

Week 3 of 6

3 Liquidity, AMMs & yield

Task

Provide liquidity on devnet, or to a pool the class builds

Blog post

Learning DeFi: my experience with liquidity pools and yield on Solana

Today

1 How a market sets a price

2 $x \cdot y = k$, by hand

3 Impermanent loss

4 Smarter curves and routing

5 Where yield really comes from

How a market sets a price

Two designs. One needs a counterparty. One doesn't.

Order book vs pool

Order book

Makers post prices. A matching engine pairs them.

	151.2	40
	150.8	25
	150.5	60
	150.2	30
spread		
	149.9	35
	149.6	50
	149.2	20
	148.8	45

Buy 90: fills 30 at 150.2, 60 at 150.5

AMM pool

Liquidity providers fill two tanks. A formula sets the price.



No counterparty waiting. The pool always quotes.


$$x \cdot y = k$$

One line of maths runs billions in volume.

The whole rule

$$\textcircled{x} \cdot \textcircled{y} = \boxed{k}$$

x tokens of SOL in the pool

y tokens of USDC in the pool

k must never go down after a trade

$$\text{Price of SOL} = y \div x = 15,000 \div 100 = \mathbf{150 \text{ USDC}}$$

One swap, worked by hand

No fee

Pool before

$$x = 100 \quad y = 100$$

$$k = 100 \times 100 = 10,000$$

You add 10 X

$$x = 110$$

k must stay 10,000

Pool solves for y

$$y = 10,000 \div 110$$

$$= 90.91$$

You receive

$$100 - 90.91$$

$$= 9.09 \text{ Y}$$

With a 0.3% LP fee

0.3% fee kept back

$$10 \times 0.997$$

$$= 9.97 \text{ X counts for pricing}$$

Pool solves for y

$$y = 10,000 \div 109.97$$

$$= 90.934$$

You receive

$$100 - 90.934$$

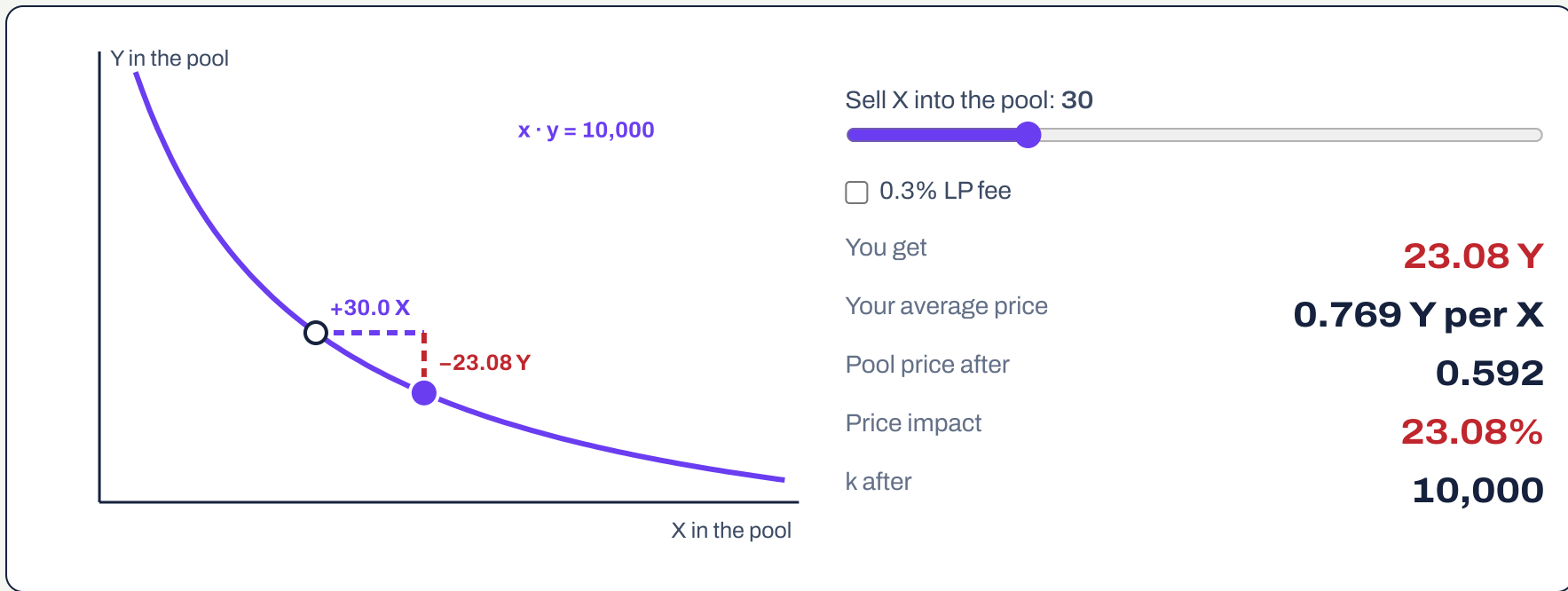
$$= 9.066 \text{ Y}$$

Pool keeps all 10 X

$$k = 110 \times 90.934$$

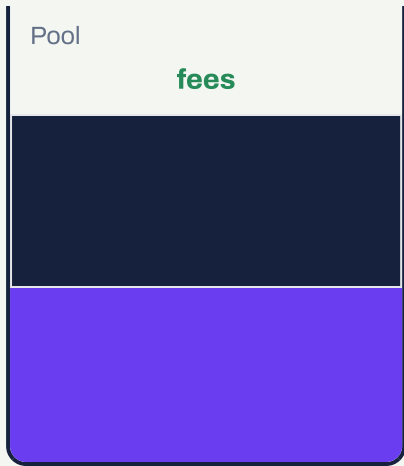
$$= 10,002.7 \text{ LPs got richer}$$

Drag it



Bigger trade, steeper slide down the curve, worse price. Tick the fee and watch k grow.

Being the pool



Deposit

Both tokens, at the pool's current ratio



Get a receipt

LP tokens (CPMM) or a position NFT (CLMM)



Earn

Every swap leaves a fee in the pool



Withdraw

Burn the receipt, take your share of both tokens



Impermanent loss

The price of being the pool everyone trades against.

Why it happens

Elsewhere

SOL jumps to **\$200**

Your pool

still says **\$100**

Arbitrageurs buy cheap SOL from the pool until it says \$200

Pool had

5 SOL + 500 USDC

Pool now has

3.54 SOL + 707 USDC

At \$200: holding = \$1,500 · your pool share = **\$1,414** · gap **-5.72%**

The table to memorise

SOL moves	Hold	LP	Impermanent loss
1.25x	\$1125	\$1118	 -0.62%
2x	\$1500	\$1414	 -5.72%
4x	\$2500	\$2000	 -20.00%
0.5x	\$750	\$707	 -5.72%

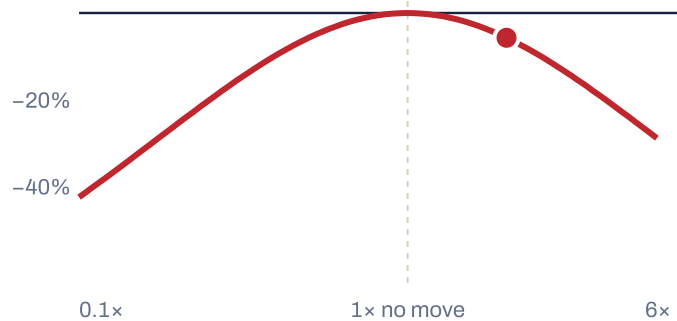
Same loss for 2x and 0.5x. Direction doesn't matter, only distance.

Can fees win it back?

SOL price moves **2.00x** (from \$100 to \$200)



0.5x 1.25x 2x 4x



-5.72%

$$IL = 2\sqrt{r} \div (1 + r) - 1$$

Just hold		\$1,500
LP position		\$1,414
LP + fees		\$1,464

Fee APR 20%



Days 90



Holding wins by \$36

SOL doubles. You were an LP in a SOL/USDC pool with no fees. Compared with just holding, you have:

A More dollars

B The same

C About 5.7% fewer dollars

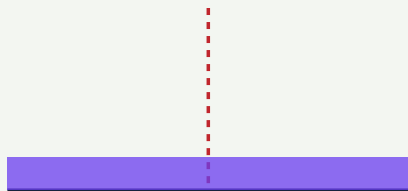
D You lost everything



Beyond $x \cdot y = k$

How Solana's DEXs put liquidity where trades happen.

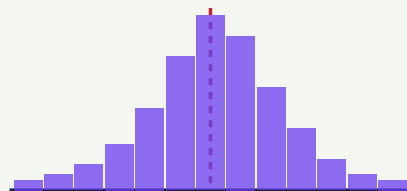
Four ways to hold liquidity



Constant product

Raydium CPMM

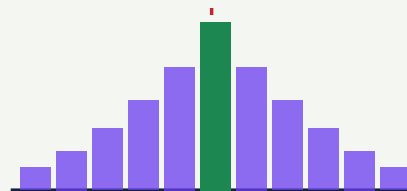
Spread over every price, zero to infinity. Simple, thin.



Concentrated

Orca Whirlpools, Raydium CLMM

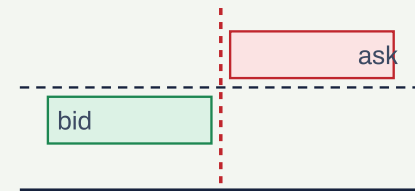
LPs pick ranges. Depth piles up near the price.



Bins

Meteora DLMM

Fixed-price buckets. No slippage inside a bin.

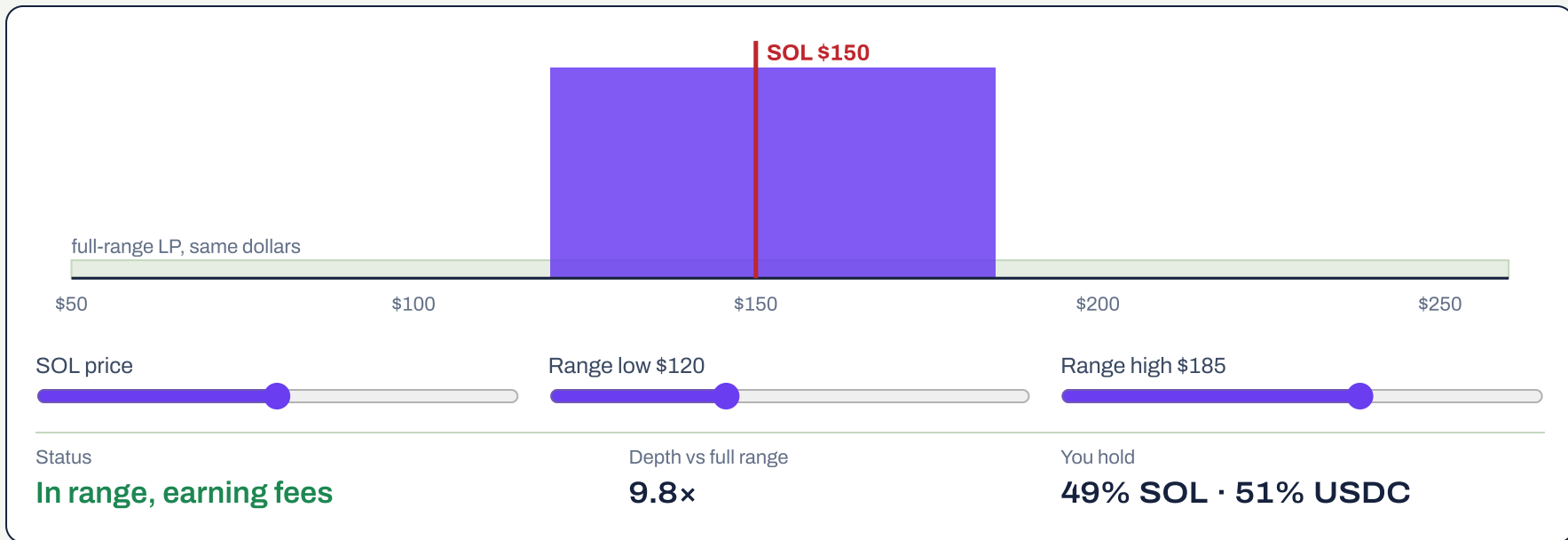


Prop AMM

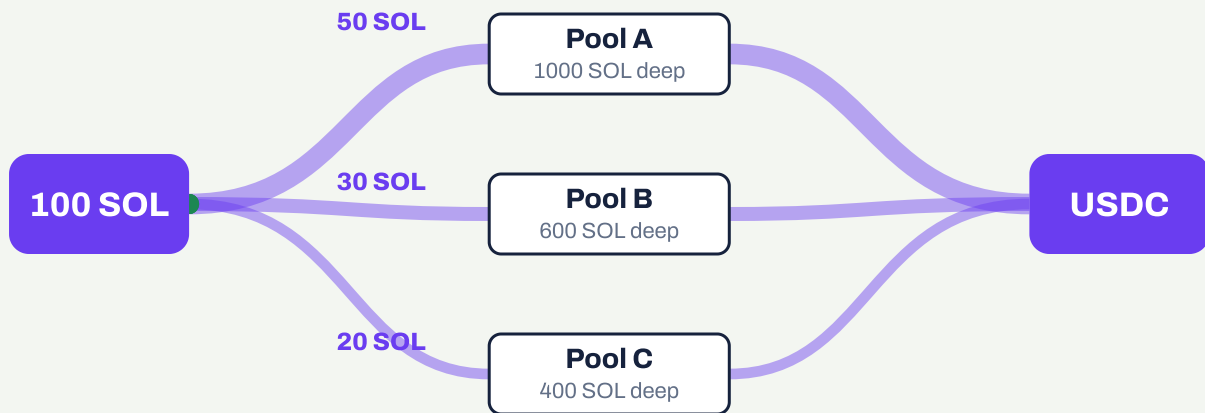
HumidiFi and others

A market maker quotes its own prices on-chain, updated every slot.

Concentrated: deeper, until it isn't



Why Jupiter splits your order



All through pool A

9.1% impact

$100 \div (1000 + 100)$

Split by depth

4.8% in each pool

$50 \div 1050 = 30 \div 630 = 20 \div 420$



Where yield comes from

Every yield has a payer. Find them.

Who pays you?

Traders swapping

Raydium, Orca, Meteora pools

Network issuance + MEV tips

JitoSOL, JupSOL, mSOL

Borrowers

Kamino Lend, Jupiter Lend

Newly printed tokens

Farm rewards

Same sources, automated

Kamino vaults

LP fees

Lasts while volume lasts

Staking

Lasts while the network runs

Lending interest

Lasts while people borrow

Token emissions

Paid by dilution. Ends when the budget does

Vaults

Adds a strategy fee and a smart-contract layer

The APY trap

Advertised APR 40%



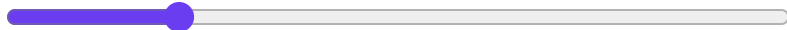
Yearly

Monthly

Weekly

Daily

Token price over the year -50.0%



APR (simple)

40.0%

↓ compounded

APY

49.1%

$(1 + \text{APR} \div n)^n - 1$

↓ priced in dollars

What you actually made

-25.4%

\$1,000 became \$746

A farm advertises 300% APY, paid in its own new token. What is the first question to ask?

A Is it audited?

B Who is paying this, and in what?

C Daily or weekly compounding?

D What is the TVL?

Before the task: two "USDC"s

USDC

Issued by Circle's devnet faucet

4zMMC9...JDncDU

What you got in week 2



**Same name, different mint. A
program only checks the mint.**

devUSDC

Minted by Orca for its devnet pools

BRjpCH...iHok1k

What the Orca devnet pool wants

On mainnet, anyone can mint a token called "USDC". **Check the mint address.**

Task: be a liquidity provider

Users

Pick the route your class can run:

- **A** Raydium devnet UI: swap, add liquidity
- **C** Add liquidity to the class pool with your two week-2 tokens

Then build the impermanent-loss table: 1.25x, 2x, 4x.

Builders

- **B** Orca SDK on devnet:
`setWhirlpoolsConfig('solanaDevnet')`, open a position in the SOL/devUSDC Whirlpool
- **C** Deploy the constant-product pool from `program-examples/tokens/token-swap` or Blueshift's Pinocchio AMM

Test the route yourself first

Raydium: devnet setting on raydium.io, program IDs from docs.raydium.io only. Orca pool: `3KBZiL...z1HvPt`

Blog post

"Learning DeFi: my experience with liquidity pools and yield on Solana"

Include your impermanent-loss table, and one yield you found and who pays it.



Teacher prep

For you, before class. Skip this live.

You, before class

1. **Work the swap by hand.** 100 / 100 pool, swap 10 X in: 9.09 Y out. Redo with 0.3% fee: 9.066.
2. **Build the IL spreadsheet.** 1.25x, 2x, 4x: LP vs holding. Reuse it in class.
3. **Learn three curve families.** CPMM, concentrated (Whirlpools, CLMM), bins (DLMM). Then how a prop AMM differs.
4. **Skim Blueshift's Pinocchio AMM** (deposit, swap) so you can point builders to it.
5. **Run your chosen task route** on a fresh wallet, 48 hours before.

Addresses and checks

What	Value
Orca Whirlpools program (devnet)	<code>whirLbMiicVdio4qvUfM5KA6Ct8VwpYzGff3uctyCc</code>
Orca SOL/devUSDC pool	<code>3KBZiL2g8C7tiJ32hTv5v3KM7aK9htpqTw4cTXz1HvPt</code>
Orca devUSDC mint	<code>BRjpCHtyQLNCo8gqRUr8jtdAj5AjPYQaoqbvcZiHok1k</code>
Circle devnet USDC mint	<code>4zMMC9srt5Ri5X14GAgXhaHii3GnPAEERYPJgZJDncDU</code>
Raydium devnet APIs	<code>api-v3-devnet.raydium.io</code> , <code>transaction-v1-devnet.raydium.io</code>
Pool example	<code>solana-developers/program-examples/tokens/token-swap</code>

Verify before class

Raydium's devnet is a setting on raydium.io (announced 15 Jul 2025). Use program IDs from docs.raydium.io, not the cpi-example README (CPMM mismatch, issue #16, still open).