Finance Quantitative

Modèle de Black-Derman-Toy

Patrick Hénaff

Version: 14 mars 2024

1 Le modèle de Black-Derman-Toy

On considère le modèle de Black, Derman et Toy décrit dans la note de cours.

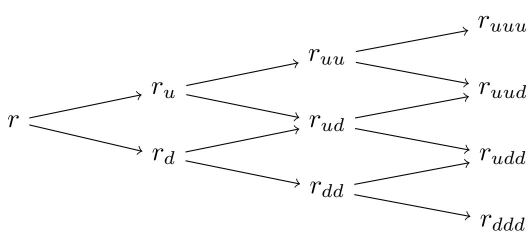


Figure 1: Black-Derman-Toy short rate tree

On doit calibrer le modèle à une courbe zero-coupon et une courbe de volatilité du taux zero-coupon.

Maturity	$z(t)$	$\beta(t)$
1	10.0	
2	11.0	19.0
3	12.0	18.0
4	12.5	17.7
5	13.0	17.5

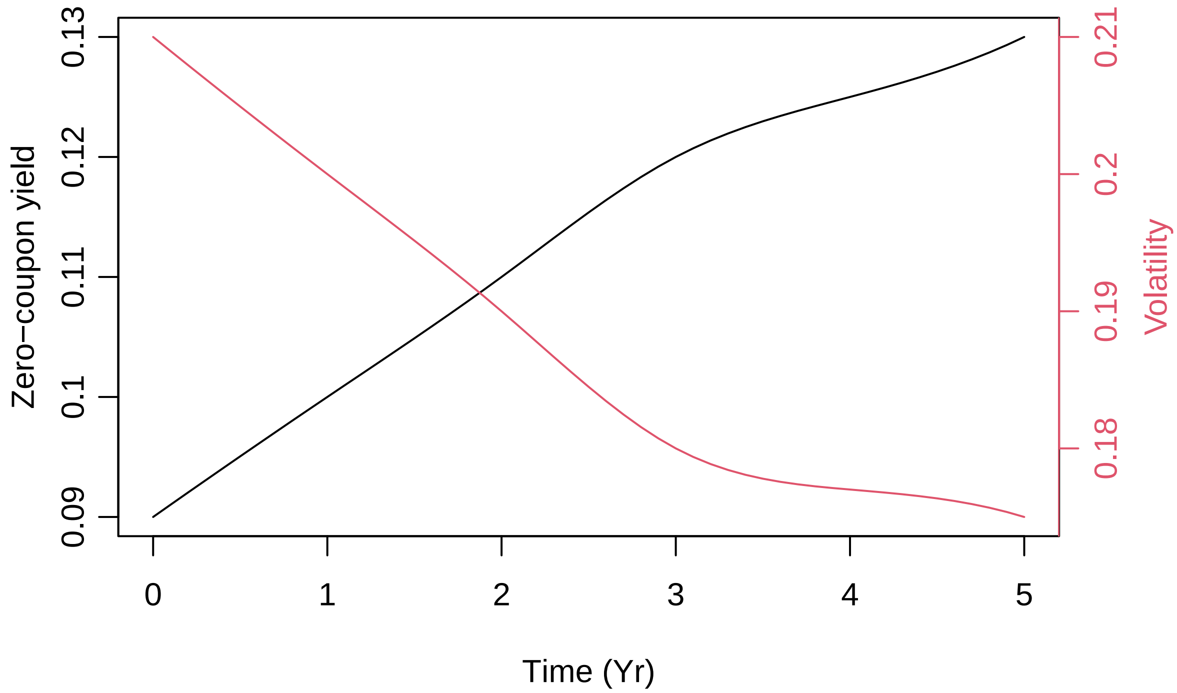
1.1 Construction d'un arbre BDT

```
z <- data.bdt$z/100
beta <- data.bdt$b/100
```

Fonctions d'interpolation pour la courbe zero-coupon et la courbe de volatilité. On ajoute un taux court à la courbe zero-coupon pour permettre une interpolation robuste.

```
zc.curve <- splinefun(seq(0,5), c(.09, z))
beta[1] <- .2
vol.curve <- splinefun(seq(0,5), c(.21, beta))

df <- function(r) {
  1/(1+r)
}
```



2 Questions

1. En suivant l'exemple donné en cours, construire un arbre de Black-Derman-Toy à 3 pas de temps, chaque pas de temps représentant 1 an.

- Pour le pas Δt, poser les equations en r_u et r_d et les résoudre avec la fonction `nleqslv` du package du même nom.
- Même question pour les pas de maturité 2 ans et 3 ans. Utiliser la volatility locale constante pour formuler le problème en en système de deux equations à deux inconnues.

2. Calculer le prix d'une obligation zero-coupon de maturité 3 ans, aux noeuds de l'arbre r_{uu}, r_{ud}, r_{dd}.

3. Calculer le prix d'un call Européen de maturité deux ans, strike 85 sur une obligation zero-coupon de maturité 3 ans.

4. Géneraliser la mise en oeuvre pour pouvoir construire un arbre de n pas, et d'incrément Δt.

2.1 Equations pour t = 2 ans

```
obj <- function(x) {
  r.d <- x[1]
  r.u <- r.d * x[2]
  res <- numeric(2)
  res[1] <- df(z[1]) * (1/2) * (df(r.u) + df(r.d)) - df(z[2]) **2
  res[2] <- (1/2) * log(r.u/r.d) - beta[2]*sqrt(1)
  res
}

sol <- nleqslv(as.vector(c(.1, 1)), obj)
r.d <- sol$x[1]
r.u <- r.d * sol$x[2]
```

Ce qui donne $r_u = 0.143$ and $r_d = 0.098$. Comme la volatilité locale est constante sur un pas de temps,

$$\frac{1}{2}\ln\left(\frac{r_{uu}}{r_{du}}\right) = \frac{1}{2}\ln\left(\frac{r_{du}}{r_{dd}}\right) = \sigma(t)\sqrt{\Delta t} = \alpha(t)$$

quelque soit le nombre de noeuds à une étape de temps, il suffit de résoudre un système de deux équations à deux inconnues: le taux court du noeud le plus bas $\hat{r}$ et le coefficient α. Ces deux variables définissent l'ensemble des taux courts d'une étape de temps dans l'arbre.