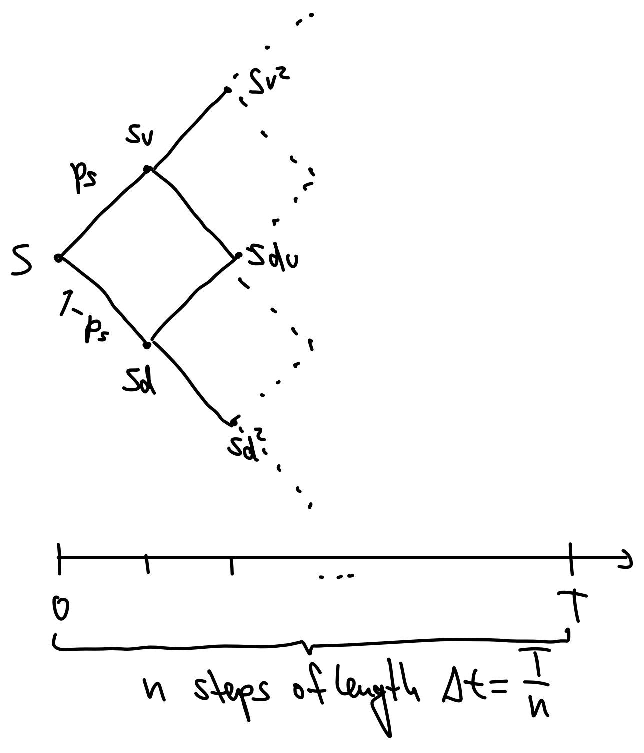
Stochastic Methods Lab, Jacobs University Bremen, Fall 2022 — Session 10, Oct. 12, 2022

Prof. Dr. Soeren Petrat

2.3 Binomial Tree Models

We repeat the binary model with n steps:

1. Stock price model:



$\Rightarrow$ stock price after j upward movements after n steps: $S_T(j \text{ up's}) = S u^j d^{n-j}$

Probability for j up movements for n steps is $P(j,n) = \binom{n}{j} p_s^j (1-p_s)^{n-j}$

$$\binom{n}{j} = \frac{n!}{(n-j)!\,j!} = \frac{n(n-1)\cdots(n-j+1)}{j!} \quad (\text{"}n \text{ choose } j\text{"})$$

(Recall: $(a+b)^n = \sum_{j=0}^{n} \binom{n}{j} a^j b^{n-j}$, the binomial theorem.)

Consistency check: Is this really a probability?

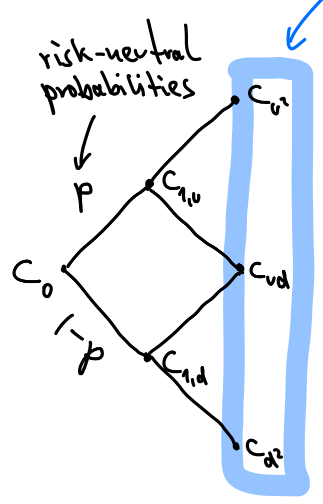
$$\sum_{j=0}^{n} P(j,n) = \sum_{j=0}^{n} \binom{n}{j} p_s^j (1-p_s)^{n-j} = \left(p_s + (1-p_s)\right)^n = 1$$ => yes (all possibilities add up to 1)

We will come back to reasonable choices of parameters u, d, p_s later.

2. Option Price Model:

here, at expiration, we know the payoff (given the possible $S(T)$)

Ex.: $n=2$



$$\left.\begin{array}{l} C_{u^2} = \max(0, Su^2 - K) \\ C_{ud} = \max(0, Sud - K) \\ C_{d^2} = \max(0, Sd^2 - K) \end{array}\right\} \text{for European calls}$$

K = strike price, $p = \frac{e^r - d}{u-d}$ $\quad$ ($C_{1,u}, C_{1,d}$ are $\underbrace{\text{"intermediate payoffs"}}_{\text{"option value at step 1"}}$, C_0 = option price)

We know from binary model how to do one step:

$$\Rightarrow C_{1,u} = e^{-r}\left(p\, C_{u^2} + (1-p)\, C_{ud}\right)$$

$$\Rightarrow C_{1,d} = e^{-r}\left(p\, C_{ud} + (1-p)\, C_{d^2}\right)$$

r = interest rate for one step

next step (from 1 to 0):

$$C_0 = e^{-r}\left(p\,C_{1,u} + (1-p)C_{1,d}\right)$$

$$= e^{-2r}\left(p^2 C_{u^2} + p(1-p)C_{ud} + (1-p)p\,C_{ud} + (1-p)^2 C_{d^2}\right)$$

$$= e^{-2r}\left(p^2 C_{u^2} + 2p(1-p)C_{ud} + (1-p)^2 C_{d^2}\right) \rightarrow \text{option price at time/step } 0$$

In this case (European call options without dividend payments) we get the closed-form formula (for n steps):

$$\boxed{C_0 = e^{-nr}\sum_{j=0}^{n}\binom{n}{j}p^j(1-p)^{n-j}\max(0, S u^j d^{n-j} - K)}$$

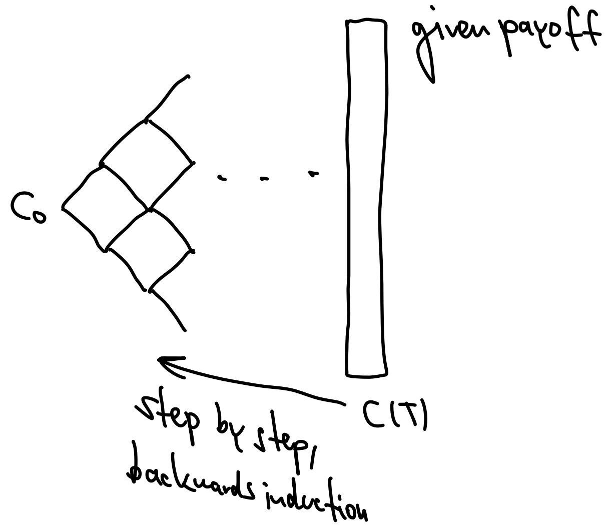
Important note: in this notation r is the interest rate for one step; for the period interest rate r_p (annual if T is in years) we have $\boxed{r = r_p\,\Delta t = r_p \frac{T}{n}}$

(so also $p = \frac{e^{r_p \frac{T}{n}} - d}{u-d}$)

Note: In the general case and for more complicated models (e.g., puts or dividend payments or discontinuous interest compounding) there might not be closed-form formulas, so it is better to have an algorithm available using $\boxed{\text{"backwards induction"}}$ (meaning: start from last column, go to step n-1, then n-2, ..., until at time 0 you get the result C_0) is still based on binomial tree.

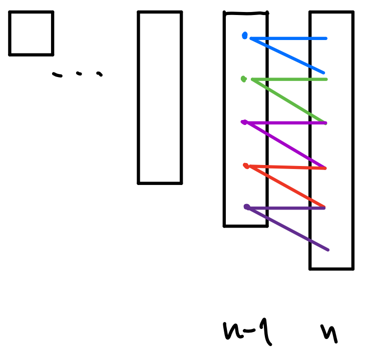
This is a very general and versatile way of option pricing.

For several steps:



Implementation in python: (Problem 2, HW 4)

- use one "for loop" to go through all the steps
- but: computation of vector/array $C(n-1)$ from vector $C(n)$ should be implemented vectorized



recall the notation vector[a:b:increment]

- to store data one could use: • one vector (length $n+1$); memory efficient
  • an $(n+1)\times(n+1)$ matrix; if all data is needed, e.g., for visualization