

**NATURE DE L'ACTIVITÉ**

Nous n'avons pas toujours les moyens suffisants pour acheter ou vendre des objets immédiatement. Il est toutefois possible d'acheter un droit d'achat (call) ou de vente (put) d'un objet pour une date ultérieure à un prix fixé à l'avance. Le pricer calcule ce prix conformément au modèle de Black & Scholes.

**COMPÉTENCES MISES EN ŒUVRE POUR LA RÉALISATION DE L'ACTIVITÉ**

- C33** Développer à l'aide d'un langage de programmation événementielle  
**C37** Mettre au point et maintenir une application

**CONDITIONS DE RÉALISATION****MATÉRIELS ET SYSTÈMES :**

PC sous Windows 2000 Professionnel  
 PC sous Windows XP Professionnel

**LOGICIELS**

Visual Studio.NET

**LANGAGES**

C#, ASP.NET, HTML

**DESCRIPTION DE L'ACTIVITÉ RÉALISÉE****SITUATION :**

L'utilisateur rentre les paramètres nécessaires au calcul du premium en vue d'afficher le prix. Le spot (S) est le prix du moment, le strike (K) le prix voulu, la maturité (T) le temps restant à maturité, la volatilité (v) la sensibilité annualisée du pri, enfin le taux (r) est le taux d'intérêt.

**DÉROULEMENT :**

Avant d'envoyer les données du formulaire, l'application enregistre le type d'option (achat ou vente) dans un ViewState, où il ira chercher la valeur au chargement de la page.

Le logiciel vérifie que les champs renseignés sont bien numériques avant d'appeler la fonction BlackScholes pour le calcul.

Le calcul du prix s'effectue de la manière suivante:

CALL :  $c(S, K, r, T, v) = S N(d_1) - K \exp(-rT) N(d_2)$   
 PUT :  $p(S, K, r, T, v) = -S N(-d_1) + K \exp(-rT) N(-d_2)$

$N(x)$  est la fonction de répartition de la loi normale centrée où :

$$d_1 = \frac{1}{v\sqrt{T}} \left[ \ln\left(\frac{S}{K}\right) + \left(r + \frac{1}{2}v^2\right)T \right] \quad \text{et} \quad d_2 = d_1 - v * \text{racine}(T)$$

$N(x) = 1 - 0.39894228 * \exp(-\text{var} * \text{var} / 2) * y ;$

$y = 1 / (1 + 0.2316419 * \text{ValeurAbsolue}(x));$

$z = (((1.330274429 * x - 1.821255978) * x + 1.781477937) * x - 0.356563782) * x + 0.31938153) * x;$

Et on obtient

Si  $x > 0$  :  $1 - 0.39894228 * \exp(-x * x / 2) * z$

Si  $x < 0$  :  $0.39894228 * \exp(-x * x / 2) * z$

**ANALYSE DES RESULTATS OBTENUS****OBJECTIF ATTEINT :**

On vérifie les valeurs une à une avec des points d'arrêts. Le résultat affiché concorde avec celui de l'exercice du livre Mathématiques Des Marchés Financiers.

**BILAN DE L'ACTIVITÉ :**

Le débogueur de Visual Studio.NET isole les erreurs et facilite grandement le développement d'application. Aussi quand la page affichait des résultats incohérents, les causes du problème étaient assez simple à trouver. Quand la page se rafraîchissait à la validation du formulaire par exemple, réinitialisant ainsi le type à Call, les forums des sites [www.asp.net](http://www.asp.net) et [www.aspalliance.com](http://www.aspalliance.com) m'ont aidé à résoudre le problème.