Calcul du prix d'un PUT et d'un CALL Européen basé sur un modèle GARCH / Comparaison avec le Modèle de Black-Scholes (B.S)

Martin Rupp

I.P.C.S.T
Institut de Préparation Aux Concours Scientifiques et Techniques

Table des matières

I:Principes Théoriques

– Rappel des définition de **Call** et de **Put** liés à une option

Un CALL sur une option permet d'exercer le droit par l'acheteur („reader“) à un moment donné d'acheter une action au vendeur („writer“) , ce dernier ne peut refuser la vente si le CALL a lieu.

Le PUT est l'“inverse“ du Call : dans ce cas c'est le vendeur et non l'acheteur qui a le droit de vendre son action et c'est l'acheteur qui ne peut refuser lachat si le PUT a lieu .

$K = Strike Price : prix de transaction a laquel se fera la vente
T = Time To maturity : Limite de validité du PUT/CALL

– Différence put/call européen et américain

PUT/CALL Américain : on peut exercer l'option avant la date de maturité (t < T)
PUT/CALL Européen : on peut exercer l'option à la date de maturité (t=T)

1. Rappel Modèle Black-Scholes

Selon le modèle de Black-Scholes , un Call européen sur une option avec un payoff de MAX(S-$K,0) avec un temps de maturité T et une valeur S(t) à la date t , aura un prix **C** à t=0 suivant la formule suivante :

$$\mathbf{C} = S_0 * N(d) - Ke^{-rT} N(d - \sigma * \sqrt{(T)})$$

avec $d=\{\ln(S_0/K)+(r+\sigma^2/2)*T\}/(\sigma*\sqrt{(T)})$

2. Rappel Modele Stochastique : GARCH

Le GARCH est un modéle recursif autoregressif et hétéroskedastic . Il permet de modéliser les séries temporelles financières .

Parmi ses divers variantes , le GARCH(p,q) , le EGARCH , etc ... nous choisissons le GARCH(1,1) avec un seul niveau de récursivité .

La formule donnant la variance conditionnelle est :

$$\sigma_{t+1}^2=\omega+\alpha*\sigma_t^2*(\epsilon_t-\lambda)^2+\beta*\sigma_t^2$$

le log-retour est donné par :

$$\ln(S_{t+1}/S_t)=r+\lambda*\sigma_{t+1}-\sigma_{t+1}^2/2+\sigma_{t+1}*\epsilon_{t+1}$$

Soit dans un cas de risque neutralisé localement :

$$\ln(S_{t+1}/S_t)=r-\sigma_{t+1}^2/2+\sigma_{t+1}*\epsilon_{t+1}$$

avec $\epsilon_t \sim N(0{,}1)$

De même , le calcul du prix **C** d'un call sur une option avec un payoff de Max(S-$K,0) à T sera donné par :

C = $e^{-rT}E(Max(S_T-K),0)$

avec $S_T=S_0 e^{Tr-\sum_{i=1}^{T}\sigma_i^2/2+\sum_{i=1}^{T}\sigma_i*\epsilon_i}$

II. Méthode de calcul

si St désigne le cours du titre au moment du CALL (i.e à T) , K le strike price, r le taux de risk free , notre problem se résout à calculer :

Valeur du Call ($K,T) = $e^{-rT}\times E(max(S_T-K,0))$

1. Calibration : Estimation des parametres du GARCH(1,1)

On peut estimer ensuite les différents paramètres du GARCH : $\alpha, \beta, \omega, \lambda$ en utilisant le maximum de vraisemblance via la fonction SOLVER de Excel (solver.dll)

sur : LOGRETURN ~ GARCH(1,1)(***N***(0,1) - lambda)^2 au moyen d'une feuille excel , on construit la série des variances et la série des erreurs associées , on en déduit ensuite les log-maximums de vraisemblances dont la somme doit être maximisée sous la contrainte $\alpha+\beta<1$

Dow	Return	Error	Variance	LogLik
679,06				
685,47	0,94%	0,90%	0,000092	8,859
682,62	-0,42%	-0,46%	0,000090	9,197
677,66	-0,73%	-0,77%	0,000081	9,054
675,73	-0,29%	-0,33%	0,000078	9,390
667,16	-1,28%	-1,32%	0,000069	8,319
660,43	-1,01%	-1,06%	0,000083	8,723
656,44	-0,61%	-0,65%	0,000087	9,107
660,53	0,62%	0,58%	0,000081	9,216
659,68	-0,13%	-0,17%	0,000075	9,484
653,86	-0,89%	-0,93%	0,000065	8,977
645,07	-1,35%	-1,40%	0,000068	8,159
643,69	-0,21%	-0,26%	0,000085	9,334
645,43	0,27%	0,23%	0,000074	9,472
645,85	0,07%	0,02%	0,000065	9,640
639,07	-1,06%	-1,10%	0,000056	8,712
639,84	0,12%	0,08%	0,000065	9,637
637,67	-0,34%	-0,38%	0,000056	9,655
629,84	-1,24%	-1,28%	0,000051	8,276

mu	0,000436	-->
beta	0,864959	
Alpha	0,135041	
const	1,000000	-->
omega	0,000000	
unc var	0,000092	
LogLik	106308,404	-->

parametres a determiner par le solveur de contraintes

cellule à maximiser

contrainte (alpha+beta <= 1

Une méthode ne faisant pas appel à l'application Excel peut être envisagée via une librairie tierce partie tel que le SDK solver de FrontLine Systems .

2. Moyen de calcul du CALL

la valeur de l'actif à maturité vaut : $S_T = S_0 e^{Tr - \sum_{i=1}^{T} \sigma_i^2/2 + \sum_{i=1}^{T} \sigma_i * \epsilon_i}$

et suit une loi conditionnelle qu'il faut estimer . Cette loi n'a **pas** de forme analytique ce qui impose le recours a la simulation .

Comment calculer $e^{-rT} \times E(max(S_T - K,0))$?

Une bonne approximation est donnée par :

$$\overline{C} = e^{-rT} \sum_{j=1}^{N} Max(S_T^j - K,0)/N$$

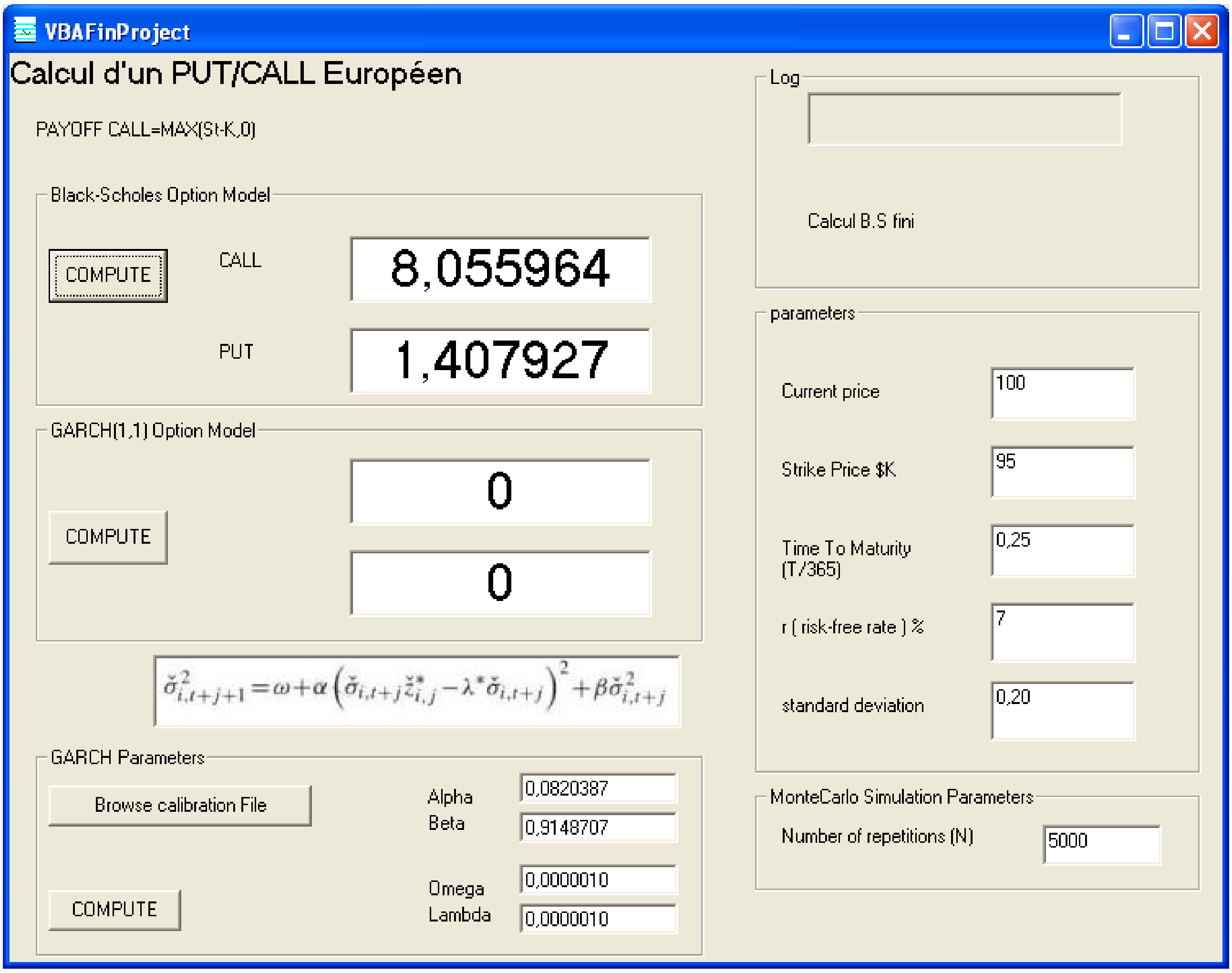
Qui sera obtenue par Simulation Monte-Carlo sur N répétitions (d'autres méthodes étant possibles)

III. Conception du logiciel

Plûtot que de coder directement en VBA dans la feuille Excel , nous choisissons d'écrire le programme en VB 6.0 .

1. Interface

Nous réalisons une Interface Graphique qui nous permet de paramétrer l'algorithme de calcul et de réaliser une comparaison directe avec le prix donné par B.S



2. fonction de calcul via la méthode Black-Scholes

On utilise la fonction suivante qui renvoie directement la valeur du call suivant la formule indiquée ci-dessus .

Call_Eur	S	x	T	r	SD
Input	Le prix de l'action initiale	Le strike price	Le temps de Maturité	Le taux de risque free	L'écart-type (σ)
Output	Le prix du Call				

```
Function Call_Eur(S, x, T, r, SD)
   Dim a As Single
   Dim b As Single
   Dim C As Single
   Dim d1 As Single
```

```
   Dim d2 As Single

   a = Log(S / x)
   b = (r + 0.5 * SD ^ 2) * T
   C = SD * (T ^ 0.5)
   d1 = (a + b) / C
   d2 = d1 - SD * (T ^ 0.5)
   Call_Eur = S * SNorm(d1) - x * Exp(-r * T) * SNorm(d2)

End Function
```

Snorm() réalise une Distribution Normale Standard Cumulée

Le PUT est calculé de manière similaire par parité CALL-PUT .

Put_Eur = x * Exp(-r * T) - S + **CallEur**

3. fonction de calcul via la méthode GARCH(1,1)

▪ Programme de calcul des parametres

Déterminer les paramètres alpha , beta , omega , lambda et sigma(t) à partir du fichier de calibration (SP 500 , Dow Jones , Nasdaq , etc ...) - a faire une seule fois

Le programme va utiliser une fonction SOLVER afin de calculer l'estimation a partir du maximum de vraisemblance

Input : Le chemin vers le fichier de calibration
Output : alpha , beta , omega , lambda et sigma(t)

▪ Programme de Calcul de S(t)

COMPUTE_S	Alp	Bet	Ome	Lam	Time	Sig	Rsk	S0	TT
Input	α	β	ω	λ	Nombre d'itérations (J)	σ	Taux de risk-free	Valeur de l'actif initiale	Temps de maturité en J/365

Output	Valeur de l'action a maturité - S(T)

En fonction des paramètres , calculer S(t) . Cela implique de calculer récursivement sigma(i) pour i=t...T avec l'injection a chaque etape i de l'innovation ϵ (loi normale)

```
Function COMPUTE_S(Alp As Single, Bet As Single, Ome As Single, Lam As Single, Time As
Integer, Sig As Single, Rsk As Single, S0 As Single, TT As Single)

Dim SUM_SS As Single
Dim SUM_SE As Single
Dim iPath As Integer
```

```
Dim Sigma As Single
Dim SSigma As Single

Dim Innov As Single


Innov = gauss()
Sigma = Sig
SSigma = Sig ^ 2
'? To Check
SUM_SS = 0
SUM_SE = 0

'get T values of epsilon ("path") - innovation without a call to randomize()
'==> call to gauss()
'compute the T values of sigma
'sigma^2=omega  +  alpha x sigma^2 x (epsilon - lambda)^2   +  beta x sigma^2

For iPath = 1 To Time

Innov = gauss()

SSigma = Ome + Alp * SSigma * ((Innov - Lam) ^ 2) + Bet * SSigma
Sigma = Sqr(SSigma)

SUM_SS = SUM_SS + SSigma
SUM_SE = SUM_SE + (Sigma * Innov)

Next iPath

COMPUTE_S = S0 * Exp((Rsk * TT) - (SUM_SS / 2) + SUM_SE)

End Function
```

- ## Programme de calcul du CALL

Utilise un generateur de MonteCarlo et calcule:

$$\overline{C} = \mathrm{e}^{-rT} \sum_{j=1}^{N} Max\left(S_T^j - K, 0\right) / N$$

calcul le PUT par parité PUT/CALL

Librairies nécessaires :

- Générateur de lois normales (Fonction Gauss())
- Générateur aleatoire (on utilisera Rnd et randomize() avec comme seed un nombre dépendant de l'itération et de Random et du temps actuel en ms)
- Efficacité du generateur de Monte-Carlo en calculant la valeur de Pi en même temps (totalement optionnel)
-

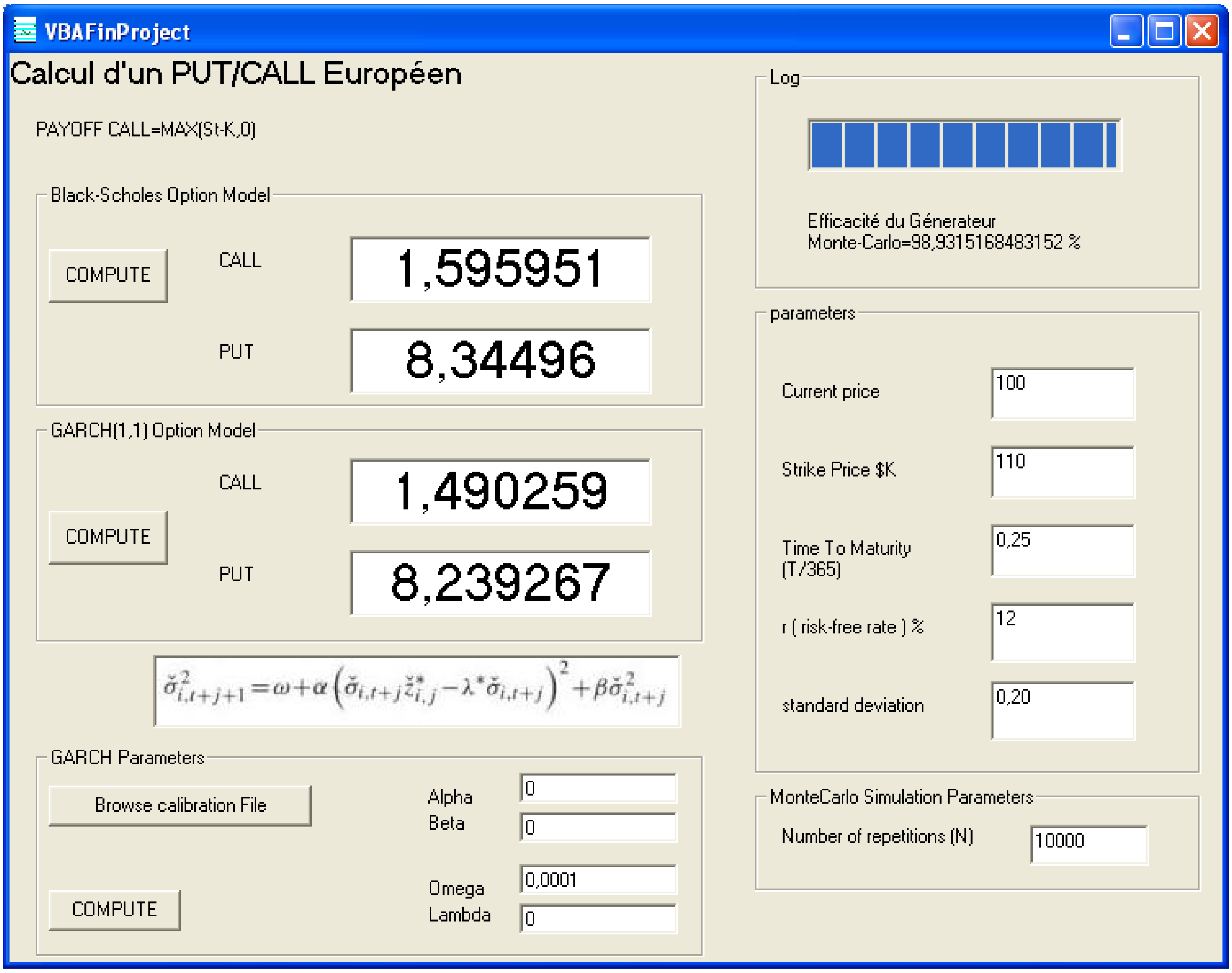
Il s'agit de la form principale qui récupere les entrées via les différentes TextBox et les convertit en Single (Util de CSng)ou Integer (Util de Cint) .

```
'*****************************************************************
'*          GARCH European PUT/CALL Price Computation                    *
'*****************************************************************

Private Sub COMPUTE_GARCH_Click()

Dim vALPHA  As Single
Dim vBETA   As Single
Dim vOMEGA  As Single
Dim vLAMBDA As Single
Dim MC As Long
Dim iMc As Integer
Dim TTM As Integer
Dim SIGMA0 As Single
Dim SSUM As Single
Dim RFR As Single
Dim SSTART As Single
Dim T_ As Single
Dim K As Single
Dim S As Single
Dim CP As Single
Dim PP As Single
Dim iBar As Integer
'Initialisation

vALPHA = CSng(ALPHA.Text)
vBETA = CSng(BETA.Text)
vOMEGA = CSng(OMEGA.Text)
vLAMBDA = CSng(LAMBDA.Text)
MC = CLng(MONTE_CARLO.Text)
T_ = CSng(TIME_TO_MATURITY.Text)
TTM = CInt(CSng(TIME_TO_MATURITY.Text) * 365)
SIGMA0 = (CSng(STANDARD_DEVIATION.Text))
RFR = CSng(RISK_FREE_RATE.Text) / 100
SSTART = CSng(CURRENT_PRICE.Text)
K = CSng(STRIKE_PRICE.Text)
SSUM = 0
S = SSTART

'for testing the random generator
Dim SPI As Single
SPI = 0

ProgressBar1.Value = 0
iBar = 0

TTM = Round(TTM)

'Start Monte Carlo Sampling

For iMc = 1 To MC

Randomize (Round(iMc * Rnd * 100))

If iBar = 50 Then
LogTxt.Caption = "Monte-carlo : " + CStr(iMc) + " /" + CStr(MC) + ",  S=" + CStr(S)
```

```
ProgressBar1.Value = 100 * (iMc / MC)
iBar = 0
LogFrame.Refresh
End If
iBar = iBar + 1

S = COMPUTE_S(vALPHA, vBETA, vOMEGA, vLAMBDA, TTM, SIGMA0, RFR, SSTART,
T_)

'test du generateur aleatoire
SPI = SPI + Estimate_pi

SSUM = SSUM + Max(S - K, 0)

Next iMc

LogTxt.Caption = "Efficacité du Génerateur Monte-Carlo=" + CStr(100 - Min(100, 100 * Abs((4 *
SPI / iMc) - 3.141))) + " %"


CP = (Exp(-RFR * T_) * SSUM) / MC

CALL_GARCH.Text = CStr(CP)
PP = (K * Exp(-RFR * T_)) - SSTART + CP
PUT_GARCH.Text = CStr(PP)

End Sub
```

IV. Calculs Numériques

Si $\alpha = \beta = 0$ on doit constater que les valeurs sont à peu près les mêmes que B.S , puisque B.S est alors un „cas particulier“ du GARCH(1,1)



Ref :

GARCH AND STOCHASTIC VOLATILITY
OPTION PRICING , Jin-Chuan Duan, Department of Finance
Hong Kong University of Science & Technology