SOFR Toolkit Supplementary Guide

Generated: 2025-09-24

Introduction

This supplement addresses additional concepts and gaps that were not fully covered in previous guides on SOFR futures, futures options, swaps, and cap/floors. It is designed to provide extra context, implementation notes, and practical considerations for building a complete pricing and risk analysis framework in Python with QuantLib/rateslib and custom modules (such as the interpolation toolkit).

1. Carry and Roll-down Nuances

Carry and roll-down attribution can be more nuanced than a frozen-curve assumption. In practice, risk systems separate:
• **Accrual Carry:** deterministic accrual of fixed vs floating legs.
• **Curve Roll-down:** value change from instruments sliding down the curve.
• **Convexity/Optionality Effects:** futures convexity, option theta, skew dynamics.
• **Cross-currency basis:** if hedging SOFR with non-USD instruments.
These should be decomposed in code into attribution buckets for PnL explain reports.

2. Futures Convexity Adjustment

SOFR futures settle to the compounded SOFR, not the instantaneous forward. Thus, a convexity adjustment is often applied when mapping between futures-implied rates and OIS forwards. In QuantLib, this is handled via convexity correction models. For daily attribution, this may be small but is significant for precise hedging.

3. Smile Dynamics and Hedging

When integrating volatility surfaces, risk systems must decide how smiles move as the underlying forward shifts:
• **Sticky-Strike:** vols fixed at absolute strikes.
• **Sticky-Delta:** vols shift with the moneyness definition.
• **Calibrated Dynamics:** SABR or SVI dynamics fitted to market quotes.
These choices affect option hedging and PnL attribution. In code, this would be implemented as a `vol_surface.get_vol(K, T, mode='sticky_delta')` style method.

4. Data Engineering Requirements

To fully operationalize this framework, robust data handling is required:
• Parsing and storing CME futures and options quotes (daily settlement, surface quotes).
• Parsing OIS swaps and SOFR curve bootstraps.
• Interpolation/cleaning using custom_interpolation (FFT, wavelets, log-linear).
• Storing volatility surfaces in gridded CSVs or databases.
• Scheduled calibration routines for SABR/SSVI parameters.
These tasks often exceed pure quant development and require ETL engineering skills.

5. Greeks Beyond First Order

Beyond Delta, Gamma, and Vega, additional Greeks matter for SOFR products:
• **Theta:** time decay (especially important for cap/floor vega positions).
• **Rho:** sensitivity to discounting curve shifts.
• **Cross-Gamma:** curvature across different curve nodes.
• **Vanna and Volga:** sensitivity of option price to vol + forward cross-moves.
These can be added in code by extending the analytic Greeks in Black-76 or by numerical bump-and-revalue techniques.

6. Risk Reporting Integration

In practice, trading desks require risk decomposition to map across:
• Futures DV01 buckets (IMM packs and bundles).
• Swap DV01s by tenor.
• Option Vegas by expiry/strike buckets.
• Cap/Floor vegas aggregated by tenor.
The Python toolkit should output a standardized risk report (Excel or database-ready) so these can be reconciled against system-of-record reports (Calypso, Murex, etc.).

7. Model Governance Considerations

Any quantitative library must undergo governance. This includes:
• Documentation of model formulas (as in these PDFs).
• Stress testing against extreme rate/vol moves.
• Benchmarking against vendor systems (Bloomberg, ICE, CME).
• Logging and reproducibility of calibration inputs/outputs.
Python prototypes should be validated before deployment to production systems.

Conclusion

This supplement closes several gaps: convexity adjustment, smile dynamics, data engineering, higher-order Greeks, and model governance. Together with prior PDFs, this provides a nearly complete playbook for building and running a SOFR derivatives pricing and risk management toolkit. Future extensions may include swaptions, cross-currency SOFR/ESTR/SONIA instruments, and Monte Carlo simulation engines for exposure and CVA analysis.