

Modèle de BOYLE :

FUNCTION PrixBOYLE

```
(Spot,           // Prix du Sous-Jacent à l'instant initial
Taux,           // Valeur du taux d'intérêt supposé constant
Volatilite,     // Valeur de la volatilité
MaturiteAnnuelle, // Maturité de l'option
Lambda,        // Paramètre Lambda
PrixExercice : Double; // Prix d'exercice
NombreDePeriodes, // Nombre de Périodes de l'arbre
CallOuPut : Integer) : Double;
VAR
i,j : Integer;
DeltaT : Double; // Durée d'une période
u,d : Double; // Coefficients de hausse et de baisse
ProbaUp, ProbaMid, ProbaDown : Double; // Probabilité de hausse,
de tendance
// moyenne et de baisse
Discount : Double; // Coefficient d'actualisation
ValeurSousJacent , ValeurOption : Array of double; // Tableau de
valeurs
//du Sous Jacents et de l'option
M, V : Double; // Coefficients Intermédiaires du calcul des
probabilités
BEGIN
// 1. Calculs Préliminaires
// Calcul de la longueur d'une période 'DeltaT'
DeltaT := MaturiteAnnuelle / NombreDePeriodes;
```

```

// Calculs des coefficients de hausse 'u' et de baisse 'd' au cours de
chaque
// période
u := Exp(Lambda * Volatilite * sqrt(DeltaT));
d := 1/u;

// Calculs des Probabilités de Hausse 'ProbaUp', de tendance
moyenne 'ProbaMid'
// et de Baisse 'ProbaDown'
M := Exp(Taux*DeltaT);
V := Sqr(M) * (Exp(Sqr(Volatilite)*DeltaT)-1);
ProbaUp := ((V+Sqr(M)-M)*u - (M-1)) / ((u-1)*(sqrt(u)-1));
ProbaDown := ((V+Sqr(M)-M)*Sqr(u) - IntPower(u,3)*(M-1)) / ((u-
1)*(sqrt(u)-1));
ProbaMid := 1 - ProbaUp - ProbaDown;

// Calculs du coefficient d'actualisation
Discount := 1 / M;

//      2. Calcul des Valeurs du Sous-Jacent 'ValeurSousJacent' à
//      l'Echéance

// Allocation de la Mémoire du Vecteur : ValeurSousJacent
SetLength(ValeurSousJacent, (2*NombreDePeriodes+1));

// Calculs des valeurs
For i:=1 To 2*NombreDePeriodes+1 Do
ValeurSousJacent[i] := Spot      * IntPower(u,
NombreDePeriodes+1-i)

//      3. Calcul des Valeurs Finales de l'Option

// Allocation de la Mémoire du Vecteur : ValeurOption
SetLength (ValeurOption, (2*NombreDePeriodes+1));

```

```

// Calculs des valeurs Finales de l'option
For i:= 1 To 2*NombreDePeriodes+1 Do
ValeurOption[i] := Max(CallOuPut*(ValeurSousJacent[i] -
PrixExercice) , 0);

// 4. Induction Arrière, Calcul de la Valeur de l'Option à
l'Instant Initial
For i:=NombreDePeriodes DownTo 1 Do
For j:=1 To 2*i Do
ValeurOption[j] := Discount * (ProbaUp * ValeurOption[j]
+ ProbaMid * ValeurOption[j+1]
+ ProbaDown * ValeurOption[j+2]);

Result := ValeurOption[1];

// Libération Mémoire des vecteurs. Se fait automatiquement en
Delphi !
ValeurSousJacent := nil ;
ValeurOption := nil ;
END;

```