QPRLib

Version: 202505
Team: RCQF, PolyU

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

目录

- **QPRLib Introduction**
- **QPRLib Installation**
- **QPRLib Products**
- **QPRLib Pricing**
  - Autocallable
  - Convertible Bond
- **QPRLib Volatility Models**
  - Black Scholes(BS)
  - Local Volatility(LV)
  - Stochastic Volatility(SV)
  - Stochastic Local Volatility(SLV)

QPRLib Introduction

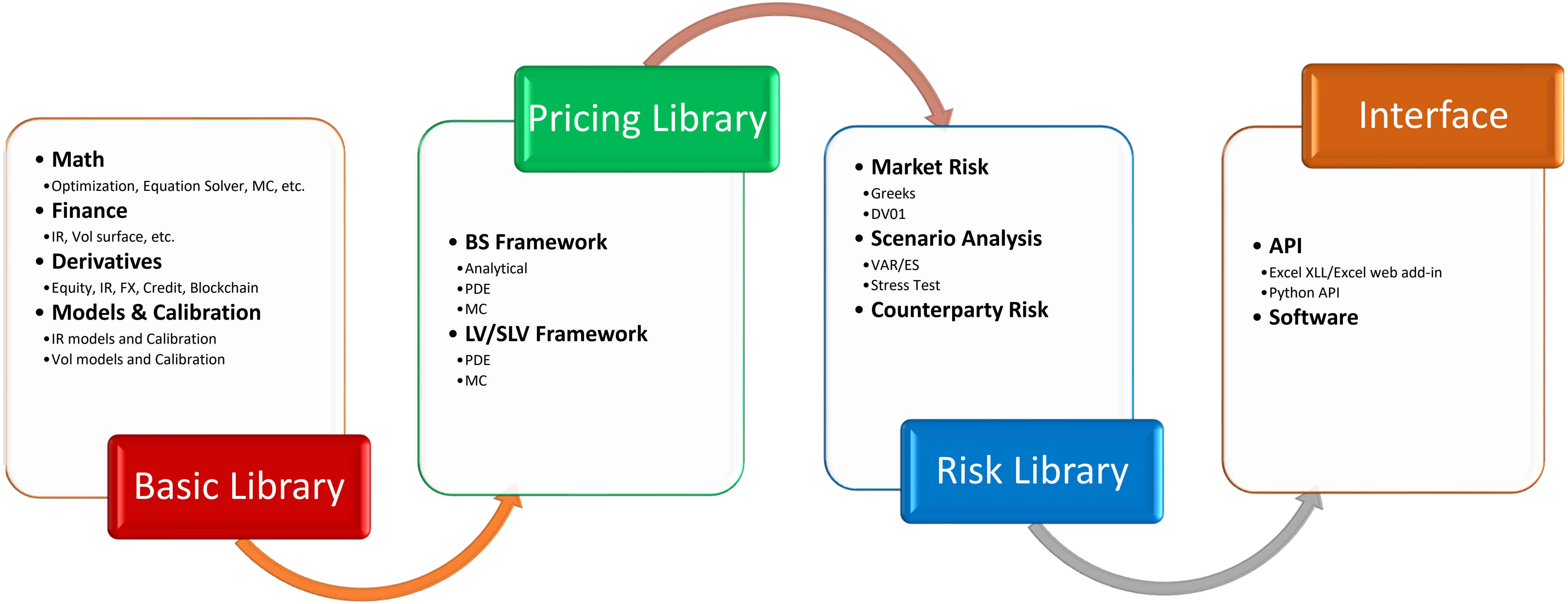
QPRLib (Quantitative Pricing Risk Library) is a foundational library developed for the **Chinese** and **Hong Kong** financial derivatives markets. It provides pricing and risk measurement capabilities for specific over-the-counter (OTC) instruments across equity and fixed-income categories, such as structured snowball products and convertible bonds.

This project is under continuous development, and we envision the final version of the QPRLib library to provide the following functionalities:

- *Capability to handle all common financial derivative instruments (e.g., fixed-income products, OTC equity-structured products, etc.);*
- *Multiple models and algorithms available for user selection within the same product category;*
- *A comprehensive analytical framework including product construction, data import, model calibration, pricing and valuation, as well as risk analysis;*
- *Diversified interfaces and seamless IT integration*

QPRLib Introduction

Structure



QPRLib Introduction

Products

Equity	IR	FX	Credit	Blockchain
☐ European/American	☐ Convertible bond	☐ European/American	☐ CDS	☐ Perpetual contract
☐ Barrier	☐ Perpetual bond	☐ Barrier	☐ CLN	
☐ Digital	☐ Floating rate bond	☐ Asian		
☐ Asian	☐ Callable and puttable bond			
☐ Lookback	☐ Swap			
☐ Warrant	☐ Swaption			
☐ Accumulator	☐ ABS			
☐ Autocallable				

QPRLib Introduction

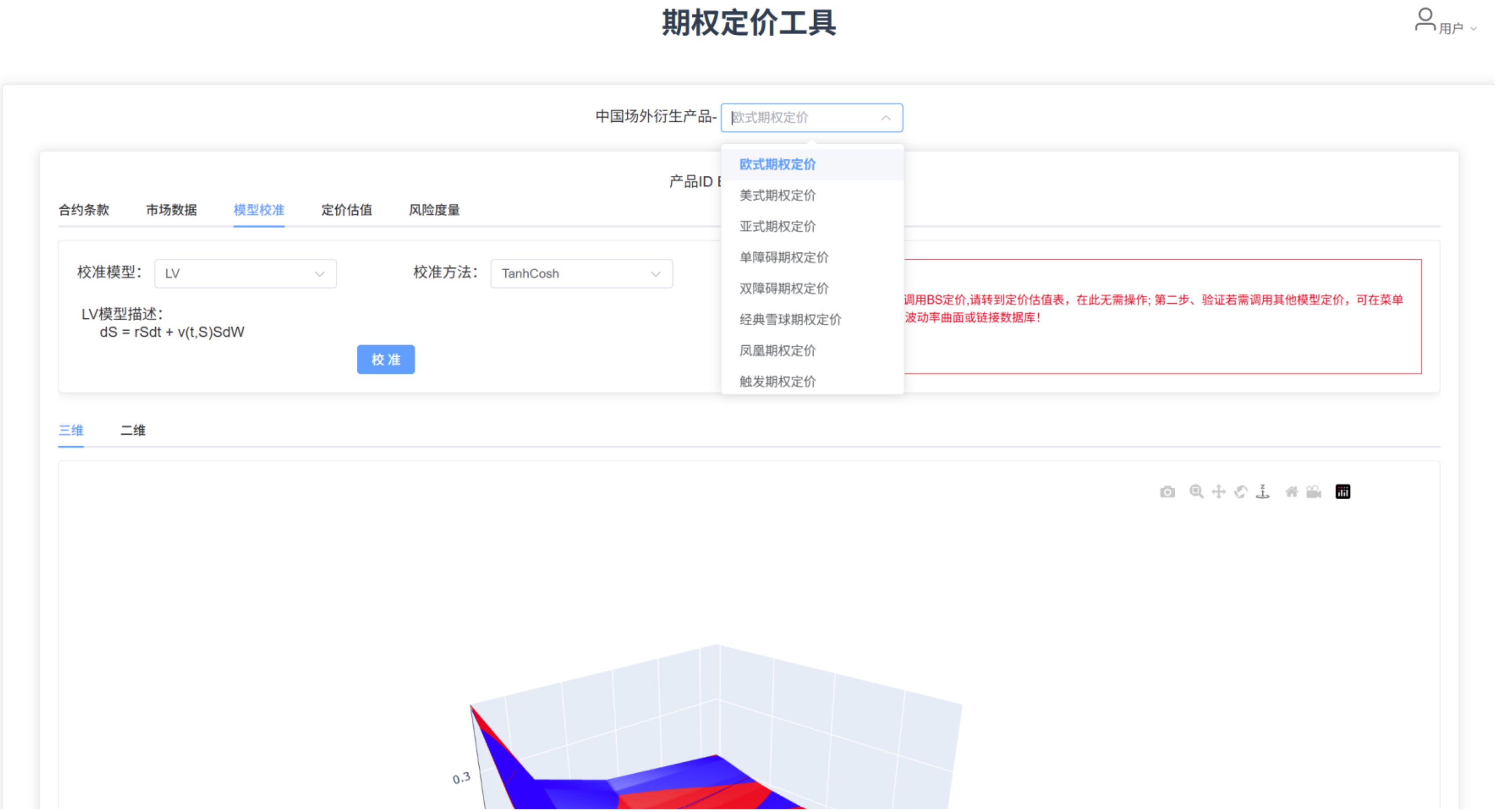
Models & Algorithms For Equity Derivatives

Products	BS			Local Vol		Heston			Stochastic Local Vol	
	Analytical	PDE	MC	PDE	MC	Analytical	PDE	MC	PDE	MC
European	√	√	√	√	√	√	√	√	√	√
American	√	√	X	√	X	X	X	X	X	X
Asian	√	√	√	X	√	X	X	√	X	√
Barrier	√	√	√	√	√	X	X	√	X	√
Digital	√	√	√	√	√	X	X	√	X	√
Lookback	√	X	√	X	√	X	X	√	X	√
Autocallable	X	√	√	√	√	X	X	√	X	√
Accumulator	X	X	√	X	X	X	X	X	X	X
Warrant	√	X	X	X	X	X	X	X	X	X
Three Interval	√	X	√	X	X	X	X	X	X	X
Shark Fin	X	X	√	X	X	X	X	X	X	X
Touch	X	X	√	X	X	X	X	X	X	X

QPRLib Access

WEB

Deployed on 47.115.61.38

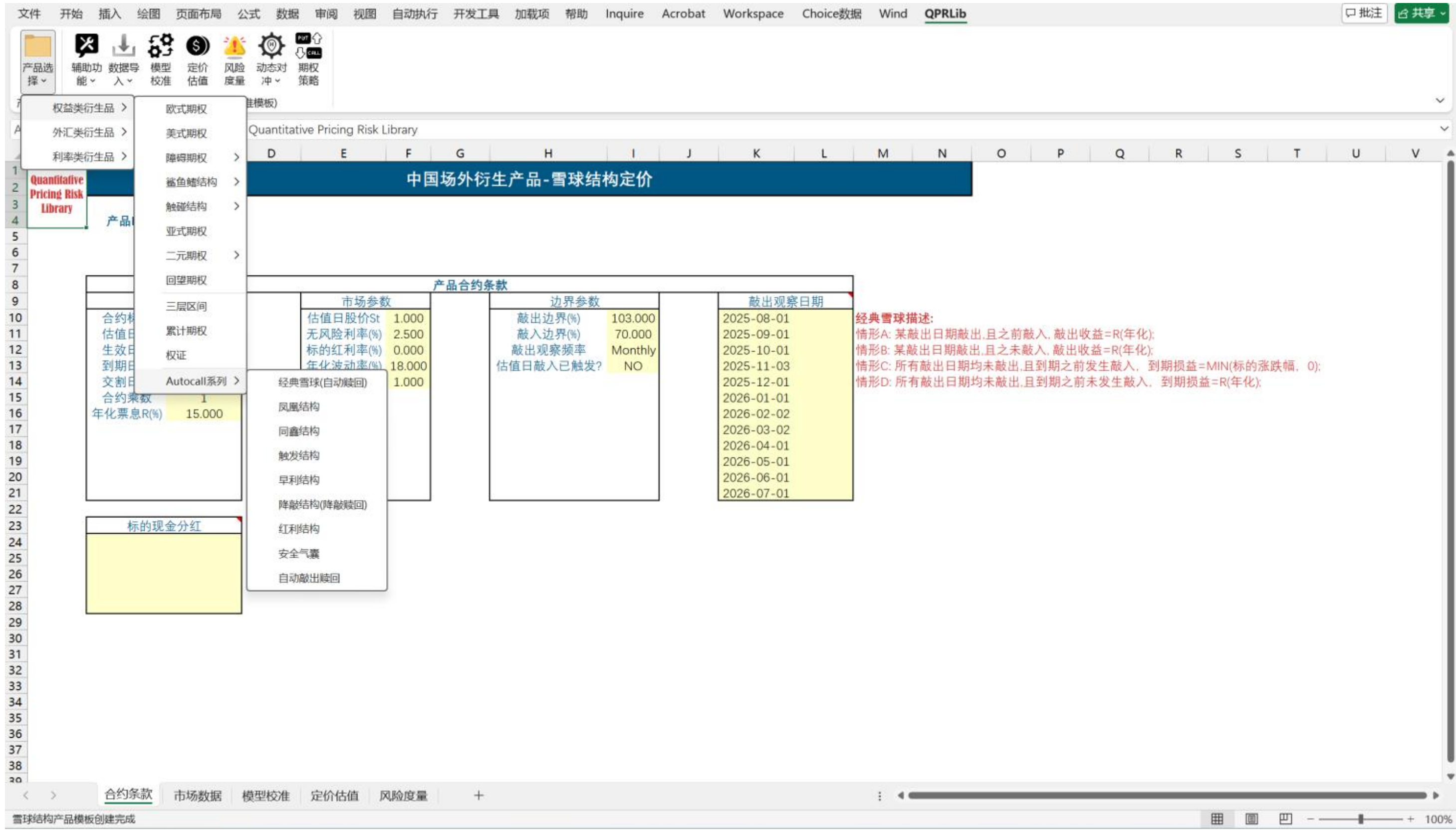


QPRLib Access

<u>Install on Computer as Excel Add-in</u>

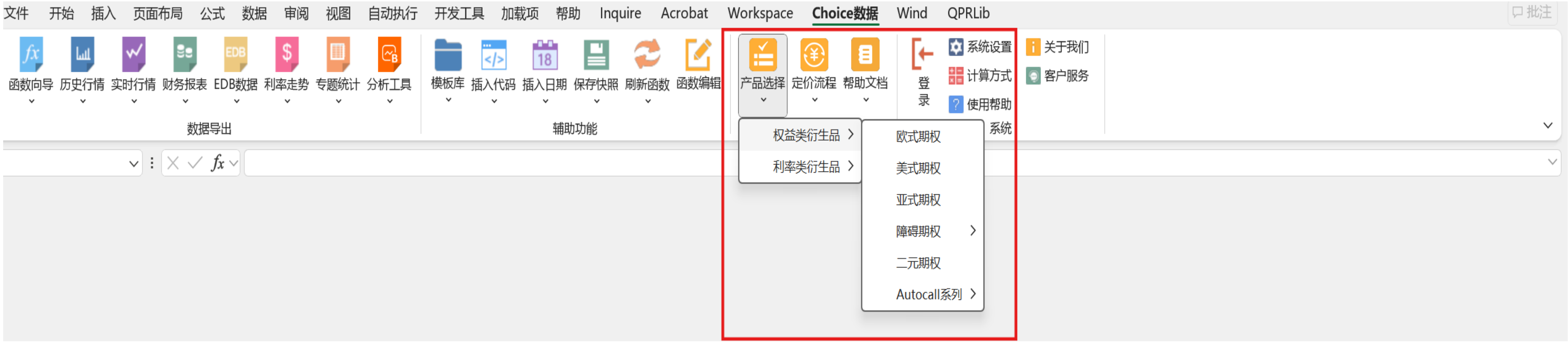
Download the installation package from the PolyU official website (https://www.polyu.edu.hk/rcqf/qprlib-cn.html) by obtaining the Installation.zip file.

After extraction, navigate to the *Installation_online* folder. Depending on whether your computer uses Excel or WPS, right-click the corresponding .bat file and run it as an administrator to complete the installation.



QPRLib Access

Choice Terminal



QPRLib Products

Euro/Amer Option

Payoff: $V_T(S_T, T) = \begin{cases} (S_T - K)^+, CALL \\ (K - S_T)^+, PUT \end{cases}$

Euro: Exercise at T;
Ame: Exercise before or at T;

Digital Option

Payoff: $V_T(S_T, T) = \begin{cases} I_{\{S_T \geq K\}}, CALL \\ I_{\{S_T \leq K\}}, PUT \end{cases}$

QPRLib Products

Barrier Options — Single Barrier

Barrier Type	Product Description	Relation between Barrier B and Initial Price S_0
UpIn	If any observation date $S_t \geq B$, product behaves as the corresponding European option; if all $S_t < B$, payoff = 0 at maturity.	$B > S_0$
UpOut	If any observation date $S_t \geq B$, product expires; if all $S_t < B$, product behaves as the corresponding European option.	
DnIn	If any observation date $S_t \leq B$, product behaves as the corresponding European option; if all $S_t > B$, payoff = 0 at maturity.	$B < S_0$
DnOut	If any observation date $S_t \leq B$, product expires; if all $S_t > B$, product behaves as the corresponding European option.	

Barrier Options — Double Barrier

Barrie r Type	Product Description	Relation between Barriers B_1, B_2 and Initial Price S_0
KnockI n	If any observation date $S_t \geq B_2$ or $S_t \leq B_1$, product behaves as the corresponding European option; if all $B_1 < S_t < B_2$, payoff = 0 at maturity.	$B_1 < S_0 < B_2$
Knock Out	If any observation date $S_t \geq B_2$ or $S_t \leq B_1$, product expires; if all $B_1 < S_t < B_2$, product behaves as the corresponding European option.	

QPRLib Products

Asian Option

Strike Type	Average Type	Payoff
Float Strike	Arithmetic	$V_T(S_T, T) = (S_T - S_A)^+$, CALL; $(S_A - S_T)^+$, PUT
Float Strike	Geometric	$V_T(S_T, T) = (S_T - S_G)^+$, CALL; $(S_G - S_T)^+$, PUT
Fixed Strike	Arithmetic	$V_T(S_T, T) = (S_A - K)^+$, CALL; $(K - S_A)^+$, PUT
Fixed Strike	Geometric	$V_T(S_T, T) = (S_G - K)^+$, CALL; $(K - S_G)^+$, PUT

- Arithmetic average: $S_A = \frac{1}{N}\sum S_i$
- Geometric average: $S_G = \left(\prod S_i\right)^{1/N}$

QPRLib Products

Lookback Option

Strike Type	Call/Put	Payoff
Float Strike	Call	$V_T(S_T, T) = S_T - S_{\min}$
Float Strike	Put	$V_T(S_T, T) = S_{\max} - S_T$
Fixed Strike	Call	$V_T(S_T, T) = (S_{\max} - K)^+$
Fixed Strike	Put	$V_T(S_T, T) = (K - S_{\min})^+$

QPRLib Products

Shark Fin——Single

UpOut Single Shark Fin (requires strike price < knock-out price):

- **Case A**: If an upward knock-out occurs during the period, the payoff = Principal + annualized upward knock-out return.
- **Case B**: If no knock-out occurs and the maturity payoff is below the strike price, the payoff = Principal + annualized guaranteed return.
- **Case C**: If no knock-out occurs and the maturity payoff lies between the strike price and the knock-out price, the payoff = Principal + annualized [guaranteed return + (underlying closing price – strike price) × participation rate].

DnOut Single Shark Fin (requires knock-out price < strike price):

- **Case A**: If a downward knock-out occurs during the period, the payoff = Principal + annualized downward knock-out return.
- **Case B**: If no knock-out occurs and the maturity payoff is above the strike price, the payoff = Principal + annualized guaranteed return.
- **Case C**: If no knock-out occurs and the maturity payoff lies between the knock-out price and the strike price, the payoff = Principal + annualized [guaranteed return + (strike price – underlying closing price) × participation rate].

Shark Fin——Double

- **Case A:** If an upward knock-out occurs during the period, the payoff = Principal + annualized upward knock-out return.
- **Case B:** If a downward knock-out occurs during the period, the payoff = Principal + annualized downward knock-out return.
- **Case C:** If no knock-out occurs and the maturity price lies between the upper and lower strike prices, the payoff = Principal + annualized guaranteed return.
- **Case D:** If no knock-out occurs and the maturity price lies between the upper strike price and the upper knock-out price, the payoff = Principal + annualized [guaranteed return + (underlying closing price – upper strike price) × upper participation rate].
- **Case E:** If no knock-out occurs and the maturity price lies between the lower knock-out price and the lower strike price, the payoff = Principal + annualized [guaranteed return + (lower strike price – underlying closing price) × lower participation rate].

QPRLib Products

Touch——Single

- **Case A: Up-Touch Type**： If the stock price touches the upper barrier during the contract period, a fixed payoff is received; otherwise, the contract expires worthless at maturity.
- **Case B: Up-No-Touch Type**： If the stock price does not touch the upper barrier at any time during the contract period, a fixed payoff is received at maturity; otherwise, the contract terminates upon touching.
- **Case C: Down-Touch Type**： If the stock price touches the lower barrier during the contract period, a fixed payoff is received; otherwise, the contract expires worthless at maturity.
- **Case D: Down-No-Touch Type**： If the stock price does not touch the lower barrier at any time during the contract period, a fixed payoff is received at maturity; otherwise, the contract terminates upon touching.

Touch——Double

- **Case A: Up-Touch Type**： If the underlying price touches either the upper barrier or the lower barrier during the contract period, a fixed payoff is received; otherwise, the contract expires worthless at maturity.
- **Case B: Up-No-Touch Type**： - If the underlying price does not touch either the upper barrier or the lower barrier during the contract period, a fixed payoff is received at maturity; otherwise, the contract terminates upon touching.

QPRLib Products

Three-Layer

Three-Layer Structure (Bullish: returns increase across intervals 1, 2, 3; Bearish: returns decrease across intervals 1, 2, 3):

- **Case A:** If the condition is met: Final underlying price / Initial price < Strike price K1, then the payoff = Annualized return of Interval 1.
- **Case B:** If the condition is met: K1 <= Final underlying price / Initial price <= K2, then the payoff = Annualized return of Interval 2.
- **Case C**: If the condition is met: Final underlying price / Initial price > Strike price K_2, then the payoff = Annualized return of Interval 3

Accumulator

- **Case A:** On the observation date, if the underlying price lies between the upper and lower bounds, the option buyer purchases a specified quantity of the underlying asset at the lower bound price.
- **Case B:** On the observation date, if the underlying price is below the lower bound, the option buyer purchases leverage × specified quantity of the underlying asset at the lower bound price.
- **Case C:** On the observation date, if the underlying price is above the upper bound, no settlement is made for that period.

QPRLib Products

Warrant

Same as European/American options.
Note 1: Exercise ratio defines how many units of underlying asset each warrant converts into.
Note 2: CBBC (Callable Bull/Bear Contracts) resale price refers to the residual value or settlement price upon mandatory redemption or maturity.

QPRLib Products

Autocall——SB

- **Case A**: On a knock-out date, if a knock-out occurs and a prior knock-in has occurred, the knock-out payoff = R (annualized).
- **Case B:** On a knock-out date, if a knock-out occurs and no prior knock-in has occurred, the knock-out payoff = R (annualized).
- **Case C**: If no knock-out occurs on any knock-out date, and a knock-in occurs before maturity, the maturity payoff = min (underlying/return,0).
- **Case D:** If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, the maturity payoff = R (annualized).

Autocall——TX

- **Case A**: If no knock-in occurs before maturity, the maturity payoff = (annualized).
- **Case B**: If a knock-in occurs before maturity and the final underlying return > , the maturity payoff = (annualized).
- **Case C**: If a knock-in occurs before maturity and the final underlying return < , the maturity payoff = underlying return (or underlying decline)

QPRLib Products

Autocall——PH

- **Case A**: On a knock-out date, if a knock-out occurs and a prior knock-in has occurred, the knock-out payoff =Jt*R .
- **Case B**: On a knock-out date, if a knock-out occurs and no prior knock-in has occurred, the knock-out payoff = Jt*R .
- **Case C**: If no knock-out occurs on any knock-out date, and a knock-in occurs before maturity, and ST>=S0, then the maturity payoff = Jt*R.
- **Case D**: If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, and ST<S0, then the maturity payoff = Jt*R + min(underlying return, 0).
- **Case E**: If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, then the maturity payoff = Jt*R .

Note: Jt represents the number of effective observation months at valuation, i.e., the number of knock-out observation dates on which the underlying closing price exceeded the knock-in price.

Autocall——TO

- **Case A**: On a knock-out date, if a knock-out occurs and a prior knock-in has occurred, the knock-out payoff = R(non-annualized).
- **Case B**: On a knock-out date, if a knock-out occurs and no prior knock-in has occurred, the knock-out payoff = R(non-annualized).
- **Case C**: If no knock-out occurs on any knock-out date, and ST>=S0, a knock-in occurs before maturity, and , then the maturity payoff = 0%.
- **Case D**: If no knock-out occurs on any knock-out date, and ST<S0, no knock-in occurs before maturity, and , then the maturity payoff = underlying decline.
- **Case E:** If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, then the maturity payoff = R(non-annualized).

QPRLib Products

Autocall——ZL

- **Case A**: On a knock-out date, if a knock-out occurs and a prior knock-in has occurred, the knock-out payoff = R1 before T; knock-out payoff = R2 after T.
- **Case B:** On a knock-out date, if a knock-out occurs and no prior knock-in has occurred, the knock-out payoff = R1 before T; knock-out payoff = R2 after T.
- **Case C:** If no knock-out occurs on any knock-out date, and a knock-in occurs before maturity, the maturity payoff = min(underlying return,0).
- **Case D**: If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, the maturity payoff = R2.

Autocall——JQ

- **Case A**: On a knock-out date, if a knock-out occurs and a prior knock-in has occurred, the knock-out payoff = R (annualized).
- **Case B**: On a knock-out date, if a knock-out occurs and no prior knock-in has occurred, the knock-out payoff = R (annualized).
- **Case C:** If no knock-out occurs on any knock-out date, and a knock-in occurs before maturity, the maturity payoff = min (underlying return ,0).
- **Case D**: If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, the maturity payoff = R (annualized).

Note: The knock-out boundary decreases proportionally with each knock-out date.

QPRLib Products

Autocall——HL

- **Case A**: On a knock-out date, if a knock-out occurs and a prior knock-in has occurred, the knock-out payoff = R (annualized).
- **Case B**: On a knock-out date, if a knock-out occurs and no prior knock-in has occurred, the knock-out payoff = R (annualized).
- **Case C**: If no knock-out occurs on any knock-out date, and a knock-in occurs before maturity, the maturity payoff = min (underlying return,0).
- **Case D**: If no knock-out occurs on any knock-out date, and no knock-in occurs before maturity, the maturity payoff = Dividend coupon (< R).

Autocall——AB

- **Case A**: If a knock-in occurs during the period, the maturity yield = underlying return – 1.
- **Case B**: If no knock-in occurs during the period, the maturity yield = max (0,underlying return - strike price percentage).

Note: If a cap clause exists, the yield is limited to the capped yield.

Autocall——AO

- **Case A**: On a knock-out date, if a knock-out occurs, the knock-out payoff = R_1 (annualized).
- **Case B**: If no knock-out occurs on any knock-out date, the maturity payoff = R_2 (annualized), where typically $R_2<R_1$.

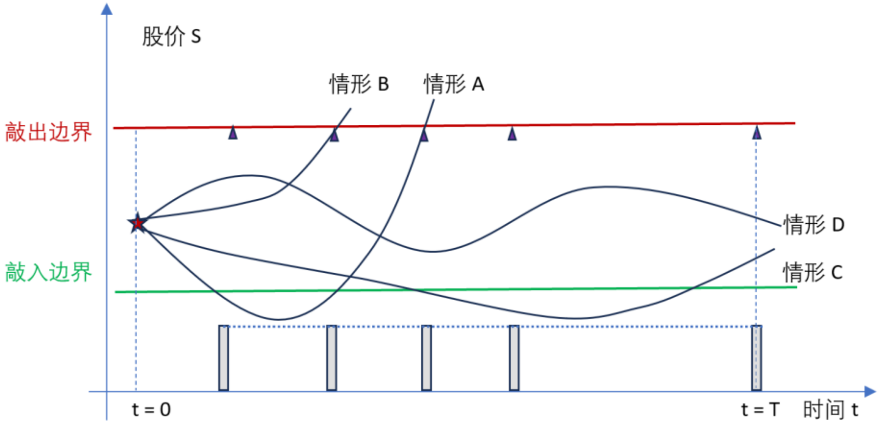
Note: Unlike the classical snowball structure, this product does not contain a knock-in clause and is capital-protected.

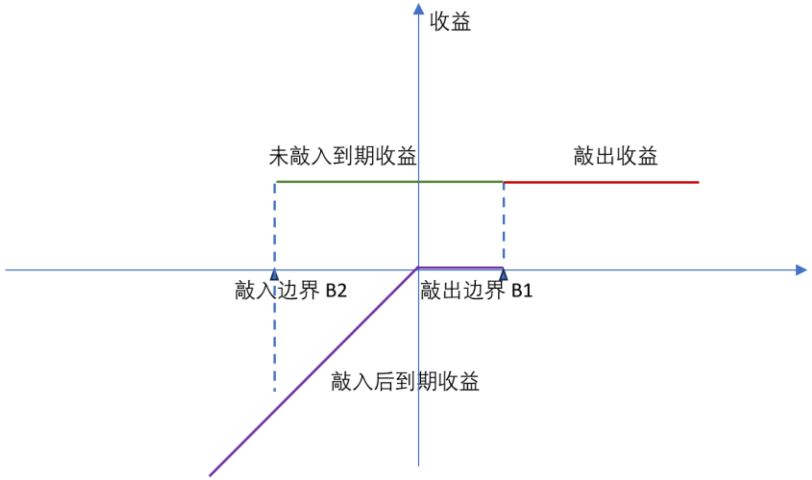
QPRLib Pricing——Autocall(SB)

Essence of the Classic Snowball:
The classic snowball essentially represents a double-barrier put option purchased by the issuing broker from investors, with **Down-and-In (DnIn)** and Up-and-Out (UpOut) features. This product combines a safety cushion, high probability of success, and relatively stable medium-to-high returns, while carrying tail risk as a nonlinear investment instrument.

Market Expectation of Purchasing a Classic Snowball:
The market outlook for buying a classic snowball is similar to that of selling a put option. The expectation is for a moderate upward trend or sideways consolidation in the market, without significant upward or downward movements

QPRLib ——Autocall(SB)



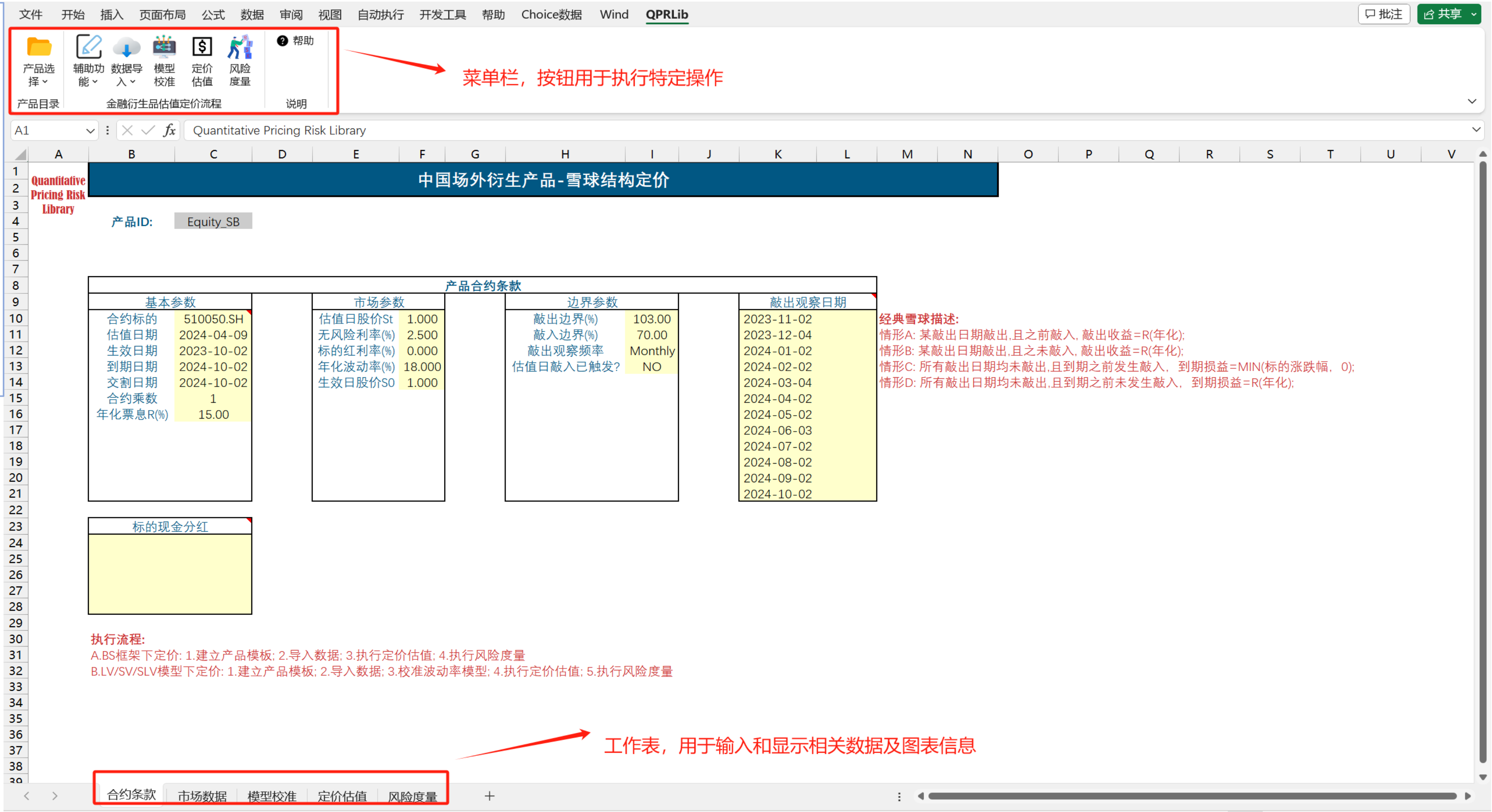


QPRLib ——Autocall(SB)

Light Yellow Background:
User-input or editable parameter area.

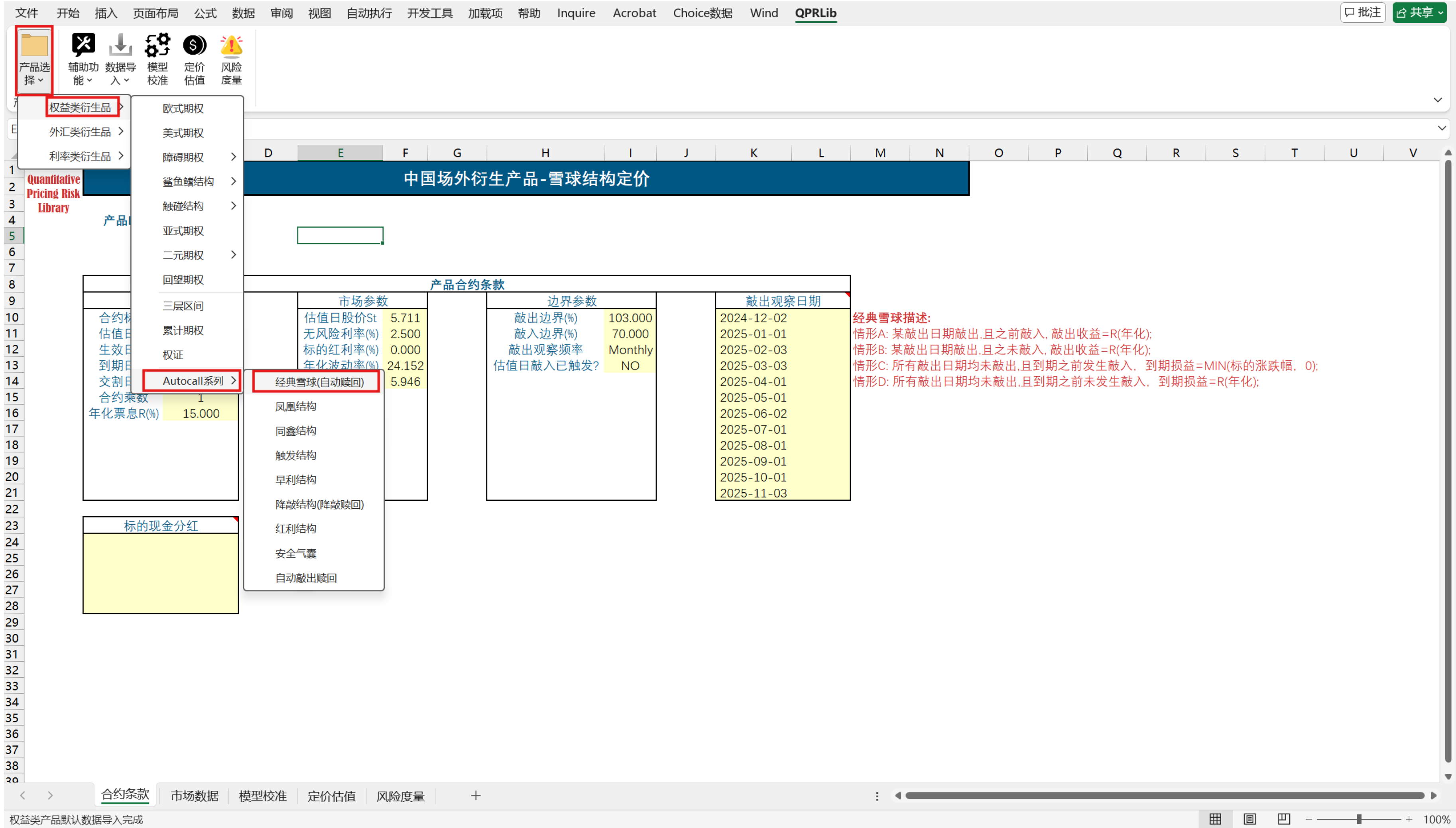
Light Gray Background:
ystem parameter area (non-editable by user).

Light Blue Background:
Result display area (non-editable by user).



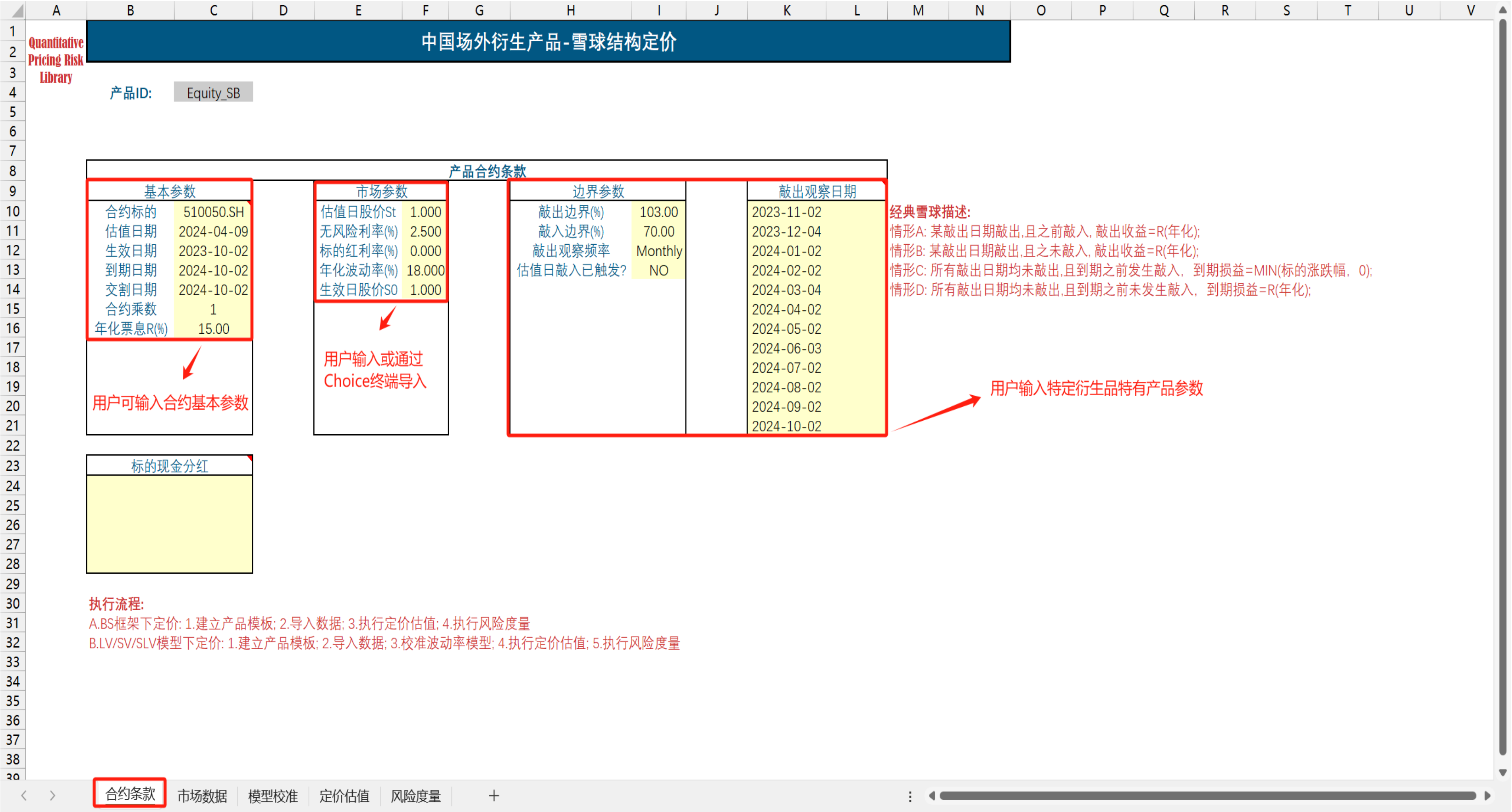
QPRLib ——Autocall(SB)

Step 1: In the menu bar, click “Product Selection” to create a product template.



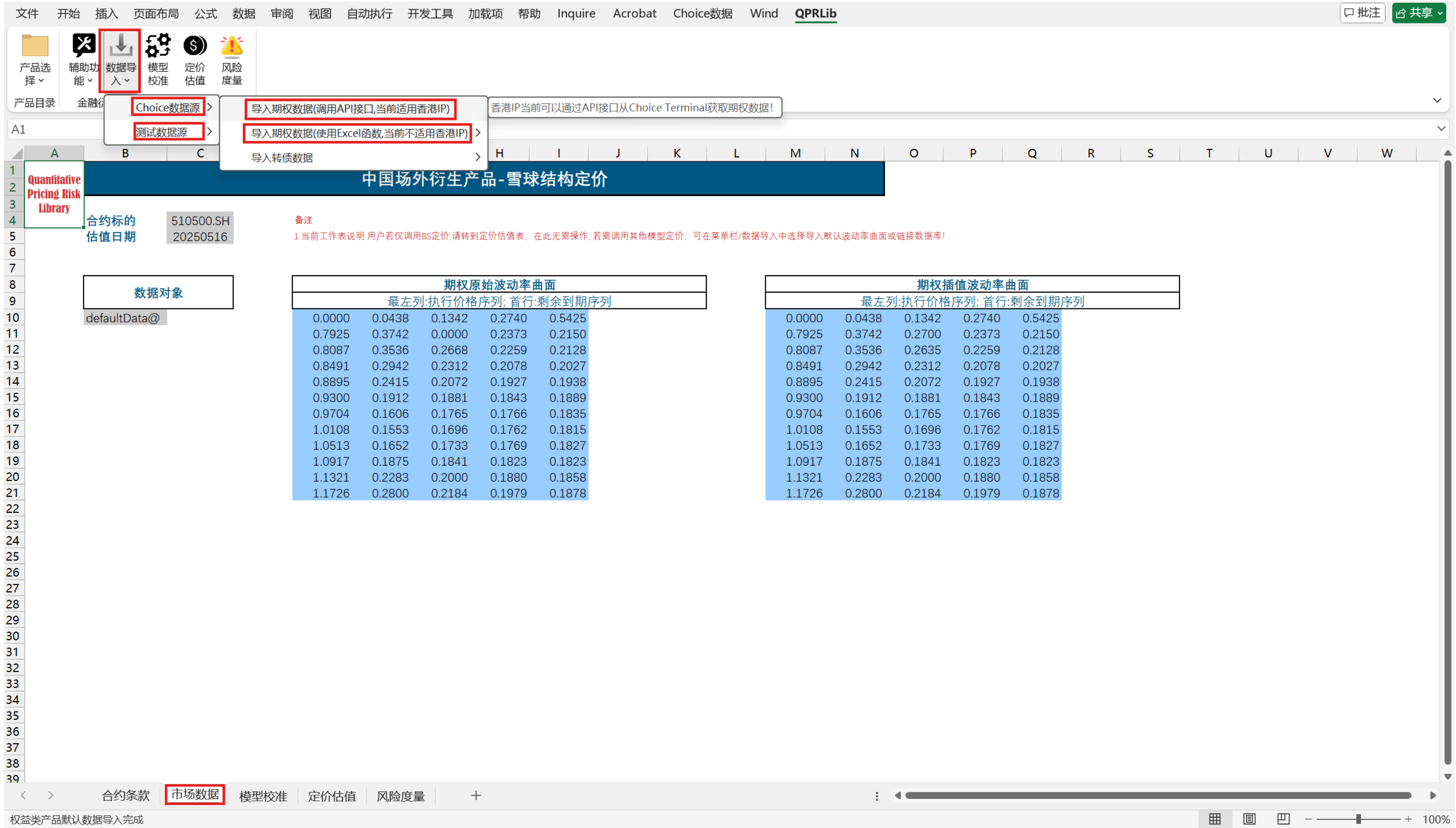
QPRLib ——Autocall(SB)

Step 2: In the “Contract Terms” worksheet, modify the corresponding product parameters.



QPRLib ——Autocall(SB)

Step 3: In the “Market Data” worksheet, execute the “Import Option Data” operation to import volatility data from Choice.



期权原始波动率曲面

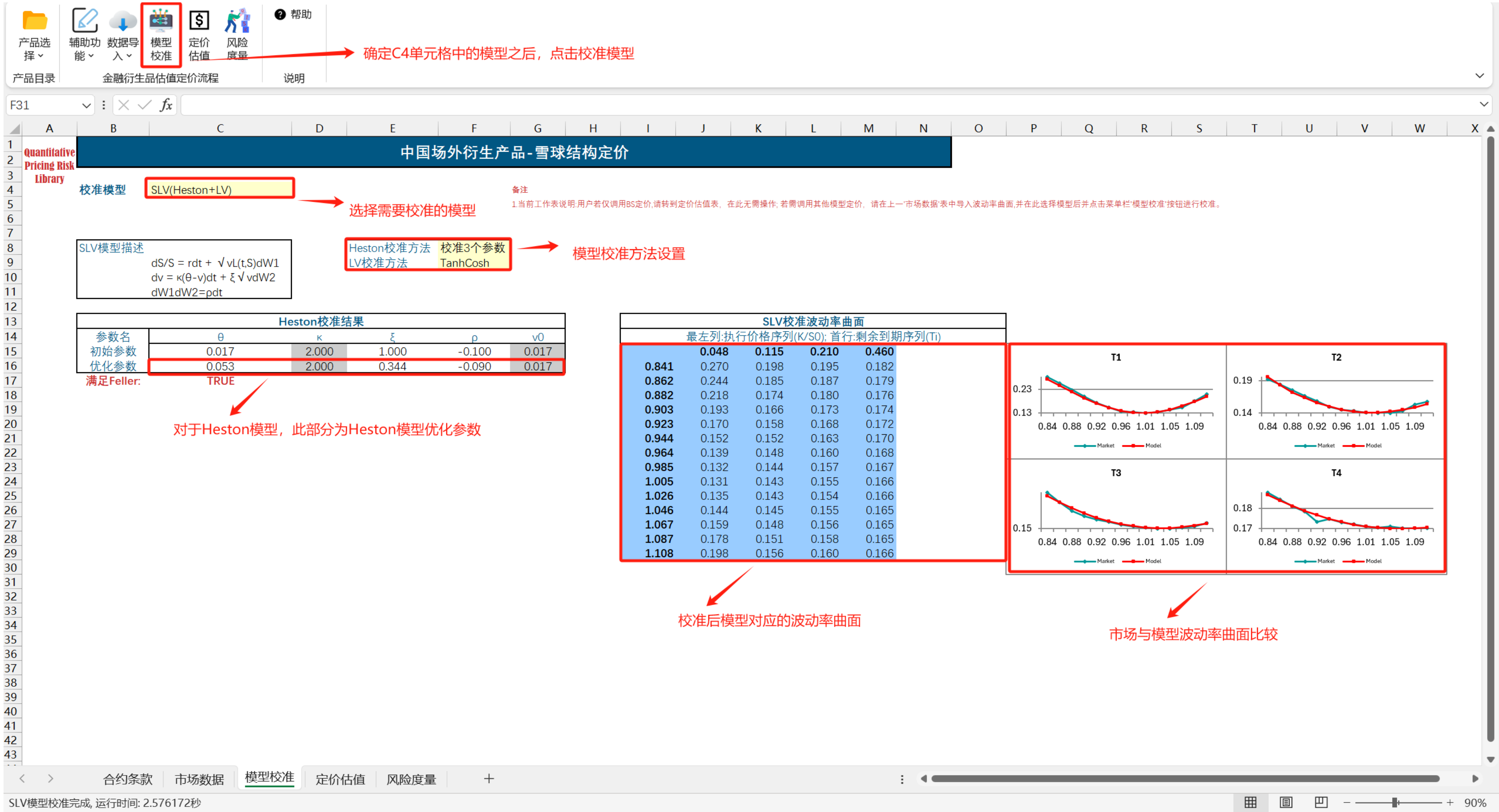
0.0000	0.0438	0.1342	0.2740	0.5425
0.7925	0.3742	0.0000	0.2373	0.2150
0.8087	0.3536	0.2668	0.2259	0.2128
0.8491	0.2942	0.2312	0.2078	0.2027
0.8895	0.2415	0.2072	0.1927	0.1938
0.9300	0.1912	0.1881	0.1843	0.1889
0.9704	0.1606	0.1765	0.1766	0.1835
1.0108	0.1553	0.1696	0.1762	0.1815
1.0513	0.1652	0.1733	0.1769	0.1827
1.0917	0.1875	0.1841	0.1823	0.1823
1.1321	0.2283	0.2000	0.1880	0.1858
1.1726	0.2800	0.2184	0.1979	0.1878

期权插值波动率曲面

0.0000	0.0438	0.1342	0.2740	0.5425
0.7925	0.3742	0.2700	0.2373	0.2150
0.8087	0.3536	0.2635	0.2259	0.2128
0.8491	0.2942	0.2312	0.2078	0.2027
0.8895	0.2415	0.2072	0.1927	0.1938
0.9300	0.1912	0.1881	0.1843	0.1889
0.9704	0.1606	0.1765	0.1766	0.1835
1.0108	0.1553	0.1696	0.1762	0.1815
1.0513	0.1652	0.1733	0.1769	0.1827
1.0917	0.1875	0.1841	0.1823	0.1823
1.1321	0.2283	0.2000	0.1880	0.1858
1.1726	0.2800	0.2184	0.1979	0.1878

QPRLib ——Autocall(SB)

Step 4: In the “Model Calibration” worksheet, select a specific pricing model, then perform the “Model Calibration” operation to obtain the calibrated model parameters and compare market results with the model outputs.



QPRLib ——Autocall(SB)

Step 5: In the “Pricing & Valuation” worksheet, set the model and calculation method, then click the “Pricing & Valuation” button to compute the results.

QPRLib ——Autocall(SB)

Step 6:
In the Risk Measurement worksheet, set the scenario analysis parameters and click the Risk Measurement button to perform the result calculation.

情景分析表格

	标的价格								
	0.600	0.700	0.800	0.900	1.000	1.100	1.200	1.300	1.400
Price(%)	-45.99%	-36.98%	-8.57%	0.38%	0.000	-2.08%	-2.25%	-2.25%	-2.25%
Delta(%)	-442.24%	-443.46%	-765.13%	-203.22%	0.000	-72.54%	-99.78%	-100.00%	-100.00%
Gamma(%)	-100.09%	-104.88%	864.28%	256.98%	0.000	-203.98%	-101.52%	-100.00%	-100.00%
Price	0.600	0.700	1.016	1.115	1.111	1.088	1.086	1.086	1.086
Delta	1.000	1.004	1.944	0.302	-0.292	-0.080	-0.001	0.000	0.000
Gamma	0.002	0.123	-24.326	-9.006	-2.523	2.623	0.038	0.000	0.000
Vega	0.000	0.000	-0.013	-0.006	-0.001	0.000	0.000	0.000	0.000
Theta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rho	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

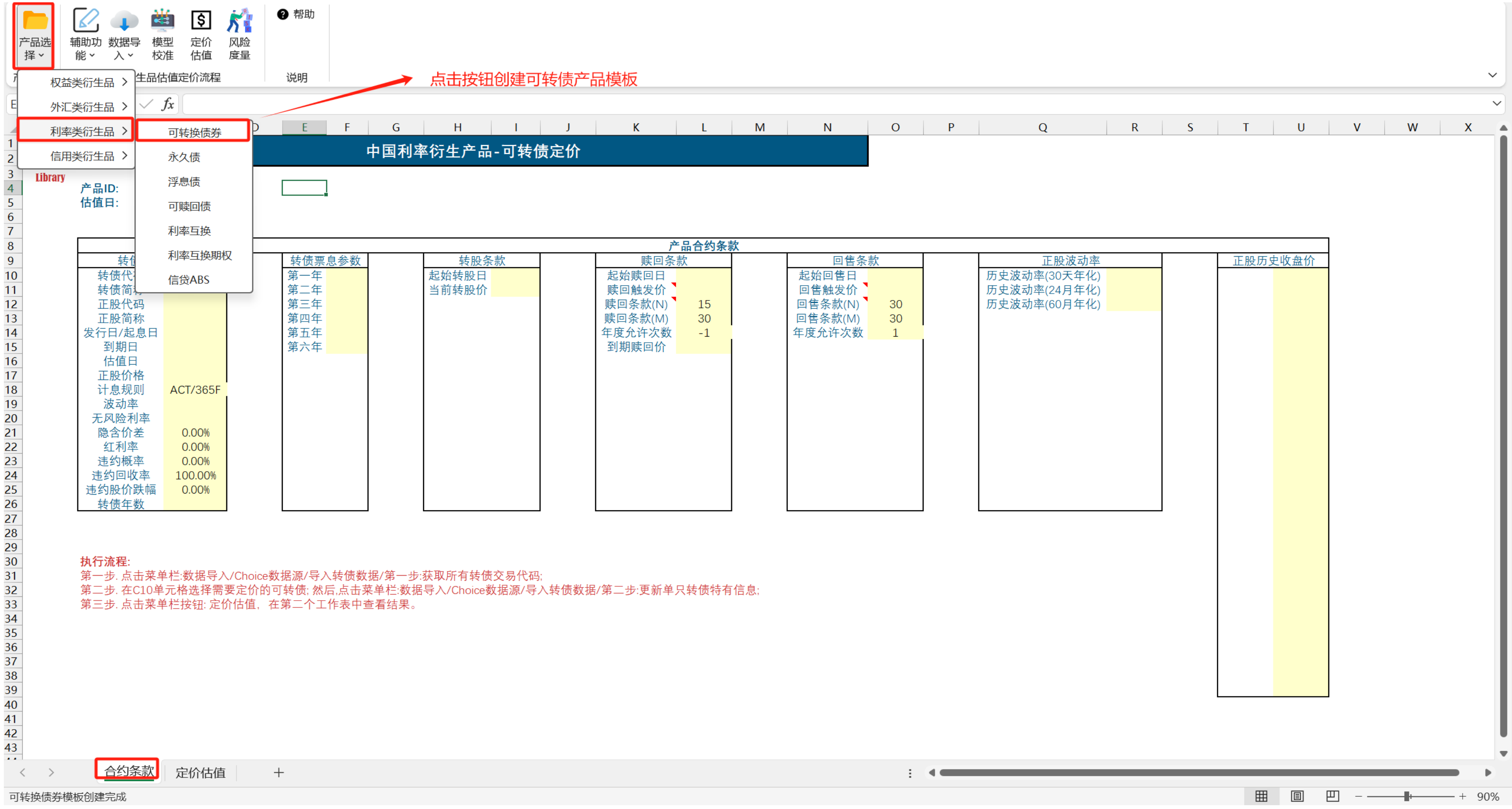
QPRLib——Convertible Bond

Convertible bonds (short for *convertible corporate bonds*, also known simply as *convertibles*) are a special type of corporate bond that can be converted into common stock at specific times and under certain conditions.

Convertible bonds combine the characteristics of both debt and equity. Like other bonds, they carry a fixed interest rate and maturity date, allowing investors to hold them until maturity and receive principal plus interest. At the same time, during the conversion period, holders have the right to convert the bonds into a certain number of shares at the predetermined conversion price.

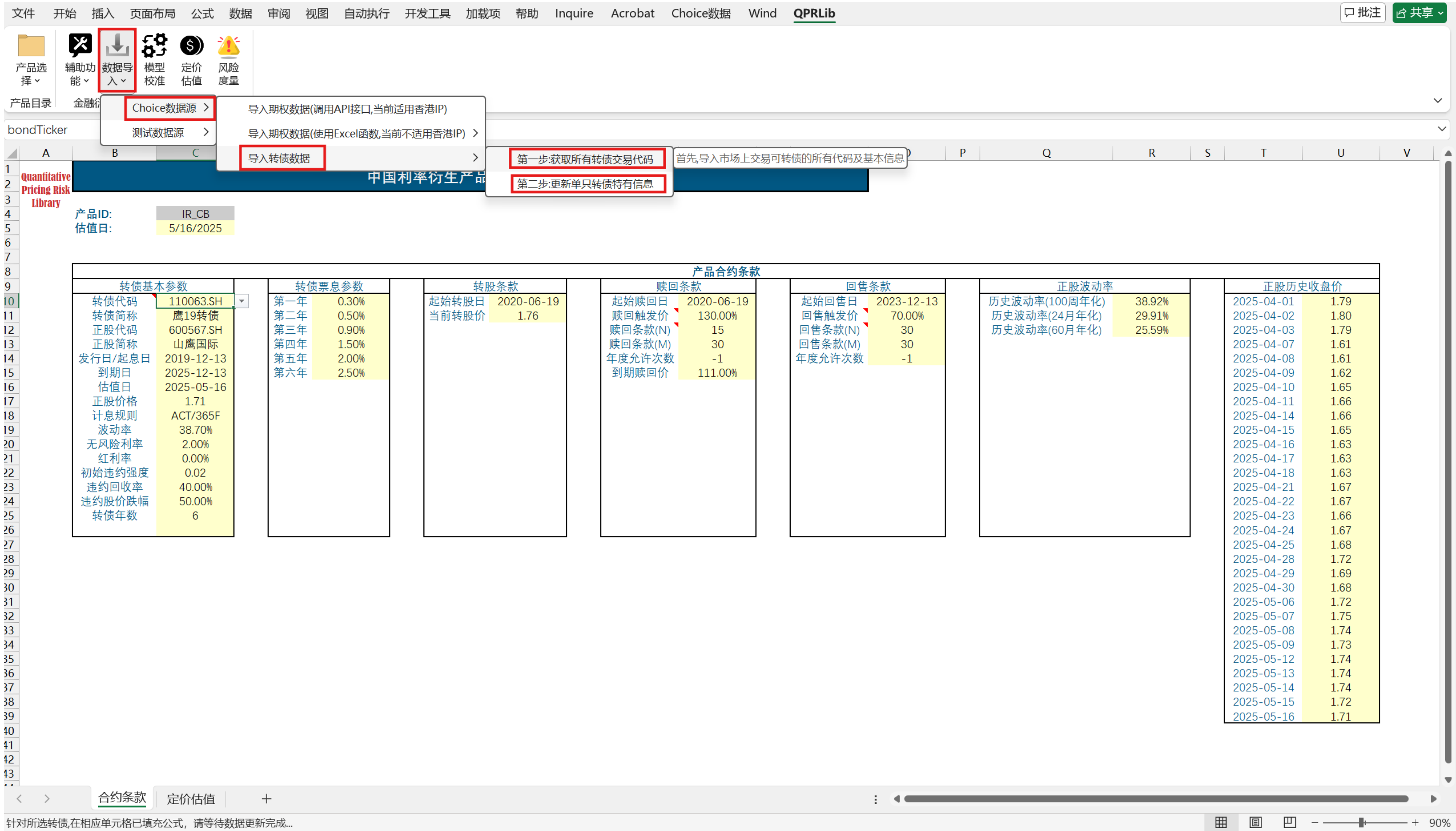
QPRLib——Convertible Bond

Step 1: Click "Product Selection / Interest Rate Derivatives / Convertible Bonds" to create a convertible bond product template.



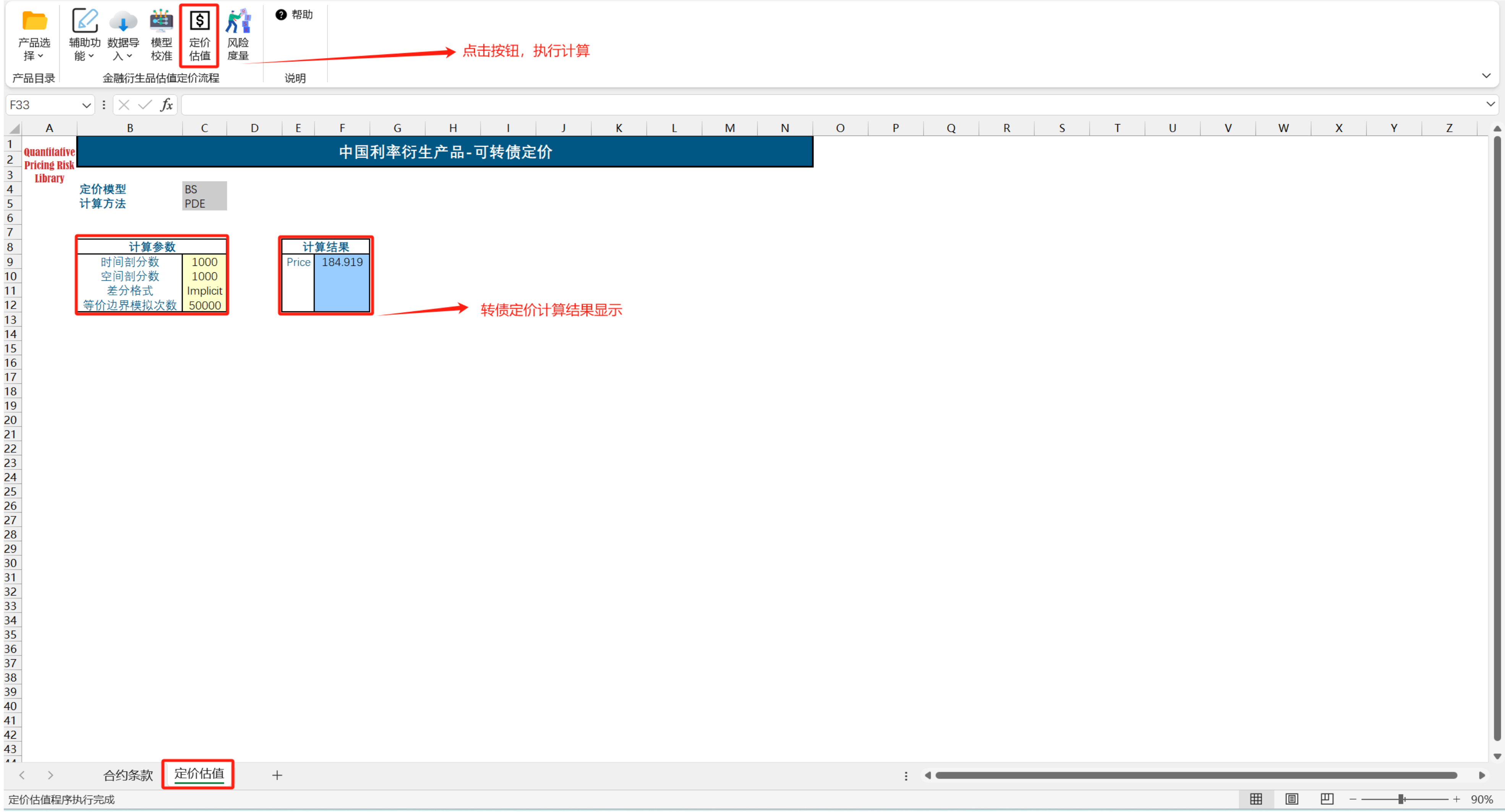
QPRLib——Convertible Bond

Step 2: After confirming the valuation date, click the “Step 1” button to import all convertible bond codes. Then, select the convertible bond code in cell C10, and click the “Step 2” button to import the specific information of the selected bond.



QPRLib——Convertible Bond

Step 3: In the “Pricing & Valuation” worksheet, click the “Pricing & Valuation” button to perform the result calculation.



Black-Scholes(BS)

◆ **Process**

Under the Black–Scholes (BS) model, the underlying asset price follows a lognormal distribution, with volatility assumed to be a constant σ :

$$\mathrm{d}S/S = (\mathrm{r} - q)dt + \sigma\, dW_{\mathrm{t}}$$

Since the option price is monotonically increasing with respect to σ , there exists a one-to-one relationship between volatility and option price. The volatility $\boldsymbol{\sigma_{imp}}$ inferred from market-traded option prices is called **implied volatility**. The collection of implied $\boldsymbol{\sigma_{imp}(K,T)}$ across different strike prices K and maturities T forms the **implied volatility surface**.

Local Volatility(LV)

- **Process**

Under the Local Volatility (LV) model, the underlying asset price follows a lognormal distribution, where volatility is a deterministic function of S and t, denoted as $\sigma_{LV}(S,t)$

$$\mathrm{d}S/S = (\mathrm{r} - q)dt + \sigma_{LV}(S,t)dW_{\mathrm{t}}$$

The LV model is generally divided into two categories:

1. Parametric models: If $\sigma_{LV}(S,t)$ is expressed in a parametric form, such as the CEV model,$(S,t) = \sigma_0 S^{\beta-1}$
2. Dupire model: Since $\sigma_{LV}(S,t)$ corresponds one-to-one with the implied volatility $\sigma_{imp}(K,T)$, the market implied volatility surface uniquely determines the local volatility function$\sigma(K,T)$;

In QPRLib, the implemented version is the Dupire model.

Local Volatility(LV)

◆ **Dupire Model**

The Dupire model is generally expressed in three forms, corresponding to derivatives with respect to: "Option Price", "Implied Volatility", "Total variance"

① $\mathbf{C}(\boldsymbol{K},\boldsymbol{T})$

$$\sigma_{LV}^2(K,T)=\frac{2(\frac{\partial C}{\partial T}+(r-q)K\frac{\partial C}{\partial K}+qC)}{K^2\frac{\partial^2 C}{\partial K^2}}$$

② $\boldsymbol{\sigma_{imp}}(\boldsymbol{K},\boldsymbol{T})$

$$\sigma_{LV}^2(K,T)=\frac{2(\frac{\partial\sigma_{\mathrm{imp}}}{\partial T}+\frac{\sigma_{\mathrm{imp}}}{2\,T}+(r-q)K\frac{\partial\sigma_{\mathrm{imp}}}{\partial K})}{K^2\left[\frac{\partial^2\sigma_{\mathrm{imp}}}{\partial K^2}-d\sqrt{T}\left(\frac{\partial\sigma_{\mathrm{imp}}}{\partial K}\right)^2+\frac{1}{\sigma_{\mathrm{imp}}}\left(\frac{1}{K\sqrt{T}}+d\frac{\partial\sigma_{\mathrm{imp}}}{\partial K}\right)^2\right]},\ d=\frac{\ln\left(\frac{S_0}{K}\right)+\left(r-q+\frac{1}{2}\sigma_{\mathrm{imp}}{}^2\right)T}{\sigma_{\mathrm{imp}}\sqrt{T}}$$

③ $\omega(\boldsymbol{K},\boldsymbol{T})=\boldsymbol{\sigma_{imp}^2 T}$ **【QPRLib Default】**

$$\sigma_{LV}^2(K,T)=\frac{\frac{\partial\omega}{\partial T}}{1-\frac{y}{\omega}\frac{\partial\omega}{\partial y}+\frac{1}{4}\left(-\frac{1}{4}-\frac{1}{\omega}+\frac{y^2}{\omega^2}\right)\left(\frac{\partial\omega}{\partial y}\right)^2+\frac{1}{2}\frac{\partial^2\omega}{\partial y^2}},\ y=\ln\left(\frac{\mathrm{K}}{F_T}\right),\ F_T=S_0e^{\int_0^T dt\,\mu(t)}$$

Local Volatility(LV)

◆ **Implied Volatility Surface Interpolation**

The Dupire model requires solving first- and second-order derivatives with respect to variables. However, market-traded points are often discrete, so the market surface must be interpolated to obtain a complete implied volatility surface. **QPRLib implements three interpolation methods for this purpose.**

① TanhCosh： $\sigma(X;\{\sigma_{ATM},A,B,C,D\}) = \sigma_{ATM} + Atanh[B(X-1)] + C\left(1-\frac{1}{\cosh[D(X-1)]}\right), X = K/S_0$

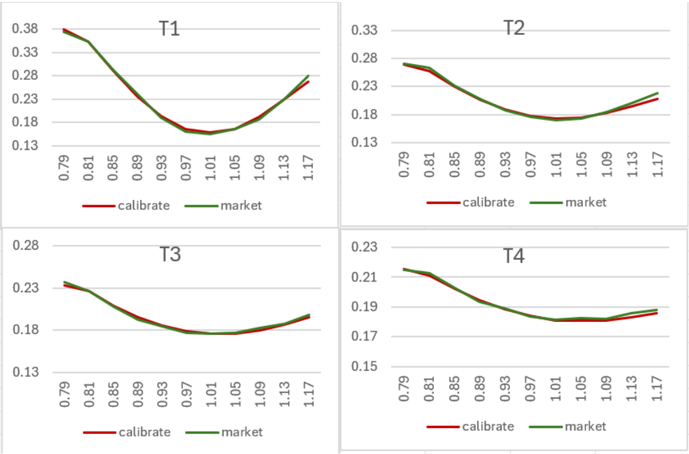
② SVI： $\omega(k;\{a,b,\rho,m,\sigma\}) = a + b\left\{\rho(k-m)+\sqrt{(k-m)^2+\sigma^2}\right\}, k = \ln\left(\frac{K}{S_0}\right), \omega$为total variance

③ CubicSpline： N/A

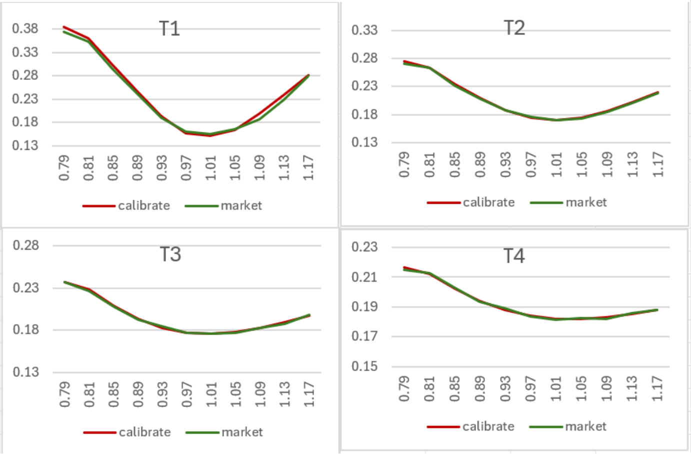
For the above implied volatility models, the **Levenberg–Marquardt** method is typically used to optimize the fixed constant parameters for each term. Once the full implied volatility surface is obtained, derivatives are taken from the complete surface to derive the corresponding local volatility.

Local Volatility(LV)

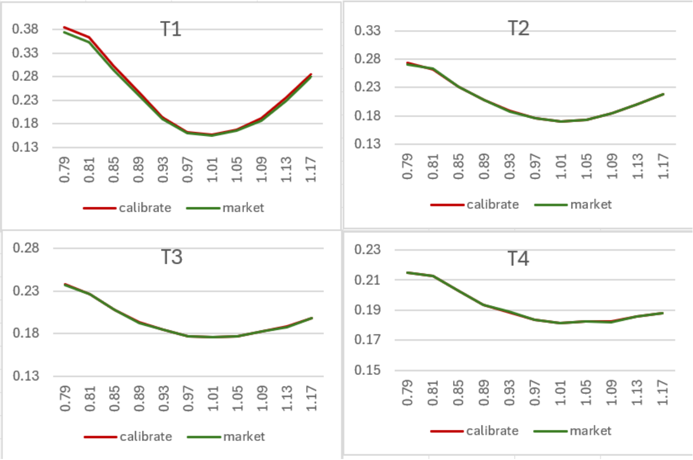
- **Based on QPRLib/Excel test data**



TanhCosh



SVI



CubicSpline

Model	L1 Error	L2 Error
TanhCosh	0.10934	0.00056
SVI	0.10142	0.00063
CubicSpline	0.07122	0.00048

Stochastic volatility(SV)

◆ **Process**

Under the Stochastic Volatility (SV) model, the volatility of the underlying asset is represented as a stochastic process (such as a square-root process). Common SV models include:

Heston

$$
\begin{aligned}
dS/S &= (\mathrm{r} - q)dt + \sqrt{y}dW_t^1 \\
dy &= \kappa(\theta - y)dt + \xi\sqrt{y}dW_t^2
\end{aligned}
$$

SABR

$$
\begin{aligned}
dF &= \sigma F^{\beta} dW_t^1 \\
d\sigma &= \gamma\sigma dW_t^2
\end{aligned}
$$

In **QPRLib**, the implemented model is the **Heston model**.

The calibration problem of the Heston model is described as follows: using the market implied volatility surface to calibrate the constant parameters $\{\kappa, \theta, \xi, \rho, y_0\}$。

Stochastic volatility(SV)

◆ **Heston calibration**

In the Heston model, there are five parameters to be determined. According to their influence on different aspects of the volatility curve (for a given maturity Ti), they can be grouped into three categories:

level: θ, y_0 --> The larger these values, the higher the overall level of volatility. They can be set as the square of the ATM volatility.

slope: ρ --> - Ranging from -1 to 1, the volatility curve shifts from downward-sloping to upward-sloping.

curvature:κ, ξ -->κ increases from 0, the overall curvature of the volatility curve flattens, and the level decreases. ξ increases from 0, the curvature of the volatility curve becomes more pronounced (the smile becomes more evident), and the level decreases.

In **QPRLib**, the parameters are optimized using the **Levenberg–Marquardt** method. The implementation provides two approaches:

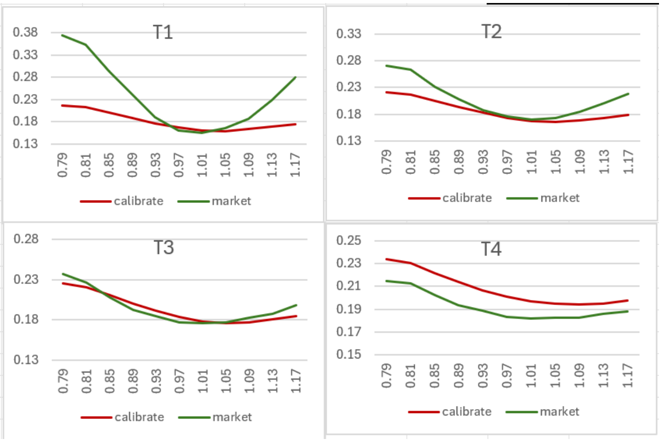
5-parameter optimization: Optimize all five parameters $\{\kappa, \theta, \xi, \rho, y_0\}$.

3-parameter optimization: Fix $\{\kappa, y_0\}$, and optimize $\{\theta, \xi, \rho\}$.

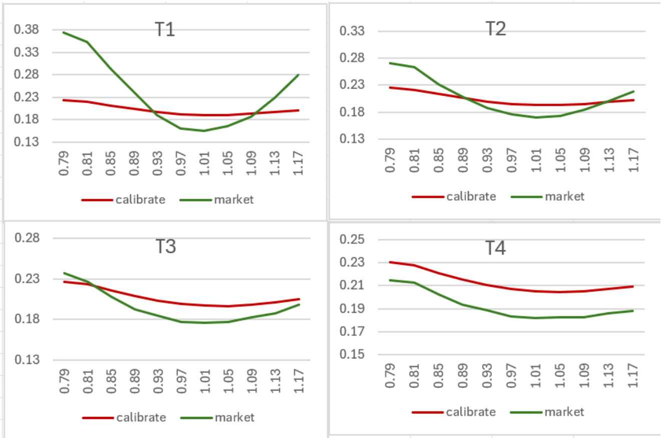
Stochastic volatility(SV)

- **Based on QPRLib/Excel test data**

Optimize 3 parameters



Optimize 5 parameters



Heston Calibration	L1 Error	L2 Error
Optimize 3 Parameters	1.1383	0.08280
Optimize 5 Parameters	1.2065	0.07196

Stochastic local volatility(SLV)

- **Process**

  Under the Stochastic Local Volatility (SLV) model, the volatility of the underlying asset is expressed as the product of a local volatility function (leverage function) and a stochastic process (such as a square-root process). In QPRLib, the SLV model is specifically defined as follows:

$$dS/S = (\mathrm{r} - q)dt + L(S,t)\sqrt{y}dW_t^1$$
$$dy = \kappa(\theta - y)dt + \xi\sqrt{y}dW_t^2$$
$$dW_t^1 dW_t^2 = \rho dt$$

  The calibration problem of the SLV model is described as follows: **using the market implied volatility surface to calibrate the constant parameters $\{\kappa, \theta, \xi, \rho, y_0\}$ as well as the function $L(S,t)$。**

Stochastic local volatility(SLV)

◆ **The Option price $V(s, y, t)$ with SLV model satisfies the PDE**

$$\partial_t V = \mathcal{L}V, S > 0, y > 0, t > 0$$
$$V_0 = (S - K)^+$$

其中, $\mathcal{L}V = \frac{1}{2}(SL)^2 y \partial_{SS} V + (r - q) S \partial_S V + \frac{1}{2}\xi^2 y \partial_{yy} V + \kappa(\theta - y)\partial_y V + \rho SL\xi y\, \partial_{Sy} V - rV$

Stochastic local volatility(SLV)

◆ **The density function P(s, y, t) of (S_t, y_t), satisfies the Fokker-Planc function**

$$\partial_T P = \mathcal{L}^* P, S > 0, y > 0, T > 0$$
$$P_0 = \delta(S_0 - S)\delta(y_0 - y)$$
$$\mathcal{L}^* P = \partial_{SS}\left(\frac{1}{2}(SL)^2 yV\right) - \partial_S\big((r-q)SV\big) + \partial_{yy}\left(\frac{1}{2}\xi^2 yV\right) - \partial_y(\kappa(\theta - y)V) + \partial_{Sy}(\rho SL\xi yV) - rV$$

To simplify the representation of subsequent numerical algorithms, we define the following operator: $\mathcal{L}^* \triangleq F_0 + F_1 + F_2$, where F_0, F_1, F_2 denote the operators for the **cross term**, the **S-direction**, and the **y-direction**, respectively.
Note that the term -rV is evenly distributed between F_1, F_2.

Stochastic local volatility(SLV)

◆ **$\mathbf{L}(\boldsymbol{S},\boldsymbol{t})$ satisfies the formula**

$$L^2(S,t) = \frac{\sigma_{LV}^2(S,t)}{E[y_t|S_t = S]} \qquad \cdots\cdots\cdots (1)$$

$$= \frac{\sigma_{LV}^2(S,t)}{\frac{\int_0^\infty y \cdot P(S,y,t)dy}{\int_0^\infty P(S,y,t)dy}} \qquad \cdots\cdots\cdots (2)$$

Here, , $\sigma_{LV}(S,t)$ denotes the **local volatility**.
The **MC (Monte Carlo) algorithm** performs model calibration by solving the expectation based on Equation (1), While the **PDE (Partial Differential Equation) algorithm** performs model calibration by solving the probability density function based on Equation (2).

Stochastic local volatility(SLV)

◆ **Based on QPRLib/Excel test data——volatility surface：**

relative strikes = {0.7925,0.8087,0.8491,0.8895,0.93,0.9704,1.0108,1.0513,1.0917,1.1321,1.1726}

maturities ={0.0438, 0.1342, 0.274, 0.5425}

surface={

{0.3742, 0.2700, 0.2373, 0.215},

{0.3536, 0.2635, 0.2259, 0.2128},

{0.2942, 0.2312, 0.2078, 0.2027},

{0.2415, 0.2072, 0.1927, 0.1938},

{0.1912, 0.1881, 0.1843, 0.1889},

{0.1606, 0.1765, 0.1766, 0.1835},

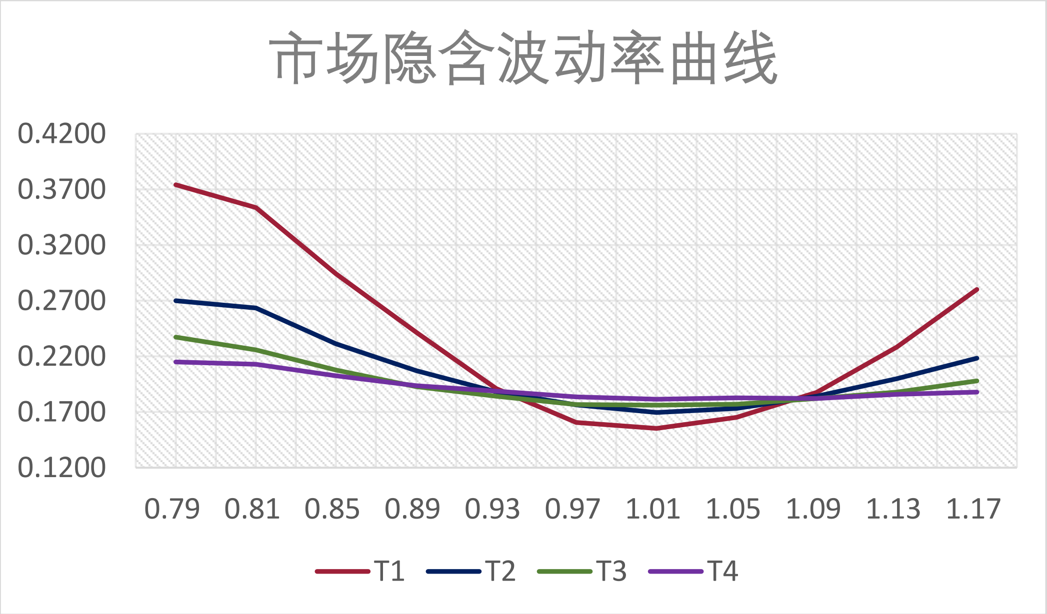
{0.1553, 0.1696, 0.1762, 0.1815},

{0.1652, 0.1733, 0.1769, 0.1827},

{0.1875, 0.1841, 0.1823, 0.1823},

{0.2283, 0.2000, 0.1880, 0.1858},

{0.2800, 0.2184, 0.1979, 0.1878}}



Heston parameters={$\kappa = 0.811, \theta = 0.13, \xi = 0.451, \rho = -0.282, y_0 = 0.025$}

Stochastic local volatility(SLV)

◆ **Based on QPRLib/Excel test data——grid parameters：**

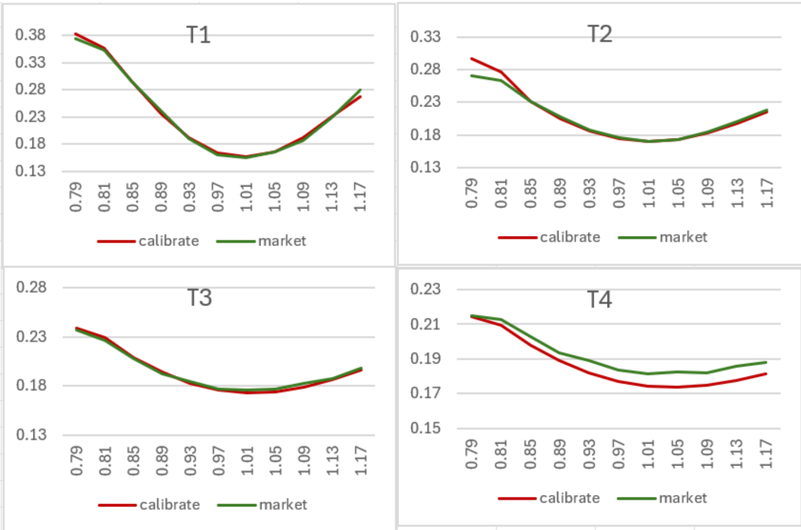
Direction	Min	Max	N	
T	0	0.5967	~45	N/A
K	0.4097	2.5129	~200	99.999942% Range (+/-5σ)
Y	0	1.0516	~300	99.999999% Range

Stochastic local volatility(SLV)

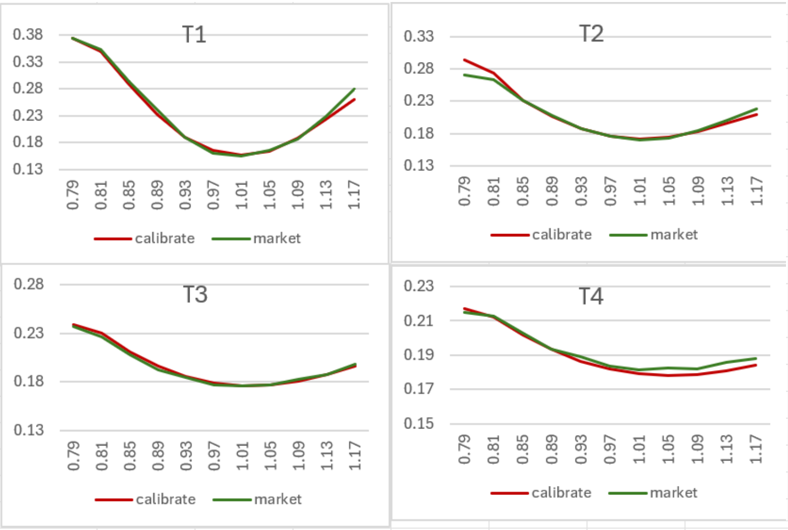
◆ **Based on QPRLib/Excel test data：**

Index	Library	Method	Grid	Time(senonds)【C++/release】	L1 Error	L2 Error	Each step
M1	QPRLib	PDE-Default	uniform	0.30913	0.18406	0.00174	2 Implicit + 1 Explicit
M2		PDE-MCS	Non-uniform in S	0.61412	0.15389	0.00149	4 Implicit + 3 Explicit
M3		*PDE-Iteration*	*Non-uniform in S*	*22.6866*	*0.33462*	*0.01154*	
M4		MC(2025-5-3 update)	Non-uniform in S	1.1296	0.18171	0.00234	2^{16} *path*
M5	QuantLib	PDE-MCS	Non-uniform in S and t	2.5648	0.48328	0.01043	4 Implicit + 3 Explicit
M6		MC	Non-uniform in S and t	1.5535	0.59856	0.01418	2^{16} *path*

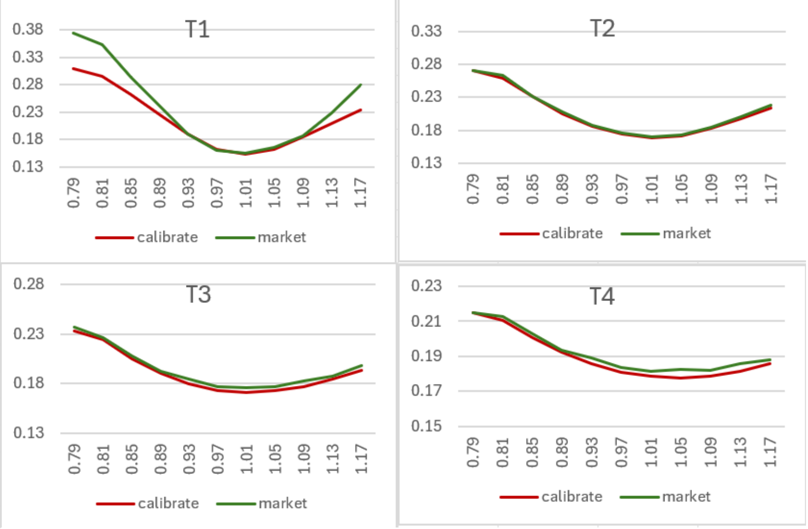
Stochastic local volatility(SLV)



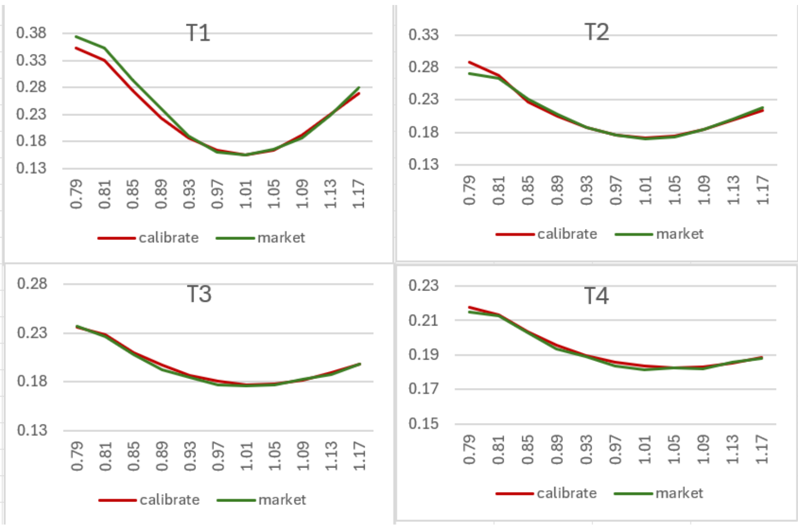
M1:QPRLib-PDE-Default



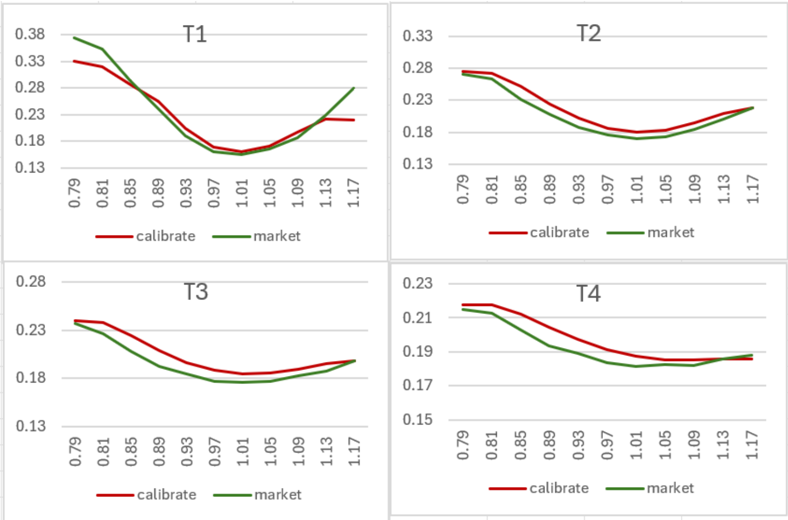
M2:QPRLib-PDE-MCS



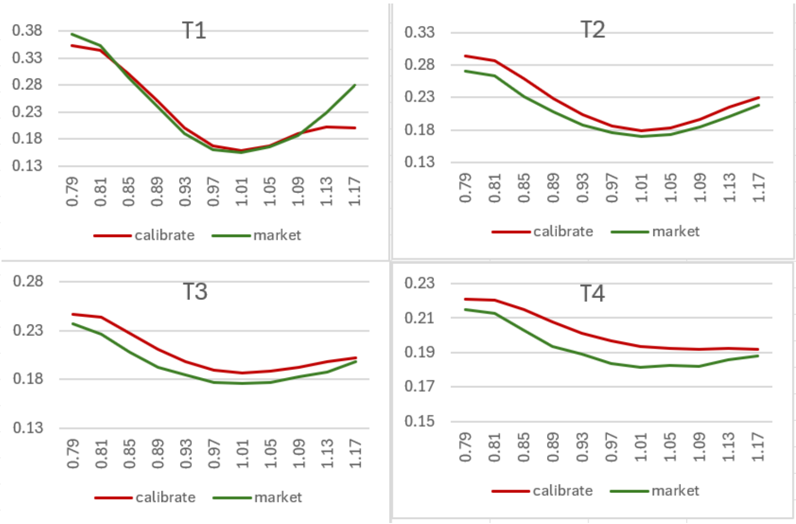
M3:QPRLib-PDE-Iteration



M4:QPRLib-MC



M5:QuantLib-PDE-MCS



M6:QuantLib-MC

Stochastic local volatility(SLV)

◆ **M1~M6 Algorithm：**

Time step	Calculate $L(S,\mathrm{t})$	PDE Method: Compute the density function at the next time step (denominator of Equation 2). MC Method: Compute the conditional expectation at the next time step (denominator of Equation 1).
$T_0 = 0$	$L(S,0) = \frac{\sigma_{LV}(S,0)}{\sqrt{y0}}$	Based on $L(S,T_0)$, Calculate $P(S,y,T_1)$ or $E\left[y_{T_1} \middle| S_{T_1} = S\right]$
$T_1 = dt$	Calculate $L(S,T_1)$	Based on $L(S,T_1)$, Calculate $P(S,y,T_2)\ or\ \left[y_{T_2} \middle| S_{T_2} = S\right]$
……	……	……

Stochastic local volatility(SLV)

◆ **Numerical schemes for solving the density function PDE in M2 and M5:**
Both M2 and M5 are based on the Modified Craig–Sneyd (MCS) scheme for computing two-dimensional PDEs, implemented respectively in QPRLib and QuantLib.

$$\begin{cases} Y_0 = P_{n-1} + \Delta t \, F(t_{n-1}, P_{n-1}) \\ Y_j = Y_{j-1} + \theta\Delta t \, (F_j(t_n, Y_j) - (F_j(t_{n-1}, P_{n-1})), (j = 1,2) \\ \hat{Y}_0 = Y_0 + \theta\Delta t \, (F_0(t_n, Y_2) - (F_0(t_{n-1}, P_{n-1})) \\ \tilde{Y}_0 = \hat{Y}_0 + (\frac{1}{2} - \theta)\Delta t \, (F(t_n, Y_2) - (F_0(t_{n-1}, P_{n-1})) \\ \tilde{Y}_j = \tilde{Y}_{j-1} + \theta\Delta t \, (F_j(t_n, \tilde{Y}_j) - (F_j(t_{n-1}, P_{n-1})), (j = 1,2) \\ P_n = \tilde{Y}_2 \end{cases}$$