

## **E.I.S.T.I.      Département Mathématiques**

**Spécialisation Génie Mathématique    2018-2019**

### **PROJET : CHAÎNES DE MARKOV EN FINANCE. MÉTHODES STOCHASTIQUES ET DÉTERMINISTES DE PRICING.**

Les modèles discrètes binomial et trinomial sont très répandues en finance. Une des applications de ces modèles est l'évaluation des options et des obligations. D'une grande simplicité, ils ont permis à des générations de traders et de market-makers d'évaluer leurs "books" avec une flexibilité suffisante pour leur permettre de gagner leurs vies, et parfois plus.

Actuellement les Chaînes de Markov construites à partir de l'arbre Binomial sont utilisés pour implémenter l'évolution des actifs sous-jacents et calculer les prix des options exotiques dont les équations n'admettent pas une formule analytique. Les Chaînes de Markov sont appliquées aussi pour générer des événements rares. Dans ce cas les algorithmes génétiques permettent de sélectionner les Chaînes intéressantes pour un problème étudié.

Les modèles fournissent les moyens les plus simples pour évaluer les options américaines. La procédure de calcul est une simple application de l'équation de Bellman, bien connue en programmation dynamique. L'arbre trinomial est la première modification que les chercheurs ont fait subir à l'arbre binomial pour la rendre plus flexible. A tout moment le prix d'un instrument financier peut enregistrer trois mouvements dans un arbre trinomial plutôt que deux comme dans l'arbre binomial. A partir d'un tel arbre on peut reproduire un processus Ornstein-Uhlenbeck ou processus de retour vers la moyenne. Un tel arbre permet d'intégrer le phénomène du smile de volatilité. Les extensions des arbres simples sont nombreuses. Plus récemment d'autres variantes de l'arbre binomial sont apparues. D'abord l'arbre quadrimomial qui est particulièrement populaire pour les options réelles. Ensuite Rubinstein a développé l'arbre implicite qui prend en compte, lors de sa construction les prix des options observés sur le marché. On applique aussi la technique de l'arbre binomial pour évaluer les titres à revenus fixes.

# 1 Introduction

## 1.1 Evolution de l'actif sous-jacent dans le Modèle Binomial

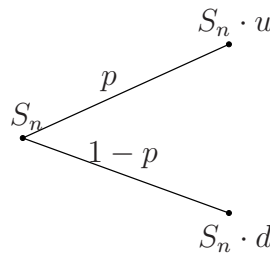
Supposons qu'une action  $S$  évolue dans le temps sur l'intervalle  $[0, T]$ . Discrétisons l'intervalle  $[0, T] : t_n = \Delta \cdot tn, \quad T = \Delta t \cdot N$ . A chaque instant  $t_n$  on fait correspondre le prix  $S_n$  :

$$t_n \rightarrow S_n, \quad t_0 \rightarrow S_0$$

L'idée principale de la Méthode Binômiale est la suivante :  $S_{n+1}$  n'admet que 2 valeurs possibles :

$$S_{n+1} = \begin{cases} u \cdot S_n \\ d \cdot S_n \end{cases}$$

On peut donc approcher la diffusion dans le temps par une chaîne de Markov (voir le cours de processus stochastiques discrets) - un arbre binômial.

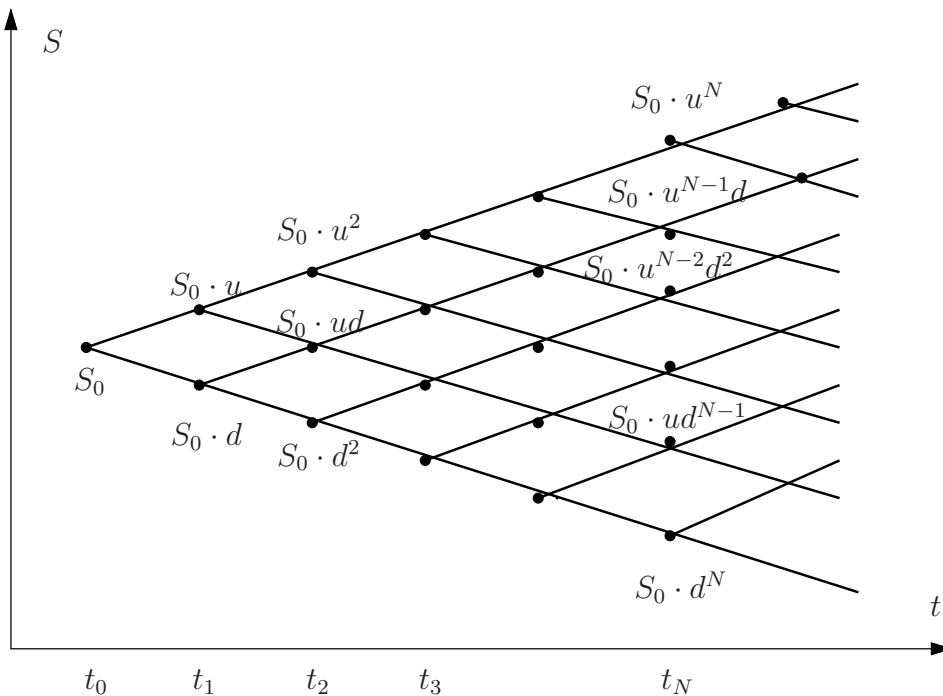


Introduisons un autre indice "i" pour calculer le prix de l'action à chaque noeud de l'arbre

$$S(n, i) = (u)^i (d)^{n-i} S_0$$

- l'indice  $n$  est l'indice temporel
- l'indice  $i$  est l'indice d'une branche qui définit la valeur de l'action parmi  $n + 1$  valeurs possibles :

$$0 \leq i \leq n$$



## 1.2 Evolution de l'actif sous-jacent dans le Modèle Trinomial

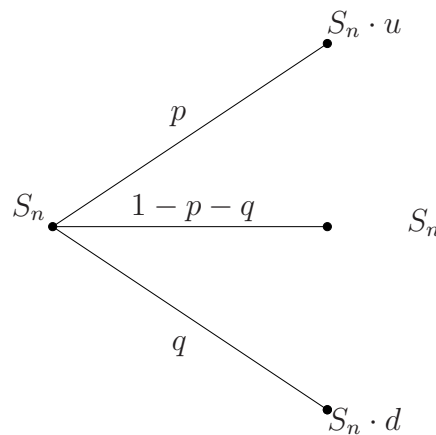
Supposons qu'une action  $S$  évolue dans le temps sur l'intervalle  $[0, T]$ . Discrétisons l'intervalle  $[0, T]$  :  $t_n = \Delta t n$ ,  $T = \Delta t N$ . A chaque instant  $t_n$  on fait correspondre le prix  $S_n$  :

$$t_n \rightarrow S_n, \quad t_0 \rightarrow S_0$$

L'idée principale de la Méthode Trinômiale est la suivante :  $S_{n+1}$  n'admet que 3 valeurs possibles :

$$S_{n+1} = \begin{cases} u \cdot S_n \\ S_n \\ d \cdot S_n \end{cases}$$

On peut donc approcher la diffusion dans le temps par une chaîne de Markov - un arbre trinomial.



Pour que l'arbre soit recombinant (un arbre est dit recombinant si le nombre de noeuds dans chaque tranche croit linéairement avec temps et non pas exponentiellement), on pose

$$ud = 1$$

Dans l'arbre trinomial on définit la probabilité historique de transition ou la matrice de transition  $P$  :

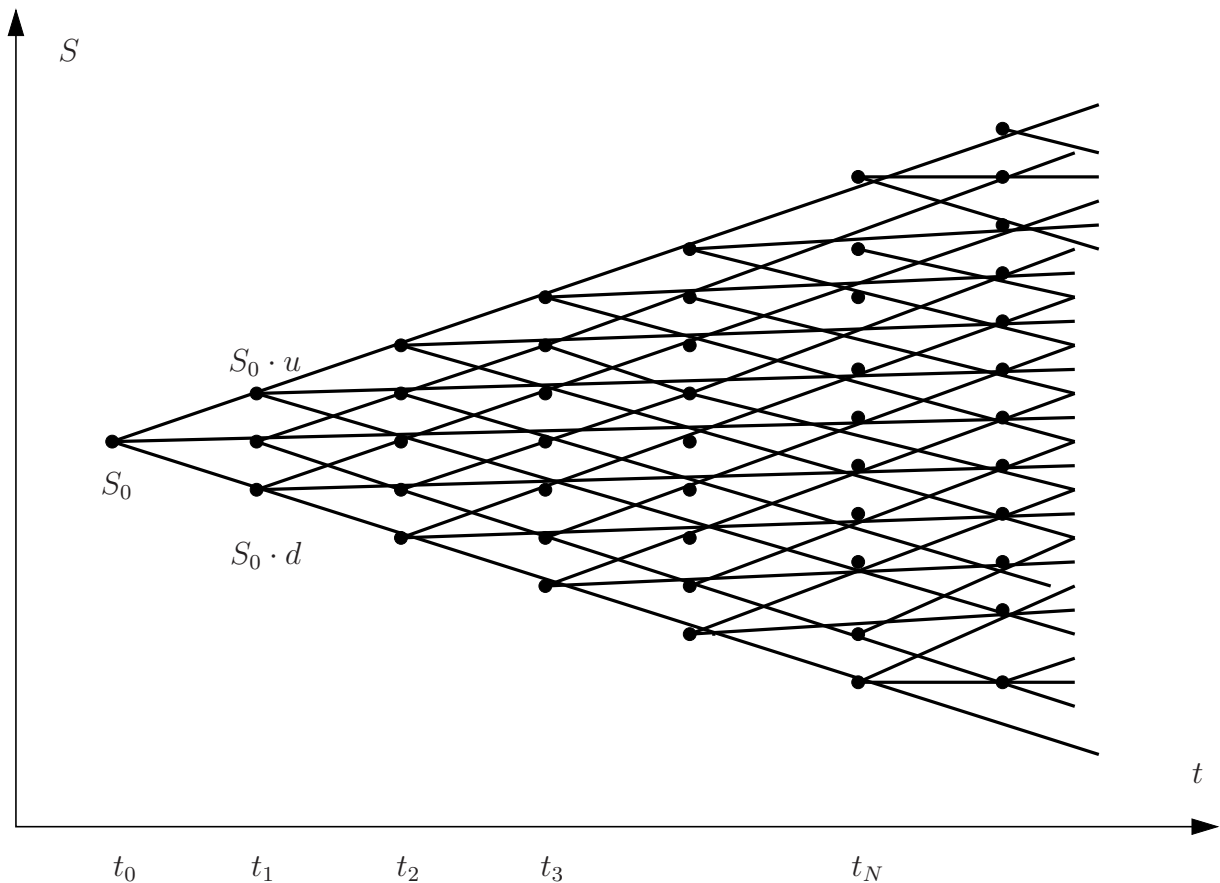
$$\begin{cases} \mathbb{P}[S_{n+1} = u \cdot x / S_n = x] = p \\ \mathbb{P}[S_{n+1} = x / S_n = x] = 1 - p - q \\ \mathbb{P}[S_{n+1} = d \cdot x / S_n = x] = q \end{cases}$$

Introduisons un autre indice "i" pour calculer le prix de l'action à chaque noeud de l'arbre

$$S(n, i) = (u)^{\frac{i}{2}} (d)^{n - \frac{i}{2}} S_0$$

- l'indice  $n$  est l'indice temporel
- l'indice  $i$  est l'indice d'une branche qui définit la valeur de l'action (ou un actif sous-jacent) parmi  $n + 1$  valeurs possibles :

$$0 \leq i \leq 2 \cdot n$$



### 1.3 Option

L'option (Call) Européenne est un contrat avec une banque que donne la possibilité ( pas l'obligation ) d'acheter une action à la date  $t = T$  avec le prix  $K$  bien défini en avance.

$K$  est le prix d'exercice, il s'appelle "strike".  $T$  est le temps d'exercice.

A l'instant  $t_N = T$  le prix de l'option est défini par la formule suivante :

$$V(T, S_N) = f_{\text{pay-off}}(S_N),$$

où  $f_{\text{pay-off}}$  s'appelle la fonction pay-off.

Pour l'option européenne

$$f_{\text{pay-off}}(x) = \max(x - K, 0)$$

Donc

$$f_{\text{pay-off}}(S_N) = \max(S_N - K, 0)$$

Supposons vous achetez au prix  $K$  une action qui vaut  $S_N$  à la date  $t = T$

- Si  $S_N > K$  vous gagnez  $S - K$
- Si  $S_N < K$  vous n'exercez pas le contrat.

#### Exercice 1

Calculer le prix de l'option à  $t = T$  pour chaque  $S_N$  et dessiner l'arbre.

Valeurs numériques :

$$\begin{cases} S_0 = 10, K = 10, u = 2, d = 1/2 \\ p = 1/3, N = 2 \end{cases}$$

Nous cherchons le prix de l'option à  $t = 0$ . C'est-à-dire nous cherchons  $V(0, S_0)$ . Ce prix est représenté par l'espérance conditionnelle :

$$V(0, S_0) = e^{-rT} E[f_{pay-off}(S_N)/S_0]$$

C'est le célèbre théorème fondamental de valorisation des actifs financiers.

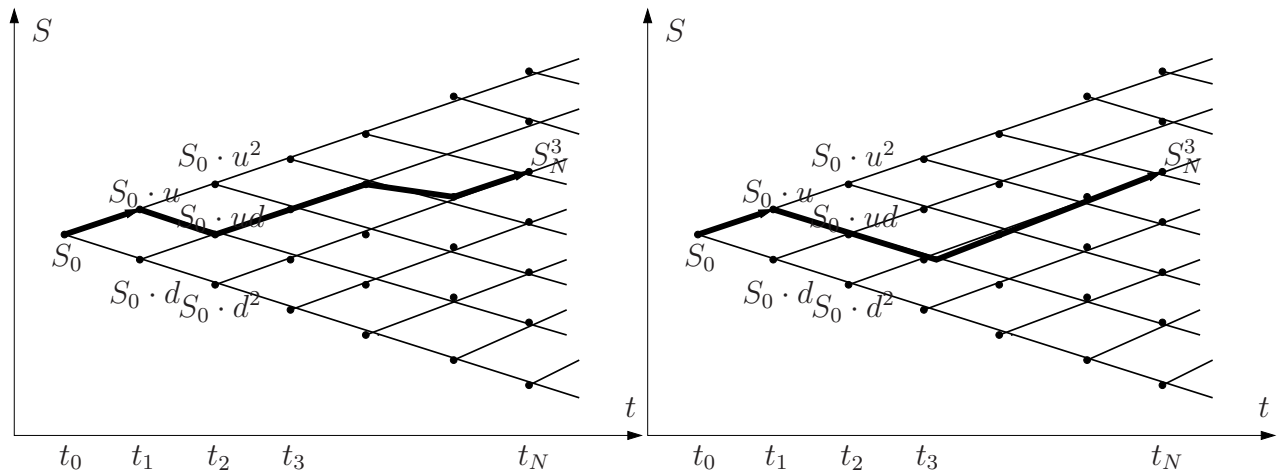
### Exercice 2

Calculer le prix de l'option à  $t = 0$  avec le modèle Binomial.

Valeurs numériques :

$$\begin{cases} S_0 = 10, K = 10, u = 2, d = 1/2 \\ p = 1/3, N = 2 \end{cases}$$

Pour grand arbre (Binomial ou Trinomial) ( $N \rightarrow \infty$ ) il n'est pas simple de calculer les probabilités de chaque branche qui conduit à une  $S_N$ . Par exemple, sur le dessin vous ne voyez que deux branches seulement.



On développe donc DEUX méthodes beaucoup plus élégantes pour calculer le prix des options  $V(0, S_0)$ . Une méthode est stochastique. Elle est basée sur les Chaines de Markov et les simulations de Monte- Carlo.

L'autre est déterministe. Cette méthode est itérative, rapide et elle représente de plus un bon exemple de programmation dynamique.

## 2 PARTIE I. MÉTHODES DE MONTE CARLO PAR LES CHAÎNES DE MARKOV

### 2.1 Convergence des Chaines de Markov

Dans l'arbre binomial on définit la probabilité historique de transition ou la matrice de transition  $P$  :

$$\begin{cases} \mathbb{P}[S_{n+1} = u \cdot x / S_n = x] = p \\ \mathbb{P}[S_{n+1} = d \cdot x / S_n = x] = 1 - p \end{cases}$$

A partir de la matrice de transition on génère des chaînes de Markov  $\{S_0, S_1, S_2, \dots, S_N\}$  qui représentent les prix de l'actif sous-jacent aux instants  $t_0, t_1, t_2, \dots, t_N$ .

Nous allons montrer que la Chaîne de Markov converge vers la loi logonormale. C'est -à-dire  $S_N$  à condition ( $N \rightarrow \infty$ ) suit la loi logonormale :

$$S_N \sim e^{c_1 \mathcal{N}(0,1) + c_2}.$$

Ce résultat est remarquable ! En effet un des postulats du modèle de Black et Scholes est la loi logonormale pour l'actif sous-jacent. Donc nous concluons que la théorie de Chaines de Markov sur l'arbre Binomial se transforme en théorie continue de Black et Scholes. La démonstration est originale et simple.

Nous avons défini le prix de l'action en chaque noeud de l'arbre. C'est la matrice  $S(n, i)$ .

$$S(n, i) = (u)^i (d)^{n-i} S_0$$

Introduisons les variables aléatoires discrètes de Bernoulli  $R_k$  indépendantes.

$$\begin{cases} \mathbb{P}[R_k = 1] = p \\ \mathbb{P}[R_k = 0] = 1 - p \end{cases}$$

Introduisons la variable aléatoire  $S_n$  qui représente un des  $n + 1$  prix possibles à l'instant  $t_n$  :

$$S_n = S_0 u^{(\sum_{k=1}^n R_k)} d^{(n - \sum_{k=1}^n R_k)} = d^n \left(\frac{u}{d}\right)^{(\sum_{k=1}^n R_k)}$$

Donc

$$\ln\left(\frac{S_n}{S_0}\right) = n \ln(d) + \ln\left(\frac{u}{d}\right) \sum_{k=1}^n R_k$$

Du théorème de Limite Centrale

$$\lim_{n \rightarrow \infty} \frac{\sum_{k=1}^n R_k - np}{\sqrt{pn(1-p)}} \rightarrow N(0, 1)$$

On en déduit que que pour grand  $n$

$$\ln\left(\frac{S_n}{S_0}\right) = n(\ln(d) + p \ln\left(\frac{u}{d}\right)) + \sqrt{n} \sqrt{p(1-p)} N(0, 1) \ln\left(\frac{u}{d}\right)$$

donc

$$S_n = S_0 e^{n(\ln(d) + p \ln(\frac{u}{d})) + \mathcal{N}(0,1)(\sqrt{n} \sqrt{p(1-p)} \ln(\frac{u}{d}))}$$

Dans le modèle de Black et Scholes

$$S_t = S_0 e^{(r-\sigma^2/2)t + \sigma W_t}$$

Ici  $W_t$  est la valeur finale du mouvement Brownien qui suit la loi Normale  $\mathcal{N}(0, t)$ . Donc à chaque instant  $t_n = \Delta t \cdot n$

$$S_n = S_0 e^{(r-\sigma^2/2)n\Delta t - \sigma\sqrt{n\cdot\Delta t}\mathcal{N}(0,1)}$$

Maintenant on peut calibrer l'arbre, c'est à dire on peut établir des relations entre les paramètres de l'arbre  $u, d, p$  et paramètres financiers : le taux d'intérêt  $r$  et la volatilité  $\sigma$ .

En effet par identification on obtient deux relations

$$\ln(d) + p \ln\left(\frac{u}{d}\right) = \left(r - \frac{\sigma^2}{2}\right)\Delta t$$

$$\sqrt{p(1-p)} \ln\left(\frac{u}{d}\right) = \sigma\sqrt{\Delta t}$$

On fixe aussi  $p = 1/2$ . En résolvant le système linéaire on obtient

$$u = e^{\Delta t(r-\frac{\sigma^2}{2})+\sigma\sqrt{\Delta t}}, \quad d = e^{\Delta t(r-\frac{\sigma^2}{2})-\sigma\sqrt{\Delta t}}.$$

Cette calibration a été proposé par Desmond J. Higham (DH). Il existe d'autres méthodes de calibration basées sur le calcul des intégrales gaussiens. Nous allons les présenter plus tard.

## 2.2 Prix de l'option par Monte Carlo

Nous cherchons le prix de l'option à  $t = 0$

$$V(0, S_0) = e^{-rT} E[f_{pay-off}(S_N) | S_0]$$

Le prix d'une option européenne au moment de temps  $t$  est donné par l'espérance conditionnelle d'une fonction pay-off  $f_{pay-off} = \max(S(T) - K, 0)$

$$V(S_0, 0) = e^{-rT} \mathbb{E}[\max(S(T) - K, 0) | S(0) = S_0] \quad (1)$$

On peut expliquer ce résultat de façon suivante : on simule un grand nombre de chaînes d'évolution de l'actif  $S(t)$  sur l'intervalle de temps  $[t, T]$ , en partant toujours de  $S_0$ . Pour chaque chemin on cherche la valeur finale  $S(T)$ . Puis on calcule la moyenne arithmétique de gains, c'est-à-dire de  $\Lambda(S(T)) = \max(S(T) - K, 0)$ . Le facteur  $e^{-rt}$  exprime le fait qu'une bank rembourse les intérêt sur l'intervalle du temps  $T - t$ .

En effet une des hypothèses fondamentales des modèles usuels est qu'il n'existe aucune stratégie financière permettant

- 1. sans investissement net de capital,
- 2. sans aucun risque,

d'acquérir une richesse certaine dans une date future. Cette hypothèse est appelée absence d'opportunités d'arbitrage.

- Un actif sans risque (obligation) produit des intérêts

$$B_T = e^{rT} B_0$$

- Un actif risqué devrait en moyenne produire la même performance qu'un investissement dans l'actif sans risque :

$$e^{rT} \cdot V(0, S_0) = \mathbb{E}[Pay\_off(S_T)|S_0] \Rightarrow V(0, S_0) = e^{-rT} \cdot \mathbb{E}[Pay\_off(S_T)|S_0]$$

Nous allons maintenant utiliser la loi de Grands Nombres et de remplacer l'espérance par la somme dans la limite  $N_{mc} \rightarrow \infty$

$$V(S_0, 0)_{estimate} = \frac{e^{-rT}}{N_{mc}} \sum_{k=1}^{N_{mc}} max(S_T^{(k)} - K, 0).$$

Pour trouver les valeurs finales des actifs simulons des Chaînes de Markov et récupérons pour chaque chaîne sa valeur finale  $S_N^{(k)}$ . On souligne que l'indice  $k$  est introduit pour compter les chaînes de Markov :  $k = 1 : N_{mc}$ . En effet simulons trois Chaînes :

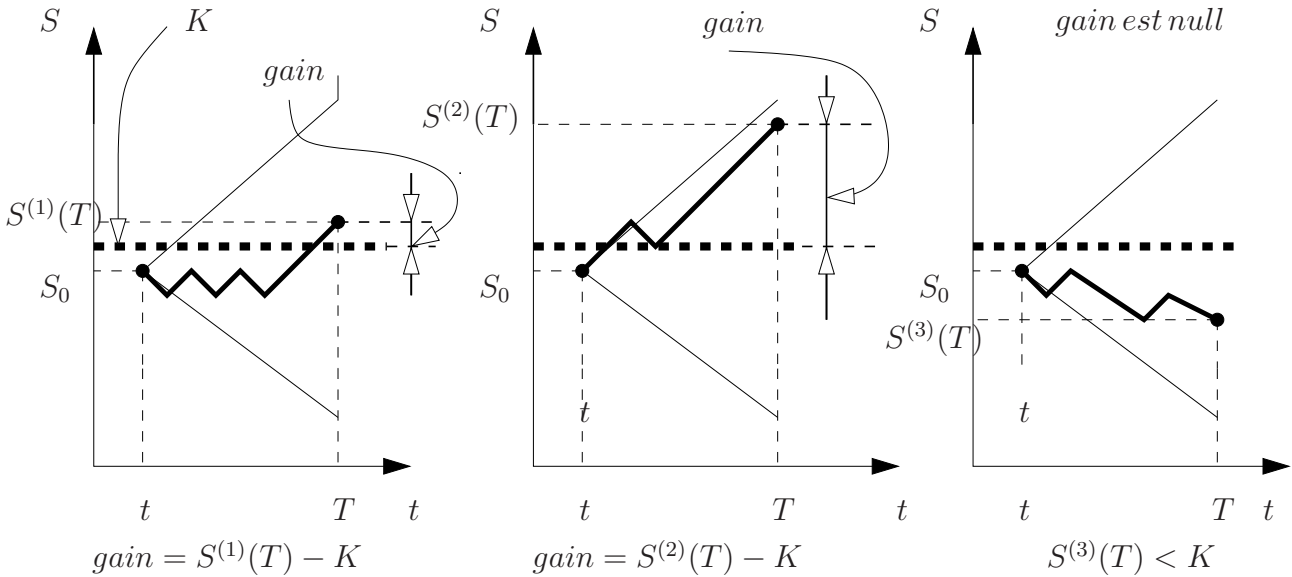


FIGURE 1 – Bases de la méthode de Monté-Carlo

Le gain moyen ( c'est le prix  $V$  de l'option) est

$$\frac{1}{3}((S^{(1)}(T) - K) + (S^{(2)}(T) - K) + 0).$$

Le prix  $V$  de l'option est

$$e^{-rT} \left( \frac{1}{3}((S^{(1)}(T) - K) + (S^{(2)}(T) - K) + 0) \right)$$

**En conclusion le prix de l'option européenne au moment  $t = 0$  sachant le prix de l'actif  $S_0$  à cet instant est donné par la moyenne arithmétique**

$$V(S_0, 0)_{estimate} = \frac{e^{-rT}}{N_{mc}} \sum_{k=1}^{N_{mc}} max(S_T^{(k)} - K, 0).$$

## 2.3 Calibration des arbres. Arbre Binomial.

La probabilité d'un saut en hauteur  $p$  et la taille des sauts  $u$  et  $d$  sont choisies de manière à ce que la Chaîne de Markov et la marche aléatoire continue aient la même espérance et la même variance (on seconde moment). C'est-à-dire que, étant donné que la valeur de l'actif est  $S_n$  au pas de temps  $t_n$ , nous assimilons les valeurs de l'espérance et de la variance attendues de  $S_{n+1}$  dans les modèles de marche aléatoire continue et de Markov discret. Par ces deux conditions on assure que Modèle discrete des Chaînes de Markov approche dans la limite  $\Delta t \rightarrow 0$  ou ( $N \rightarrow \infty$ ) le modèle continue de Black et Scholes.

- Dans le modèle Binomial

$$\begin{aligned}\mathbb{E}[S_{n+1}|S_n] &= (pu + (1-p)d)S_n \\ \mathbb{E}[(S_{n+1})^2|S_n] &= (pu^2 + (1-p)d^2)(S_n)^2\end{aligned}$$

- Dans le modèle de Black et Scholes

$$S_{n+1} = S_n e^{(r-\sigma^2/2)\Delta t + \sigma W_{\Delta t}}$$

ou  $W_{\Delta t}$  est le mouvement Brouwnien qui suit la loi  $N(0, \Delta t)$  et

$$E[S_{n+1}|S_n] = S_n e^{r\Delta t}$$

De la théorie de probabilité l'espérance conditionnelle s'exprime à l'aide de la fonction de densité :

$$E[S_{n+1}|S_n] = S_n \int_{-\infty}^{+\infty} e^{(r-\sigma^2/2)\Delta t + \sigma x} f_W(x) dx$$

$f_W(x)$  est la fonction de densité de Mouvement Brouwnien

$$f_W(x) = \frac{1}{\sqrt{2\pi\Delta t}} e^{-\frac{x^2}{2\Delta t}}.$$

Il reste de calculer l'intégrale. Nous avons aussi deuxième relation :

$$E[(S_{n+1})^2|S_n] = (S_n)^2 e^{2r\Delta t + \sigma^2\Delta t}$$

Finalement on arrive au système de deux equations avec trois inconnues :

$$\begin{cases} pu + (1-p)d = e^{r\Delta t} \\ pu^2 + (1-p)d^2 = e^{2r\Delta t + \sigma^2\Delta t} \end{cases}$$

Il faut imposer une condition supplémentaire. Il existe plusieurs choix.

- **Calibration de Cox, Ross, Rubinstein (CRR).**

Pour avoir l'arbre recombinaut (ce que concerne surtout l'arbre trinomial) on impose souvent  $ud = 1$ .

Nous avons maintenant 3 inconnus et 3 equations et on peut trouver le résultat suivant :

$$\begin{cases} A = \frac{1}{2}(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}) \\ u = A + \sqrt{A^2 - 1} \\ d = A - \sqrt{A^2 - 1} \\ p = \frac{e^{r\Delta t} - d}{u - d} \end{cases}$$

- **Calibration de Jarrow-Rudd (JR).** On choisit  $p = 1/2$ .  
Suite aux simples manipulations on obtient

$$\begin{cases} u = e^{r\Delta t}(1 - \sqrt{e^{\sigma^2\Delta t} - 1}) \\ d = e^{r\Delta t}(1 + \sqrt{e^{\sigma^2\Delta t} - 1}) \\ p = 1/2 \end{cases}$$

## 2.4 Calibration des arbres. Arbre Trinomial.

La famille des paramètres très populaire est la suivantes :

$$\begin{cases} u = e^{\sigma\sqrt{2\Delta t}} \\ d = e^{-\sigma\sqrt{2\Delta t}} \\ A = e^{\frac{r\Delta t}{2}}; \quad B = e^{-\sigma\sqrt{\frac{\Delta t}{2}}} \\ p = \left(\frac{A-B}{1/B-B}\right)^2 \\ q = \left(\frac{1/B-A}{1/B-B}\right)^2 \end{cases}$$

## 3 Travail à faire en utilisant la méthode "Monte-Carlo par Chaînes de Markov"

Pour les calculs numériques utiliser les valeurs suivantes :

$$\begin{cases} K = 10 \\ N = 20 \quad \text{ou} \quad N = 100 \\ \sigma = 0.5, r = 0.1 \\ T = 0.5 \end{cases}$$

### 3.1 Calibration de l'arbre Binomial par 3 méthodes

- 1. Montrer les formules des CCR et vérifier que  $p \leq 1$
- Calculer l'intégrale gaussienne et montrer :

$$\begin{cases} ud = 1 \\ A = \frac{1}{2}(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}) \\ u = A + \sqrt{A^2 - 1} \\ d = A - \sqrt{A^2 - 1} \\ p = \frac{e^{r\Delta t} - d}{u - d} \end{cases}$$

- 2. Montrer les formules des JR :

$$\begin{cases} p = 1/2 \\ u = e^{r\Delta t}(1 - \sqrt{e^{\sigma^2\Delta t} - 1}) \\ d = e^{r\Delta t}(1 + \sqrt{e^{\sigma^2\Delta t} - 1}) \end{cases}$$

- 3. Répéter la démonstration de DH.

### 3.2 Simulation des Chaînes de Markov sur l'arbre Binomial

- Tracer l'arbre binomial  $S(n, i)$  (les points) en **SCILAB** ou **Matlab** pour les deux familles des paramètres :  $u, d, p, q$  (CCR et DH).

- Simuler les Chaînes de Markov de l'actif.

Pour cela

- Discrétiser l'intervalle  $[0, T]$  sur  $N$  parties :  $(\Delta t = T/N, t_n = n \Delta t)$ .

$$S_{t_0} = S_0, \quad S_{t_n} = S_n, \quad S_{t_N} = S(T)$$

On répète que l'évolution de l'actif sous-jacent  $S$  est un processus stochastique  $S_t, t \in [0, T]$  Markovien. A un instant  $t_{n+1}$  l'actif  $S_{n+1}$  n'admet que 2 valeurs possibles avec les probabilités connues :

$$S_{n+1} = \begin{cases} u \cdot S_n \\ d \cdot S_n \end{cases}$$

- Fixer  $S_0$ .

- Simuler  $S_1$  sachant que  $S_1$  est une variable aléatoire discrete que ne prenne que deux valeurs :

$S_1$	$uS_0$	$dS_0$
$p$	1/2	1/2

- Simuler  $S_2, S_3, \dots, S_N$  en s'inspirant du cours MCMC pour  $N=10$ .
- Tracer une Chaîne  $t_n \rightarrow S_n$  et superposer ce graphe sur l'arbre pour  $N=10$ .
- Tracer plusieurs Chaînes et superposer ces graphes sur l'arbre pour  $N=10$ .

### 3.3 Calcul de prix de l'option dans le modèle Binomial

- Créer une fonction pay-off de l'option Européenne, Call :

$$\left\{ \begin{array}{l} \text{function}[f]= \text{Payoff\_Europ\_Call}(S) \\ f = \max(S - K, 0) \\ \text{endfunction} \end{array} \right.$$

- Créer une fonction qui génère une Chaîne de Markov et retourne sa valeur finale.

- Créer une fonction qui calcule le prix du Call de l'option Européenne pour  $S_0$  fixe. Utiliser  $N = 100, N_{mc} = 100$  et  $N_{mc} = 1000$ .

$$\left\{ \begin{array}{l} \text{function}[prix]= \text{Prix\_Europ\_S0\_fixe}(S_0) \\ \quad \text{sum} = 0 \\ \quad \text{for } n = 1 : N_{mc} \\ \quad \quad S = \{ \text{Appler la fonction qui genere la valeur finale d'une Chaîne de Markov} \} \\ \quad \quad \text{sum} = \text{sum} + \text{Payoff\_Europ\_Call}(S) \\ \quad \quad \text{end for} \\ \quad \text{prix} = e^{-rT} \text{sum} / N_{mc} \\ \quad \text{endfunction} \end{array} \right.$$

- Calculer le prix du Call pour  $S_0 = 10$ .
- Créer la fonction qui calcule le prix de l'option Européenne pour  $S_0$  quelconque, par exemple pour chaque valeur de  $S_0 = 0, 0.5, 1, 1.5, 2, \dots, 20$ .

$$\left\{ \begin{array}{l} \text{fonction [prix\_option]=Prix\_Call\_Europ( )} \\ \quad \text{for } k = 1 : 41 \\ \quad \quad S0(k) = (k - 1) \cdot 0.5 \\ \text{prix\_option(k) = Prix\_Europ\_S0\_fixe(S0(k))} \\ \quad \text{end} \\ \quad \text{plot(S0,prix\_option)} \\ \text{endfonction} \end{array} \right.$$

- Tracer le graphe du prix du Call de l'option Européenne :  $S_0 \rightarrow V(t = 0, S_0)$  pour  $N_{mc} = 100$ ,  $N_{mc} = 1000$  et  $N_{mc} = 10000$   
Utiliser `plot (S0,prix_option)`.

### 3.4 Calcul de prix de l'option à l'aide de l'arbre Trinomial

#### 3.4.1 Calibration de l'arbre Trinomial

Verifier les formules :

$$\left\{ \begin{array}{l} u = e^{\sigma\sqrt{2\Delta t}} \\ d = e^{-\sigma\sqrt{2\Delta t}} \\ A = e^{\frac{r\Delta t}{2}}; \quad B = e^{-\sigma\sqrt{\frac{\Delta t}{2}}} \\ p = \left( \frac{e^{\frac{r\Delta t}{2}} - e^{-\sigma\sqrt{\frac{\Delta t}{2}}}}{e^{\sigma\sqrt{\frac{\Delta t}{2}}} - e^{-\sigma\sqrt{\frac{\Delta t}{2}}}} \right)^2 \\ q = \left( \frac{e^{\sigma\sqrt{\frac{\Delta t}{2}}} - e^{\frac{r\Delta t}{2}}}{e^{\sigma\sqrt{\frac{\Delta t}{2}}} - e^{-\sigma\sqrt{\frac{\Delta t}{2}}}} \right)^2 \end{array} \right.$$

Pour cela  
 ◦ Calculer

$$E[S_{n+1}|S_n]$$

dans le modèle trinomial et l'identifier avec  $S_n e^{r\Delta t}$  obtenu de modèle Black et Scholes.

◦ Calculer  $E[(S_{n+1})^2|S_n]$  dans le modèle trinomial et l'identifier avec  $e^{2r\Delta t + \sigma^2\Delta t}$  obtenu de modèle Black et Scholes.

◦ Verifier que la famille  $(p, q, u, d)$  satisfait jusqu'au l'ordre  $(\Delta t)^2$  relations obtenues dans les deux questions précédentes.

◦ Indication : utiliser le développement limité d'ordre  $(\Delta t)^2$  .

3.4.2 Répéter toutes les simulations des Chaines de Markov

3.4.3 Répéter les calculs des prix de l'option Européenne

3.4.4 Comparer les résultats

## 4 PARTIE II. PROGRAMMATION DINAMIQUES SUR LES ARBRES

### 4.1 Equation dynamique

Il est important de comprendre que  $\forall$  fonction  $f$  bornée l'espérance conditionnelle

$$E[f(S_{n+1})/S_n = x] = \sum_{y \in \{xu, xd\}} P(x, y) f(y)$$

Donc

$$E[f(S_{n+1})/S_n = x] = pf(u \cdot x) + (1 - p)f(d \cdot x)$$

**Pour pouvoir implémenter numériquement la méthode** introduisons une fonction

$$v(n, x) : \mathbb{N} \times \mathbb{R}_+ \rightarrow \mathbb{R}$$

avec les propriétés

$$\begin{cases} v(N, y) = f_{\text{pay-off}}(y) \\ v(n, x) = e^{-r\Delta t} \sum_{y \in \{xu, xd\}} P(x, y) v(n+1, y) \end{cases}$$

L'équation

$$v(n, x) = e^{-r\Delta t} \sum_{y \in \{xu, xd\}} P(x, y) v(n+1, y)$$

s'appelle l'équation dynamique pour l'arbre binomial. On en déduit que

$$e^{-r(T-t_n)} E[f_{\text{pay-off}}(S_N)/S_n] = v(n, S_n)$$

et

$$e^{-rT} E[f_{\text{pay-off}}(S_N)/S_0] = v(0, S_0)$$

**Alors le prix de l'option à  $t = 0$**

$$V(0, S_0) = v(0, S_0)$$

Avant de faire la démonstration il est important de comprendre que  $\forall$  fonction  $f$  bornée l'espérance conditionnelle

$$E[f(S_{n+1})/S_n = x] = \sum_{y \in \{xu, xd\}} P(x, y) f(y)$$

Donc

$$E[f(S_{n+1})/S_n = x] = pf(u \cdot x) + (1 - p)f(d \cdot x)$$

## Démonstration

Soit  $x = S_n$  et par l'équation dynamique

$$v(t_n, S_n) = e^{-r\Delta t}(p v(t_{n+1}, uS_n) + (1-p) v(t_{n+1}, dS_n))$$

Les prix à la maturité sont connus :

$$v(t_N, S_N) = \max(S_N - K, 0)$$

Montrons que

$$v(0, S_0) = e^{-rT} E[f_{\text{pay-off}}(S_N)/S_0]$$

En effet on applique d'abord l'équation dynamique, puis la définition de l'espérance conditionnelle :

$$\begin{aligned} v(t_{N-1}, S_{N-1}) &= e^{-r\Delta t}(p v(t_N, uS_{N-1}) + (1-p) v(t_N, dS_{N-1})) = \\ &= e^{-r\Delta t} E[v(t_N, S_N)/S_{N-1}] = e^{-r\Delta t} E[f_{\text{pay-off}}(S_N)/S_{N-1}] \end{aligned}$$

$$\begin{aligned} v(t_{N-2}, S_{N-2}) &= e^{-r\Delta t}(p v(t_{N-1}, uS_{N-2}) + (1-p) v(t_{N-1}, dS_{N-2})) = e^{-r\Delta t} E[v(t_{N-1}, S_{N-1})/S_{N-2}] = \\ &= e^{-2r\Delta t} E[E[f_{\text{pay-off}}(S_N)/S_{N-1}]/S_{N-2}] = e^{-2r\Delta t} E[f_{\text{pay-off}}(S_N)/S_{N-2}] \end{aligned}$$

On continue le raisonnement jusqu'au  $t = 0$

## 4.2 Algorithme

**A chaque noeud de l'arbre binomial caractérisé par les indices  $(n, i)$  on associe le prix de l'actif  $S(n, i)$  et le prix de l'option  $v(n, i)$ .**

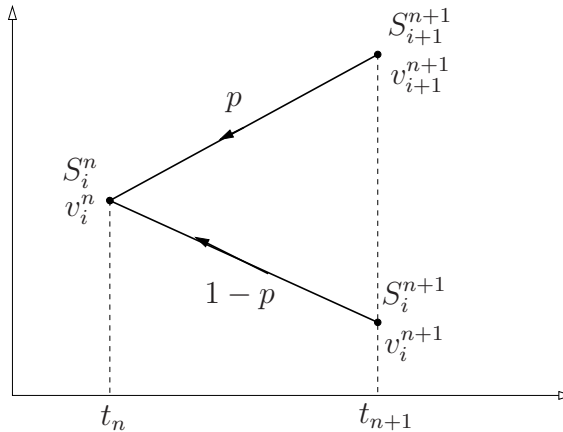
Explicitons l'équation dynamique

$$v(n, x) = e^{-r\Delta t}(p v(n+1, xu) + (1-p) v(n+1, xd))$$

On remplace

$x = S_i^n$  par l'indice  $i$  qui définit le prix de l'option à l'instant  $t_n$  et

on remplace  $xu = S_{i+1}^{n+1}$  et  $xd = S_i^{n+1}$  par les indices  $i+1$  et  $i$  qui définissent les prix de l'option à l'instant  $t_{n+1}$ .



On obtient

$$v(n, i) = e^{-r\Delta t}(pv(n + 1, i + 1) + (1 - p)v(n + 1, i))$$

On applique cette équation en remontant le temps :

1) On définit le prix de l'option à la maturité :

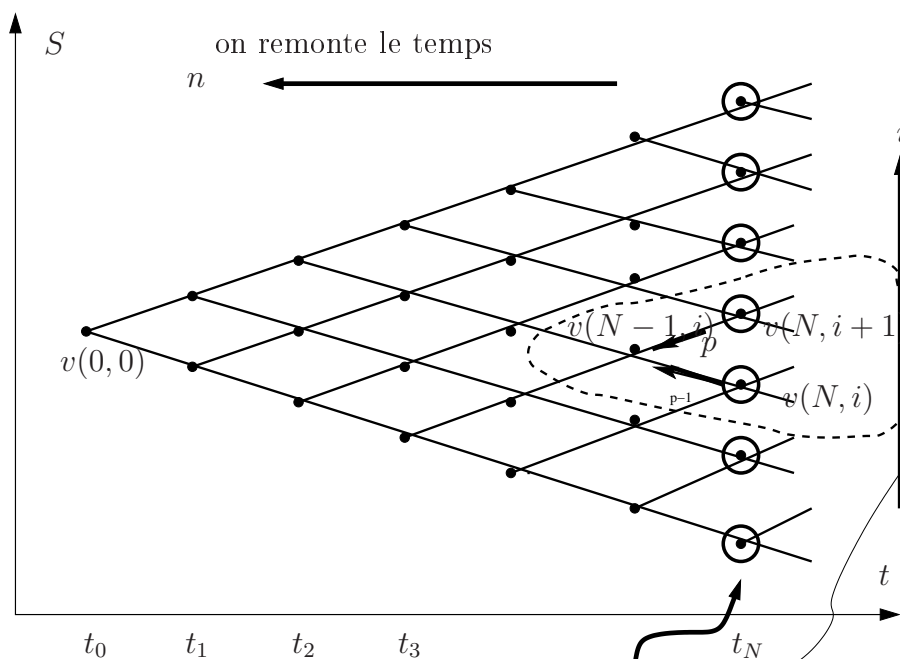
$$v(N, i) = f_{pay-off}(N, i) \quad i = 0 : N$$

Pour le Call Européen

$$f_{pay-off}(N, i) = \max(S(N, i) - K, 0).$$

2) On forme deux boucles

$$\left\{ \begin{array}{l} \text{for } n = N - 1 : -1 : 0 \\ \quad \text{for } i = 0 : 1 : n \\ \quad \quad v(n, i) = e^{-r\Delta t}(p \cdot v(n + 1, i + 1) + (1 - p) \cdot v(n + 1, i)) \\ \quad \quad \text{end} \\ \text{end} \end{array} \right.$$



Calcul du prix des options sur chaque niveau temporel a partir de la maturité

En effet, on tourne l'algorithme à la main :

$$\text{si } n = N - 1, \quad i = 0 \quad v(N - 1, 0) = e^{-r\Delta t}(p \cdot v(N, 1) + (1 - p) \cdot v(N, 0))$$

$$\text{si } n = N - 1, \quad i = 1 \quad v(N - 1, 1) = e^{-r\Delta t}(p \cdot v(N, 2) + (1 - p) \cdot v(N, 1))$$

...

$$\text{si } n = N - 1, \quad i = N - 1 \quad v(N - 1, N - 1) = e^{-r\Delta t}(p \cdot v(N, N) + (1 - p) \cdot v(N, N - 1))$$

Les prix  $v(N, 0), v(N, 1), \dots, v(N, N)$  sont connus (prix à la maturité). On a calculé donc tout les prix de l'option à l'instant  $t_{N-1}$ . On pose puis  $n = N - 2$  et on continue vers  $n = 0$ .

### 4.3 Grecques

A l'aide du modèle Binomial on peut calculer les Grecques. Par exemple, Delta de l'option Européenne est définie comme

$$\Delta = \frac{\partial V}{\partial S}$$

Cette notion est très importante pour les traders.

- $\Delta_{n+1} - \Delta_n$  exprime la quantité des actions qu'il faut acheter ( si  $\Delta_{n+1} - \Delta_n > 0$ ) entre les moments  $t_{n+1}$  et  $t_n$  ou vendre ( si  $\Delta_{n+1} - \Delta_n < 0$ ) pour couvrir une option ( protéger une option). En anglais cette procédure s'appelle Hedging.

Gamma de l'option Européenne est définie comme

$$\Gamma = \frac{\partial^2 V}{\partial S^2}$$

### 4.4 Bond zero-coupon ou l'obligation

Une obligation  $P(t, T)$  est un contrat signé à un instant  $t$  délivrant 1 euro à la date  $t = T$ .

$$P(T, T) = 1$$

Parmi les obligations, seules les zéro-coupon permettent d'éliminer réellement tout risque de taux entre deux dates  $t$  et  $T$ .

Notre but est de trouver le prix du Bond zero-coupon  $P(t, T)$  à l'instant  $t$

- Si le taux d'intérêt est une constante  $r_t = r$  alors

$$P(t, T) = e^{-r(T-t)}$$

- Si le taux d'intérêt est une fonction déterministe du temps  $r_t$  alors

$$P(t, T) = e^{-\int_t^T r_s ds}$$

- Si le taux d'intérêt est un processus stochastique alors

$$P(t, T) = E[e^{-\int_t^T r_s ds} | r(t) = r]$$

## 5 Travail à faire en utilisant la méthode "Programmation dynamique"

Pour les calculs numériques utiliser les valeurs suivantes :

$$\left\{ \begin{array}{l} K = 10 \\ N = 20 \quad \text{ou} \quad N = 100 \\ \sigma = 0.5, r = 0.1 \\ T = 0.5 \end{array} \right.$$

## 5.1 Pricing de l'Option Européenne.

### 5.1.1 Idées

- Définir les prix de l'action  $\forall i$  à l'instant  $t_N = T$ , c'est-à-dire à la maturité.

$$S(N, i) = S_0(u)^i(d)^{N-i}$$

- Définir la fonction pay-off et les prix de l'option à  $t = T$

$$v(N, i) = f_{\text{pay-off}}(S(N, i)) = \max(S(N, i) - K, 0).$$

- Introduire la matrice  $v(n, i)$ .

Utiliser l'équation dynamique

$$v(n, i) = e^{-r\Delta t}(pv(n+1, i+1) + (1-p)v(n+1, i))$$

et programmer en remontant le temps : faire deux boucles par rapport à "n" et par rapport à "i".

- Trouver  $V(0, S_0) \equiv v(0, 0)$  le prix de l'option européenne.
- Trouver  $V(0, S_0)$  le prix de l'option européenne pour  $p = 1/3$  et comparer avec le résultat précédent.
- Ajouter une boucle supplémentaire qui va varier  $S_0$  et tracer le graphe

$$S_0 \rightarrow V(0, S_0)$$

### 5.1.2 Implémentation

- On crée une fonction qui calcule le prix de l'option Européenne pour une valeur fixe  $S_0$  : **fonction[prix]=option-europ-S0-fixe(S0)** avec le paramètre de sortie **prix**.
- On fixe les paramètres :  $r, N, T, p, u, d, K, \delta t = T/N$ .

$$u = e^{\Delta t(r - \frac{\sigma^2}{2}) + \sigma\sqrt{\Delta t}}, \quad d = e^{\Delta t(r - \frac{\sigma^2}{2}) - \sigma\sqrt{\Delta t}}$$

$$\left\{ \begin{array}{l} \text{for } i = 0 : N \\ S(N, i) = S_0(u)^i(d)^{N-i}, \\ v(N, i) = \max(S(N, i) - K, 0) \\ \text{end} \end{array} \right.$$

- Finalement on définit la fonction **function[prix0]=option-europ-S0-fixe(S0)** et on applique équation dynamique en remontant le temps :

$$\left\{ \begin{array}{l} \mathbf{function[prix0]} = \text{option-europ-S0-fixe}(S_0). \\ \text{definitions des parametres} \\ \text{definitions des valeurs de l'actif et de l'option a la matutite} \\ \mathbf{for} \quad \mathbf{n} = \mathbf{N} - \mathbf{1} : -\mathbf{1} : \mathbf{0} \\ \quad \mathbf{for} \quad i = 0 : 1 : n \\ \quad \quad v(n, i) = e^{-r\Delta t}(p \cdot v(n + 1, i + 1) + (1 - p) \cdot v(n + 1, i)) \\ \quad \quad \mathbf{end} \\ \quad \quad \mathbf{end} \\ \quad \mathbf{prix0} = v(0, 0) \\ \mathbf{endfunction} \end{array} \right.$$

- On cherche **prix0=v(0,0)** pour  $S_0 = 10$ . Pour cela on appelle la fonction par **option-europ-S0-fixe(10)**.

En Scilab (Matlab) l'indice d'une matrice ne peut pas être égale à zero. On doit donc commencer l'indice  $n$  de 1 (finir par  $N + 1$ ) et l'indice  $i$  de 1 :

$$n = 1 : N + 1 \quad i = 1 : n$$

Donc on obtient

$$\left\{ \begin{array}{l} \mathbf{function[prix0]} = \text{option-europ-S0-fixe}(S_0). \\ \text{definitions des parametres} \\ \text{definitions des valeurs de l'actif et de l'option a la matutite} \\ \mathbf{for} \quad \mathbf{n} = \mathbf{N} : -\mathbf{1} : \mathbf{1} \\ \quad \mathbf{for} \quad i = 1 : 1 : n \\ \quad \quad v(n, i) = e^{-r\Delta t}(p \cdot v(n + 1, i + 1) + (1 - p) \cdot v(n + 1, i)) \\ \quad \quad \mathbf{end} \\ \quad \quad \mathbf{end} \\ \quad \mathbf{prix0} = v(1, 1) \\ \mathbf{endfunction} \end{array} \right.$$

- Créer la fonction qui calcule le prix de l'option Européene pour chaque valeur de  $S_0$  fonction **[price-option]=option-europ()** avec le vecteur de sortie **price-option**.

$$\left\{ \begin{array}{l} \mathbf{function} \text{ [price-option]=option-europ}() \\ \quad \mathbf{for} \quad k = 1 : 41 \\ \quad \quad S_0(k) = 0.5 * (k - 1) \\ \quad \quad \text{price-option}(k) = \text{option-europ-S0-fixe}(S_0(k)) \\ \quad \quad \mathbf{end} \\ \quad \mathbf{plot}(S_0, \text{price-option}) \quad (\mathbf{Plot en SCILAB}) \\ \quad \quad \mathbf{xlabel} \quad 'prix de S_0' \\ \quad \quad \mathbf{ylabel} \quad 'prix de loption' \\ \quad \quad \mathbf{title} \quad 'Prix de loption Europeenne' \\ \quad \mathbf{endfunction} \end{array} \right.$$

Remarquez que le paramètre de sortie **price-option** est un vecteur. **S0** l'est aussi. De même taille. Il est légitime donc d'utiliser **plot**.

- Comparer les résultats des prix avec ceux obtenue par MCMC.
- Calculer le prix de l'option  $v(t, S_t)$  à un instant quelconque  $t$ . Réfléchissez comment calculer ces prix. Tracer une surface de prix de l'option pour un quelconque  $t_n$  et une quelconque valeur de l'actif  $S_t$ . Utiliser la fonction Scilab **surf**.

Indication. Créer un nouveau arbre de longueur  $T - t$  avec un sommet qui commence de  $S_t$  et définir le paramètre  $\Delta t = \frac{T-t}{N}$

## 5.2 Option Américaine.

Une personne qui possède une option américaine peut l'exercer à n'importe quel moment  $t_n \in [0, T]$ . Donc la fonction pay-off s'écrit de la forme

$$Call_{pay-off} = \max(S(n, i) - K, 0)$$

$$Put_{pay-off} = \max(K - S(n, i), 0)$$

A chaque noeud d'arbre on doit implémenter pour chaque  $t_n$  une fonction de comparaison entre la valeur intrinsèque du pay-off et la valeur de l'option obtenues par la programmation dynamique.

$$v_{am}(n, i) = \max(Put_{pay-off}(S(n, i)), e^{-r\Delta t}(pv_{am}(n+1, i+1) + (1-p)v_{am}(n+1, i)))$$

$$v_{am}(n, i) = \max(\max(K - S(n, i), 0), e^{-r\Delta t}(pv_{am}(n+1, i+1) + (1-p)v_{am}(n+1, i)))$$

- Trouver le prix de l'option Américaine Call, Put et tracer le graphe

$$S_0 \rightarrow V_{americaine}(0, S_0)$$

- Tracer dans le même système de coordonnées les options Américaines Call, Put et les option Européennes pour les comparer.

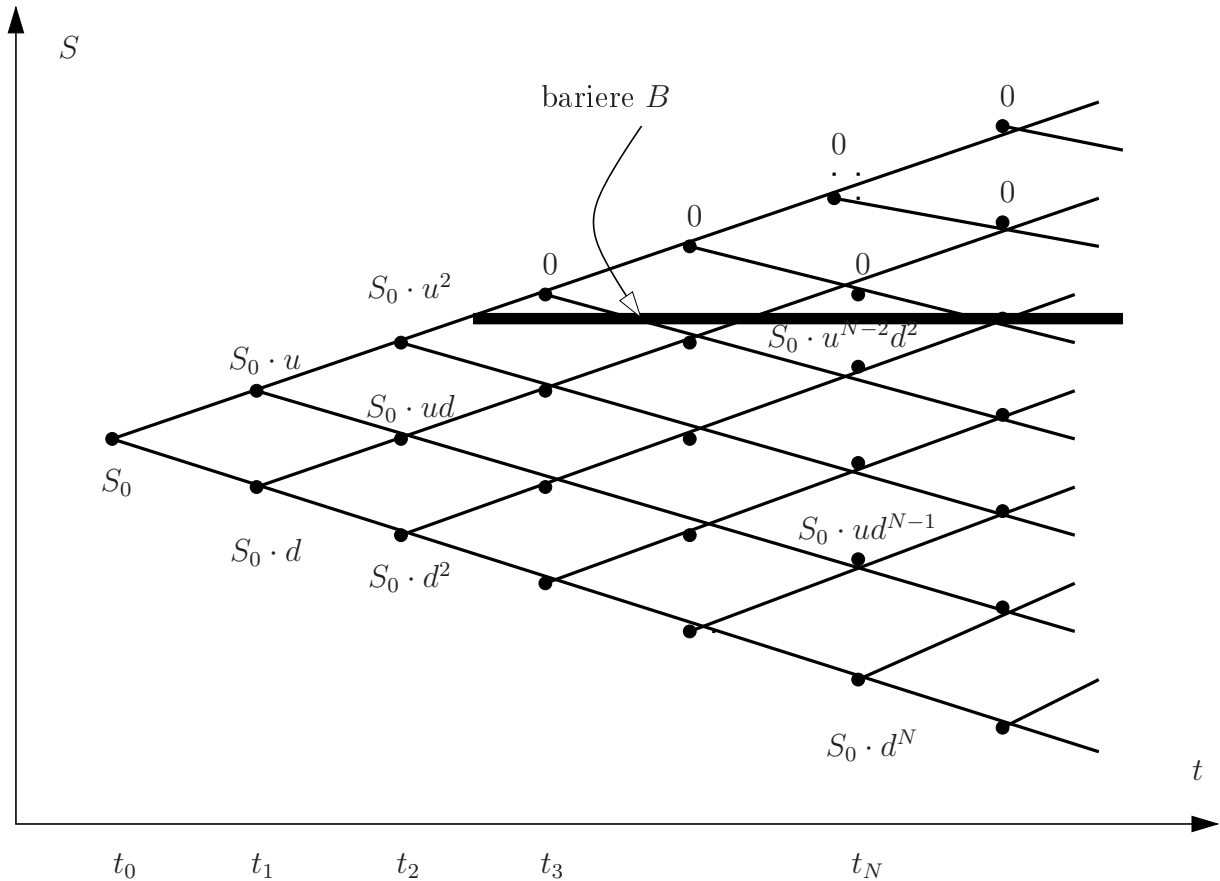
- Calculer le prix de l'option Américaines Put  $v(t, S_t)$  à un instant quelconque  $t$ . Tracer une surface de prix de l'option pour un quelconque  $t_n$  et une quelconque valeur de l'actif  $S_i$ . Utiliser la fonction Scilab **surf**.

### 5.3 Option Exotique : Barrière Knock-Out.

On définit l'option barrière de la façon suivante : Si  $S(t_n)$  touche une Barrière  $B$  :

$$S(t_n) \geq B$$

la vie de l'option prend fin et on donne la valeur zéro au prix de l'option.



- Programmer l'option Barrière Knock-Out. Pour cela a) modifier la fonction pay-off de l'option :

$$\text{Si } S(N+1, i) \geq B \quad v_{\text{barriere}}(N+1, i) = 0$$

- b) modifier la programmation dynamique :

$$\text{Si } S(n, i) > B \quad \text{alors } v(n, i) = 0$$

- tracer le graphe

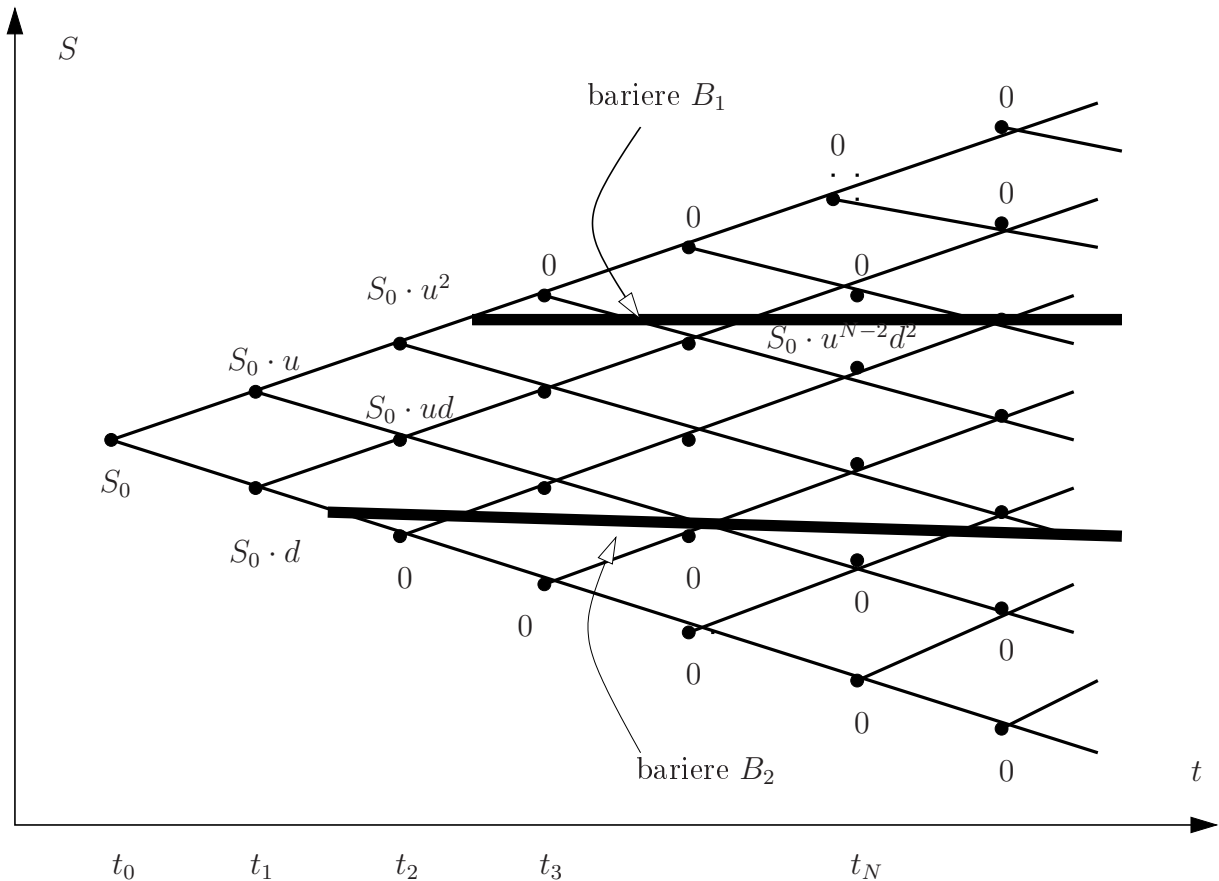
$$S_0 \rightarrow V_{\text{barriere}}(0, S_0)$$

## 5.4 Option Exotique : Double Barrière Knock-Out.

On définit l'option double-barrière par l'idée suivante : Si  $S(t_n)$  touche deux Barrières  $B_1$  et  $B_2$  :

$$S(t_n) \geq B_1 \quad \text{ou} \quad S(t_n) \leq B_2$$

la vie de l'option prend fin et on donne la valeur zéro au prix de l'option.



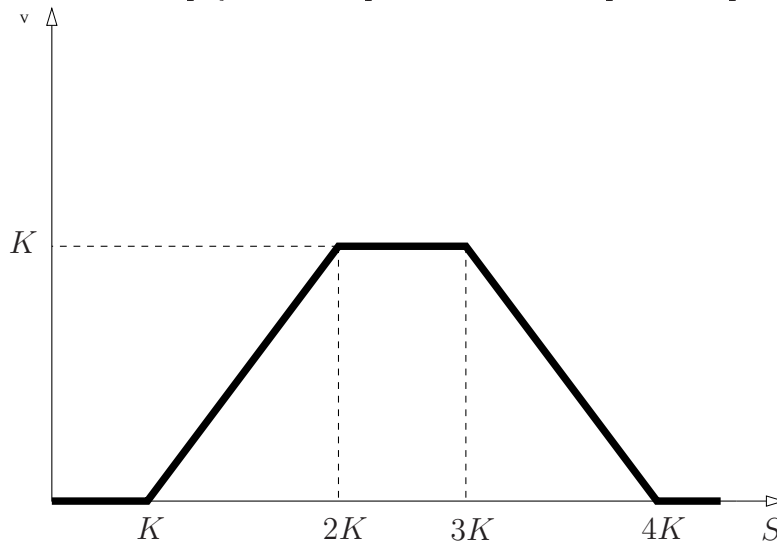
- Programmer l'option Double Barrière Knock-Out.
- Calculer le prix de l'option Double Barrière Knock-Out  $v(t, S_t)$  à un instant quelconque  $t$ . Tracer une surface de prix de l'option pour un quelconque  $t_n$  et une quelconque valeur de l'actif  $S_i$ . Utiliser la fonction Scilab **surf**.

Valeurs numériques :

$$\left\{ \begin{array}{l} K = 10, \\ p = 1/2, N = 200 \\ \sigma = 0.4, r = 0.1 \\ T = 0.5, B_1 = 17, B_2 = 9 \end{array} \right.$$

## 5.5 Option "Condor"

La fonction pay-off de l'option Condor est présenté par le graphe :



- Ecrire le programme qui calcule le prix de l'option "Condor".
- Tracer le graphe  $S_0 \rightarrow V(0, S_0)$  de l'option.
- Utilisez les valeurs suivantes :

$$\begin{cases} L = 7K \\ K = 10 \\ T = 0.5 \\ r = 0.1 \\ \sigma = 0.5 \\ N = 99 \end{cases}$$

## 5.6 Les Grecques

On calcule les Grecques de façon suivante : On remplace la dérivée par la différence finie :

$$\Delta = \frac{\partial V}{\partial S}(0, S_0) = \frac{V(0, S_0 + h) - V(0, S_0 - h)}{2h}$$

$$\Gamma = \frac{\partial^2 V}{\partial S^2}(0, S_0) = \frac{V(0, S_0 + h) + V(0, S_0 - h) - 2V(0, S_0)}{h^2}$$

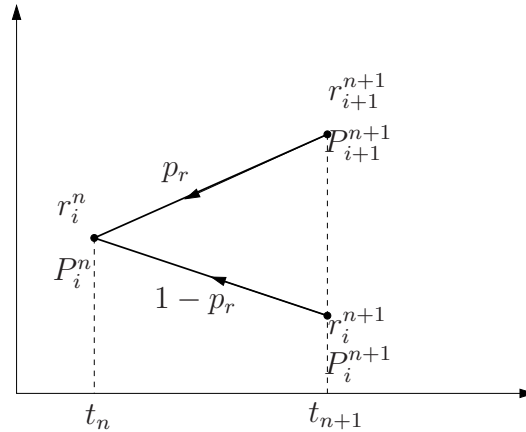
- Programmer deux arbres binomiaux avec le prix initial  $S_0 + h$  et  $S_0 - h$ . Pour  $h$  prenez une petite valeur 0.1.
  - Calculer les Grecques pour les différentes options pour une  $S_0$  fixe.
  - Tracer le graphe

$$S_0 \rightarrow \Delta(0, S_0)$$

pour les différentes options.

## 5.7 Pricing du Bond zero-coupon

- On construit l'arbre des taux d'intérêt.
- A chaque noeud de l'arbre binomial caractérisé par les indices  $(n, i)$  on associe le taux d'intérêt  $r(n, i)$  et le prix du Bond  $P(n, i)$ .



On obtient

$$P(n, i) = e^{-r(n,i)\Delta t} (p P(n+1, i+1) + (1-p) P(n+1, i))$$

- On définit le prix du Bond à la maturité :

$$P(N, i) = 1 \quad i = 0 : N$$

- Utilisez les valeurs suivantes :

$$\begin{cases} r_0 = 0.3 \\ p_r = 1/2 \\ u_r = 1.1 \\ d_r = 0.9 \\ T = 0.5 \\ N = 99 \end{cases}$$

et calculer le prix du  $P(0, T)$  à  $t=0$ .

- Regarder les Bonds avec les maturités différents

$$T = [0, T_1, T_2, \dots, T_M], \quad T_k = \Delta T \cdot k, \quad k = 0 : M$$

et tracer le graphe :

$$T \longrightarrow P(0, T)$$

et le comparer avec celui de taux fixe :

$$T \longrightarrow P_f(0, T) = e^{-r_0(T-t)}.$$

- Essayer d'autres valeurs pour les variables  $u_r, d_r$ . Existe-il des valeurs critiques ? Pour calibrer (dans le projet ce n'est pas demandé) l'arbre du taux d'intérêt on utilise les modèles suivants :

- Vasicek
- Cox - Ingersoll - Ross
- Ho - Lee
- Hull - White

## 5.8 Pricing de l'option Européenne avec les taux d'intérêt stochastique

- On construit l'arbre du taux d'intérêt.
- Quel arbre de l'actif vous choisissez, CRR ou JR ?
- Ecrire l'équation dynamique avec les taux d'intérêt stochastique.
- Ecrire le programme qui calcule le prix de l'option Européenne avec les taux d'intérêt stochastique.
- Tracer le graphe  $S_0 \rightarrow V(0, S_0)$  de l'option et comparer le graphe avec celui trouvé au paragraph 3 (Option Européenne avec les taux d'intérêt  $r_0$  fixe).
- Utilisez les valeurs suivantes :

$$\left\{ \begin{array}{l} r_0 = 0.3 \\ p = 1/2 \\ u_r = 1.1 \\ d_r = 0.9 \\ T = 0.5 \\ N = 99 \\ \sigma = 0.4 \end{array} \right.$$

## 5.9 Arbre Trinomial

### 5.9.1 Evolution de l'actif sous-jacent dans le modèle trinomial

Répetons des informations sur l'arbre Trinomial.

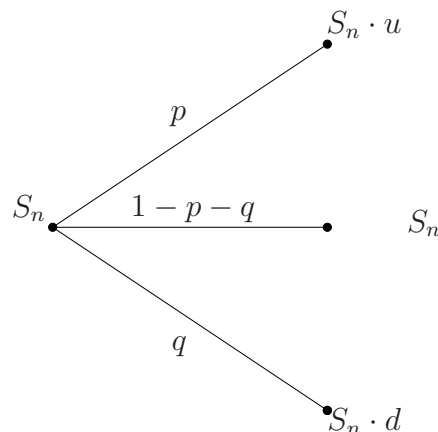
Supposons qu'une action  $S$  évolue dans le temps sur l'intervalle  $[0, T]$ . Discrétisons l'intervalle  $[0, T] : t_n = \Delta t n, \quad T = \Delta t N$ . A chaque instant  $t_n$  on fait correspondre le prix  $S_n$  :

$$t_n \rightarrow S_n, \quad t_0 \rightarrow S_0$$

L'idée principale de la Méthode Trinômiale est la suivante :  $S_{n+1}$  n'admet que 3 valeurs possibles :

$$S_{n+1} = \begin{cases} u \cdot S_n \\ (1 + r\Delta t) \cdot S_n \\ d \cdot S_n \end{cases}$$

On peut donc approcher la diffusion dans le temps par une chaîne de Markov - un arbre trinomial.



Pour que l'arbre soit recombinant (un arbre est dit recombinant si le nombre de noeuds dans chaque tranche croit linéairement avec temps et non pas exponentiellement), on pose

$$ud = 1$$

Dans l'arbre trinomial on définit la probabilité historique de transition ou la matrice de transition  $P$  :

$$\begin{cases} \mathbb{P}[S_{n+1} = u \cdot x / S_n = x] = p \\ \mathbb{P}[S_{n+1} = (1 + r\Delta t) \cdot x / S_n = x] = 1 - p - q \\ \mathbb{P}[S_{n+1} = d \cdot x / S_n = x] = q \end{cases}$$

Introduisons un autre indice "i" pour calculer le prix de l'action à chaque noeud de l'arbre

$$S(n, i) = (u)^{\frac{i}{2}} (d)^{n - \frac{i}{2}} S_0$$

- l'indice  $n$  est l'indice temporel
- l'indice  $i$  est l'indice d'une branche qui définit la valeur de l'action parmi  $n + 1$  valeurs possibles :

$$0 \leq i \leq 2 \cdot n$$

### 5.9.2 Pricing via l'arbre Trinomial

A l'instant  $t_N = T$  le prix de l'option est défini par la formule suivante :

$$V(T, S_N) = f_{pay-off}(S_N).$$

Pour l'option européenne

$$f_{pay-off}(S_N) = \max(S_N - K, 0)$$

**Nous cherchons le prix de l'option à  $t = 0$ . C'est-à-dire nous cherchons  $V(0, S_0)$ .**

Ce prix est représenté par l'espérance conditionnelle :

$$V(0, S_0) = e^{-rT} E[f_{pay-off}(S_N)/S_0]$$

Considérons une branche dans l'arbre trinomial. Il est important de comprendre que  $\forall$  fonction  $f$  bornée l'espérance conditionnelle

$$E[f(S_{n+1})/S_n = x] = \sum_{y \in \{xu, xd, x(1+r\Delta t)\}} P(x, y) f(y)$$

Donc

$$E[f(S_{n+1})/S_n = x] = pf(ux) + qf(dx) + (1 - q - p)f(x)$$

**Nous cherchons le prix de l'option à  $t = 0$**

$$V(0, S_0) = e^{-rT} E[f_{pay-off}(S_N)/S_0]$$

**Pour pouvoir implémenter numériquement la méthode** introduisons une fonction

$$v(n, x) : \mathbb{N} \times \mathbb{R}_+ \rightarrow \mathbb{R}$$

avec les propriétés

$$\begin{cases} v(N, y) = f_{pay-off}(y) \\ v(n, x) = e^{-r\Delta t} \sum_{y \in \{xu, xd, x(1+r\Delta t)\}} P(x, y) v(n+1, y) \end{cases}$$

L'équation

$$v(n, x) = e^{-r\Delta t} \sum_{y \in \{xu, xd, x(1+r\Delta t)\}} P(x, y) v(n+1, y)$$

s'appelle l'équation dynamique pour l'arbre trinomial. On en déduit que

$$e^{-r(T-t_n)} E[f_{pay-off}(S_N)/S_n] = v(n, S_n)$$

et

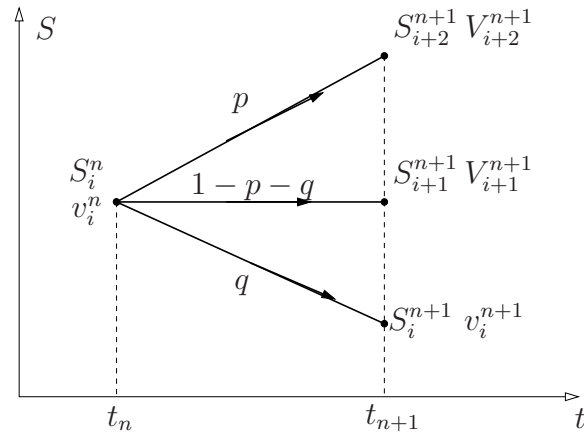
$$e^{-rT} E[f_{pay-off}(S_N)/S_0] = v(0, S_0)$$

**Alors le prix de l'option à  $t = 0$**

$$V(0, S_0) = v(0, S_0)$$

### 5.9.3 Algorithme

A chaque noeud de l'arbre trinomial caractérisé par les indices  $(n, i)$  on associe le prix de l'actif  $S(n, i)$  et le prix de l'option  $V(n, i)$ .



Explicitons l'équation dynamique :

$$V(n, i) = e^{-r\Delta t} (p(n, i) \cdot V(n + 1, i + 2) + q(n, i) \cdot V(n + 1, i) + (1 - p(n, i) - q(n, i)) \cdot V(n + 1, i + 1))$$

; On applique cette équation en remontant le temps.

- Répéter l'algorithme dynamique pour  $S_0$  fixe
- Répéter les calculs des prix de l'option Européenne. Tracer les graphes
- Comparer les résultats sur les prix de l'option Européenne avec ceux obtenues à partir de l'arbre Binomial.

### References.

1. The mathematics of Financial Derivatives. Paul Wilmott, Sam Howison, Jeff Dewynne. Cambridge University Press
2. On Quantitative Finance. Volume 1. Paul Wilmott. Wiley.