Simulation

Projet M2 - IF Apprentissage

A rendre le 24 janvier
Mises à jour et documents:
http://www.lamsade.dauphine.fr/∼saunier/m2if

Ce sujet est relativement minimaliste. Un (léger) effort de recherche peut être nécessaire pour réaliser vos implémentations. Ce sont des problèmes largement documentés. Quelques notes sont rassemblées sur *http ://www.lamsade.dauphine.fr/∼saunier/m2if.*

L'objectif de ce projet est la réalisation d'un programme complet à partir de méthodes mathématiques.

Monte Carlo

Implémenter une fonction de pricing d'options asiatiques dont le paiement dépend du chemin de prix, par une méthode de Monte Carlo (call).

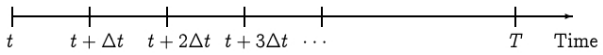

FIGURE 1 – $\Delta_t = \frac{T-t}{N}$

Au temps t, le prix est P. Pour simuler le chemin de prix en N périodes, chaque pas de la séquence est :
$P_{t+\Delta_t} = P_t e^{(r-\frac{1}{2}\sigma^2)\Delta t+\sigma\sqrt{\Delta t}.x}$ avec x une variable aléatoire suivant une distribution normale [1].

La fonction de paiement choisie est une moyenne géométrique :
$payoff = max(0, exp(\frac{\sum_0^{N-1} ln(P_n)}{N}) - K)$ avec $P = \{P_0...P_{N-1}\}$ le chemin de prix.

Votre fonction prend en paramètre le nombre de périodes N et le nombre de simulations n à effectuer. La méthode consiste à simuler n chemins de prix aléatoires, et calculer la moyenne actualisée des prix de l'option pour chacun des chemins :
$e^{-r(T-t)} \frac{\sum_1^n payoff(P)}{n}$

1. Justifier dans le rapport le générateur aléatoire que vous choisissez.

Modèle trinomial à n périodes

Le modèle trinomial est similaire au modèle binomial. Pour chaque période, trois branches sont calculées au lieu de deux.

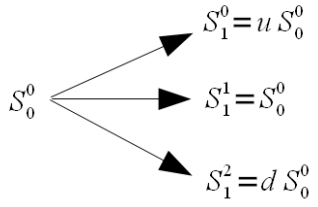

FIGURE 2 – arbre trinomial, deux périodes.

Nous étudions le cas appliqué aux options américaines, pour un **put**. L'évolution du prix peut être approchée par un arbre binomial :

$$S_{t+h} = \begin{cases} u.S_t, & p_u \\ S_t, & 1 - p_u - p_d \\ d.S_t, & p_d \end{cases}$$

Nous donnons les initialisations suivantes :
$u = e^{\sigma\sqrt{3\Delta t}},\ d = \frac{1}{u}$
$p_d = \frac{1}{6} - \sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - \frac{\sigma^2}{2}\right),\ p_u = \frac{1}{6} + \sqrt{\frac{\Delta t}{12\sigma^2}}\left(r - \frac{\sigma^2}{2}\right),\ p_m = 1 - p_u - p_d = \frac{2}{3}$

Algorithme : Il s'agit de construire l'ensemble des noeuds de l'arbre (valeurs S_n^m de S au n-ième pas pour le m-ième noeud), puis de remonter jusqu'à la racine pour trouver la valeur de l'option.

Notons C_n^m le prix d'une option américaine à l'instant n étant donné que l'action vaut S_n^m. Pour le calcul des feuilles, on a $C_n^m = max(0, K - S_n^m)$ avec K le prix du strike.

On peut ensuite remonter l'arbre par la formule d'itération suivante :

$$C_n^m = max(K - S_n^m, e^{-r\Delta t}(p_d C_{n+1}^{m+2} + p_m C_{n+1}^{m+1} + p_u C_{n+1}^m)) \tag{1}$$

avec r le taux d'intérêt par période. La racine de l'arbre donne l'évaluation de l'option.

Implémenter une fonction qui calcule le prix d'une option Put en utilisant le modèle trinomial à périodes multiples, le nombre de périodes étant un paramètre.

Excel et VBA

Créer une macro VBA permettant d'appliquer les méthodes précédentes à un ensemble (petit) de valeurs crédibles rassemblées dans une feuille Excel :

– Pour la partie Monte Carlo, vous choisirez un panel d'options. Pour une de ces options, votre programme VBA doit générer une courbe [2] sur la valeur de l'option en fonction du nombre de simulations. Etudier à partir de quelle nombre de tirages la courbe se stabilise.
– Idem pour le calcul par arbre trinomial, en fonction du nombre de périodes.

Rendu

Vous devez réaliser une partie de ce projet dans un des langages appartenant au programme du cours, à savoir C/C++, C#, ou Python ; et une partie en VBA (au moins la partie liée à Excel pour la saisie et présentation des données), à vous de juger dans quelles proportions. La partie en langage tiers devra être utilisée par votre programme VBA.

Le travail peut être effectué en monôme ou binôme. Le projet est de toute manière évalué en conséquence du nombre d'étudiants le réalisant.

Rappel important : En informatique, et plus encore hors du domaine *stricto sensu* universitaire, le principal critère de référence est que votre programme marche, c'est à dire qu'il **compile**, qu'il **s'exécute** et qu'il **fasse ce qui est demandé**. Il vaut mieux un programme qui ne prenne en compte qu'une partie des fonctionnalités demandées mais qui tourne, plutôt qu'un programme sensé tout faire mais ne marchant pas.

Deux autres critères sont la qualité de votre rapport, et la qualité algorithmique de votre programme.

Code Source

Le code doit être clair et commenté [3]. Pensez à nommer vos variables et fonctions avec des noms explicites (*nbSimulations* est plus clair que n, par exemple).

Vous devez rendre le code source de vos projets sur support informatique [4] (cd, clé usb, mail...).

Rapport

Parallèlement au code de votre programme, le (court) rapport doit mettre en lumière ce qui n'y est pas visible. C'est un projet d'informatique, vos explications doivent donc cibler les problématiques de programmation plus que les problématiques des modèles de pricing.

Il vous est donc demandé de mettre en lumière –au moins– les points suivants :
– Vue globale du projet et de sa structure
– État de votre projet : Qu'est-ce qui est implémenté, qu'est-ce qui marche ?
– Identification et justification des choix principaux d'implémentation.

Il est inutile de recopier l'énoncé du sujet dans le rapport, ou d'y mettre l'intégralité de votre code source.

2. La génération de graphique doit être inclue dans la macro.
3. Sans outrance.
4. Qui vous sera rendu s'il s'agit d'un support réinscriptible.