The Geometry of Belief: Understanding the Euclidean Norm ICBS

Introduction

This paper is written for developers building prediction markets, researchers in mechanism design, and traders who want to understand ICBS mechanics. We assume familiarity with basic calculus (partial derivatives) and cryptocurrency concepts (AMMs, bonding curves). No prior knowledge of ICBS is required.

Standard prediction markets bound prices to [0,1] because shares settle to fixed payouts of \$0 or \$1. The Euclidean norm Inversely Coupled Bonding Surface (ICBS) [1] presents an alternative mechanism where settlement rebalances reserves rather than paying fixed amounts. Prices are thus not bounded to [0,1] and do not directly represent probabilities. Instead, the ratio of reserves encodes the market's probability forecast.

At settlement, the actual outcome (0 or 1) determines how reserves rebalance between YES and NO sides. Accurate predictions gain purchasing power, inaccurate predictions lose it.

Inverse Coupling

In traditional bonding curves, buying token A doesn't affect token B's price. ICBS couples them: buying YES pushes NO's price down, and vice versa. The Euclidean norm cost function $C(s_{YES}, s_{NO}) = \lambda\sqrt{s_{YES}^2 + s_{NO}^2}$ implements this through geometric distance, producing circular iso-cost curves where movement along any direction changes both prices.

The Cost Function

Core Formula

The cost function defines the total cost to mint the current token supplies. It's not the asset itself, but a mathematical constraint that determines prices (via derivatives) and ensures the invariant $\text{TVL} = C$. The Euclidean norm ICBS uses:

$$C(s_{YES}, s_{NO}) = \lambda \cdot \sqrt{s_{YES}^2 + s_{NO}^2} \tag{1}$$

where s_{YES} and s_{NO} are the supplies of YES and NO tokens, and λ is a fixed scaling constant. Geometrically, this measures the Euclidean distance from the origin in the (s_{YES}, s_{NO}) plane. Every point on a circle of radius r costs $\lambda \cdot r$. Picture concentric circles in (s_{YES}, s_{NO}) space: trading moves outward from the origin, with each circle representing a constant cost level.

Lambda: The Unit of Account

At deployment, someone deposits D dollars (e.g., \$100) and allocates it between YES and NO sides. For a 50/50 split at initial price $p_0 = \$1$, this creates $s_{YES} = s_{NO} = 50$ tokens. The scaling constant λ is then derived to ensure the cost function equals the deposit:

$$\lambda = \frac{D}{\sqrt{s_{YES}^2 + s_{NO}^2}} \tag{2}$$

This fixes λ permanently (e.g., $\lambda = 100/\sqrt{50^2 + 50^2} = 1.414$), establishing a consistent unit of account. Markets of different sizes have identical percentage-based price dynamics, enabling cross-market comparison.

Marginal Prices

Prices emerge as partial derivatives, the slopes of the cost landscape:

$$p_{YES} = \frac{\partial C}{\partial s_{YES}} = \lambda \cdot \frac{s_{YES}}{\sqrt{s_{YES}^2 + s_{NO}^2}} \tag{3}$$

$$p_{NO} = \frac{\partial C}{\partial s_{NO}} = \lambda \cdot \frac{s_{NO}}{\sqrt{s_{YES}^2 + s_{NO}^2}} \tag{4}$$

Each token's price increases with its own supply but is suppressed by the opposing side, the essence of inverse coupling.

Reserves and the Invariant

Virtual reserves are defined as $r = s \cdot p$:

$$r_{YES} = s_{YES} \cdot p_{YES} = \lambda \cdot \frac{s_{YES}^2}{\sqrt{s_{YES}^2 + s_{NO}^2}} \tag{5}$$

$$r_{NO} = s_{NO} \cdot p_{NO} = \lambda \cdot \frac{s_{NO}^2}{\sqrt{s_{YES}^2 + s_{NO}^2}} \tag{6}$$

The total value locked becomes:

$$\text{TVL} = r_{YES} + r_{NO} = \lambda \cdot \sqrt{s_{YES}^2 + s_{NO}^2} \tag{7}$$

This equals the cost function exactly. This **on-manifold property** ensures solvency: total claimable value always matches what the vault holds. During settlement, reserves can rebase while preserving the invariant, enabling reward redistribution without minting or burning tokens.

Properties and Implications

The Euclidean norm ICBS has several key properties that make it well-suited for prediction markets:

1. Inverse Coupling. The two sides compete directly. Mathematically:

$$\frac{\partial p_{YES}}{\partial s_{NO}} = -\lambda \cdot \frac{s_{YES} s_{NO}}{(s_{YES}^2 + s_{NO}^2)^{3/2}} < 0 \tag{8}$$

Buying NO directly lowers YES's price, and vice versa. This creates genuine opposition between beliefs rather than independent liquidity pools.

2. Self-Scaling Liquidity. Buys move supply further from the origin in (s_{YES}, s_{NO}) space (sells move toward it). In active two-sided markets, net movement is outward. Since TVL $= \lambda \sqrt{s_{YES}^2 + s_{NO}^2}$, trading activity automatically grows liquidity. No external LPs needed. Markets bootstrap from minimal deposits and scale organically with volume. This differs fundamentally from LMSR[1], where the subsidy parameter b caps liquidity, and from AMMs, where LPs must deposit capital. In ICBS, trading itself creates liquidity.

3. Geometric Simplicity. Iso-cost curves are circles: every point at distance r from the origin costs $\lambda \cdot r$. This makes the mechanism intuitive and the math tractable. Only square roots, no fractional powers or exponentials.

4. Fixed Unit of Account. Since λ is fixed at deployment, prices scale consistently. A $100 pool and a $10,000 pool have the same percentage price impact for equivalent percentage trades. This enables:

- Cross-market comparison

- Predictable trading dynamics as markets grow

- Consistent pricing behavior across different market sizes

5. Arbitrarily Large Returns. Prices range from 0 to λ:

$$\lim_{s_{YES} \to \infty, s_{NO} \text{ fixed}} p_{YES} = \lambda \tag{9}$$

Early traders who buy near zero can see prices approach λ, yielding arbitrarily large (though bounded by λ) returns. Unlike LMSR's fixed [0,1] probability bounds, this rewards early conviction rather than just tracking consensus.

6. Settlement Compatibility. The on-manifold property (TVL always equals cost function) ensures solvency through settlement. Combined with virtualization, reserves can rebase according to the actual outcome without minting or burning tokens.

---

[1]Logarithmic Market Scoring Rule: a prediction market mechanism where a market maker subsidizes liquidity with a fixed budget b, bounding maximum loss but capping liquidity growth.

Settlement Mechanism

At resolution, the actual outcome determines how reserves rebalance between YES and NO sides. The market's current prediction is:

$$q = \frac{r_{YES}}{r_{YES} + r_{NO}} \tag{10}$$

For a binary outcome where the event either happens ($x = 1$) or doesn't ($x = 0$), settlement factors scale each side's reserves based on the outcome:

$$f_{YES} = \frac{x}{q} \tag{11}$$

$$f_{NO} = \frac{1 - x}{1 - q} \tag{12}$$

If the event happens ($x = 1$), YES holders gain value ($f_{YES} > 1$, $f_{NO} < 1$). If it doesn't ($x = 0$), NO holders gain value.

For example, suppose a market predicting whether it will rain tomorrow has $r_{YES} = \$65$ and $r_{NO} = \$35$ (so $q = 0.65$). If it doesn't rain ($x = 0$), then $f_{YES} = 0/0.65 = 0$ and $f_{NO} = 1/0.35 = 2.857$. Post-settlement: $r'_{YES} = \$0$, $r'_{NO} = \$100$. YES holders lose everything, NO holders gain all value.

Virtualization via Square-Root Scaling

Settlement adjusts prices without burning tokens. Scale parameters σ convert display supplies to virtual supplies: $s_{\text{virtual}} = s_{\text{display}}/\sigma$. During settlement, σ scales by the *square root* of settlement factors:

$$\sigma'_{YES} = \frac{\sigma_{YES}}{\sqrt{f_{YES}}} \tag{13}$$

$$\sigma'_{NO} = \frac{\sigma_{NO}}{\sqrt{f_{NO}}} \tag{14}$$

Why square roots? Reserves equal supply times price: $r = s \cdot p$. To make reserves scale by f while keeping supply constant, we need prices to scale by f. But prices depend on virtual supply *ratios* through the cost function. By scaling σ by $\sqrt{f}$, virtual supplies scale by $\sqrt{f}$, which makes reserves (supply^2 via the norm) scale by exactly f.

Reserves rebalance directly:

$$r'_{YES} = r_{YES} \cdot f_{YES} \tag{15}$$

$$r'_{NO} = r_{NO} \cdot f_{NO} \tag{16}$$

Total vault balance: $r'_{YES} + r'_{NO} = r_{YES} + r_{NO}$ (preserved). Capital flows from incorrect predictions to correct ones. Token holders gain or lose purchasing power based on accuracy.

References

[1] Nick Williams and Vitalik Buterin, *Better Curation via Inversely Coupled Bonding Surfaces*, Ethereum Research, 2020. https://ethresear.ch/t/better-curation-via-inversely-coupled-bonding-surfaces/7613/2