Finance Quantitative

TP: Modèle Trinomial

Patrick Hénaff

Version: 30 mars 2023

L'objectif de ce TP est de construire un modèle trinomial par agrégation de deux pas d'un modèle binomial de type Jarrow-Rudd.

On rappelle les paramètres du modèle binomial de Jarrow-Rudd, qui est charactérisé par des probabilités de transition $p = q = 1/2$: $u = e^{\mu \Delta t + \sigma \sqrt{\Delta t}}$, $d = e^{\mu \Delta t - \sigma \sqrt{\Delta t}}$.

avec $\mu = (r - d) - \frac{1}{2}\sigma^2$.

Questions

Calculer les paramètres d'un arbre trinomial constitué par l'agrégation de deux pas de temps d'un arbre binomial de Jarrow-Rudd.

Le pas de temps du modèle trinomial est Δt, par agrégation de deux pas binomiaux de $\Delta t/2$. Partant du noeud S_0, on peut atteindre les noeuds $S'_2 = S_0 u^2, S'_1 = S_0 ud, S'_0 = S_0 d^2$ avec les probabilités $p = q = 1/4$

Construire un arbre de 200 pas, maturité 1 an pour le processus log-normal

$$dS_t = rdt + \sigma dW$$

avec les paramètres suivants:

\$S_0\$	1e+02
\$\sigma\$	2e-01
\$r\$	2e-02

```
S.0 <- 100
r <- .02
sigma = .2
T <- 1
N <- 200
dt.bin <- (T/N)/2
mu <- r - sigma^2 / 2
u.bin <- exp(mu*dt.bin + sigma*sqrt(dt.bin))
d.bin <- exp(mu*dt.bin - sigma*sqrt(dt.bin))
u <- u.bin^2
d <- d.bin^2
m <- u.bin*d.bin
p <- 1/4
```

```
q <- 1/4
dt <- T/N
```

Comme on doit valoriser une option de type Européenne, il suffit de construire un vecteur des noeuds de l'arbre à maturité pour pouvoir effectuer une récursion inverse. L'indice des pas de temps commence à $n = 0$. A l'étape n, on a $2n + 1$ noeuds.

```
iExp <- (1:(2*(N+1)-1))-(N+1)
uu <- exp(sigma*sqrt(2*dt))
S <- exp(mu*T) * S.0 * uu^iExp
```

Déterminez une manière parcimonieuse de représenter cet arbre, adaptée à la résolution de la question suivante.

Vérification: Forward:

```
df <- exp(-r*dt)
Fwd  <- S
for(i in seq(N,1,-1)) {
  l <- length(Fwd)
Fwd <- df*(p * Fwd[3:l] + (1-p-q) * Fwd[2:(l-1)] + q * Fwd[1:(l-2)])
}
P.Fwd <- Fwd[1]
print(paste("Valeur actualisée de S.T: ", round(P.Fwd, 3)))
```

```
## [1] "Valeur actualisée de S.T:  100"
```

Vérification: valorisation d'un call ATM.

```
df <- exp(-r*dt)
K <- 100
Call <- pmax(S-K, 0)
for(i in seq(N,1,-1)) {
  l <- length(Call)
Call <- df*(p * Call[3:l] + (1-p-q) * Call[2:(l-1)] + q * Call[1:(l-2)])
}
P.Tri <- Call[1]
P.ATM <- GBSOption("c", S.0, K, Time=1, r, b=r, sigma)@price
print(paste("call ATM: Trinomial: ", round(P.Tri,3), " BS: ", round(P.ATM,3)))
```

```
## [1] "call ATM: Trinomial:  8.911  BS:  8.916"
```

Valorisation d'une option "chooser".

Une option "chooser" de maturité t_2 et strike K donne le droit au détenteur de choisir si l'option est un call ou un put, à un moment t_1 de la vie de l'option préalablement défini. A ce moment là, l'option "chooser" vaut $\max(C(K,t_2), P(K,t_2))$, où $C(K,t_2)$ $(P(K,t_2))$ est la valeur en t_1 d'un call (put) de strike K de maturité t_2.

1. Calculer la valeur d'une option "chooser" de strike $K = 100$, avec $t_2 = 1$ an, $t_1 = t_2/2$.

2. Montrer que l'option "chooser"peut être répliqué par un portefeuille statique, et calculez sa valeur analytiquement. Comparez vos deux estimations. \end{enumerate}

Version 1: Calcul des valeurs du call et du put au temps t_1.

```
# Valeurs du call et du put à la date de maturité
K <- 100
Call <- pmax(S-K, 0)
Put <- pmax(K-S, 0)
df <- exp(-r*dt)

# Recursion inverse jusqu'à t.1
for(i in seq(from=N, to=N/2,by=-1)) {
  l <- length(Call)
  Call <- df*(p * Call[3:l] + (1-p-q) * Call[2:(l-1)] + q * Call[1:(l-2)])
  Put <- df*(p * Put[3:l] + (1-p-q) * Put[2:(l-1)] + q * Put[1:(l-2)])
}

# At t.1, choisir max(call, put)
CP <- pmax(Call, Put)
# Recursion inverse jusqu'à t.0
for(i in seq(from=(N/2)-1, to=1, by=-1)) {
  l = length(CP)
  CP <- df*(p * CP[3:l] + (1-p-q) * CP[2:(l-1)] + q * CP[1:(l-2)])
}
```

La valeur de l'option "chooser" est: 13.537, ou 13.542 avec reduction de variance.

Version 2: Solution analytique

On utilise la relation de parité call-put:

$$C + Ke^{-rT} = P + S$$

En t_1, on calcule le maximum entre un call et un put de maturité t_2: $\max(C(t_1, t_2, K), C(t_1, t_2, K))$. On utilise la relation de parité call-put pour éliminer $P(t_1, t_2, K). La valeur en t_1 devient:

$$V(t_1) = \max(C(t_1, t_2, K), C(t_1, t_2, K) + Ke^{-r(t_2-t_1)} - S(t_1)) = C(t_1, t_2, K) + \max(0, Ke^{-r(t_2-t_1)} - S(t_1))$$

Dans la dernière expression, on reconnait la valeur d'exercice d'un put de maturité t_1 et de strike $Ke^{-r(t_2-t_1)}$.

L'option "chooser" est donc finalement un portefeuille composé: - d'un call de strike K, maturité t_2 - d'un put de strike $Ke^{-r(t_2-t_1)}$, maturité t_1.

La valeur analytique est donc:

```
t.2 <- 1
t.1 <- 0.5
V.1 <- GBSOption("c", S.0, K, Time=t.2, r, b=r, sigma)
V.2 <- GBSOption("p", S.0, X=K*exp(-r*(t.2-t.1)), Time=t.1, r, b=r, sigma)
```

La valeur analytique de l'option "chooser" est: 13.563, très proche de la valeur donnée par le modèle trinomial.