# Volatility Surfaces, Skews, and Smiles in SOFR Options and Cap/Floors

Generated: 2025-09-24

## Introduction

This document provides a detailed overview of how volatility surfaces, skews, and smiles feed into the pricing of SOFR futures options and cap/floor instruments. It covers the mathematical foundations (Black-76 model, volatility parameterization), the practical role of smiles and skews, and how such surfaces would hypothetically integrate into a Python pricing framework such as the one developed for SOFR products. The goal is to provide analysts with both theoretical and implementation-level insight into volatility modeling and its application in interest rate derivatives.

## 1. Volatility Surfaces

A volatility surface is a 2D or 3D representation of implied volatilities across strikes (moneyness) and maturities. For SOFR products, surfaces are usually constructed from market quotes of options on SOFR futures (short end) and caps/floors (longer end).

Mathematically, the implied volatility surface is represented as:

$$\sigma = \sigma(K, T)$$

where:

- $K$ = strike
- $T$ = maturity
- $\sigma$ = implied volatility

In practice, surfaces are bootstrapped from sparse market quotes using interpolation methods (linear, spline, SVI, SABR). The volatility surface determines the input $\sigma$ in the Black pricing formula used for both futures options and caplets.

## 2. Volatility Skews and Smiles

The volatility smile refers to the U-shaped pattern of implied volatilities as a function of strike (for a fixed maturity). A skew refers to an asymmetric smile, where volatilities are higher for either low strikes (put-skew) or high strikes (call-skew).

For example, in interest rate markets, cap/floor implied vols often show smirks (downward slopes), reflecting market demand for downside protection. This is embedded in pricing via:

$$V = f(F, K, T, \sigma(K,T))$$

so that option value depends on the local volatility corresponding to its strike and tenor.

In the code, this would manifest as retrieving the correct $\sigma$ from a volatility surface data structure, e.g., `sigma = vol_surface.get_vol(K, T)`.

## 3. Black-76 with Volatility Surface

The Black-76 formula for pricing a call option on a forward (futures or caplet) is:
$$ C = DF \cdot (F N(d_1) - K N(d_2)) $$
$$ d_1 = \frac{\ln(F/K) + 0.5\sigma^2T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T} $$
where:
- $F$ = forward rate
- $K$ = strike
- $DF$ = discount factor
- $\sigma$ = implied volatility (from the surface)
- $T$ = time to maturity

If $\sigma$ is taken directly from the volatility surface as a function of strike and maturity, the option price reflects the smile/skew. In the Python codebase, this corresponds to replacing fixed volatility inputs with surface lookups.

## 4. Cap/Floor Pricing with Volatility Smile

A cap is priced as the sum of caplets, each priced with Black-76 using forward rates and discount factors. The volatility smile enters through:
$$ \text{Caplet Price} = DF \cdot \Delta \cdot (F N(d_1) - K N(d_2)) $$
where $\sigma$ is selected based on each caplet's strike and maturity bucket from the surface. For a floorlet, the put version of Black-76 is used with the same volatility lookup.

In implementation, a loop across reset periods would pull the correct volatility for each caplet maturity and strike combination. For example:
```python
for t in maturities: sigma = vol_surface.get_vol(K, t) price += black_caplet(F, K, sigma, t, DF)
```

## 5. Integration into the Python Code

In the custom pricing library, volatility surfaces can be represented as gridded objects with interpolation (bilinear, spline, SABR calibration). Hypothetically, the following hooks can be inserted:

- **For Futures Options:** In `black76_fut`, replace `sigma` parameter with a surface lookup.
- **For Caplets:** In `price_caplet_black`, replace fixed `sigma` with a function call to surface.
- **For Swaptions (future extension):** Surfaces extend naturally to strike/tenor matrices.

Pseudo-code integration:
```python
class VolSurface: def __init__(self, strikes, maturities, vols): self.grid = (strikes, maturities, vols) def get_vol(self, K, T): return interp2d(self.grid, K, T)
```

## 6. Advanced Volatility Models

While Black-76 assumes lognormal rates, practitioners often use SABR (Stochastic Alpha Beta Rho) or SVI (Stochastic Volatility Inspired) parameterizations to fit entire volatility surfaces. These models enforce no-arbitrage and provide smoothness. In a Python workflow, these models can be calibrated offline and plugged into the surface object.

Example SABR implied volatility formula:
$$ \sigma_{SABR}(F,K) \approx \alpha \frac{1}{(F K)^{(1-\beta)/2}} \cdot \frac{z}{x(z)} \cdot \left[1 + ( (1-\beta)^2/24 \ln^2(F/K) + ... )T \right] $$
where parameters ($\alpha,\beta,\rho,\nu$) are calibrated to market quotes.

## 7. Risk and Greeks under a Vol Surface

Greeks (Delta, Gamma, Vega, Theta) are all affected by volatility surfaces. In particular, Vega is highly surface-dependent, as it measures sensitivity to the local volatility at a given strike and maturity. Smile dynamics also play a role: risk systems need to capture how the surface moves when the underlying forward shifts (sticky-strike vs sticky-delta).

In code, Greeks computed from Black-76 would be parameterized by the volatility returned by the surface. This provides realistic risk attribution in PnL explains for options and cap/floor positions.

## Conclusion

Volatility surfaces, skews, and smiles are critical to accurate pricing and risk management of SOFR options and cap/floor instruments. The Black-76 model serves as the core formula, with volatility determined not as a flat input but as a function of strike and maturity. Integrating volatility surfaces into the pricing library allows for realistic valuation, PnL attribution, and hedging analysis. Future enhancements include SABR/SSVI fits, dynamic smile models, and cross-asset calibration.