FINA4150

OTC ACCUMULATOR ON CRYPTO – WRITTEN REPORT

Group Project Question 3

Chan Cheuk Hang 1155170210

Ho Man Hin 1155193238

Lin Chun Tat 1155193613

Ng Tsz Ho 1155175257

December 5, 2025

Contents

Chapter 1

Executive Summary

As of 19 November 2025, with ETH spot at \$3,050, we priced a 13-week weekly-settled accumulator featuring 2× gearing (7 ETH accumulated when spot $\geq$ Forward Price, 14 ETH when below), a 110% knock-out effective from week 4, and a 3-week guaranteed accumulation period. The contract is structured to have zero upfront premium.

Using the ETH option chain from Deribit, we constructed seperate bid and ask implied volatility surfaces, calibrated Dupire local volatility models in time and log-moneyness space, valued the product via 100,000 Monte Carlo paths.

The fair Forward Price that makes the contract zero-cost is:

- Bid side (client sells accumulator to desk): 84.2329% of spot, \$2,569.10
- Ask side (client buys accumulator from desk): 83.7660% of spot, \$2,554.86

The sub-100% strike reflects the pronounced downside volatility skew in ETH options, which significantly raises the probability of double accumulation on weak weeks while the 110% knock-out caps upside risk.

Chapter 2

Data Preparation

2.1 Option Data

The project uses ETH option data from Deribit.com and processes it in four ways:
API IV (`Accumulator.py`)

- Retrieve option mark prices from the Deribit API.
- Compute implied volatility by inverting the Black–Scholes formula.

Mark IV (`Accumulator_markiv.py`)

- Use Deribit's "mark implied volatility" directly from the API response.
- No inversion required; smoother surface.

CSV Mid IV (`Accumulator_csv.py`)

- Use exported Deribit option data.
- Implied vol is taken as (bid IV + ask IV)/2.

CSV Bid/Ask IV (`Accumulator_csv_bidask.py`)

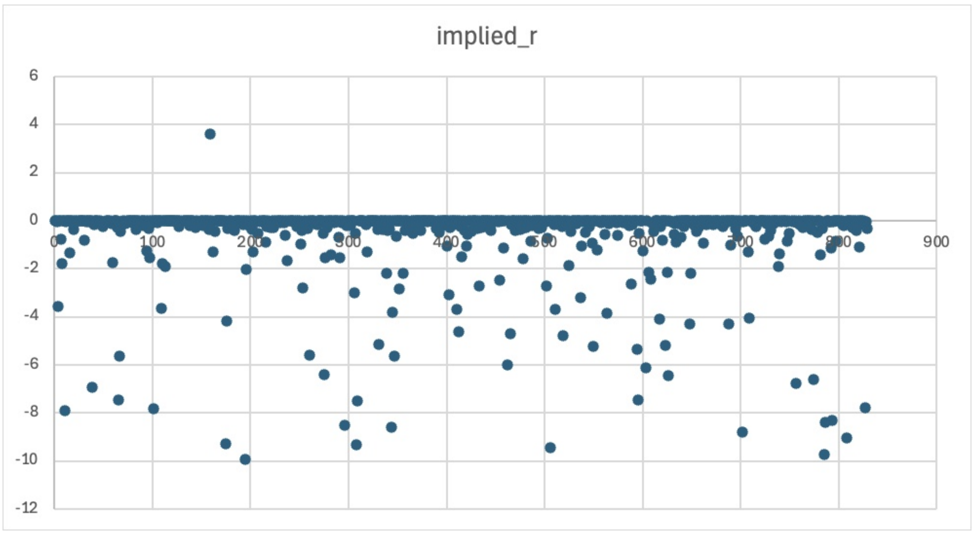
- Plot the implied vol separately for bid IV and ask IV.
- Allows analysis of the bid–ask volatility spread and liquidity effects.

For each dataset, the following fields are extracted and aligned: strike, maturity, option type, bid, ask, mark price, mark IV, and ETH spot.

Options with zero quotes, missing values, or extremely far OTM strikes are filtered out.

We used four different IV datasets (API, mark IV, mid, bid/ask) to cross-check noise and stabilize the volatility surface before calibration.

2.2 Risk-Free Rate



Since the implied interest rates computed from market option prices cluster very close to zero, we set $r = 0$ as a reasonable and stable approximation.

Chapter 3

Implied Volatility

3.1 Mathematical Foundation

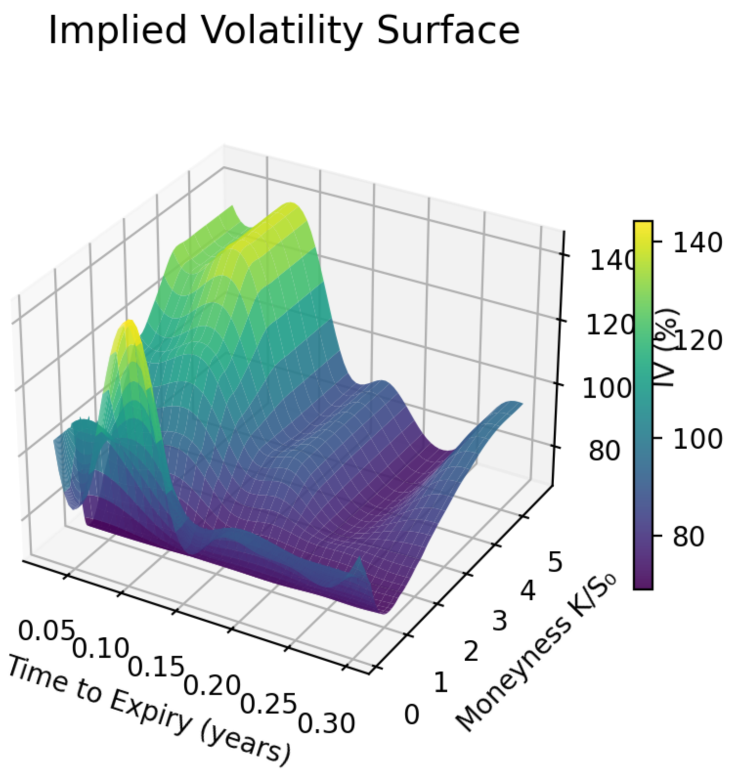
Mathematically, implied volatility (σ_{imp}) is obtained by numerically solve for σ_{imp} in the following equation:

$$C = f(\sigma_{\text{imp}}, \cdot)$$

where C is the value of the option and f is a pricing model depending on σ_{imp} . However, since Deribit offers dataset for σ_{imp}, we used the dataset instead. We confirmed the validity of the dataset by solving for the σ_{imp} using Newton's method.

3.2 Methodology

We extracted the σ_{imp} directly from Deribit for every listed ETH option converted to decimal. Then, we interpolated the IV onto a regular grid by first organizing the IV as a function of log-moneyness $m \triangleq \ln\left(\frac{K}{S_0}\right)$ and the time to expiry T, then fit a smooth bivariate cubic spline. Using Python, we produced the following surface for σ_{imp} where the axis uses moneyness M instead of m for display purpose.



3.3 Interpretation

There are few points to note from the graphs itself:

- Skew is strong as time to expiry gets closer, in particular for $T \in [0.05, 0.15]$. It indicates that the risk is higher for close-to-expiry ETH options. However, the skew flattens once it go beyonds the cutting point $T = 0.15$
- The skew is extremely inverted. As moneyness increases, the σ_{imp} increases, which signifies a huge market demand in bidding crash protection and the fear of the traders facing a violent reversal.

Chapter 4

Local Volatility

4.1 Model Selection

Since the Accumulator contract contains a Knock-Out barrier at 110% of the initial spot price, the valuation is highly sensitive to the volatility skew. A standard Black-Scholes model with constant volatility would fail to capture the market's implied probability of the asset hitting the barrier. To address this, we utilized the Dupire Local Volatility model, which extracts a deterministic, time-and-level dependent volatility surface $\sigma_{\text{loc}}(T, K)$ consistent with current market option prices.

4.2 Mathematical Formulation

We implemented the classic Dupire equation, which relates the local variance to the partial derivatives of the Call price surface $C(K, T)$:

$$\sigma_{\text{loc}}^2(K,T) = \frac{\frac{\partial C}{\partial T} + (r-q)K\frac{\partial C}{\partial K} + qC}{\frac{1}{2}K^2\frac{\partial^2 C}{\partial K^2}}$$

Given that the risk-free rate r and dividend yield q are set to zero for this analysis, the equation simplifies to the ratio of the time-decay of the option price to the curvature of the price with respect to the strike:

$$\sigma_{\text{loc}}(K,T) = \sqrt{\frac{\frac{\partial C}{\partial T}}{\frac{1}{2}K^2\frac{\partial^2 C}{\partial K^2}}}$$

4.3 Numerical Implementation

The calibration was performed in Python using a discretized grid approach. The implementation steps were as follows:

4.3.1 Grid Construction

We constructed a dense rectangular grid to ensure numerical stability:

- Time Dimension (T): 50 linearly spaced points from 0.01 years to 0.50 years
- Space Dimension (K): 90 points spaced uniformly in log-moneyness ($m = \ln(K/S_0)$), covering the range $m \in [-0.9,\, 0.9]$

4.3.2 Computation of Partial Derivatives

Directly differentiating market prices can be unstable. Instead, we computed the derivatives semi-analytically using the Black-Scholes Greeks and the implied volatility surface constructed in the previous section:

Time Derivative ($\partial C/\partial T$): We applied the total derivative rule, combining the standard Black-Scholes Theta Θ_{BS} with the sensitivity to changes in implied volatility (Vega):

$$\frac{\partial C}{\partial T} \approx \Theta_{\text{BS}} + \nu_{\text{BS}} \frac{\partial \sigma_{\text{imp}}}{\partial T}$$

The term $\partial \sigma_{\text{imp}}/\partial T$ was approximated using central finite differences on the IV spline

Strike Derivative $\left(\partial^2 C \middle/ \partial K^2\right)$: We compute derivatives in log-moneyness space (m) first, then converted them to strike space (K) using the chain rule:

$$\frac{\partial^2 C}{\partial K^2} = \frac{1}{K^2}\left(\frac{\partial^2 C}{\partial m^2} - \frac{\partial C}{\partial m}\right)$$

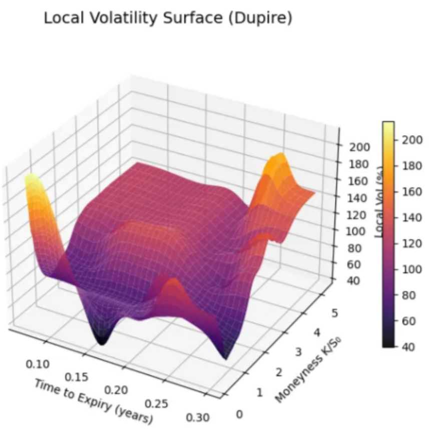
4.3.3 Surface Regularization (Smoothing)

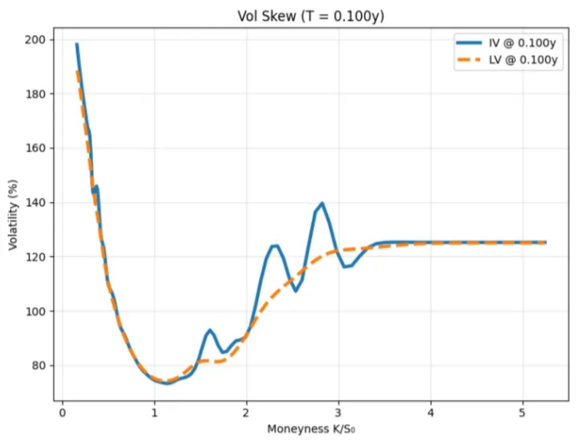
To ensure a stable Monte Carlo simulation, we applied a two-stage smoothing process to the resulting local volatility matrix:

- Median Filter: A 3×3 kernel was applied to remove isolated outliers without blurring sharp edges
- Gaussian Filter: We applied a Gaussian kernel with $\sigma_T = 1.8$ (time axis), and $\sigma_K = 2.5$ (strike axis) to ensure the surface is C^1 continuous

4.3.4 Boundary Constraints

To prevent numerical explosion at the grid boundaries, the final local volatility values were clipped to a realistic range of $[2\%, 400\%]$. Any remaining NaN values resulting from arbitrage violations were filled via linear interpolation.
Graph produced:





Interpretation:

- Local Volatility Surface:
  There is a distinct, high-volatility "ridge" traversing the surface at the 0.20–0.25 year maturity mark (roughly 2.5 months out). Unlike a standard term structure which might gently slope, this specific hump implies the market is pricing in a significant discrete event in that specific window that is expected to drive volatility higher regardless of the strike price.

- Vol Skew:
  The Local Volatility (Orange) curve cuts through the noise, presenting a smooth, theoretical volatility progression. It effectively filters out the idiosyncratic noise of individual strike prices to reveal the underlying volatility structure, which is a steep descent from the downside and a gradual rise (convexity) towards the upside.

Chapter 5

Monte Carlo Stimulation

5.1 Underlying Price Dynamics

We model the ETH price using the local volatility extension of geometric Brownian motion:

$$\mathrm{d}S_t = rS_t\,\mathrm{d}t + \sigma_{\mathrm{loc}}(t,\,S_t)S_t\,\mathrm{d}W_t$$

where

- $r = 0$ (Explained in data preparation)
- $\sigma_{\mathrm{loc}}(t,\,S_t)$ is the deterministic local volatility function via the Dupire formula

5.2 Path Discretisation

We simulate paths on a weekly grid ($\Delta t = 7/365.25$ years) using the exact Euler–Maruyama scheme:

$$S_{t+\Delta t} = S_t \exp\left[\left(r - \frac{1}{2}\sigma^2_{\mathrm{loc}}(t, S_t)\right)\Delta t + \sigma_{\mathrm{loc}}(t, S_t)\sqrt{\Delta t}\,Z\right], \qquad Z \sim \mathcal{N}(0,1)$$

At each weekly step (in week loop), local volatility is re-evaluated at the current calendar time t and the simulated spot level S_t using the calibrated Dupire surface (cubic spline in time and log-moneyness), illustrated in following code:

```
for w in range(weeks):
t = w * dt
S_c = S[active, w]
logm_c = np.log(S_c / S0)    # log-moneyness
vol = lv_surf(t, logm_c, grid=False)
vol = np.clip(vol, 0.05, 4.0)

dW = np.random.normal(0, sqrt_dt, size=vol.shape)
S[active, w+1] = S[active, w] * np.exp((r - 0.5*vol**2)*dt + vol
```

5.3 Implementation of Contract Features

100,000 paths (fixed seed 42) are simulated. The term-sheet logic is coded as follows:

Initial settings (outside the weeks loop)

- KO=110% of initial spot

```
np.random.seed(seed)
fp = fp_pct * S0
ko = 1.10 * S0
weeks, dt = 13, 7/365.25
sqrt_dt = np.sqrt(dt)

S = np.full((N, weeks+1), S0)
payoff = np.zeros(N)
active = np.ones(N, dtype=bool)
week_stds = []
```

Weakly gearing (shares = 7 if settle $\geq$ FP else 14)

```
settle = S[active, w+1]
weekly_shares = np.where(settle >= fp, 7.0, 14.0)
```

Cash settlement each week

- Aggergate the difference between the settle price and forward price * the shares accumulated to the payoff

```
weekly_payoff = weekly_shares * (settle - fp)
disc = np.exp(-r * (w+1)*dt)
payoff[active] += weekly_payoff * disc
```

KO logic

- KO only from week 4 onward because of the 3-week guaranteed period
- Early termination on KO by deactivated paths from knock-out week onwards

```
if w >= 3:   # i.e., weeks 4 to 13 (w = 3,4,...,12)
ko_hit = settle >= ko
if ko_hit.any():
# Paths that knock out: deactivate them (no more accumulation)
active[active] = ~ko_hit
if not active.any():
break
```

5.4 Solving the Zero-Cost Forward Price

- `brentq` finds the unique root of Expected payoff (FP) = 0
- The wide bracket [0.60, 1.10] safely contains the solution in all market conditions
- Convergence tolerance 10^{-5} i.e. final precision better than 0.001% of spot

The fair Forward Price that makes the contract zero-cost is:

- Bid side (client sells accumulator to desk): 84.2329% of spot, \$2,569.10
- Ask side (client buys accumulator from desk): 83.7660% of spot, \$2,554.86