Project 03 — CRR Binomial Tree & Discrete Delta Hedging

A mini-course + report template

Quant Finance Portfolio (Karim)

December 19, 2025

Contents

1 What you build in Project 03

In this project you implement and validate a **Cox–Ross–Rubinstein (CRR) binomial lattice** and connect it to hedging.

- Build the **CRR stock price tree** with parameters (u, d, p).

- Price **European call/put** by **backward induction**.
- Compute **node deltas** (replicating strategy in the tree).
- Compare the binomial price to **Black–Scholes** and show **convergence** as $N \to \infty$.
- Simulate **discrete-time delta hedging** (rebalancing M times) and quantify the **replication error** distribution vs rebalancing frequency.

One-sentence pitch. "I built a CRR lattice pricer, computed replicating deltas, verified convergence to Black–Scholes, and simulated discrete delta hedging to quantify replication error as rebalancing becomes more frequent."

2 Prerequisites (math you must know)

This is the minimal toolkit to fully understand the rest.

2.1 Discrete-time random processes & filtrations (minimal)

A **price process** in discrete time is $(S_i)_{i=0,\dots,N}$. A **filtration** $(\mathcal{F}_i)$ is the information available up to time i. A trading strategy is $\mathcal{F}_i$-measurable: you can only trade using information you have.

2.2 No-arbitrage & risk-neutral pricing (discrete-time idea)

In a one-period model (from t to $t+\Delta t$), **no-arbitrage** is equivalent to the existence of a **risk-neutral measure** $\mathbb{Q}$ such that the **discounted price** is a martingale:

$$\mathbb{E}^{\mathbb{Q}}\left[\frac{S_{t+\Delta t}}{e^{r\Delta t}} \,\middle|\, \mathcal{F}_t\right] = S_t.$$

Intuition: under $\mathbb{Q}$, the expected growth rate used for pricing is the risk-free rate r.

2.3 Replication & self-financing portfolios

A (stock, cash) portfolio at time t_i is

$$\Pi_i = \Delta_i S_i + B_i,$$

where Δ_i is the number of shares, and B_i is the cash position. **Self-financing** means changes in holdings are financed internally (no external cash injection):

$$\Pi_{i+1} = \Delta_i S_{i+1} + B_i e^{r\Delta t}.$$

Replication means choosing (Δ_i, B_i) so that $\Pi_N = \text{payoff}(S_N)$.

2.4 Binomial tree basics

After i steps, the CRR model has $i+1$ possible nodes (i,j) (with j up moves). Combinatorics: $\mathbb{P}(j \text{ up moves in } i \text{ steps}) = \binom{i}{j} p^j (1-p)^{i-j}$.

2.5 Basic calculus for hedging intuition

In continuous time, delta hedging is related to the derivative $\Delta = \partial V/\partial S$. In discrete time, we approximate delta and accept a **non-zero hedging error**.

3 Model setup: CRR binomial dynamics

We discretize time: $t_i = i\Delta t$ with $\Delta t = T/N$.

3.1 CRR up/down factors

The CRR choice links the tree to a lognormal diffusion limit:

$$u = e^{\sigma\sqrt{\Delta t}}, \qquad d = \frac{1}{u} = e^{-\sigma\sqrt{\Delta t}}.$$

Stock evolution:

$$S_{i+1} = \begin{cases} uS_i & \text{(up)} \\ dS_i & \text{(down)}. \end{cases}$$

3.2 Risk-neutral probability

No-arbitrage requires $d < e^{r\Delta t} < u$, and the risk-neutral probability is

$$p = \frac{e^{r\Delta t} - d}{u - d}, \qquad 0 < p < 1.$$

Discount factor per step: $\text{disc} = e^{-r\Delta t}$.

Sanity check (must pass). If $p \notin (0,1)$, parameters are inconsistent (arbitrage in the tree). Fix by adjusting Δt, σ, or r.

4 European option payoffs

For strike K and maturity T:

$$\text{Call payoff } = (S_T - K)^+, \qquad \text{Put payoff } = (K - S_T)^+, \qquad x^+ = \max(x, 0).$$

5 Pricing by backward induction (replication)

Let $V_{i,j}$ be the option value at node (i,j). At maturity:

$$V_{N,j} = \Phi(S_{N,j}).$$

Backward induction uses the risk-neutral expectation:

$$V_{i,j} = e^{-r\Delta t}\Big(p\,V_{i+1,j+1} + (1-p)\,V_{i+1,j}\Big).$$

Key idea. In the binomial model, pricing and replication are equivalent:

$$V_{i,j} = \text{(discounted) risk-neutral expected continuation value.}$$

5.1 Node delta (replicating strategy in the tree)

At node (i,j), define the one-step replicating delta as the slope between the two next nodes:

$$\Delta_{i,j} = \frac{V_{i+1,j+1} - V_{i+1,j}}{S_{i+1,j+1} - S_{i+1,j}}.$$

Then cash $B_{i,j}$ is chosen so that

$$V_{i,j} = \Delta_{i,j} S_{i,j} + B_{i,j}.$$

This is the discrete-time replication strategy.

6 Convergence to Black–Scholes

As $N \to \infty$ (so $\Delta t \to 0$), the CRR binomial model converges to the Black–Scholes model (diffusion limit). Practically:

- binomial price $\to$ Black–Scholes price,
- binomial delta $\to$ Black–Scholes delta (for smooth payoffs, away from maturity singularities).

7 Discrete delta hedging under GBM (simulation)

The binomial model replicates *exactly* inside the tree. In continuous Black–Scholes, continuous hedging replicates exactly. But in practice, hedging is discrete: you rebalance M times.

7.1 Underlying path simulation (GBM)

A standard geometric Brownian motion (GBM) is

$$\frac{\mathrm{d}S_t}{S_t} = \mu\,\mathrm{d}t + \sigma\,\mathrm{d}W_t.$$

In risk-neutral simulations, use $\mu = r$.

7.2 Discrete-time hedging strategy

Let $t_k = kT/M$. Define the Black–Scholes delta at time t_k as $\Delta_k = \Delta^{BS}(S_{t_k}, T - t_k)$. Start with initial option price C_0, hold Δ_0 shares, and put the remainder in cash:

$$B_0 = C_0 - \Delta_0 S_0.$$

Between rebalancing dates, cash accrues at r. At t_{k+1}, update delta and finance the trade from cash:

$$B_{k+1} \leftarrow B_k e^{r\Delta t} - (\Delta_{k+1} - \Delta_k) S_{t_{k+1}}.$$

Terminal hedging portfolio:

$$\Pi_T = \Delta_M S_T + B_M.$$

Replication (hedging) error:

$$\varepsilon = \Pi_T - \Phi(S_T).$$

Main empirical message. As rebalancing frequency increases ($M \uparrow$), the error distribution tightens: $\mathrm{Std}(\varepsilon)$ decreases (typically at a rate comparable to $\propto 1/\sqrt{M}$ in many settings).

8 Implementation notes (what your code is doing)

- **CRR parameters:** $\Delta t = T/N$, $u = e^{\sigma\sqrt{\Delta t}}$, $d = 1/u$, $p = \frac{e^{r\Delta t} - d}{u - d}$, $\mathrm{disc} = e^{-r\Delta t}$.
- **Tree build:** store a triangular matrix for $S_{i,j} = S_0 u^j d^{i-j}$.
- **Pricing:** set payoff at maturity, apply backward induction.
- **Deltas:** compute $\Delta_{i,j}$ locally from (V, S) at child nodes.
- **Hedging simulation:** simulate GBM paths, rebalance BS delta at M dates, compute error ε.
- **Visualization:** interactive Plotly tree for small N, plus convergence and histogram charts saved as HTML in `assets/`.

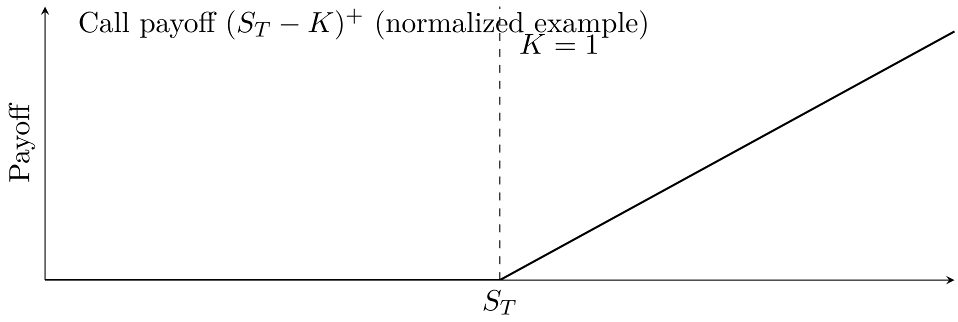
9 Sanity checks you should always do

- **Probability:** verify $0 < p < 1$.
- **Bounds (calls/puts):** $0 \le C_0 \le S_0$, $0 \le P_0 \le Ke^{-rT}$.
- **Monotonicity:** call price increases with S_0 and σ.
- **Put–call parity:** $C_0 - P_0 = S_0 - Ke^{-rT}$ (same inputs).
- **Convergence:** binomial price approaches BS as N grows.

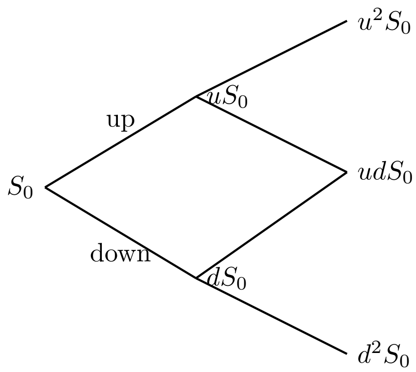
10 Overleaf plots

These are intentionally *conceptual* (fast to compile) and are great for explanations.

10.1 Call payoff diagram

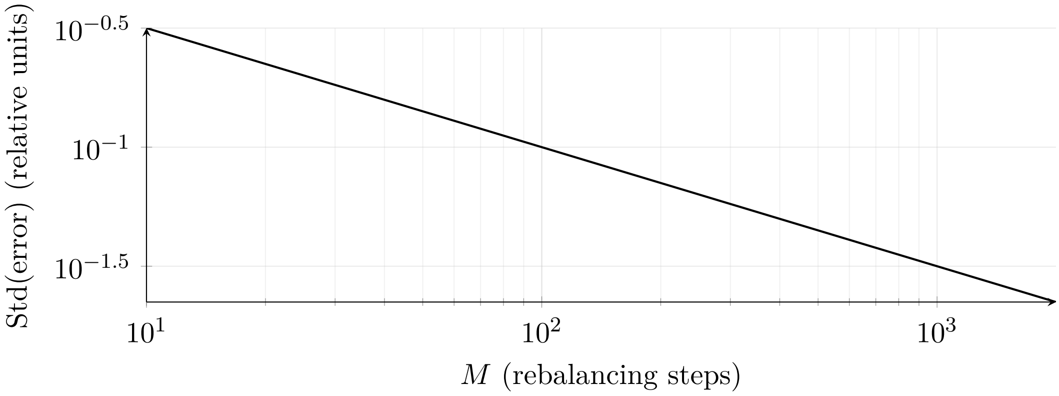


10.2 Tree intuition (small N sketch)



10.3 Why hedging error shrinks with more rebalancing (illustration)

If $\mathrm{Std}(\varepsilon) \propto M^{-1/2}$, a log–log plot is a straight line of slope $-1/2$.



11 Interview pitch (what to say in 20 seconds)

Pitch. "I implemented a Cox–Ross–Rubinstein binomial tree: calibrated (u, d, p), built the lattice, priced European calls/puts by backward induction, and computed node deltas as the replicating strategy. Then I validated convergence to Black–Scholes as the number of steps increases. Finally, I simulated discrete Black–Scholes delta hedging on GBM paths and quantified the replication error distribution versus rebalancing frequency, showing that more frequent re-hedging reduces risk."