Floating Odds eXchange Model

Joshua Ng

Introduction

FoX is a protocol for making bets with floating odds. As opposed to fixed odds, the markets in FoX have odds that fluctuate dynamically over time – in response to factors such as the volume of bets placed, new information about the event, and changes in market sentiment.

It is designed for bets to happen continuously without the need for a counterparty or a traditional order book. As bets are represented by fungible tokens, they are also fully composable with existing Defi protocols.

Markets can be created for questions with binary outcomes: Yes or No (E.g. Will it rain today?) In our model, each market has 2 tokens – one that represents each side of the bet (y token and n token). To place a bet, you buy that side of the token, and each purchase triggers a mint that increases the token supply. Winners then split the pot of deposits proportionally to the amount of tokens they hold on the winning side.

1) For Efficient Price Discovery

Our model introduces the following formula – which allows for more predictable, steady price changes when the balance shifts – and factors in relative demand of each side of the token into each token's price.

Formula: $P_y = \frac{S_y}{S_y + S_n}$

P_y = Price of token y S_y = Supply of token y S_n = Supply of token n

This graph shows the price of S_y as its supply increases (assuming the supply of S_n is constant)
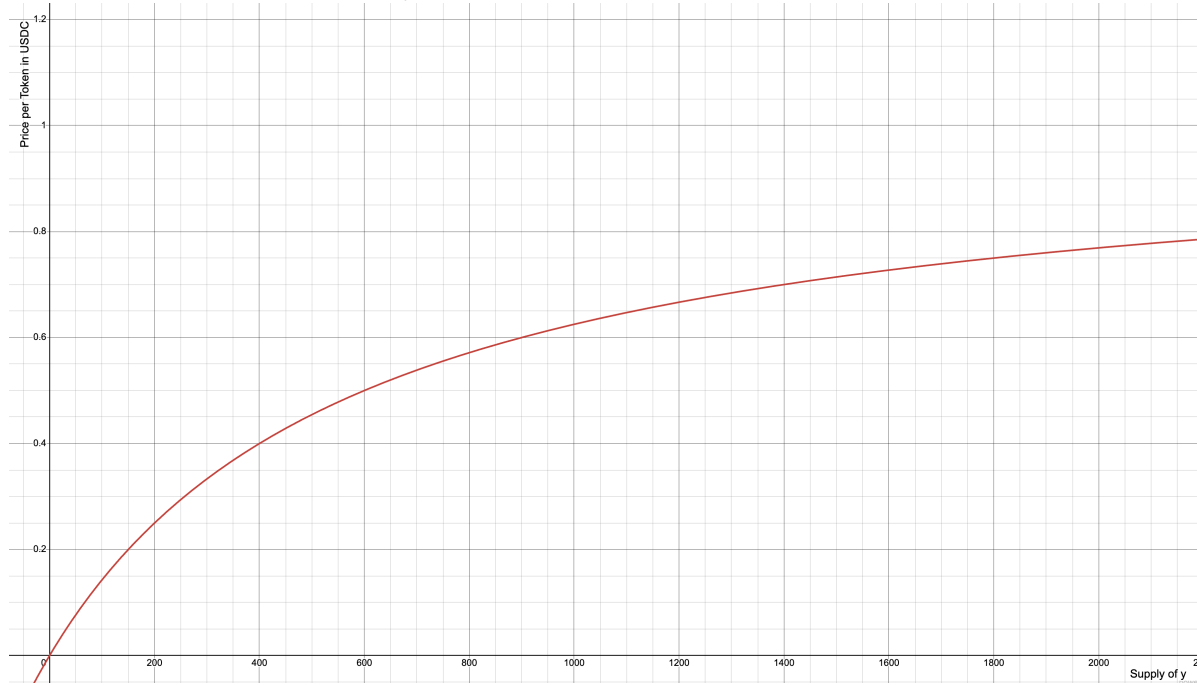
The price of token n can be calculated with the formula which will create a similar graph:

$$P_n = \frac{S_n}{S_y + S_n}$$

2) For Real-time Market Sentiment

The sum of $P_y + P_n$ will always add up to \$1 because the prices are derived as a proportion of the total supply of both, making it a natural reflection of the current market consensus of each outcome's probability. As new information enters the market, purchases for a particular side of the token will cause a repricing and adjustment of odds.

3) Direct Exposure to the Desired Outcome

Often in other prediction markets, users are required to mint both outcome tokens ("Yes" and "No") when making a trade. In ours, users are only buying the token they want ("Yes" or "No") There is no need to mint and automatically sell the opposite token.

This allows users to focus solely on the outcome they believe in, and not be forced to engage in unnecessary transactions with slippage and suboptimal pricing in low liquidity situations.

Technical Overview

Calculating total cost of purchase

The integral of the curve is used to calculate the total price paid for your purchase.

If a trader wants to buy token n, the price is calculated with this formula where Sy, the supply of y is a constant as it doesn't change.

$$\int_{S_{no}}^{S_{n1}} \frac{S_n}{S_y + S_n} dS_n = \left[S_n - S_y \ln \left(S_n + S_y \right) \right]_{S_{n0}}^{S_{n1}}$$

S_{n0} = Initial Supply S_{n1} = Final Supply S_n = Supply of token n S_y = Supply of token y

Take for example a trader wants to buy 200 n tokens from a market that has 600 y and 900 n. The total amount paid will be calculated as follows and indicated by the shaded area under the graph:

$$\int_{900}^{1100} \frac{S_n}{600 + S_n} dS_n = \left[S_n - 600 \ln \left(S_n + 600 \right) \right]_{900}^{1100} \approx 124.90$$
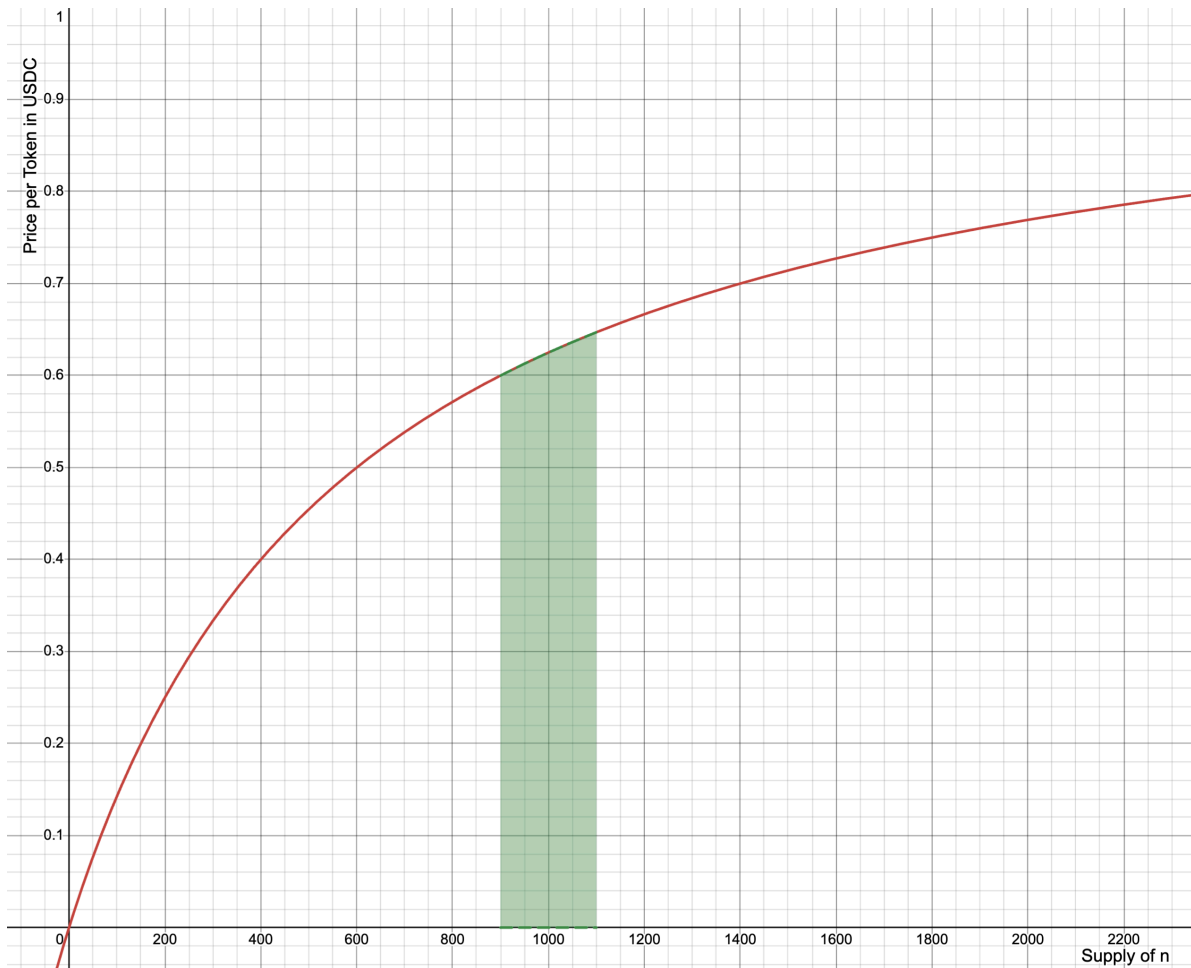
Market Creation

A creator initialises a market with some initial context (question, estimated end date). In the alpha stage, a resolver address is selected by the creator, with the authority to set the outcome when it is determined. The creator also deposits the initial liquidity by buying tokens from both sides of the market. To offset the risk that creators take in holding a position, they can set creator fees.

Initial Liquidity

To minimise price impact when seeding an unbalanced market, deposits are made in the following sequence, in 1 atomic transaction:

To minimise price impact when seeding an unbalanced market, deposits are made in the following sequence, in 1 atomic transaction:

Creator wants to create a market with 500 n tokens and 1000 y tokens. 1. First, 500 n and 500 y will be purchased in equal amounts. for 500 USDC 2. Then we purchase the remaining 500 y tokens, which will be calculated as follows. $\int_{500}^{1000} \frac{S_y}{S_y+500} dS_y = \left[S_y - 500 \ln(S_y + 500) \right]_{500}^{1000} \approx$ 297.27 3. Thus the final price is 797.27 USDC

Liquidity Depth and Price Impact

The more deposits there are, the greater the liquidity depth, which would make it easier for traders to place larger bets without as much price impact. Some examples: In a market with 500 y and 300 n tokens seeded, the initial price is $0.375. From here, 100 n purchased will bring the price of n up to $0.444, which is an 18.5% price impact. If you seed the market in the same ration but with 2000 y and 1200 n, 100 n purchased creates a price impact of only 5.1%

By seeding more liquidity, the creator is taking on more risk and can potentially make a loss. For example, if they choose to seed heavily towards the side that lost, they will make a loss. If the market creator doesn't put enough liquidity, the pool will have too little liquidity at the start and traders cannot make big trades without causing a huge price impact, causing the market to move slowly.

A Devnet calculator is available here, and you can enter different amounts to see the corresponding changes in price impact: https://fox.jup.ag/market/766PJ9idbugMgs7xbLKCwg2zio1JH2ZnQcouaaLtqZoG

Market Resolution and Winnings

The resolver sets the outcome as one of the following: Yes, No, Undecided (Refund). Users who hold the token representing the winning side get to claim from the pot of deposits according to their share of the winning token, after creator fees are deducted. The token representing the losing side will not be able to claim anything from our contract.

If the resolution sets the outcome as undecided, the pool will be split amongst all n and y token holders equally and the fee will not be charged.

As tokens are fungible, they can also be traded on any external pools at any time, at a price determined by the market. This allows traders who buy the token to cash out before the market is resolved by selling it in external pools.
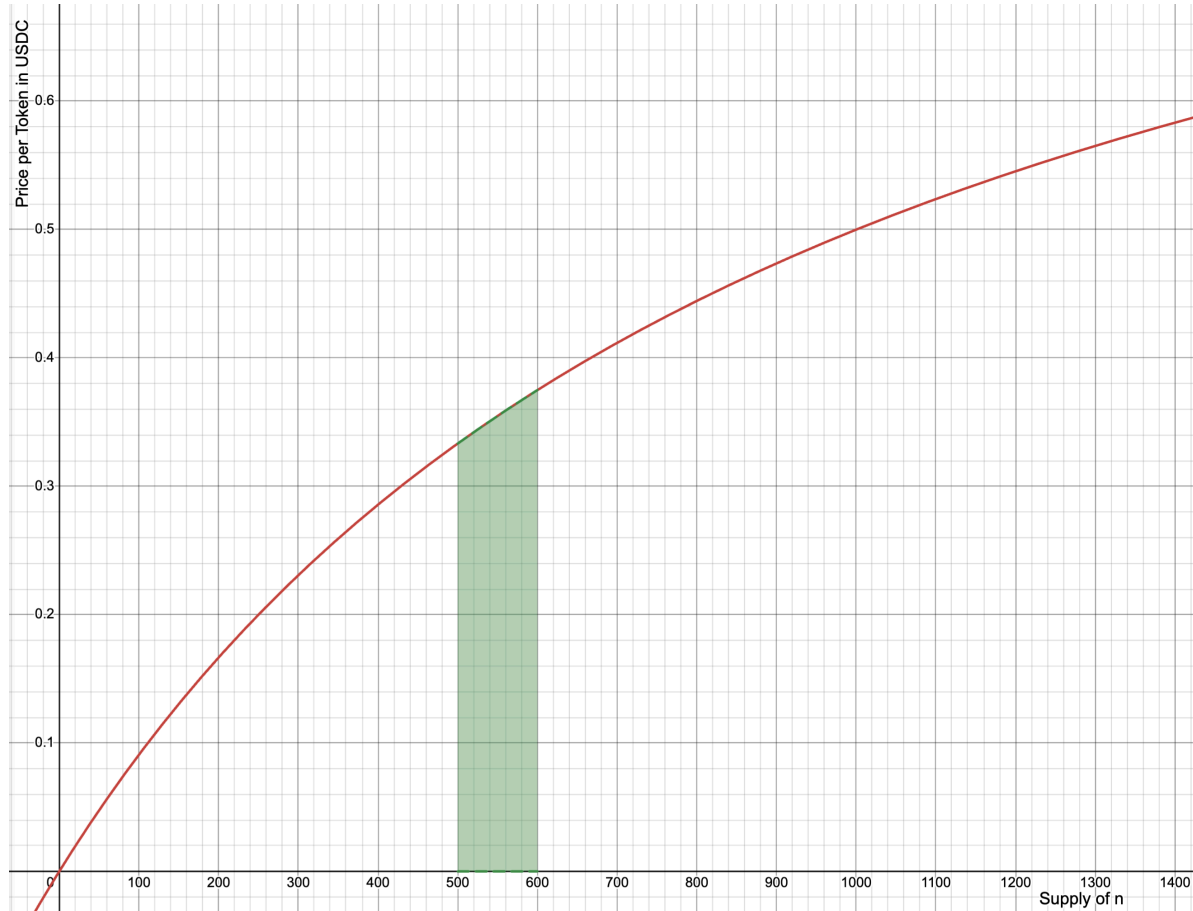
Example:

The creator creates a market and seeds liquidity in it, buying 500 N and 1000Y. The price to seed this amount of liquidity is 797.27 USDC.

A trader then wants to mint 100 token n from the contract. The price to mint the tokens are

$$\int_{500}^{600} \frac{S_n}{S_n+1000} dS_n = [S_n - 1000\ln(S_n + 1000)]_{500}^{600} \approx 35.46$$

The price is calculated using the area under the bonding curve where Sy is constant

Since N is the token with less demand(fewer N has been minted than Y) here, it is sold for cheaper at 0.3546 USDC per token.

Another trader wants to buy 200N and the price to buy is represented with the area from 600 to 800 under the same graph since the supply of Y did not change:

$$\int_{600}^{800} \frac{S_n}{S_n+1000} dS_n = [S_n - 1000\ln(S_n + 1000)]_{600}^{800} \approx 82.22$$
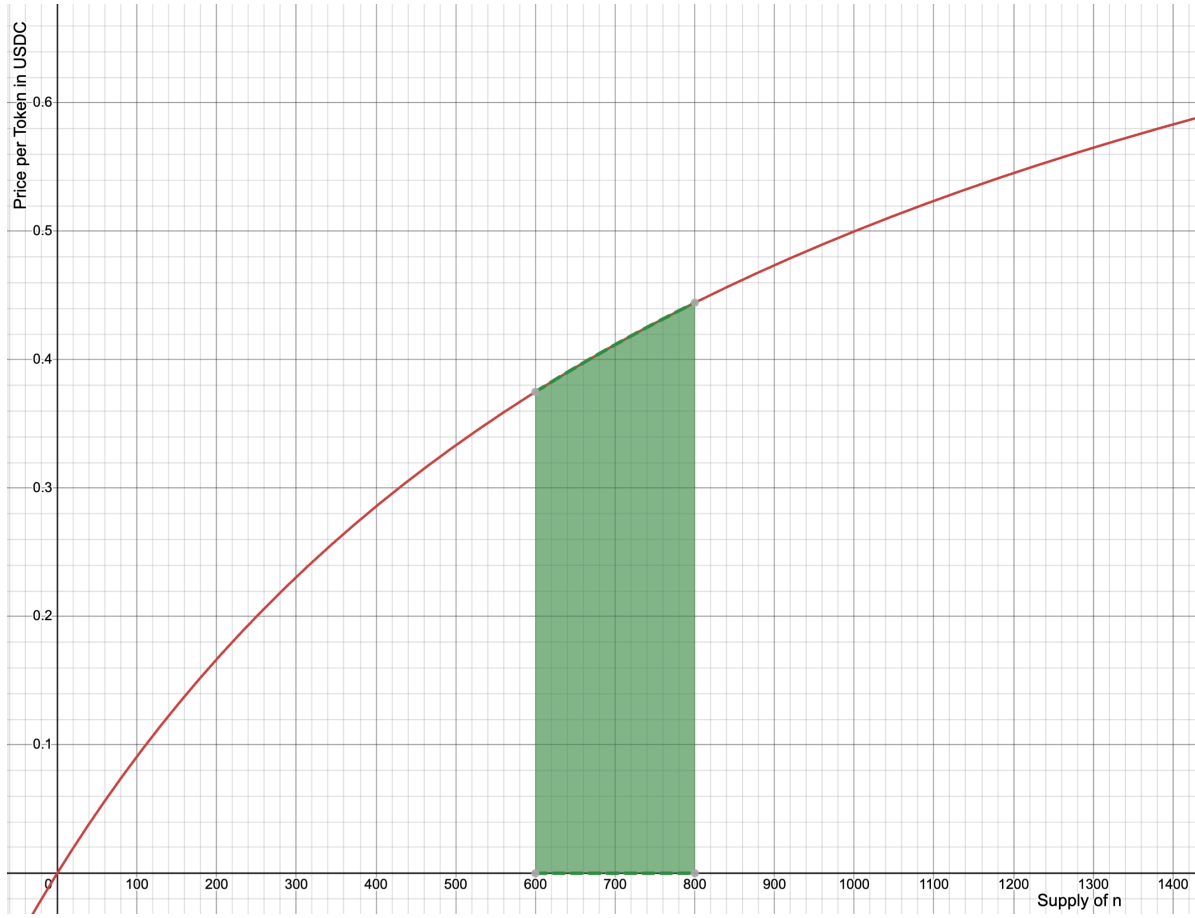
When a trader wants to buy 400 token y, the price is calculated with the same formula, but instead with Sy as the variable and Sn as the constant:

$$\int_{1000}^{1400} \frac{S_y}{S_y + 800} dS_y = \left[S_y - 800 \ln(S_y + 800) \right]_{1000}^{1400} \approx 239.46$$
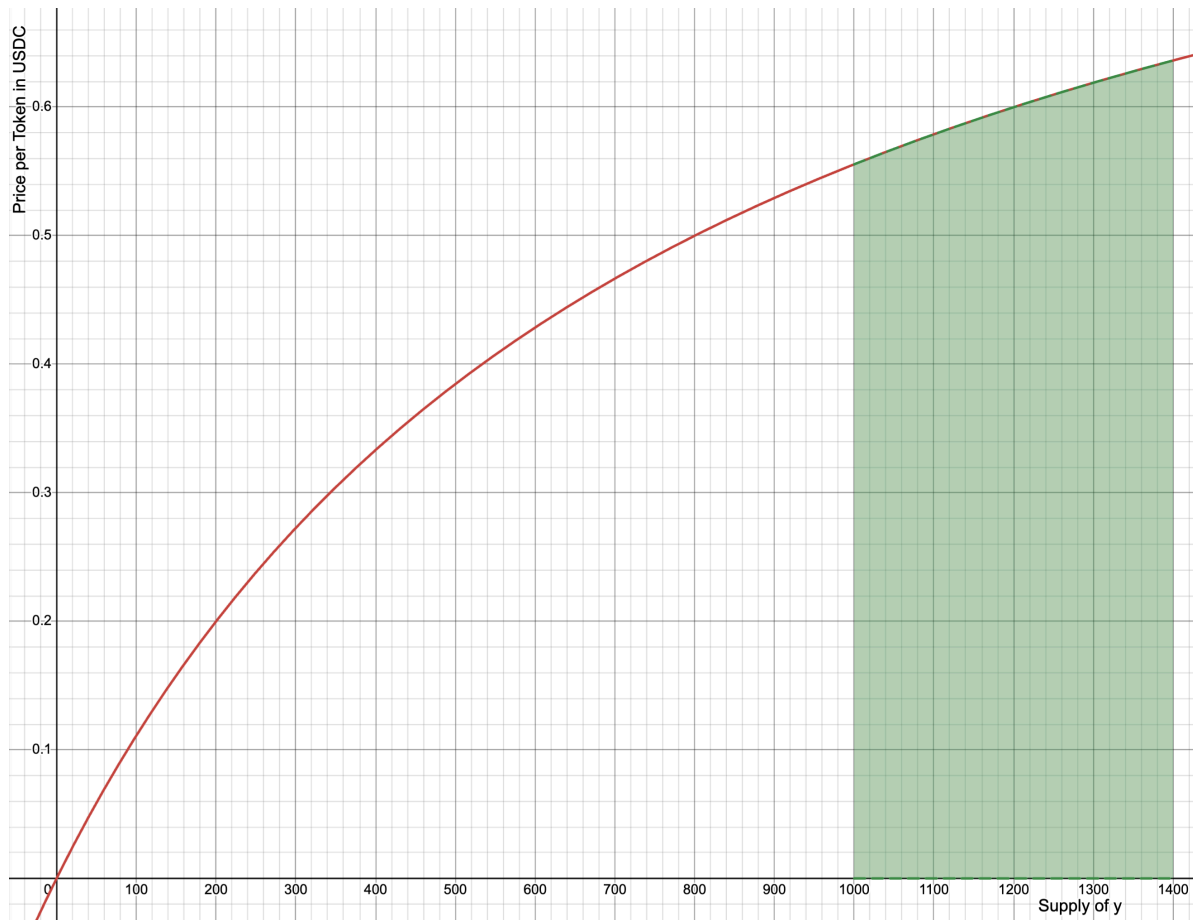
When the pool closes, the pot is split amongst the winners. In this case there is 1,154.41 USDC in the pot. If the result is yes or no, the 1% is deducted and the rest is split amongst the winners. If yes wins, the payout will be 0.816 USDC and if no wins, the payout is 1.43 USDC. If it is inconclusive, the creator fee will not be charged and the pot will be split amongst all holders of n and y.

Next steps

Fox is an experimental protocol for us to make bets on fun questions with the community. At this alpha stage, markets are created by us.

Over time, we might change the curve behind the program and further develop market mechanics. We are also interested in developing a community-based resolution and arbitration process so that we can allow anyone to create their own market on questions they find interesting, and tap into the full integration of the protocol with Jupiter.