

# Spécifications du composant 6

## Constructeur de chemin

**Groupe 4**  
Magali BIT  
Laurent MARY  
Rémy MILIA

Version doc	Date	Auteur(s)	Modifications
1.0	21/02/2015	Magali BIT Laurent MARY Rémy MILIA	Version initiale
1.1	14/03/2015	Laurent MARY	Précisions dans l'enchaînement des appels
1.2	04/04/2015	Magali BIT Laurent MARY Rémy MILIA	Signatures des fonctions Valeurs de retours attendues (fonctionnement nominal et cas d'erreurs)
1.3	11/04/2015	Laurent MARY	Petite precision dans l'appel vers le GNA Gaussien Ajout de headers de fonctions à des fins de test
1.4	14/04/2015	Laurent MARY	Ajustement pour l'appel au composant 3

# 1. Contexte

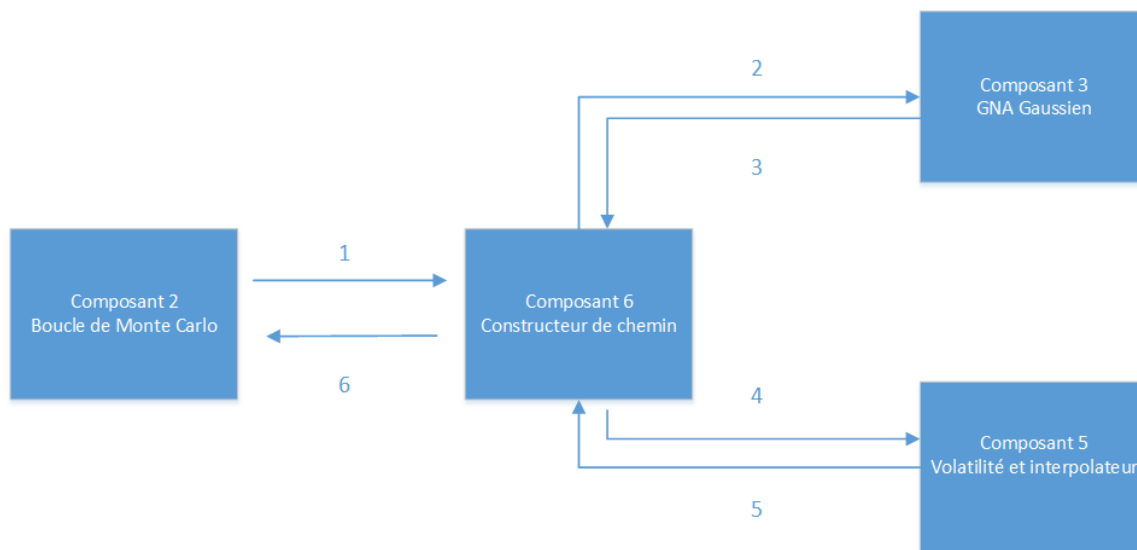
L'objectif de cette spécification est de décrire les fonctionnalités et le comportement attendu d'un composant n°6 « Constructeur de chemin » qui sera compilé sous forme d'une DLL et qui sera utilisé dans le cadre d'une application permettant le pricing d'options (européennes, américaines ou bermudéennes).

## 2. Objectif du composant

Le composant « Constructeur de chemin » devra être capable à partir du spot de départ  $S_0$  et de la maturité  $T$  de fournir un vecteur de taille  $size$  (vecteur allant de 0 à  $T-1$ ) correspondant à la valeur estimée de l'actif sous-jacent pour chaque jour de cotation.

## 3. Fonctionnement

### a. Enchaînement des appels



1. Le composant « Constructeur de chemin » est appelé par le composant « Boucle de Monte Carlo », ce dernier lui fournit en paramètre la valeur du spot de départ  $S_0$  et la maturité  $T$  attendue (par défaut 504 jours = 252 jours annuels de cotation \* 2 ans)

2. Le « Constructeur de chemin » commence par initialiser le GNA Gaussien en appelant la fonction `initialize()` et appelle le « GNA Gaussien » avec la fonction `double normalRandom(504)`

3. Le « GNA Gaussien » suite à l'appel précédent lui renvoie un vecteur de variables aléatoires suivant une loi normale gaussienne de moyenne 0 et de variance 1 qu'on nommera  $N_t \forall t \in$

$N^* \cap [0; T - 1]$   $T$  étant la date de maturité exprimée en jours (les différentes valeurs sont donc stockées dans un `vector<double>`  $N$  de taille  $T=504$ ).

4. Le « Constructeur de chemin » interroge le composant « Volatilité et interpolateur » par le biais de la fonction double `getLocalVol(double strike, double maturity)`, il lui fournit en paramètre le strike  $S_t/100$  et la maturité  $t$ . Cet appel sera répété autant de fois qu'il y a de jours de cotations soit  $T (=504)$  fois.

5. Le « Constructeur de chemin » récupère la volatilité locale  $\sigma_t$  en  $t$  ( $t$  désignant le temps courant) et la stocke dans un `vector<double>`  $\sigma$  (qui a une taille de  $T=504$ ).

6. Le module « Constructeur de chemin » renvoie au module « Boucle de Monte Carlo » un vecteur contenant les valeurs estimées du sous-jacent pour chaque jour de cotation.

### a. Calcul de la valeur du sous-jacent en $t$

La valeur du sous-jacent  $S$  en  $t+1$  dépend de la valeur calculée sous-jacent en  $t$ , de la volatilité locale  $\sigma_t$  et de la variable aléatoire  $N_t$ .

La formule permettant de calculer le prix du sous-jacent est la suivante :

$$S_{t+1} = S_t \left( 1 + \frac{\sigma_t}{\sqrt{252}} N_t \right)$$

La valeur  $t$  correspond à fois au temps et à la position dans le vecteur généré.

## 4. Header

Ce composant met à disposition une fonction polymorphe permettant d'obtenir un chemin (vecteur de « double » de taille  $T + 1$ ,  $T$  désignant la maturité) à partir d'un spot de départ et d'une maturité :

**`vector<Double> getChemin()`**

Le spot de départ vaut 100 et la maturité sera de 504 jours de cotation (le vecteur renvoyé aura donc 505 valeurs, la première valeur du vecteur étant le spot et la dernière étant la valeur du sous-jacent à maturité).

Les fonctions suivantes sont également à implémenter à des fins de test :

`Vector<double> getN() ; // renvoie le tableau N`

`Vector<double> getSigma() ; // renvoie le tableau sigma`

## 5. Cas d'erreurs

Dans tous les cas, lorsque le composant rencontre une erreur, il renvoie (« throw ») une chaîne de caractères :

Tableau 1 Correspondance entre type d'erreur et valeur renvoyée

Libellé de l'erreur	Code d'erreur
Spot de départ (strike) négatif ou nul (non respect de l'hypothèse log-normale)	« spot_neg_null »
Spot de départ (strike) trop grand	« spot_too_big »
Maturité négative ou nulle	« maturity_neg_null »
Taille du vecteur renvoyée par le GNA gaussien incorrecte	« incorrect_vector_size »
Au moins une valeur renvoyée par le GNA gaussien n'est pas comprise entre 0 et 1	« incorrect_vector_value »
Le prix $S_t$ du sous-jacent est négatif à une valeur $t$ donnée	« negative_price_value »

On remarquera que dès le composant rencontre une erreur, il se termine immédiatement et renvoie le code erreur correspondant.