Interest Rate Curves Explained

Explanation of the Hybrid Forward Curve Construction Process

This document describes the **Hybrid Forward Bootstrap** methodology used to construct a smooth forward rate curve using splines. It combines **bootstrapping**, **interpolation**, and **spline fitting** techniques to ensure that the curve:

- Accurately reprices financial instruments.
- Provides smooth transitions between different time points.
- Captures the curvature of the forward rate term structure.

---

1. Overview of the Approach

- The spline curve is built using **n+1 nodes** corresponding to different maturities.
- Each node is characterized by:
  - A **time** from valuation.
  - A **discount factor**.
  - A **continuously compounded forward rate**.
  - A **curvature coefficient** (used in the spline construction).
- The first node corresponds to the valuation date, where:
  - The time is **zero**.
  - The discount factor is **1**.

Step-by-Step Approach

1. **Bootstrap a rough estimate** of forward rates using a **constant forward spline**.
2. **Refine the forward rates** using a piecewise **linear** spline.
3. **Use quadratic splines** to introduce curvature for a smoother final curve.

---

2. Step 1: Bootstrapping Forward Rates

The first step involves **bootstrapping** using a **constant forward spline**.

- This method provides initial forward rates f_i **for the first n nodes**.
- An initial estimate of **5%** is assumed for the forward rate at the first node.
- Future nodes are iteratively bootstrapped based on previous estimates.

$$f_i = \tilde{f}_{i-1} + \frac{t_{i-1} - t_{i-2}}{t_i - t_{i-2}} (\tilde{f}_i - \tilde{f}_{i-1})$$

where:

- $\tilde{f}$ are forward rates from the constant forward bootstrap.
- This ensures that initial estimates are available before refining the curve.

---

3. Step 2: Constructing the Hybrid Forward Spline

The forward curve is built using different **spline functional forms**:

1. **Constant Forward Spline**:
   - Assumes the forward rate is constant within each segment:
   $$f(t) = f_{i-1}, \quad \text{for } t \in [t_{i-1}, t_i]$$
   - This is the simplest method but lacks smoothness.
2. **Linear Forward Spline**:
   - The forward rate changes **linearly** between nodes:
   $$f(t) = f_{i-1} + \frac{f_i - f_{i-1}}{t_i - t_{i-1}} (t - t_{i-1})$$
   - Provides a smooth but **piecewise-linear** approximation.
3. **Quadratic Forward Spline**:
   - Introduces a **curvature term** to improve smoothness:
   $$f(t) = f_{i-1} + \frac{f_i - f_{i-1}}{t_i - t_{i-1}} (t - t_{i-1}) + c_i (t - t_{i-1})$$
   - The **curvature coefficient** c_i is bootstrapped iteratively.

Spline Integral and Derivatives

Each spline formulation provides:

- **The integral of the forward rate function**:

$$F(t) = \int f(t)dt$$

- **The derivative of the integral**, used in the Newton-Raphson root-finding method.

These expressions help ensure smooth transition and correct pricing of instruments.

---

4. Step 3: Solving for Forward Rates via Newton-Raphson

Since forward rates must **reprice instruments**, an iterative solver is required.

The system of equations is solved using the **Newton-Raphson method**, which updates the forward rate estimates iteratively:

1. Compute the pricing error:

$$\epsilon_{i,j} = \sum_{k=0}^{n_i} p_{i,k} df_{i,k} + df_{i-1} \sum_{k=n_i}^{n_{i-1}} p_{i,k} \exp(-F_i(t_{i,k}; x_{i,j}))$$

2. Compute the adjustment term:

$$\Delta_{i,j} = df_{i-1} \sum_{k=n_i}^{n_{i-1}} p_{i,k} \frac{\delta F}{\delta x_i}(t_{i,k}; x_{i,j}) \exp(-F_i(t_{i,k}; x_{i,j}))$$

3. Update the forward rate estimate:

$$x_{i,j+1} = x_{i,j} + \frac{\epsilon_{i,j}}{\Delta_{i,j}}$$

This ensures that the calculated forward rates result in correct instrument pricing.

---

5. Summary of the Hybrid Forward Curve Approach

- **Step 1: Bootstrapping** a rough forward curve using a constant forward spline.
- **Step 2: Refining the curve** using linear and quadratic splines.

- **Step 3: Solving forward rates iteratively** using Newton-Raphson.
- **Ensuring smoothness** via the spline curvature coefficients.

Why Use This Approach?

1. **Improves Accuracy**: More realistic term structure modeling.
2. **Ensures Smoothness**: No sharp kinks in the curve.
3. **Bootstraps Instruments Correctly**: Ensures correct pricing.
4. **Flexible**: Can incorporate different spline techniques.

Would you like me to implement this numerically in Python for better intuition?