

Projet

Objectif : L'objectif de ce projet est de comparer différents modèles stochastiques de valorisation d'option Call Vanilla en VBA.

Base de données : Aucune base de données sous Access n'est fournie mais vous pouvez stocker chaque calcul dans une base.

Interface : Le projet doit être réalisé en VBA sous forme de formulaires. Tous les calculs doivent être réalisés en VBA **et non dans des cellules Excel**. Seuls les graphiques pourront être présentés dans des feuilles Excel.

Organisation : Les projets doivent être réalisés par groupe de 2 étudiants (au maximum). Veillez à me remettre des projets qui ne ressemblent pas à ceux des voisins !!!

Projet : L'objectif de ce projet est de comparer différents modèles stochastiques de valorisation d'option Call Vanilla en VBA.

- S_0 désigne le prix initial de l'option
- μ désigne la tendance du sous-jacent
- σ désigne la volatilité du sous-jacent
- K désigne le Strike de l'option
- T désigne la maturité de l'option
- r désigne le taux sans risque
- $dW_t \sim N(0,1)$

Le pay off d'une option Call Vanilla est :

$$[S_T - K]^+$$

Le prix d'une option Call Vanilla est :

$$e^{-r.T} \cdot E[[S_T - K]^+]$$

- 1) Créez une fonction **BlackScholes**($S_0, K, \mu, \sigma, T, r$) donnant le prix d'une option Call Vanilla selon le modèle de Black- Scholes.

Aide : Rappel du modèle de Black-Scholes pour une option Call Vanilla :

- $dS_t = \mu \cdot S_t \cdot dt + \sigma \cdot S_t \cdot dW_t$
- $C(S_0, K, r, t, \sigma) = S_0 \cdot \mathcal{N}(d_1) - K \cdot e^{-r \cdot T} \cdot \mathcal{N}(d_2)$
- $\mathcal{N}(x) = \int_{-\infty}^x \frac{1}{\sqrt{2 \cdot \pi}} \cdot \text{Exp}\left(-\frac{1}{2} u^2\right) \cdot du$
- $d_1 = \frac{1}{\sigma \cdot \sqrt{T}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) \cdot T \right]$
- $d_2 = d_1 - \sigma \cdot \sqrt{T}$

Nous allons maintenant valoriser l'option Call Vanilla sous des modèles alternatifs par la méthode de Monté Carlo.

On considère :

- N le nombre de périodes simulées entre t_0 et T
- $h = t_{i+1} - t_i = \frac{T}{N}$ le pas de temps entre deux prix simulés
- N' nombre de séries de prix simulées
- λ nombre moyen d'occurrences d'évènements exceptionnels
- κ intensité de l'impact des évènements exceptionnels
- $dP_t \sim P(\lambda)$

- 2) Créez une fonction **Normal**(N, N') générant une matrice de N lignes et N' colonnes contenant que des variables aléatoires suivant la loi Normale de moyenne nulle et de volatilité 1.

*Aide : Vous pouvez utiliser la fonction **Application.WorksheetFunction.NormInv(X, 0, 1)***

- 3) On suppose que le prix du sous-jacent suivent dorénavant un mouvement brownien arithmétique.

$$dS_t = \mu \cdot dt + \sigma \cdot dW_t$$

Faites un fonction **PrixArithmetique**(S_0, μ, σ, W) produisant la série de prix du sous-jacent pour chaque pas h à partir d'un vecteur de valeurs aléatoires W

Attention, il faut interdire les prix négatifs en figeant le prix à 0 dès qu'il atteint la limite 0.

- 4) Faites une fonction **PayOffMCArithmetiques**($S_0, K, N, N', T, \mu, \sigma$) donnant le Pay Off moyen du Call Vanilla sur N' simulations en utilisant la fonction **Normal**(N, N') et la fonction **PrixArithmetique**(S_0, μ, σ, W).
- 5) Créez une fonction **Poisson**(N, N', λ) générant une matrice de N lignes et N' colonnes contenant que des variables aléatoires suivant la loi de Poisson.

Aide : Vous pouvez créer des valeurs aléatoires suivant une loi de poisson avec l'algorithme suivante:

```

y ← 0, g ← 1
Loop :
  xy ~ U(0,1)
  g ← g * xy
  If (g > e-λ):
    y ← y + 1
  Else:
    return y

```

- 6) On suppose que le prix du sous-jacent suivent dorénavant un processus à saut.

$$dS_t = \mu \cdot dt + \sigma \cdot dW_t + \kappa \cdot dP_t$$

Faites une fonction **PrixSaut**($S_0, \mu, \sigma, \kappa, W, P$) produisant la série de prix du sous-jacent pour chaque pas h à partir d'un vecteur de valeurs aléatoires W et d'un vecteur de valeurs aléatoires P .

Attention, il faut interdire les prix négatifs en fixant le prix à 0 dès qu'il atteint la limite 0.

- 7) Faites une fonction **PayOffMCSaut**($S_0, K, N, N', T, \mu, \sigma, \kappa, \lambda$) donnant le Pay Off moyen du Call Vanilla sur N' simulations en utilisant la fonction **Normal**(N, N'), la fonction **Poisson**(N, N', λ) et la fonction **PrixSaut**($S_0, \mu, \sigma, \kappa, W, P$).

- 8) Faites une fonction **PriceCallVanilla**($S_0, K, N, N', T, r, \mu, \sigma, \kappa$) affichant les prix du Call Vanilla selon les trois modèles précédents en utilisant les fonctions **BlackScholes**($S_0, K, \mu, \sigma, T, r$), **PayOffMCArithmetiques**($S_0, K, N, N', T, \mu, \sigma$) et **PayOffMCSaut**($S_0, K, N, N', T, \mu, \sigma, \kappa, \lambda$).

Attention, dans la fonction **PayOffCallVanilla**, on ajustera les paramètres d'entrée μ et σ par la périodicité appliquée aux fonctions :

$$\text{BlackScholes}(S_0, K, \frac{\mu}{252}, \frac{\sigma}{\sqrt{252}}, T, r)$$

$$\text{PayOffMCArithmetiques}(S_0, K, N, N', T, \frac{\mu \cdot h}{252}, \frac{\sigma \cdot \sqrt{h}}{\sqrt{252}})$$

$$\text{PayOffMCSaut}(S_0, K, N, N', T, \frac{\mu \cdot h}{252}, \frac{\sigma \cdot \sqrt{h}}{\sqrt{252}}, \kappa, 1 - (1 - \lambda)^{h/252})$$

- 9) Réalisez un formulaire permettant à l'utilisateur d'insérer les paramètres de l'option Call Vanilla. Un bouton permettra d'afficher les prix de l'option Call Vanilla selon les différents modèles. Les paramètres par défaut seront les suivant :

- $S_0 = 100$
- $K = 110$
- $T = 500$
- $N = 50$
- $N' = 50$
- $r = 2\%$
- $\mu = 10\%$
- $\sigma = 15\%$
- $\lambda = 5\%$
- $\kappa = -20\%$

- 10) Dans une seconde fenêtre, faites afficher un graphique de convergence des Pay Off pour les différents modèles en fonction des paramètres définis dans le formulaire.