Projet Quantlib

Module 304 - IMT ATLANTIQUE
11/02/2019

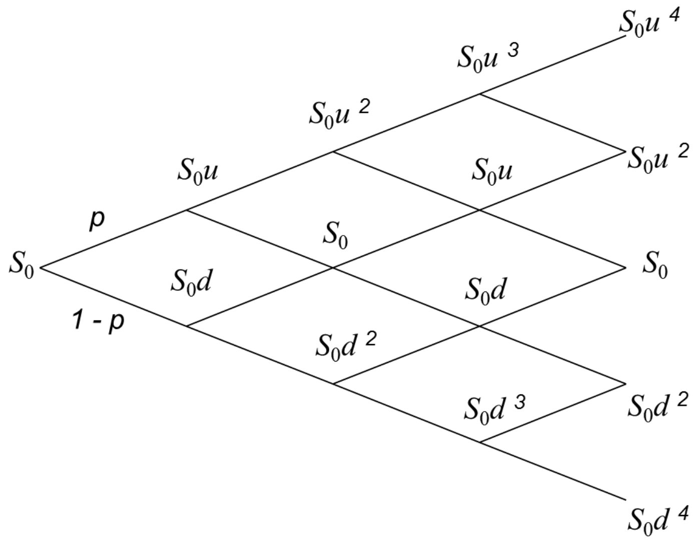
NSKRBA Nikola
SHEN Sheng
ZHAO Fei
MANCHENO KAJJIOU Yanis

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

I - Introduction

Afin de valoriser une option sur les marchés financiers, il existe 3 méthodes couramment utilisées : le calcul par le modèle de Black&Scholes (appelé B&S dans la suite) créée par Fisher Black et Myron Scholes puis développé par Robert C.Merton en 1973, la simulation de Monte Carlo et la méthode de l'arbre binomial développée par Cox, Ross et Rubinstein en 1979.
Le modèle de B&S modélise l'actif comme un processus stochastique à temps continu et est la méthode la plus utilisée car elle a le mérite d'être rapide à calculer et facile à comprendre. Néanmoins elle ne peut être utilisée que pour les options européennes et elle suppose plusieurs hypothèses de départ qui ne sont pas toujours vérifiées dans la réalité comme l'invariabilité du taux d'intérêt sans risque et de la volatilité.
La simulation de Monte Carlo permet de valoriser toutes les options notamment les options "path-dependant" comme les options asiatiques, mais elle requiert un temps de calcul assez conséquent par rapport à B&S et à l'arbre binomiale.
Enfin la méthode de l'arbre binomial suit un processus stochastique à temps discret et permet de valoriser rapidement les options simple comme les options européennes, américaines et bermudiennes, elle est moins rapide que la méthode de B&S mais plus précise.

Nous allons ici nous intéresser à la méthode de l'arbre binomial: pour calculer la valeur de l'option au temps *t0* on trace un arbre de scénarii en affectant une probabilité *p* si la valeur de l'actif augmente et *(1-p)* si elle diminue, puis on réitère sur un nombre de pas choisis jusqu'à la maturité. On obtient donc un arbre avec un nombre de feuilles égale au nombre de pas choisis et qui représentent les valeurs possibles de l'actif à maturité:



On peut donc ensuite calculer la valeur de l'option à chaque noeud final et ensuite remonter l'arbre jusqu'au noeud initial pour trouver la valeur de l'option à la date à laquelle on veut la valoriser.
Pour appliquer la méthode binomiale à une option nous disposons de la librairie C++ Quantlib dirigée par Ballabio Luigi et Ferdinand Ametrano.

II - Position du problème

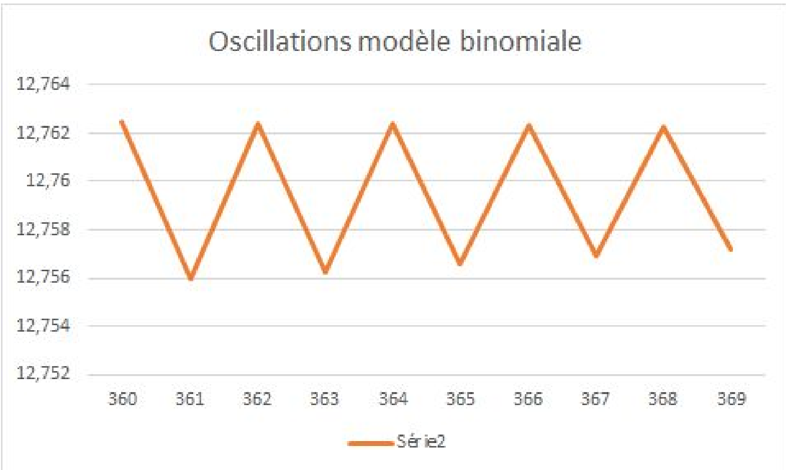
Nous avons à notre disposition une classe permettant de calculer la valeur d’une option vanille : *BinomialVanillaEngine.* Celle-ci utilise la **méthode binomiale** telle que décrite en introduction. On part de l’hypothèse qu’en disposant d’un portefeuille contenant différents actifs, sa valeur future peut prendre seulement deux états, soit dans une situation haussière ou bien une situation baissière des prix. Dans cette optique, l’utilisation du modèle est adéquate et reflète bien une grande partie de la réalité.
La valeur de l’option est de la forme :
C0 = S0 * phi - K/(1+r)n* phi où phi distribution binomiale, S le prix du sous jacent, et K le strike.

Lors de la modélisation numérique, nous rencontrons le problème suivant : nous observons des <u>oscillations</u> du prix de l’option selon le nombre de *timestep* que nous choisissons (cf. graphe ci-dessous avec les données du sujet):

time steps	value	time steps	value
360	12.7624691555	365	12.756572671
361	12.7559902568	366	12.7623429479
362	12.7624336546	367	12.7568998734
363	12.7562325066	368	12.7622865313
364	12.7623909426	369	12.7572141581

Table 1: Value of an American put option on a binomial Cox-Ross-Rubinstein tree for different numbers of time steps. $u = 100$, $k = 110$, $T = 1$, $r = 3\%$, $q = 0$, $\sigma = 20\%$.

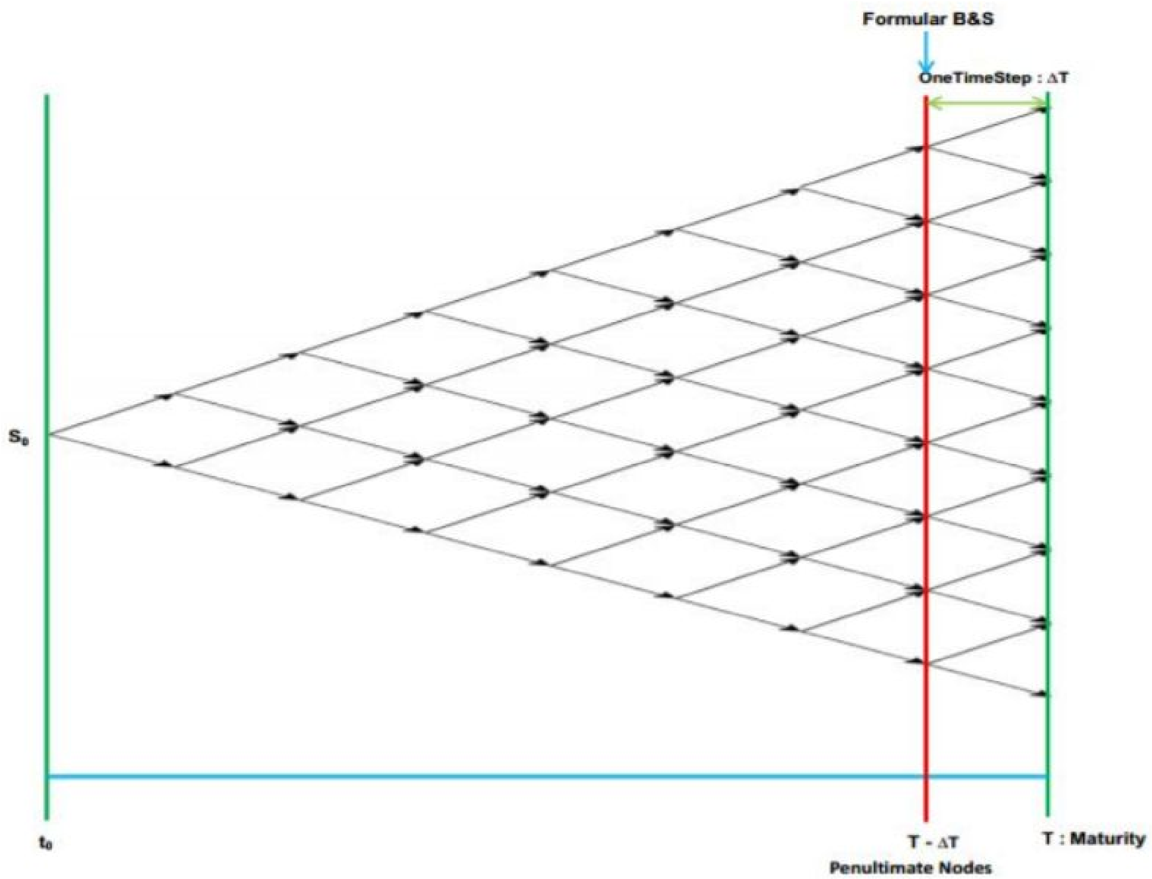
.



Nous pouvons toutefois remarquer que le modèle semble converger. En effet, nous avons : (Vi+1-Vi) > (Vp+1-Vp) avec i<p. En extrapolant cette modélisation, on peut donc conclure qu’avec un timestep très grand, nous pourrions diminuer l’erreur sur le prix avec des oscillations toujours plus petites. Néanmoins, cela irait de pair avec une augmentation exponentiel du temps de calcul.

III - La solution proposée

Ainsi pour résoudre ce problème d'oscillation, Chung and Shackleton ont proposé en 1996 une autre solution. Celle-ci consiste à calculer les prix de l'option pour chacun des pénultièmes noeuds de l'arbre binomiale par la méthode de Black & Scholes. Ceci est rendu possible par le fait qu'à cet endroit, l'option peut être considérée comme une option européenne de maturité *dt_* (un timestep). Autrement dit, elle ne peut pas être exercée entre les noeuds pénultièmes et la maturité T.



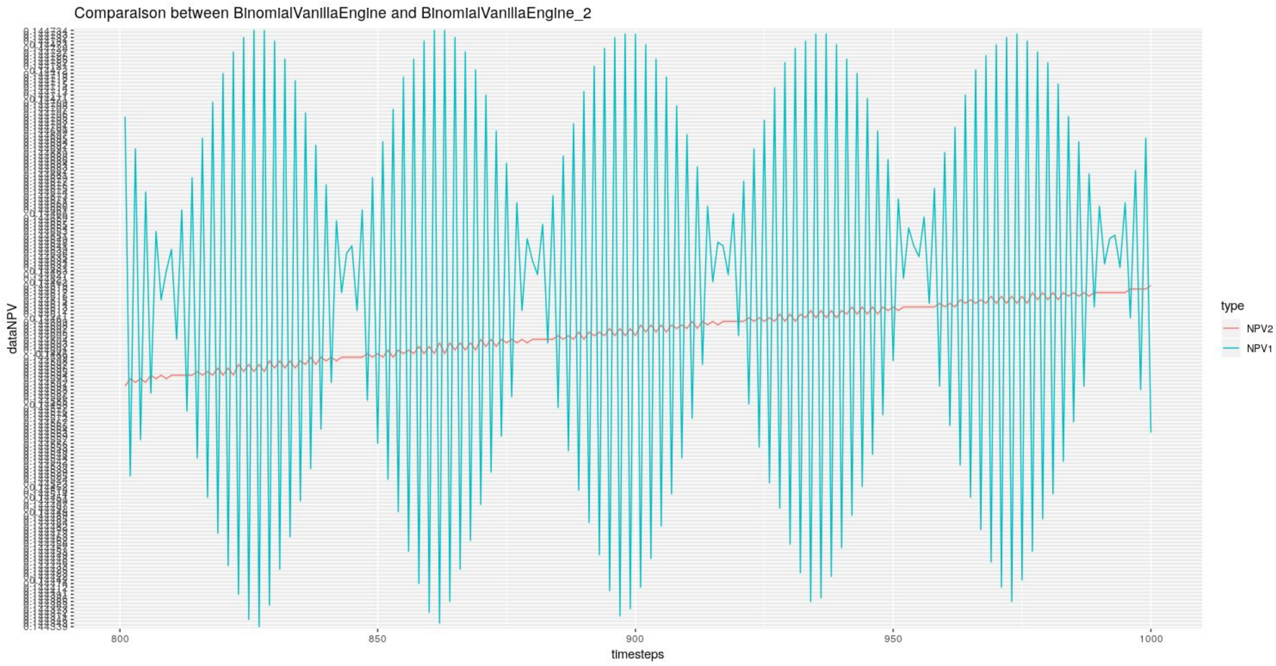
Détaillons succinctement le processus mis en jeu :

- nous parcourons l'arbre afin de récupérer le prix du sous jacent à l'ensemble des avant-derniers noeuds : *lattice -> underlying(timestep -1)*
- nous calculons le prix de l'option par la méthode de Black & Scholes avec en paramètre le prix du sous jacent déterminé précédemment.
- enfin, nous appliquons la méthode *rollback()* à la valeur de l'option obtenue pour remonter à la valeur recherchée, selon le principe de la méthode binomiale.

```
for (int i =0 ; i<Integer(lattice->size(timeSteps_ -1)); i++) {
    BlackScholesCalculator bsCalculator(payoff->optionType(), strike_, lattice->underlying(timeSteps_ -1 , i), growth, vol, discount_) ;
    option.values()[i] = bsCalculator.value() ;
```

IV - Résultats

Après initialisation des valeurs relatives à notre option, nous appliquons notre méthode NPV() et nous stockons l'ensemble des résultats dans la variable NPV2 que nous traçons ensuite à l'aide d'un code R.



La courbe bleue représente la valorisation de l'option avant modification, et la courbe rouge représente la valorisation après introduction de la méthode développée par Chung and Shackleton.
Il apparaît clairement que les oscillations ont été fortement réduites.

V - Conclusion

En se basant sur la solution proposée par Chung and Shackleton et en modifiant la classe *BinomialVanillaEngine* nous avons donc réussi à réduire les oscillations en convergence de la valeur de l'option calculée par la méthode de l'arbre binomial.