Technical Specification: American Options Consensus Engine (AOCE)

Yash Saxena

December 27, 2025

1 Introduction

The American Options Consensus Engine (AOCE) is a high-frequency valuation and decision-making framework designed to optimize the early exercise of American-style derivatives. Unlike European options, the American variant includes an "early exercise premium." This system identifies the **Absolute Payoff Maxima** by triangulating three distinct mathematical models and enforcing a rigorous statistical safety filter.

2 System Architecture

The system is divided into four functional layers:

2.1 Data Ingestion Layer (Current Backtesting Methodology)

- **Market Pricing:** Utilizes the *Tiingo Python SDK* for adjusted closing prices.
- **Risk-Free Rate (r):** Fetched dynamically via *pandas_datareader* from the Federal Reserve Economic Data (FRED), specifically the 1-Month Treasury Constant Maturity Rate (GS1M) (Section 7.1).
- **Volatility (σ):** Realized volatility is calculated using a rolling 30-day window of logarithmic returns: $\sigma_{ann} = \text{std}(\ln(P_t/P_{t-1})) \times \sqrt{252}$.

2.2 Safety & Filtering Layer (The Blackout)

A 2 week exclusion window (blackout radius can be changed) is applied around corporate actions (earnings and dividends). This mitigates "Jump-Diffusion" risks where the Black-Scholes assumption of continuous price paths is violated.

2.3 Valuation Layer (The Consensus)

The engine computes the option value using three methodologies:

- **Analytical:** Barone-Adesi Whaley (BAW) (Section 8.3).
- **Discrete:** Cox-Ross-Rubinstein (CRR) Binomial Tree (Section 8.2).
- **Continuous:** Crank-Nicolson Finite Difference Method (CN-PDE) (Section 8.1).

2.4 Risk Prediction Layer

Before execution, the system calculates the **Probability of Profit (PoP)** using the Cumulative Distribution Function (CDF) of d_2. Tickers with $PoP < 25\%$ (again user defined threshold) are disqualified to preserve capital. (Refer to Section 6 for more details)

3 Asset Price Dynamics: Geometric Brownian Motion (GBM)

The underlying asset price S_t is assumed to follow a Geometric Brownian Motion (GBM). Its evolution is described by the Stochastic Differential Equation (SDE):

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

The solution to this SDE for the asset price at a future time t, given an initial price S_0, is:

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right)$$

The term $-\frac{\sigma^2}{2}$ is the **Ito's Lemma correction term**, also known as the **convexity adjustment**. It arises because the logarithm is a concave function; when transitioning from arithmetic return to geometric return, this term accounts for the effect of volatility on the average growth rate of the log-price. Specifically:

$$\ln(S_t) \sim N\left(\ln(S_0) + \left(\mu - \frac{\sigma^2}{2}\right)t, (\sigma\sqrt{t})^2\right)$$

4 Option Pricing: Black-Scholes Model

The Black-Scholes model determines the theoretical premium used as the cost basis.

$$C_t = S_0\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2), \quad P_t = Ke^{-r(T-t)}\Phi(-d_2) - S_0\Phi(-d_1)$$

where $d_1 = \frac{\ln(S_0/K)+(r+\frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$ and $d_2 = d_1 - \sigma\sqrt{T-t}$. In this implementation, the risk-free return rate r is dynamically referenced from the 13-week Treasury Bill yield via API to ensure the growth basis reflects current macroeconomic conditions.

5 Implied Volatility (Volatility Solver)

Since σ cannot be solved for analytically, the Newton-Raphson method is employed in `vol_solver.py`. The update rule is:

$$\sigma_{n+1} = \sigma_n - \frac{C_{BS}(\sigma_n) - C_{mkt}}{\text{Vega}}$$

where Vega $= S_0\phi(d_1)\sqrt{T}$. This convergence provides the market-implied diffusion coefficient for the PoP calculation.

6 Probability of Profit (PoP)

The PoP is the statistical likelihood that an option trade will be profitable at expiration.

6.1 Call Option PoP

Profitability occurs if $S_T > X_C$, where $X_C = K + \text{Premium}$:

$$P(S_T > X_C) = 1 - \Phi\left(\frac{\ln(X_C) - \left[\ln(S_0) + (\mu - \frac{\sigma^2}{2})(T-t)\right]}{\sigma\sqrt{T-t}}\right)$$

6.2 Put Option PoP

Profitability occurs if $S_T < X_P$, where $X_P = K - \text{Premium}$:

$$P(S_T < X_P) = \Phi\left(\frac{\ln(X_P) - \left[\ln(S_0) + (\mu - \frac{\sigma^2}{2})(T-t)\right]}{\sigma\sqrt{T-t}}\right)$$

7 Transition to the Real-World Measure (P)

In the context of directional backtesting, the drift parameter μ is transitioned from the risk-neutral rate r to a **Real-World Drift** derived from historical observations. This represents a shift from arbitrage-free pricing to **statistical expectancy modeling**.

7.1 Historical Estimator

The drift is estimated via the expected value of the logarithmic returns over a lookback window L:

$$\hat{\mu} = \frac{1}{L\Delta t}\sum_{i=1}^{L}\ln\left(\frac{S_i}{S_{i-1}}\right) + \frac{\sigma^2}{2}$$

The inclusion of the $\sigma^2/2$ term (Convexity Adjustment) is required to transform the observed geometric growth rate back into the arithmetic drift parameter utilized in the SDE:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

7.2 Threshold Sensitivity

Utilizing $\hat{\mu}$ significantly expands the distribution of PoP values. In regimes of high momentum where $\hat{\mu} \gg r$, the probability density function shifts aggressively, increasing the frequency of $PoP > \tau = 25\%$ signals. This subjects the strategy to **Model Risk**, as the validity of the trade is contingent on the persistence of the observed drift.

8 Exercise Strategy Optimization Methods

8.1 Crank-Nicolson Finite Difference (Continuous Time)

We solve the Black-Scholes PDE:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV = 0 \tag{1}$$

The Crank-Nicolson method applies a second-order accurate central difference in space and a trapezoidal rule in time. Let $V_{i,j}$ be the value at price node i and time node j. The implicit scheme results in a tridiagonal matrix:

$$-a_i V_{i-1}^j + (1+b_i)V_i^j - c_i V_{i+1}^j = a_i V_{i-1}^{j+1} + (1-b_i)V_i^{j+1} + c_i V_{i+1}^{j+1} \tag{2}$$

At each time step, we enforce the American constraint: $V_i^j = \max(V_i^j, \text{Intrinsic Value})$.

8.2 Cox-Ross-Rubinstein Binomial Tree (Discrete Time)

We discretize the price path into N steps. The factors u and d are:

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}}, \quad p = \frac{e^{(r-q)\Delta t} - d}{u-d} \tag{3}$$

We iterate backward from T to $t=0$:

$$V_{node} = \max\left(\text{Intrinsic}, e^{-r\Delta t}[pV_{up} + (1-p)V_{down}]\right) \tag{4}$$

8.3 Barone-Adesi Whaley (Analytical Approximation)

BAW approximates the early exercise premium $f(S)$. For a Put:

$$V_{am}(S) = V_{euro}(S) + A_1(S/S^*)^{q_1} \quad \text{for } S > S^* \tag{5}$$

S^* is the critical price where immediate exercise is optimal.

9 Implementation Logic for Testing (As of Dec 27, 2025)

9.1 Recreating the Data Flow

Developers must ensure that the `TreasuryClient` reindexes the monthly FRED data to a daily frequency using `ffill()`. Without this, the `main.py` loop will fail to find a risk-free rate for daily trading steps.

9.2 The Consensus Execution Trigger

The system triggers an exercise if and only if:

1. $S \leq S^*$ (for Puts) or $S \geq S^*$ (for Calls).
2. The time-value $\theta \approx 0$. Mathematically: $|V_{PDE} - \text{Intrinsic}| < \epsilon$ where $\epsilon = 0.005$.
3. PoP calculated by `RiskPredictor` is above the survival threshold.

10 Developer Summary

To deploy this project: 1. Install dependencies (`tiingo`, `pandas_datareader`). 2. Implement the `solve_banded` logic in `engines.py` for the PDE solver to ensure $O(N)$ efficiency. 3. Use the 30-ticker S&P 500 universe for diversified backtesting across sectors.