Automated Market Makers

The Formula That Replaced the Order Book

BSc Blockchain, Crypto Economy & NFTs

Digital Finance Program

Follow Luca as he reverse-engineers
Uniswap for a hackathon project.

What you will learn:

The constant-product formula $x \cdot y = k$
How slippage depends on pool size
Impermanent loss: why LPs can lose money
Uniswap V1 through V4 evolution

Spring Semester 2026

Contents

1 How to Use This Primer

This primer is designed for students with basic programming knowledge who want to understand AMMs deeply enough to build one. It assumes familiarity with high-school algebra and comfort reading code.

Suggested reading path:

1. Read sections 1–4 to build intuition.
2. Work through the formulas in Section 4 with pen and paper.
3. Read Section 5 ("Deep Dive: Building a Minimal AMM") while optionally following along in a Solidity editor.
4. Attempt the practice problems (Section 9). At least half should be doable without re-reading.
5. Use the solutions appendix to check your work.

Reading time: 3–4 hours for content, 1–2 hours for exercises, and optionally 5–10 hours if you decide to build the minimal AMM described in Section 5.

What you need:

- A calculator (or Python/JavaScript REPL for the formula work).
- Optionally: Foundry or Hardhat installed for hands-on code experiments.
- A Sepolia testnet wallet (free) if you want to deploy your own contracts.
- No real funds required. Everything can be done with testnet ETH.

Luca's advice: "The math looks scary until you plug in real numbers. Do the worked examples by hand at least once. After that, it becomes obvious."

2 Meet Luca

Luca's Story

Luca is a second-year BSc Computer Science student. He is obsessed with hackathons. Last week his roommate swapped ETH for USDC in ten seconds on his phone. No exchange account, no order book, no waiting.
"How does that work without a counterparty?" Luca asks.
"A formula sets the price," his roommate says. "The liquidity sits in a pool. Anyone can trade against it."
Luca opens his IDE. He has four weeks until the next hackathon. His goal: understand AMMs deeply enough to build a minimal DEX.

3 The Story: Why Order Books Fail On-Chain

Luca's Story

Luca's first instinct is to replicate the NYSE. He sketches an on-chain order book: users submit buy and sell orders; the contract matches them. Then he estimates the gas costs. Every order placement, cancellation, and modification is a blockchain transaction. At $1.50 per transaction, a single market maker adjusting prices 100 times per hour would spend $150/hr in gas—just to quote.
"This is insane," Luca mutters. "There has to be a better way."

The on-chain order book problem is real. Traditional exchanges process millions of order updates per second at near-zero cost. Blockchains process 15–30 transactions per second on L1, each costing gas. AMMs solve this by replacing the order book with a mathematical formula.

Definition: Automated Market Maker (AMM)

An **Automated Market Maker** is a smart contract that holds reserves of two (or more) tokens in a **liquidity pool** and uses a deterministic pricing function to execute trades. No order book, no matching engine—just math.

3.1 Where the Idea Came From

AMMs did not appear out of nowhere. The intellectual lineage includes:

- **Robin Hanson's market scoring rules (2002).** Hanson proposed logarithmic scoring rules for prediction markets—a mathematical way to always quote a price without a counterparty.
- **Bancor (2017).** The first on-chain AMM with continuous token models, though it struggled with UX and had a major hack.
- **Vitalik Buterin's Reddit post (2016).** Buterin proposed using $x \cdot y = k$ for a decentralized exchange, years before Uniswap was built.
- **Uniswap V1 (2018).** Hayden Adams turned the theory into production code. A single solo developer changed the DEX landscape forever.

Luca is struck by how simple the idea is. "No one had to invent new math," he thinks. "They just needed the courage to deploy it."

Definition: Liquidity Pool

A **liquidity pool** is a smart contract holding token reserves deposited by **liquidity providers (LPs)**. Traders swap against the pool; LPs earn a share of trading fees proportional to their contribution.

Explain Like I'm 5

Imagine a seesaw with two buckets of tokens. When you take tokens from one bucket (buy), you must add tokens to the other bucket (sell). The seesaw's balance determines the price. The more you take, the higher the price rises.

4 Key Concepts

4.1 The Invariant Idea

Before the specific formula, consider the general idea. An AMM is defined by an **invariant**—a mathematical condition that must hold at all times, regardless of trading activity.

Definition: Market-Making Invariant

A **market-making invariant** is a function $f(x, y) = k$ that:
- Holds constant during all trades (trades move along the curve).
- Changes only when liquidity is added or removed (deposits/withdrawals shift the curve).
- Determines the price via its partial derivatives: price of x in terms of y equals $-\partial f/\partial x \div \partial f/\partial y$.

Different AMMs use different invariants. Uniswap uses $f = x \cdot y$. Balancer uses $f = \prod_i x_i^{w_i}$.

Curve uses a hybrid combining linear and constant-product terms.

Explain Like I'm 5

An invariant is the rule the pool always follows. Think of it like a rubber band: you can stretch it into different shapes (different reserve ratios), but the total tension remains the same. Trading moves you along the stretched band. Adding liquidity means getting a bigger, less stretchy band.

4.2 The Constant-Product Formula

Definition: Constant-Product Formula

The simplest AMM uses the **constant-product invariant**:

$$x \cdot y = k$$

where x is the reserve of token A, y is the reserve of token B, and k is a constant that changes only when liquidity is added or removed (not during trades).

Explain Like I'm 5

Think of k as the "size of the pool." Trading moves tokens between the two sides, but the product stays the same. A bigger k means a deeper pool and less price impact per trade.

4.3 Slippage

Definition: Slippage

Slippage is the difference between the expected price of a trade and the actual execution price. In AMMs, slippage increases with trade size relative to pool depth: larger trades move the price more.

Worked Example: Small Pool vs. Large Pool

Pool A (small): 10 ETH + 20,000 USDC ($k = 200{,}000$).
Buy 1 ETH: new reserves = 9 ETH + $200{,}000/9 = 22{,}222$ USDC.
Cost = $22{,}222 - 20{,}000 = 2{,}222$ USDC per ETH. Spot was \$2,000. Slippage: 11.1%.

Pool B (large): 10,000 ETH + 20,000,000 USDC ($k = 2 \times 10^{11}$).
Buy 1 ETH: cost = $\frac{2\times10^{11}}{9{,}999} - 20{,}000{,}000 \approx 2{,}000.2$ USDC. Slippage: 0.01%.
Pool depth matters enormously.

4.4 Impermanent Loss

Definition: Impermanent Loss (IL)

Impermanent loss is the difference in value between holding tokens in an AMM pool versus simply holding them in a wallet. It occurs whenever the price ratio between the two tokens changes after deposit. The loss is "impermanent" because it reverses if the price returns to the original ratio.

Explain Like I'm 5

You deposit equal value of ETH and USDC into a pool. If ETH doubles, the pool auto-sells some ETH (rebalancing). You end up with fewer ETH than if you had just held. That "missed gain" is impermanent loss.

Worked Example: IL at 2x Price Change

Deposit: 1 ETH ($2,000) + 2,000 USDC = $4,000 total.
ETH doubles to $4,000. If you had just held: 1 ETH ($4,000) + 2,000 USDC = $6,000.
In the pool: the AMM rebalances to $\approx$ 0.707 ETH + 2,828 USDC = $5,657.
IL = $6,000 − $5,657 = $343, or **5.72%**.
The LP must earn more than $343 in fees to break even.

4.5 Fee Tiers

Definition: Fee Tiers

Uniswap V3 introduced multiple **fee tiers** for pools of the same token pair:
- **0.01%** – stablecoin pairs (USDC/USDT)
- **0.05%** – correlated pairs (ETH/stETH)
- **0.30%** – standard pairs (ETH/USDC)
- **1.00%** – exotic or volatile pairs

Lower fees attract more volume; higher fees compensate LPs for IL risk.

4.6 Concentrated Liquidity

Definition: Concentrated Liquidity

In Uniswap V3, LPs can concentrate their capital within a specific **price range** $[p_a, p_b]$ instead of spreading it across the entire price curve $(0, \infty)$. This provides up to 4,000x better capital efficiency but requires active management.

Worked Example: Concentrated vs. Full-Range

LP deposits $10,000 into the ETH/USDC pool.
- **Full-range (V2 style):** liquidity spread from $0 to ∞. Effective depth: minimal.
- **Concentrated (V3):** range $1,800–$2,200. The $10,000 acts like $200,000 of full-range liquidity within that band.

Trade-off: if ETH moves outside the range, the position earns zero fees and is 100% in the less valuable token.

Self-Assessment Checkpoint

Can you explain $x \cdot y = k$ in one sentence? Can you calculate the output of a trade given pool reserves? Can you explain why impermanent loss occurs? If all three: continue. Otherwise, re-read Section 3.

5 How It Works: The Math

Luca's Story

Luca opens a blank Solidity file. He needs to implement three core functions: `swap()`, `addLiquidity()`, and `removeLiquidity()`. Time to learn the math.

5.1 Swap Output Formula

Formula: AMM Swap Output

Given reserves (x, y) and input amount Δx (before fees):

$$\Delta y = y - \frac{k}{x + \Delta x} = \frac{y \cdot \Delta x}{x + \Delta x}$$

After a fee f (e.g., $f = 0.003$ for 0.30%):

$$\Delta y_{\text{net}} = \frac{y \cdot \Delta x \cdot (1 - f)}{x + \Delta x \cdot (1 - f)}$$

Worked Example: Full AMM Trade

Pool: 100 ETH + 200,000 USDC. $k = 20{,}000{,}000$. Fee = 0.30%.
Luca wants to buy ETH with 10,000 USDC.
Step 1: Effective input after fee: $10{,}000 \times 0.997 = 9{,}970$ USDC.
Step 2: New USDC reserve: $200{,}000 + 9{,}970 = 209{,}970$.
Step 3: New ETH reserve: $20{,}000{,}000/209{,}970 = 95.251$ ETH.
Step 4: ETH output: $100 - 95.251 = 4.749$ ETH.
Step 5: Effective price: $10{,}000/4.749 = \$2{,}105.7$ per ETH.
Spot price was $200{,}000/100 = \$2{,}000$. Slippage: 5.3%.
The 0.30% fee ($30) goes to LPs. The 5.3% slippage is the price impact.

5.2 Impermanent Loss Formula

Formula: Impermanent Loss

If the price ratio changes by factor r (i.e., new price = $r\times$ original price):

$$\text{IL} = \frac{2\sqrt{r}}{1 + r} - 1$$

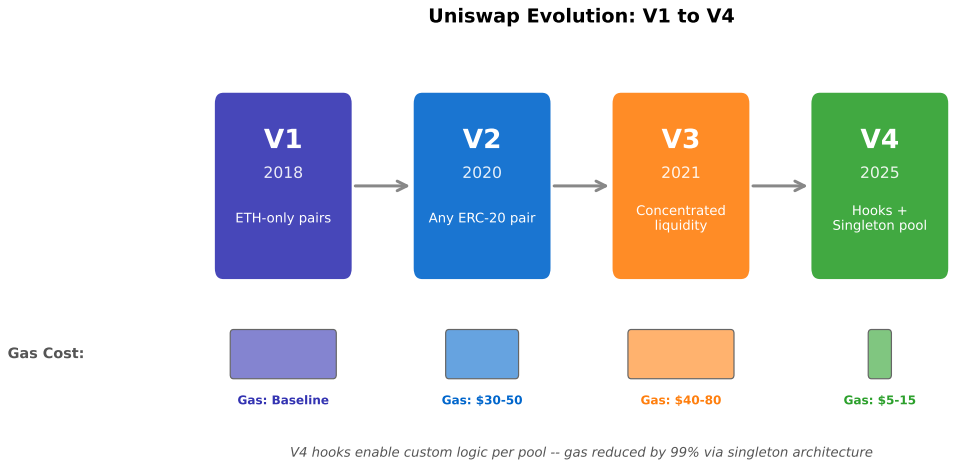
This is always negative (a loss) when $r \neq 1$.

Worked Example: IL at Various Price Multipliers

Price Change (r)	IL Formula	IL (%)
$1.25\times$ ($+25\%$)	$\frac{2\sqrt{1.25}}{2.25} - 1$	-0.60%
$1.50\times$ ($+50\%$)	$\frac{2\sqrt{1.50}}{2.50} - 1$	-2.02%
$2.00\times$ ($+100\%$)	$\frac{2\sqrt{2.00}}{3.00} - 1$	-5.72%
$3.00\times$ ($+200\%$)	$\frac{2\sqrt{3.00}}{4.00} - 1$	-13.40%
$5.00\times$ ($+400\%$)	$\frac{2\sqrt{5.00}}{6.00} - 1$	-25.46%

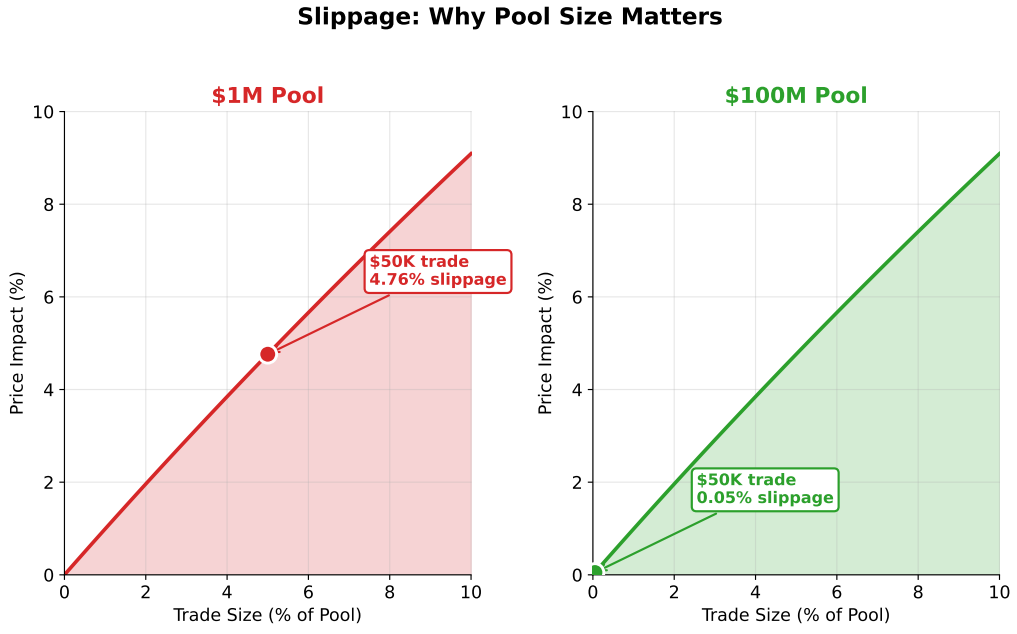
At $5\times$ price change, the LP has lost a quarter of their potential value.

Uniswap Version Evolution



V4 hooks enable custom logic per pool -- gas reduced by 99% via singleton architecture

Slippage Comparison: Small vs. Large Pools



Larger pools absorb trades with minimal price impact -- deep liquidity protects traders

Key Takeaway

AMMs replace order books with a simple invariant ($x \cdot y = k$). The price emerges from the ratio of reserves. Slippage is the cost of moving along the curve. Impermanent loss is the cost of the pool rebalancing your position.

5.3 Beyond Constant Product: StableSwap

Definition: StableSwap Invariant (Curve)

Curve Finance uses a hybrid formula optimized for assets expected to trade near 1:1 (stablecoins or correlated pairs):

$$A \cdot n^n \sum x_i + D = A \cdot D \cdot n^n + \frac{D^{n+1}}{n^n \prod x_i}$$

where A is an amplification coefficient. For small deviations from parity, the curve behaves almost linearly (very low slippage). For large deviations, it reverts to constant-product behavior.

Explain Like I'm 5

Constant product is a seesaw—any imbalance has a price. StableSwap is a bowl with a flat bottom and steep sides. Tokens near the middle swap with almost zero slippage; tokens near the edges swap normally.

Worked Example: Curve vs. Uniswap for Stablecoin Swap

Swap 1M USDC for USDT in a pool with $100M total liquidity.

AMM	Slippage	USDT Received
Uniswap V2 ($x \cdot y = k$)	~1.0%	990,000
Curve StableSwap ($A = 100$)	~0.01%	999,900

Curve is 100x more efficient for like-for-like swaps. This is why nearly all stablecoin trading volume routes through Curve.

5.4 DEX Aggregators and Routing

Definition: DEX Aggregator

A **DEX aggregator** (1inch, Matcha, Cow Swap) routes a single user trade across multiple AMMs to find the best effective price. A large swap might be split across 5+ pools simultaneously to minimize slippage.

Worked Example: Route Splitting

A user wants to sell 500 ETH for USDC. No single pool has enough liquidity for zero-slippage execution.

The aggregator routes:

- 180 ETH through Uniswap V3 ETH/USDC (0.05% tier)
- 140 ETH through Curve tricrypto pool
- 110 ETH through Balancer weighted pool
- 70 ETH through SushiSwap V2

Combined effective price: within 0.3% of spot. A direct swap through one pool might have been 2–5% worse.

5.5 Derivation: Where $x \cdot y = k$ Comes From

The constant-product formula is not arbitrary. It emerges from two design goals:

1. **No-arbitrage pricing.** For infinitesimally small trades, the price should be smooth and continuous.
2. **Bounded liquidity.** The pool should never run out of a token entirely, regardless of trade size.

The simplest function satisfying both properties is $f(x, y) = x \cdot y = k$:

- As $x \to 0$ (all of token A removed), $y \to \infty$ to maintain k. Impossible: the pool runs out first.
- Marginal price at (x, y): $P = y/x$. Buying a tiny amount dx costs $dy = y \cdot dx/x +$ smaller terms.
- The price curve is strictly convex: larger trades pay a higher marginal price. This is slippage.

Worked Example: Marginal vs. Average Price

Pool: 100 ETH + 200,000 USDC. Marginal (spot) price = 200,000/100 = \$2,000.
Buy 1 ETH (small trade): average price ≈ \$2,020 (1% premium).
Buy 10 ETH (medium): average price ≈ \$2,222 (11% premium).
Buy 50 ETH (large): average price ≈ \$4,000 (100% premium).
The larger the trade, the more the average price diverges from the marginal price. This is why AMMs are not suitable for very large trades without routing through aggregators.

5.6 Fee Revenue and LP Economics

Formula: LP Daily Fee Income

$$\text{Daily fee income} = \text{Pool volume} \times \text{Fee rate} \times \frac{\text{Your liquidity}}{\text{Total liquidity}}$$

Worked Example: LP Profitability Calculation

You provide \$10,000 to a pool with \$5M total liquidity and 0.30% fee. The pool processes \$2M in daily volume.
Daily income: \$2M × 0.003 × (10,000/5,000,000) = \$2,000 × 0.002 = \$4/day.
Annual income: \$4 × 365 = \$1,460, or **14.6% APR**.
But if ETH/USDC moves 2x during the year, IL = 5.7% of position = \$570. Net return: \$1,460 − \$570 = \$890, or 8.9%. Still profitable, but well below the headline 14.6%.

6 Real-World Cases

6.1 Arbitrage: The Invisible Hand

Definition: AMM Arbitrage

An **arbitrageur** profits from price differences between an AMM and an external reference market. If Uniswap prices ETH at $2,010 while Coinbase prices it at $2,000, a bot buys ETH on Coinbase and sells on Uniswap, capturing the $10 spread until the two prices converge.

Explain Like I'm 5

Arbitrageurs are the invisible hand that keeps the AMM price honest. Every time prices drift, they swoop in for profit—and pull the price back to the "correct" value. Without arbitrageurs, AMMs would quickly become unusable.

Worked Example: Arbitrage Opportunity

Binance spot price for ETH: $2,000. Uniswap pool price: $2,020 (someone just made a large buy).
An arbitrageur:
1. Buys 10 ETH on Binance for $20,000.
2. Sells 10 ETH on Uniswap for approximately $20,100 (slippage pushes price down during the sale).
3. Gross profit: $20,100 − $20,000 = $100.
4. Subtract: exchange fees, gas, bridge costs, capital holding costs.
5. Net profit: perhaps $40–$60.

After the arbitrage trade, the Uniswap price has fallen back near $2,005. Next trader gets a fair price.

Historical Case Study: Uniswap V1 to V4

- **V1 (Nov 2018):** Built by one developer (Hayden Adams). ETH-only pairs. Proof of concept.
- **V2 (May 2020):** Any ERC-20/ERC-20 pair. Flash swaps. Time-weighted average price (TWAP) oracle.
- **V3 (May 2021):** Concentrated liquidity. Multiple fee tiers. Up to 4,000x capital efficiency vs. V2.
- **V4 (Jan 30, 2025):** "Hooks"—custom plugins that execute before/after swaps. Singleton contract (one contract for all pools). Dynamic fees. Net-settlement for gas savings.

In seven years, Uniswap went from a toy project to processing over $1 trillion in cumulative volume.

Historical Case Study: The LP Profitability Crisis

A 2023 study found that **only 37.2% of Uniswap V3 LPs were profitable** after accounting for impermanent loss. The reasons:
- Concentrated liquidity amplifies IL within the chosen range.
- Retail LPs set ranges and forget; professional market makers actively rebalance.
- MEV bots exploit stale LP positions via just-in-time (JIT) liquidity attacks.

Key Takeaway

Active management is essential in V3/V4. Passive "set and forget" LP strategies are profitable only in low-volatility pairs (stablecoin/stablecoin).

7 The LP's Dilemma

Luca's Story

Luca's friend Elena is a passive investor. She asks: "Should I provide liquidity to earn yield?" Luca hesitates before answering. The honest answer depends on many things: the asset pair, volatility expectations, her willingness to monitor positions, and her alternative uses of capital.

Definition: LP Decision Framework

A rational LP should deposit into a pool if and only if:

$$\text{Expected fee yield} > \text{Expected IL} + \text{Opportunity cost} + \text{Smart contract risk premium}$$

In practice:

- **Stablecoin pools** (USDC/USDT): near-zero IL, low fee yield. Usually profitable for passive LPs.
- **Correlated pairs** (stETH/ETH, wBTC/BTC): low IL, moderate yield. Generally safe.
- **Volatile major pairs** (ETH/USDC): significant IL, decent yield. Profitable only with active management.
- **Volatile exotic pairs** (random memecoin/ETH): massive IL, high yield. Rarely profitable long-term.

Worked Example: Stablecoin LP vs. Volatile LP

Elena has \$10,000. Two options:
Option A: USDC/USDT on Curve. Annual yield: ∼3–5%. IL: essentially zero. Expected return: \$300–\$500/year.
Option B: ETH/USDC on Uniswap V3 (narrow range). Annual fee yield: ∼20%. Expected IL: ∼8% (assuming moderate volatility). Opportunity cost if ETH rises: ∼5%. Expected return: \$700/year—but highly variable.
Elena's choice depends on her risk tolerance. Option A is nearly risk-free in dollar terms. Option B has more upside but can underperform in bad scenarios.

Key Takeaway

"LP yield" in headlines usually ignores impermanent loss. Always calculate expected IL before committing capital. A "20% APY pool" can easily underperform a 3% APY stablecoin pool if volatility is high.

8 What Can Go Wrong

Luca's Story

Luca's hackathon mentor warns him: "Building a DEX is easy. Making LPs not lose money—that's the hard part."

Important Caveat

Impermanent loss exceeding fees. In volatile markets, IL can dwarf fee income. An LP in an ETH/USDC pool during a 50% crash loses 5.7% to IL. If the pool earns only 3% in fees annually, the LP is underwater.

Important Caveat

Rug pulls in new token pools. A scammer creates a token, adds liquidity to a Uniswap pool, waits for buyers, then removes all liquidity. Buyers are left with worthless tokens. In 2024, rug pulls accounted for hundreds of millions in losses.

Common Misconception: "Providing liquidity is free money"

LP fees look attractive, but they come with IL risk, smart contract risk, and the opportunity cost of not simply holding the tokens. In many pools, a simple buy-and-hold strategy outperforms LP positions.

Common Misconception: "Bigger pools always earn more fees per LP"

Fee income per LP depends on *volume relative to liquidity*. A \$100M pool with \$1M daily volume generates less fee income per dollar deposited than a \$10M pool with the same \$1M daily volume. The ratio volume/TVL is what matters.

9 The Cryptoeconomics Lens

Luca's Story

Before submitting his hackathon project, Luca steps back to analyze AMMs as an economic system.

Before applying the six questions, Luca writes a note to himself: *AMMs succeed because they align the incentives of three different groups—traders, LPs, and arbitrageurs—without any of them trusting each other.*

Definition: The Three-Sided AMM Market

Every AMM involves three participant types:

- **Traders** want the best price. They benefit from deep liquidity.
- **LPs** want fee income without losing principal to IL. They benefit from high volume and low volatility.
- **Arbitrageurs** profit from price discrepancies. They benefit from price movements and inefficiencies.

These three groups have partially conflicting interests. The genius of AMM design is that each

group's self-interested actions contribute to the system's overall function: traders provide volume, LPs provide liquidity, and arbitrageurs enforce pricing accuracy.

Cryptoeconomics Lens

Apply the six cryptoeconomics questions to AMMs:

1. **PROBLEM:** On-chain order books are prohibitively expensive. AMMs replace them with a formula, enabling permissionless trading with minimal on-chain operations.
2. **INCENTIVES:** LPs deposit tokens to earn trading fees. Traders pay fees for instant execution without counterparty risk. Arbitrageurs keep AMM prices aligned with external markets (and earn profit for the service).
3. **BENEFITS / COSTS:** Traders get 24/7 permissionless access; they pay slippage and fees. LPs earn fees; they pay impermanent loss. Arbitrageurs extract value from price discrepancies.
4. **FAILURE MODE:** IL exceeding fees, oracle-based attacks, rug pulls, MEV extraction. The 2023 LP profitability study showed that most retail LPs lose money on V3.
5. **DESIGN:** Constant product ($x \cdot y = k$) vs. StableSwap (Curve's amplified invariant) vs. concentrated liquidity (V3). Each optimizes for different asset types and volatility profiles.
6. **ALTERNATIVES:** Intent-based trading (no pool, solvers find best execution), hybrid order-book/AMM models (dYdX), and request-for-quote systems could reduce IL and MEV.

10 Deep Dive: Building a Minimal AMM in Pseudo-Code

Luca decides to sketch a minimal AMM in pseudo-code. He wants to see exactly what a DEX contract must track.

10.1 State Variables

Worked Example: AMM Contract State

```
contract SimpleAMM {
  uint256 public reserveA;    // amount of token A
  uint256 public reserveB;    // amount of token B
  uint256 public totalShares; // total LP shares issued
  mapping(address => uint256) public shares;
  uint256 public constant FEE = 3;  // 0.3% (basis: 1000)
}
```

Just five state variables are enough to implement a working AMM. Compare that to an order-book exchange, which would need arrays for every order, matching logic, and cancellation handling.

10.2 addLiquidity Function

Worked Example: addLiquidity Logic

```
function addLiquidity(uint256 amountA, uint256 amountB) {
  if (totalShares == 0) {
    // First LP sets the initial ratio
    sharesMinted = sqrt(amountA * amountB);
  } else {
```

```
    // Subsequent LPs must match the existing ratio
    require(amountA * reserveB == amountB * reserveA);
    sharesMinted = (amountA * totalShares) / reserveA;
  }
  reserveA += amountA;
  reserveB += amountB;
  totalShares += sharesMinted;
  shares[msg.sender] += sharesMinted;
}
```

Key insight: the first LP establishes the price ratio. All subsequent LPs must deposit in that ratio (or they lose value to arbitrageurs).

10.3 swap Function

Worked Example: swap Logic

```
function swap(uint256 amountIn, bool aToB) {
  if (aToB) {
    uint256 amountInAfterFee = amountIn * (1000 - FEE) / 1000;
    uint256 amountOut = (reserveB * amountInAfterFee)
                      / (reserveA + amountInAfterFee);
    reserveA += amountIn;
    reserveB -= amountOut;
    transfer(tokenA, contract, amountIn);
    transfer(tokenB, msg.sender, amountOut);
  } // else symmetric for B to A
}
```

This is the entire trading engine. In real code there are safety checks (reentrancy guards, slippage bounds, deadline checks), but the math above is the core.

10.4 Deployment Economics

Worked Example: Gas Cost of Deployment

A minimal AMM contract compiled in Solidity costs roughly:
- Deployment: ∼1.5M gas ($30–$300 on L1, pennies on L2).
- Add liquidity: ∼150k gas ($3–$30 on L1).
- Swap: ∼120k gas ($2–$25 on L1).

Luca's hackathon budget ($50) covers deployment, 10 liquidity additions, and 20 test swaps on Sepolia testnet.

10.5 What Gets Harder in Production

Definition: Production-Grade AMM Challenges

Moving from a minimal AMM to a production system introduces many challenges:
- **Multiple tokens per pool.** Balancer supports up to 8 tokens with arbitrary weights.
- **Fee tiers.** Uniswap V3's multiple fee tiers require per-pool accounting.
- **Concentrated liquidity.** V3's tick-based system increases contract complexity 10x.
- **Oracle integration.** Exposing time-weighted prices to other protocols.

- **Gas optimization.** Shaving 10% off gas can save users millions per month across all pools.
- **Flash loans.** Supporting flash loans within the same AMM contract.
- **Hooks (V4).** Allowing third-party plugins at key lifecycle points.

Key Takeaway

Writing a minimal AMM is a weekend project. Writing a safe, efficient, production-grade AMM is a career. The gap is risk, optimization, and defensive programming against adversaries you have not yet imagined.

11 Practice Problems

Discovery Exercise 1

Basic swap. A pool has 50 ETH and 100,000 USDC. Calculate the output if a trader sells 5,000 USDC (ignore fees). What is the effective price per ETH?

Discovery Exercise 2

Fee impact. Repeat Exercise 1 with a 0.30% fee. How much less ETH does the trader receive?

Discovery Exercise 3

Slippage comparison. The same trader sells 5,000 USDC in two pools: Pool A has 50 ETH + 100K USDC, Pool B has 500 ETH + 1M USDC. Compute slippage in each. What is the ratio?

Discovery Exercise 4

IL calculation. You deposit 2 ETH (at \$2,000) and 4,000 USDC into a pool. ETH rises to \$3,000. Calculate: (a) the value of your position if you had just held, (b) the value in the pool, (c) the impermanent loss in dollars and percentage.

Discovery Exercise 5

IL break-even. Using the IL formula, at what annual fee rate does an LP break even if ETH moves 2× over the year? Express as APR.

Discovery Exercise 6

Concentrated liquidity. An LP provides \$10,000 to a V3 ETH/USDC pool with range \$1,900–\$2,100. If the current price is \$2,000 and stays in range, the position acts like \$200,000 of V2 liquidity. What is the capital efficiency multiplier? What happens if ETH drops to \$1,800?

Discovery Exercise 7

Fee tier selection. Explain why a USDC/USDT pool uses the 0.01% tier while an ETH/SHIB pool uses 1.00%.

Discovery Exercise 8

Rug pull detection. List three on-chain signals that a new token pool might be a rug pull.

Discovery Exercise 9

V3 vs. V4. Uniswap V4 introduces "hooks." Give two examples of hook functionality and explain how each benefits traders or LPs.

Discovery Exercise 10

Cryptoeconomics analysis. Arbitrageurs extract value from AMM pools. Is this a feature or a bug? Argue both sides in 3–4 sentences each.

12 Glossary of AMM Terms

Active LP
A liquidity provider who regularly rebalances their position to stay within profitable price ranges.

AMM
Automated Market Maker. A smart contract that prices trades using a formula rather than an order book.

Arbitrage
Simultaneous buying and selling of an asset across markets to profit from price differences.

Aggregator
A service (1inch, Matcha, CowSwap) that routes trades across multiple AMMs to find the best price.

Concentrated liquidity
Uniswap V3 feature allowing LPs to provide liquidity only within chosen price ranges.

Constant product formula
$x \cdot y = k$. The core invariant of Uniswap V1/V2.

CowSwap
A batch auction DEX that matches user trades peer-to-peer before falling back to AMMs.

Curve
A DEX specialized in stablecoin and correlated-asset swaps using a hybrid invariant.

DEX
Decentralized Exchange. A protocol that allows token swaps without a custodian.

Fee tier
The trading fee charged by a pool (0.01%, 0.05%, 0.30%, 1.00% in Uniswap V3).

Full-range LP
An LP who provides liquidity across the entire price curve, from 0 to ∞ (V2 style).

Hook
Uniswap V4 feature: custom code executed at key lifecycle points of a pool.

Impermanent loss (IL)
The loss incurred by an LP when token prices diverge from the deposit ratio.

JIT liquidity
Just-in-time liquidity. An MEV strategy where a searcher adds liquidity, captures fees from a known

large trade, then removes liquidity—all in one block.

K (invariant)
The constant product of reserves in a pool. Changes only when liquidity is added or removed.

Liquidity pool
A smart contract holding reserves of two or more tokens, against which users can trade.

LP
Liquidity Provider. A user who deposits tokens into a pool to earn fees.

LP token
A receipt token representing a share of a liquidity pool.

Order book
A list of buy and sell orders with prices and quantities. The traditional exchange model.

Passive LP
An LP who deposits liquidity and does not actively manage the position.

Price impact
The amount by which a trade moves the AMM price. Equivalent to slippage for small trades.

Reserves
The amounts of each token held in a liquidity pool.

Router
A contract that handles the user-facing logic for swaps, routing through multiple pools.

Singleton
Uniswap V4's contract architecture: one contract holds all pools, reducing deployment costs.

Slippage
The difference between expected and actual execution price on a trade.

Slippage tolerance
The maximum slippage a user is willing to accept before a trade reverts.

Spot price
The current marginal price of a pool: y/x for constant product pools.

StableSwap
Curve's invariant, optimized for low slippage between stable assets.

Tick
A discrete price level in Uniswap V3, between which liquidity is allocated.

Uniswap
The largest DEX, pioneer of AMM-based trading.

V1 / V2 / V3 / V4
Versions of Uniswap, each adding significant features.

Solutions

Exercise 1. $k = 50 \times 100{,}000 = 5{,}000{,}000$. New USDC $= 100{,}000 + 5{,}000 = 105{,}000$. New ETH $= 5{,}000{,}000/105{,}000 = 47.619$. Output $= 50 - 47.619 = 2.381$ ETH. Effective price $= 5{,}000/2.381 = \$2{,}100$ per ETH. Spot was \$2,000; slippage $= 5.0\%$.

Exercise 2. Effective input $= 5{,}000 \times 0.997 = 4{,}985$. New USDC $= 104{,}985$. New ETH $= 5{,}000{,}000/104{,}985 = 47.626$. Output $= 2.374$ ETH. The fee costs $2.381 - 2.374 = 0.007$ ETH ($\approx \$14$).

Exercise 3. Pool A: slippage $= 5.0\%$ (from Exercise 1). Pool B: $k = 500{,}000{,}000$. New USDC $= 1{,}005{,}000$. New ETH $= 500M/1{,}005{,}000 = 497.512$. Output $= 2.488$ ETH. Effective price $= \$2{,}009.6$. Slippage $= 0.48\%$. Pool A has $\approx 10.4\times$ the slippage of Pool B, because it is $10\times$ smaller.

Exercise 4. (a) Hold: $2 \times \$3{,}000 + \$4{,}000 = \$10{,}000$. (b) Pool: $k = 2 \times 4{,}000 = 8{,}000$. At \$3,000/ETH, pool rebalances to $\sqrt{8{,}000/3{,}000} = 1.633$ ETH and $\sqrt{8{,}000 \times 3{,}000} = 4{,}899$ USDC. Value $= 1.633 \times 3{,}000 + 4{,}899 = \$9{,}798$. (c) IL $= \$10{,}000 - \$9{,}798 = \$202$, or 2.02%. Matches formula: $r = 1.5$, IL $= 2\sqrt{1.5}/2.5 - 1 = -2.02\%$.

Exercise 5. At $r = 2$, IL $= 5.72\%$. The LP needs annual fees $\geq 5.72\%$ APR to break even. In practice, this requires high volume relative to TVL.

Exercise 6. Capital efficiency $= \$200{,}000/\$10{,}000 = 20\times$. If ETH drops to \$1,800 (below range), the position converts entirely to ETH and earns zero fees until price re-enters the range.

Exercise 7. USDC/USDT rarely deviates from 1:1, so IL is near zero and LPs can afford low fees. ETH/SHIB is extremely volatile; LPs need 1% fees to compensate for high IL risk. Fee tier should match the asset pair's volatility.

Exercise 8. (1) Deployer holds >90% of liquidity tokens (can drain the pool). (2) Token contract has hidden mint functions or transfer restrictions. (3) No lock on liquidity tokens (LP can remove at any time). Additional signals: unverified contract, no audit, anonymous team.

Exercise 9. Example hooks: (1) *Dynamic fee hook* – adjusts the fee based on volatility (benefits LPs by increasing fees during volatile periods). (2) *TWAP order hook* – executes large orders over time to reduce slippage (benefits traders).

Exercise 10. *Feature:* Arbitrageurs keep AMM prices aligned with the global market, which benefits all traders and prevents the pool from becoming mispriced. Without them, the AMM price could diverge permanently. *Bug:* Arbitrageurs extract value from LP positions by trading right after price changes, capturing value that should flow to LPs. This is a form of MEV that reduces LP profitability and contributes to the finding that 62.8% of V3 LPs are unprofitable.

Further Reading

Foundational papers:

- Adams, H., Zinsmeister, N., & Robinson, D. (2020). *Uniswap v2 Core*. The original V2 whitepaper introducing pair contracts and flash swaps.
- Adams, H., Zinsmeister, N., Salem, M., Keefer, R., & Robinson, D. (2021). *Uniswap v3 Core*. Concentrated liquidity mathematics.
- Angeris, G., & Chitra, T. (2020). "Improved Price Oracles: Constant Function Market Makers." *ACM AFT*. Formalization of AMM pricing.
- Angeris, G., Kao, H.-T., Chiang, R., Noyes, C., & Chitra, T. (2019). "An Analysis of Uniswap Markets." Foundational analysis of $x \cdot y = k$.

Empirical studies:

- Lehar, A., & Parlour, C. A. (2023). "Decentralized Exchange: The Uniswap Automated Market Maker." *Journal of Finance* (forthcoming).
- Heimbach, L., Wang, Y., & Wattenhofer, R. (2022). "Risks and Returns of Uniswap V3 Liquidity Providers." Showed 37.2% LP profitability.

Tools to explore:

- **Uniswap interface** (`app.uniswap.org`) – Try small swaps on L2 to see V3 concentrated liquidity in action.
- **Revert Finance** (`revert.finance`) – Analytics for V3 LP positions, including IL tracking.
- **APY.vision** – LP profitability tracker.
- **Curve Finance** (`curve.fi`) – Try a stablecoin swap to see the StableSwap invariant in action.
- **Foundry / Hardhat** – Solidity development frameworks for building your own AMM.

Luca's hackathon plan:
Luca decides his minimum viable product is (1) a single-pair AMM with add/remove liquidity and swap functions, (2) deployed on Sepolia testnet, (3) a minimal web frontend using ethers.js. Total code: ∼300 lines of Solidity plus 200 lines of JavaScript. Total cost: $0 (testnet). His learning goal: understand every line of code, including the math behind fee distribution and LP token accounting.