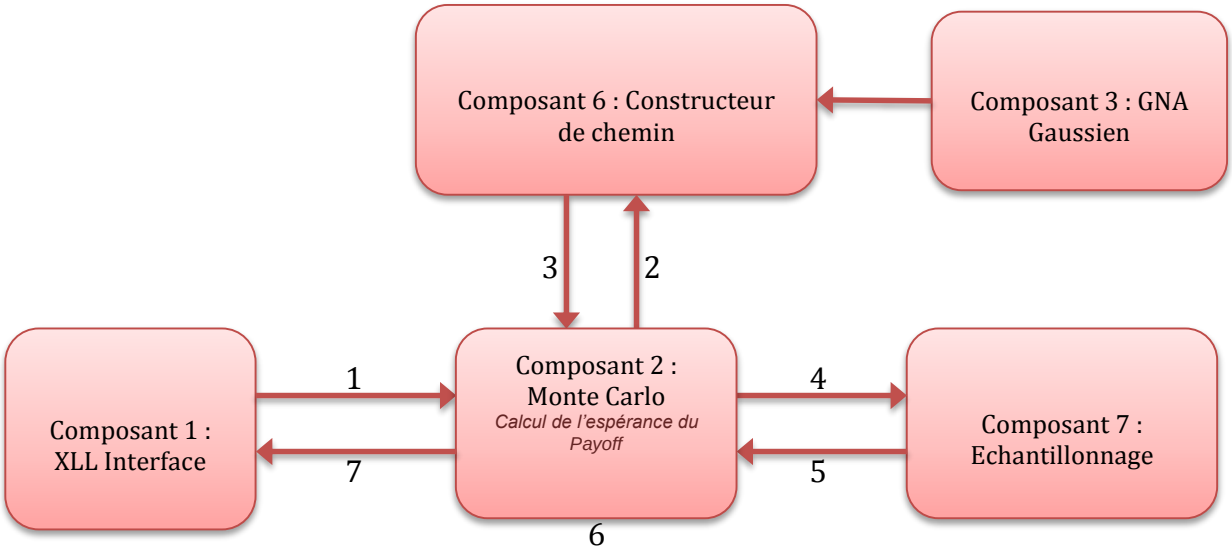
Groupe 3
Lauren BARTHELEMY
Maxime BETTY
Long DO
Thardsajini SATKUNARAJAH

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 »

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 :



1 – Réception des données du composant 1
2 – Appel du composant 6
3 – Le composant 6 retourne une liste de valeurs au composant 2
4 – Appel du composant 7
5 – Le composant 7 retourne une valeur
6 – Le composant 2 calcul l'espérance du Payoff
7 – Le composant 2 retourne l'espérance du Payoff 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.

2 Les étapes

2.1 Réception du composant 1

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

2.2 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* ».

2.2.2 Appel du composant 7

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

2.3 Calcul de l’espérance

A partir des N PayOff reçus, nous calculons l’espérance du PayOff.

2.4 Envoi au module 1

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

3 Description des interfaces

Pour le composant, les fonctions doivent être définies comme suit :

3.1 Etape 2.1

L'implémentation de cette méthode peut se faire de deux manières :

Nom Fonction	Type de retour
GetInfos ()	***double [4]***

Ou bien,

Nom Fonction	Type de retour
GetMaturité ()	***int***
GetIterations ()	***long***
GetStrike ()	***double***
GetTypePayOff ()	***int***

3.2 Etape 2.2

3.3.1 Etape 2.2.1

Nom Fonction	Type de retour
GetVecteur()	***double [504] (élément négociable)***

3.3.2 Etape 2.2.2

Nom Fonction	Type de retour
GetPayOff(int TypePayOff, double[504] Vecteur)	***double***

3.3 Etape 2.3

Nom Fonction	Type de retour
EsperancePayOff (double[n] VecteurPayOff)	***void***

3.4 Etape 2.4

Nom Fonction	Type de retour
SendRes(double Res)	***bool***

4 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) :

4.1 Etape 2.1

Implémentation 1 :

Nom Fonction : *GetInfos ()*	
Erreur : tableau incomplet	***-1***
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

Ou bien,

Implémentation 2 :

Nom Fonction : *GetMaturité ()*	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

Nom Fonction : *GetIterations ()*	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

Nom Fonction : *GetStrike ()*	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

Nom Fonction : *GetTypePayOff ()*	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

4.2 Etape 2.2

4.3.1 Etape 2.2.1

Nom Fonction : *GetVecteur()*	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

4.3.2 Etape 2.2.2

Nom Fonction : *GetPayOff(int TypePayOff, double[504] Vecteur)*	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***

Erreur : données manquantes	***-4***
Erreur : valeur supérieure à 1 000 000	***-5***

4.3 Etape 2.3

Nom Fonction : ***EsperancePayOff (double[n] VecteurPayOff)***	
Erreur : mauvais type de données	***-2***
Erreur : données négatives	***-3***
Erreur : données manquantes	***-4***

4.4 Etape 2.4

Nom Fonction : ***SendRes(double Res)***	
Si valeur de retour = false	***Message d'erreur indiquant que l'envoi du résultat au module 1 a échoué***