# Breakout Trading Strategy

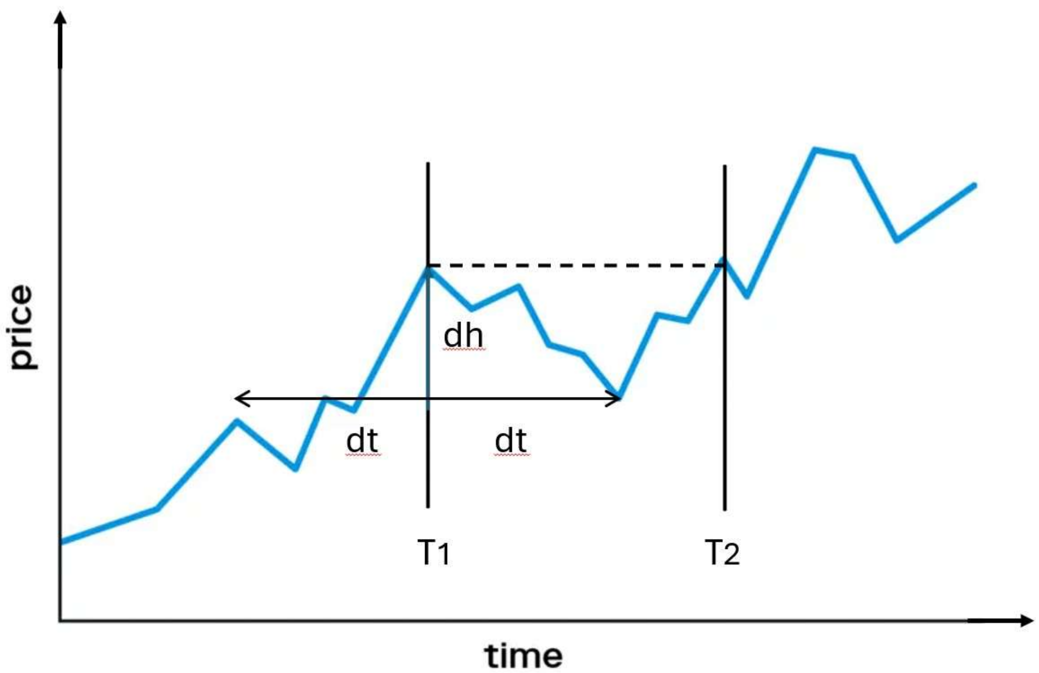
Denis Laurichesse (April 2025), denis.laurichesse@gmail.com

## Introduction

Breakout trading strategies exploit a well-documented phenomenon observed across a wide range of asset classes, including currencies, indices, commodities, and cryptocurrencies. Local extrema — that is, local maxima and minima — naturally define **support** and **resistance** levels in price evolution. By systematically identifying these levels and entering positions when they are reached, traders aim to capture movements where the price continues beyond an established range, thus profiting from sustained directional trends.

This section formalizes the detection and selection of local extrema using two parameters: a time window $dt$ and a height threshold $dh$.

The following figure provides a synthesis of the key components involved in breakout detection.



# Extrema Detection

## Local Maximum Detection

Let $p(t)$ denote the signal (e.g., price) as a function of time $t$. A local maximum is detected at a time $T_1$ if the following two conditions are satisfied:

- **Local Supremum Condition**: The value at $T_1$ must be the maximum within the symmetric window $[T_1 - dt, T_1 + dt]$:

$$p(T_1) = \max_{t \in [T_1 - dt, T_1 + dt]} p(t)$$

- **Height Separation Condition**: There must exist at least one point $t^- \in [T_1 - dt, T_1]$ and one point $t^+ \in [T_1, T_1 + dt]$ such that:

$$p(t^-) \leq p(T_1) - dh \quad \text{and} \quad p(t^+) \leq p(T_1) - dh$$

**Persistence**: Once detected, the maximum at $T_1$ remains valid until a new higher maximum is found at a later time $T_2$ such that:

$$p(T_2) > p(T_1)$$

## Maximum Selection at a Given Time

At any evaluation time $T$, all valid maxima detected within a retrospective window $[T_0, T]$ (with $T_0 < T$) are considered. Among these, the selected maximum corresponds to the one with the lowest amplitude:

$$\hat{T} = \arg \min_{T_1 \in [T_0, T]} p(T_1)$$

## Local Minimum Detection

The detection of local minima follows the same principles as maxima, with reversed inequalities:

- **Local Infimum Condition**:

$$p(T_1) = \min_{t \in [T_1 - dt, T_1 + dt]} p(t)$$

- **Height Separation Condition**:

$$p(t^-) \geq p(T_1) + dh \quad \text{and} \quad p(t^+) \geq p(T_1) + dh$$

**Persistence**: A detected minimum at $T_1$ remains valid until a new lower minimum is found at a time $T_2$ such that:

$$p(T_2) < p(T_1)$$

### Minimum Selection at a Given Time

At any evaluation time $T$, all valid minima within the window $[T_0, T]$ are considered. The selected minimum corresponds to the one with the highest amplitude:

$$\hat{T} = \arg\max_{T_1 \in [T_0, T]} p(T_1)$$

## Trading Rules (Long Side)

This section describes the trading logic for the **long side**. The rules for the short side are symmetric, obtained by reversing the maxima and minima criteria and switching long to short positions.

### Stop Placement and Update

At each evaluation time $T$, the following steps are applied:

1. **Selection of the Valid Maximum**: The valid maximum is selected according to the procedure described above.

2. **Stop Management**:

   - **No Active Stop**: If no stop order is currently active, a new stop order is placed at the selected maximum.
   - **Active Stop Exists**: If a stop order exists and the newly selected maximum is lower than the current stop level, the stop is updated and repositioned at the new maximum.
   - **Higher Maximum**: If the newly selected maximum is higher than the current stop, no update is performed.

3. **Stop Order Parameters**:

   - A **Take Profit** target defining the profit realization level.
   - A **Stop Loss** level, initially set and then dynamically trailed upward as the position moves favorably.

4. **Position Management**:

   - If no position is open, stop orders are managed as described (placement or update).
   - **If a position is already active, no new stop orders are placed.** Only the existing stop linked to the open position is maintained and dynamically adjusted.

## Backtest

The following backtest evaluates the breakout trading strategy applied to the Gold CFD (symbol **XAUUSD**). Historical data were downloaded from Interactive Brokers (IBKR) and are available in the file *`get_IB_XAUUSD.py`*. The data cover the period from **December 2020** to the present.

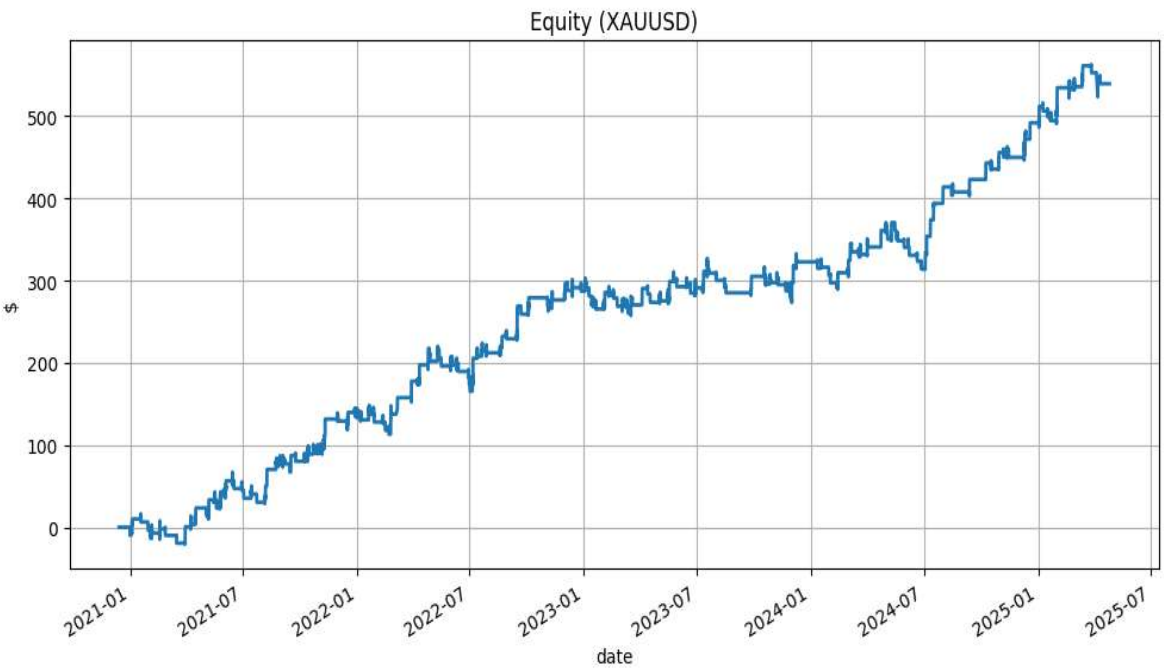
The backtest was performed using the Jupyter notebook *`breakout_strategy.ipynb`*, with the following settings:

- Bar size: 5 minutes
- Window parameter *dt* (number of bars): 720
- Height threshold *dh* (USD): 20
- Take Profit (USD): 20
- Stop Loss (USD): 10

The results are characterized by smooth and consistent performance:

- Average profit per trade: $2.78
- Yearly mean profit: $183
- Yearly standard deviation: $91
- Sharpe ratio: 2.01

These outcomes highlight a favorable balance between profitability and risk control under the selected parameters.



## Asset Classes

The breakout trading strategy can be extended to other asset classes, such as currencies, stock indices, commodities, and cryptocurrencies. However, the **optimal parameter tuning** — including $dt$, $dh$, take profit, and stop loss — varies substantially depending on the asset's volatility and market structure.

**Summary of Tuning for Assets:**

| Asset | Bar Size | $dt$ (bars) | $dh$ | Take Profit / Stop Loss |
|---|---|---|---|---|
| Gold (XAUUSD) | 5 min | 720 | 20 ($) | 20 / 10 ($) |
| S&P 500 (US500) | 5 min | 96 | 100 ($) | 200 / 20 ($) |
| Nasdaq 500 (USTEC) | 5 min | 96 | 30 ($) | 60 / 7 ($) |
| Bitcoin (BTCUSD) | 6 hours | 80 | 200 ($) | 500 / 100 ($) |
| EUR/USD | 1 day | 20 | 0.005 ($) | 0.02 / 0.005 ($) |
| GBP/USD | 1 day | 20 | 0.005 ($) | 0.02 / 0.005 ($) |
| AUD/USD | 1 day | 20 | 0.005 ($) | 0.02 / 0.005 ($) |
| USD/JPY | 1 day | 20 | 0.5 (¥) | 2 / 0.5 (¥) |
| USD/CAD | 1 day | 20 | 0.005 ($) | 0.02 / 0.005 ($) |
| EUR/JPY | 1 day | 20 | 0.5 (¥) | 2 / 0.5 (¥) |
| AUD/JPY | 1 day | 20 | 0.5 (¥) | 2 / 0.5 (¥) |
| GBP/JPY | 1 day | 20 | 0.5 (¥) | 2 / 0.5 (¥) |