Pricing Quanto Options

- Advanced C++ Final Project -

Project Report

Omar Faruque

Fordham University
Gabelli School of Business

QFGB8960 Advanced C++ for Finance

May 1, 2025

Abstract

Quanto options are cross-currency derivatives that have payoffs in a fixed domestic currency but depend on foreign asset performance, requiring adjustments for asset–foreign exchange (FX) correlation. This report describes the implementation and validation of three numerical methods for pricing European quanto options within the C++ quantitative finance library, `qflib version 1.0.0`. We implement and validate three numerical pricing methods within the library: an analytical closed-form Black-Scholes solution, a Monte Carlo simulation framework, and a finite difference PDE solver. Our implementation incorporates the critical quanto adjustment factors, particularly the asset-FX correlation effect that modifies standard option pricing dynamics. Through rigorous testing across a range of parameters, we verify numerical convergence, confirm theoretical consistency through quanto call-put parity, and examine the economic impact of asset-FX correlation. Results demonstrate high numerical accuracy across all three methods, with relative errors below 5×10^{-5}, confirming the robustness of our implementation. This project contributes a valuable extension to `qflib`, enabling comprehensive quanto option analysis for practical financial applications.

Introduction

Quanto options are financial derivatives whose underlying asset is denominated in a foreign currency, but the payoff is made in the investor's domestic currency at a fixed exchange rate. This structure allows investors to gain exposure to foreign asset performance while eliminating direct currency risk. The term "quanto" originates from "quantity adjusting," referring to how the payoff adjusts in domestic currency based on the foreign asset's performance.

A European quanto call or put on an asset denominated in currency CCY1 pays $(\phi(S(T)-K))^+$ units of payoff currency CCY2, where $\phi = 1$ for a call and $\phi = -1$ for a put. The primary complication in pricing such options lies in capturing the correlation between the foreign asset and the exchange rate, which is not handled by the standard Black-Scholes framework.

This project implements and validates three numerical methods for pricing European quanto options:

1. An analytical solution using modified Black-Scholes formulas
2. A Monte Carlo simulation incorporating quanto-adjusted drift
3. A finite difference Partial Differential Equation (PDE) solver with adjusted parameters

Each method is integrated into the `qflib` C++ library and exposed through Python interfaces for practical use and testing.

Theoretical Framework

Quanto-Adjusted Stochastic Process and Forward Price

Consider a foreign asset whose price $S(t)$ is denominated in a foreign currency. The asset follows a geometric Brownian motion under its local risk-neutral measure:

$$dS(t) = (r_f - q)S(t)\,dt + \sigma_S S(t)\,dW_f(t) \tag{1}$$

where:

- $S(t)$: price of the foreign asset at time t
- r_f: foreign risk-free interest rate
- q: continuous dividend yield on the asset
- σ_S: asset volatility
- $W_f(t)$: Brownian motion under the foreign risk-neutral measure

Let $X(t)$ denote the exchange rate (domestic currency per unit of foreign currency). Under the domestic risk-neutral measure, the dynamics of $\ln X(t)$ are:

$$d\ln X(t) = (r_d - r_f - \tfrac{1}{2}\sigma_Q^2)dt + \sigma_Q\,dW_X^d(t) \tag{2}$$

where:

- $X(t)$: exchange rate
- r_d: domestic risk-free interest rate
- σ_Q: exchange rate volatility
- $W_X^d(t)$: Brownian motion under the domestic measure

To price a quanto derivative (payoff in domestic currency, asset in foreign currency), we need the dynamics of $S(t)$ under the domestic risk-neutral measure. Using Girsanov's theorem and assuming correlation ρ between $W_f(t)$ and $W_X^d(t)$, the asset price dynamics transform to:

$$dS(t) = (r_f - q + \rho\sigma_S\sigma_Q)\, S(t)\, dt + \sigma_S S(t)\, dW_S^d(t) \tag{3}$$

This results in a modified drift due to the correlation between the asset and FX rate. The term $\rho\sigma_S\sigma_Q$ is the quanto adjustment.

The quanto forward price under domestic measure is then:

$$F^Q(t,T) = \mathbb{E}^d[S(T) \mid \mathcal{F}_t] = S(t) \cdot \exp\left[(r_f - q + \rho\sigma_S\sigma_Q)(T-t)\right] \tag{4}$$

Alternatively, in the domestic risk-neutral world, cash flows grow at r_d and are discounted accordingly. To maintain consistency with Black-Scholes notation, we interpret the adjusted drift as a modified dividend yield q':

$$dS(t) = (r_d - q')S(t)\, dt + \sigma_S S(t)\, dW_S^d(t) \tag{5}$$

Matching drift terms in the two formulations:

$$r_d - q' = r_f - q + \rho\sigma_S\sigma_Q$$

Solving for q', we obtain the quanto-adjusted dividend yield:

$$q' = r_d - r_f + q - \rho\sigma_S\sigma_Q \tag{6}$$

This adjusted dividend yield is used in analytical, Monte Carlo, and PDE pricing formulations for quanto options.

Pricing Approaches

Quanto options require an adjustment to the dynamics of the foreign asset to reflect its payoff in domestic currency. This is done through two key modifications to the standard Black-Scholes framework:

- Replace the growth rate of the asset with a quanto-adjusted drift: $r_f - q + \rho\sigma_S\sigma_Q$
- Replace the dividend yield with an effective yield: $q' = r_d - r_f + q - \rho\sigma_S\sigma_Q$

These adjustments account for the correlation between the asset and FX rate, motivating both the forward value and the risk-neutral drift in the domestic measure.

We implement three distinct pricing methods, each implementing the quanto effects in its own way:

1. **Analytical**: Applies the Black-Scholes formula directly to the quanto-adjusted forward, which gives us a fast closed-form price.
2. **Monte Carlo**: Simulates paths under the modified drift and aggregates discounted payoffs to estimate price and uncertainty.
3. **PDE**: Solves the Black-Scholes PDE using finite differences with the adjusted dividend yield, producing a price and the entire value surface.

Implementation

Analytical Pricer

File: `qflib/pricers/simplepricers.hpp`

A new function declaration was added:

- `double quantoEuropeanOptionBS(payoffType, spot, strike, timeToExp, discRate, growthRate, divYield, assetVol, fxVol, correl)`

This function prices European quanto options in the Black-Scholes model. It incorporates:

- Domestic discount rate
- Foreign growth rate
- Dividend yield
- Correlation between asset and FX returns
- Asset and FX volatilities

File: `qlfib/pricers/simplepricers.cpp`

The corresponding implementation of `quantoEuropeanOptionBS(...)` was added. It extends the standard Black-Scholes pricing logic to handle quanto features. The key differences and computations in the implementation are:

- **Input Parameters:**
  - `payoffType` — integer, $+1$ for call, -1 for put.
  - `spot` — initial asset price S_0.

  - `strike` — strike price K.
  - `timeToExp` — time to maturity T (in years).
  - `discRate` — domestic discount rate r_d.
  - `growthRate` — foreign growth rate r_f.
  - `divYield` — asset dividend yield q.
  - `assetVol` — asset volatility σ_S.
  - `fxVol` — FX volatility σ_Q.
  - `correl` — correlation ρ between asset and FX returns.

- **Quanto Forward Computation:** The foreign asset is adjusted into domestic terms using the quanto forward:

$$F^Q = S_0 \cdot \exp\left((r_f - q + \rho\sigma_S\sigma_Q)T\right)$$

  where:

  - $r_f - q$ is the natural drift of the foreign asset under foreign risk-neutral measure.
  - $\rho\sigma_S\sigma_Q$ is the quanto drift term.

- **Discount Factor:** The domestic discount factor is computed as:

$$D(T) = e^{-r_d T}$$

- **Option Price:** Using the standard Black-Scholes formula with lognormal dynamics, the adjusted forward is plugged into:

$$d_1 = \frac{\ln(F^Q/K) + \frac{1}{2}\sigma_S^2 T}{\sigma_S\sqrt{T}}, \quad d_2 = d_1 - \sigma_S\sqrt{T}$$

  Then, the option price is:

$$\text{Price} = D(T) \cdot \left[\phi F^Q \cdot N(\phi d_1) - \phi K \cdot N(\phi d_2)\right]$$

  where ϕ depends on whether it's a call or a put.

- **Return:** The function returns a `double` scalar representing the present value of the quanto option.

Monte Carlo Pricer

New File: `qflib/pricers/bsmcquantopricer.hpp`

Defines the `BsMcQuantoPricer` class for Monte Carlo pricing of European quanto options.

Key elements:

- Accepts a product, domestic and foreign yield curves
- Takes asset vol, FX vol, correlation, and dividend yield
- Uses an Euler path generator
- Exposes:
  - `simulate(...)` for running the simulation
  - `processOnePath(...)` for evolving a single path

New File: `qflib/pricers/bsmcquantopricer.cpp`

Implements the complete logic for pricing quanto European options via Monte Carlo simulation:

- **Path Generator Construction**:
  - Selects an Euler path generator based on the user-specified uniform random number generator (URNG) type in `McParams`.
  - Supported URNGs include:
    * `MINSTDRAND`
    * `MT19937`
    * `RANLUX3`
    * `RANLUX4`
  - The path generator is configured to simulate the asset price.
- **Discount Factor Precomputation**:
  - For each payment time, the discount factor is computed using the domestic discount yield curve.
  - These factors are stored in the vector `discfactors_`.
- **Quanto-Adjusted Drift and Volatility Calculation**:
  - The solver loops through all fixing times to compute:
    * The adjusted drift:

$$\text{drift}_i = \left(r_f - q + \rho \sigma_S \sigma_Q - \frac{1}{2}\sigma_S^2 \right) \cdot (t_i - t_{i-1})$$

    * The diffusion term (volatility standard deviation):

$$\text{stdev}_i = \sigma_S \cdot \sqrt{t_i - t_{i-1}}$$

  - These values are stored in the vectors `drifts_` and `stdevs_`.

- The forward rate r_f is obtained from the growth yield curve.

- **Path Evolution via Euler Scheme**:
  - A matrix is allocated to store one sample path.
  - At each time step, the path is updated as:

$$S_{t_{i+1}} = S_{t_i} \cdot \exp(\text{drift}_i + \text{stdev}_i \cdot Z_i)$$

    where $Z_i \sim \mathcal{N}(0,1)$ is drawn from the URNG.

- **Payoff Evaluation**:
  - The generated path is passed to the product's `eval()` method.
  - This function populates the internal vector `payamts_` with the product's payoffs at each pay time.

- **Present Value Computation**:
  - The final pathwise value is computed by summing the discounted payoff cashflows:

$$\text{PV} = \sum_i \texttt{discfactors\_}[i] \cdot \texttt{payamts\_}[i]$$

- **Return Value**:
  - The method `processOnePath()` returns a `double` value representing the discounted present value (PV) of the payoff along one simulated price path.
  - This scalar PV is passed to the `StatisticsCalculator` in `simulate()` to compute and return the Monte Carlo mean (estimated option price) and standard error.

PDE Pricer

File: `qflib/methods/pde/pde1dsolver.hpp`

Adds a 10-argument constructor to enable quanto PDE pricing:

```
Pde1DSolver(SPtrProduct, SPtrYieldCurve, SPtrYieldCurve, double,
double, double, double, double, Pde1DResults&, bool)
```

This constructor accepts:

- Growth curve (foreign market)
- Asset and FX volatilities
- Correlation between asset and FX

Also introduces:

- Boolean `isQuanto_`
- Scalars: `assetVol_`, `fxVol_`, `correl_`

File: `qflib/methods/pdepde1dsolver.cpp`

The additional quanto-aware constructor computes the effective dividend yield using the formula:

$$q' = r_d - r_f + q - \rho\sigma_S\sigma_Q$$

where:

- r_d is the domestic interest rate (obtained from the discount yield curve),
- r_f is the foreign growth rate (from the growth yield curve),
- q is the original dividend yield,
- ρ is the correlation between the asset and the FX rate,
- σ_S, σ_Q are the asset and FX volatilities, respectively.

This adjusted dividend yield is assigned to the internal vector `divyields_`, which is later used in the PDE drift calculation. The value is computed once in the constructor and no further changes are needed to the PDE stepping logic.

The rest of the PDE implementation, including methods such as `solveFromStepToStep()`, `initValLayers()`, and `evalProduct()`, remains unchanged. These functions use the modified dividend yield transparently through the existing interface, allowing the same numerical solver to handle both standard and quanto products without modification.

- **Return Value**:
  - The pricing result is stored in `results.prices[0]`.

Wrapper Functions

File: `pyqflib/pyfunctions5.hpp`

Defines Python-callable interfaces for quanto pricing.

Analytical Wrapper: `pyQfQuantoEuroBS`

This function provides a Python interface to the analytical Black-Scholes formula for pricing European quanto options, implemented in `quantoEuropeanOptionBS`.

- Accepts scalar inputs: spot price, strike, time to expiration, domestic discount rate, foreign growth rate, dividend yield, asset volatility, FX volatility, and correlation between asset and FX returns.
- Internally calls the analytical pricer which computes the quanto-adjusted forward price:

  $$F^Q = S_0 \cdot e^{(r_f - q + \rho\sigma_S\sigma_Q)T}$$

  and applies the standard Black-Scholes formula.

- No yield curves or market handles are required; all inputs are passed directly as numerical values.

Return Value:

- A single scalar value representing the analytical price of the European quanto call or put option.

Monte Carlo Wrapper: `pyQfQuantoEuroBSMC`

This function wraps the Monte Carlo pricer `BsMcQuantoPricer` and is used to simulate the price of a European quanto option under the Black-Scholes model.

- Inputs include:
  - Basic option parameters (payoff type, strike, time to expiration, spot).
  - Curve names for the domestic discount curve and foreign growth curve.
  - Market data: dividend yield, asset and FX volatilities, correlation.
  - Monte Carlo parameters (URNG type, path generator, number of paths, etc.).
- Internally:
  1. Retrieves the specified yield curves from the global `market()` object.
  2. Constructs a `EuropeanCallPut` product.
  3. Instantiates `BsMcQuantoPricer` with all inputs.
  4. Uses a `MeanVarCalculator` to run the simulation and aggregate results.

Return Value:

- A Python dictionary with two keys:
  - `"Price"`: Monte Carlo estimate of the option price (sample mean)
  - `"StdErr"`: Standard error of the estimate

PDE Wrapper: `pyQfQuantoEuroBSPDE`

This function exposes the finite difference PDE-based pricing of quanto European options using the `Pde1DSolver` class.

- Accepts all standard option and market parameters as well as:
  - Names of the discount and growth yield curves
  - PDE solver configuration dictionary: number of spot/time nodes, θ scheme value, spot range (in std devs), etc.
  - A boolean flag to optionally return the full solution grid
- Internally:

1. Constructs a `EuropeanCallPut` product
2. Instantiates `Pde1DSolver` with quanto parameters
3. Applies the quanto-adjusted dividend yield
4. Solves the PDE backward from maturity to time zero

Return Value:

- If `allresults` is `False`:
  - `"Price"`: scalar solution at the initial spot
- If `allresults` is `True`:
  - `"Price"`: scalar solution at the initial spot
  - `"Times"`: time grid used in the PDE
  - `"Spots"`: spot price grid
  - `"Values"`: 2D grid of option values over time and spot

Verification and Testing

Quanto Forward Convergence

We verified that as the strike of the quanto call approaches zero, its price converges to the discounted quanto forward:

```
strike_near_zero = 1.00e-10
Discounted Quanto Forward = 101.00501671
Quanto Price = 101.00501671
rel_error = 9.70e-13
```

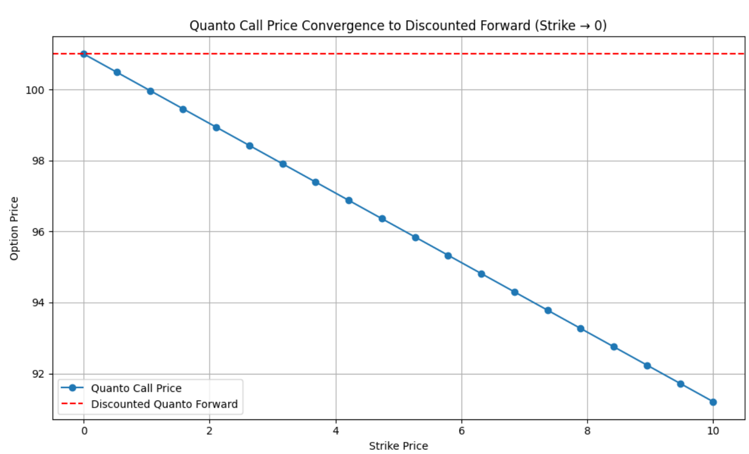

Figure 1: Quanto Convergence as Strike → 0

Method Comparison

We compared all three pricing methods for a standard test case:

- Spot: 100, Strike: 100
- Asset volatility: 0.2, FX volatility: 0.15, Correlation: 0.5
- Domestic rates: [0.01, 0.012, 0.02, 0.023, 0.03, 0.035]
- Foreign (growth) rates: [0.015, 0.018, 0.025, 0.03, 0.035, 0.04]
- Maturities: [0.25, 0.5, 1.0, 2.0, 5.0, 10.0] years
- Dividend yield: 0.01

Results:

- Analytical: 9.50800966
- Monte Carlo (30M paths): 9.50758241 (Rel. Error: 4.49e-05)
- PDE (25600 nodes): 9.50780061 (Rel. Error: 2.20e-05)

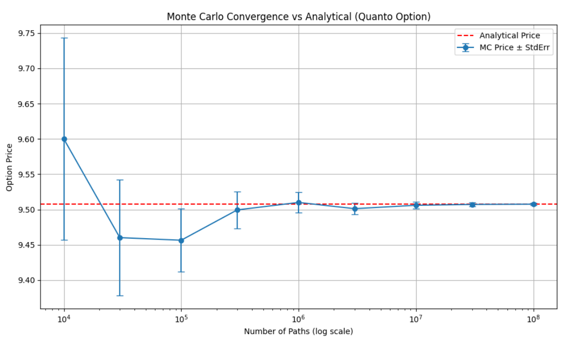

Figure 2: Monte Carlo Convergence

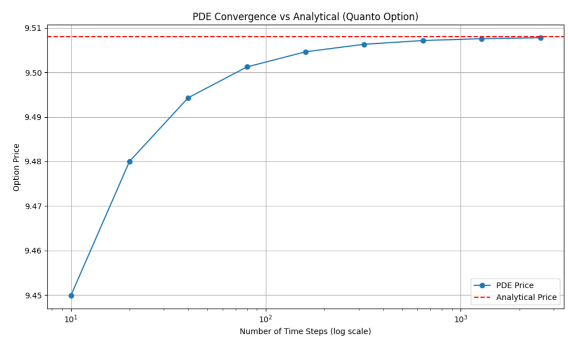

Figure 3: PDE Convergence

This demonstrates excellent agreement between all three methods, confirming the implementation's accuracy with enough paths/steps.

Convergence Analysis

We conducted convergence studies for both the Monte Carlo and PDE methods:

Monte Carlo Convergence

- Tested path counts from 10,000 to 30,000,000
- Standard error decreases with $\sqrt{N}$ as expected
- Final relative error: 4.49e-05

PDE Convergence

- Tested grid refinement from 10 to 2560 time steps
- Corresponding spot nodes from 100 to 25600
- Final relative error: 2.20e-05

Research Questions

Quanto Call-Put Parity

Theoretical Derivation

For regular options, the call-put parity is:

$$\text{Call} - \text{Put} = D(T) \times (F(T) - K) \tag{7}$$

For quanto options, the adjusted parity becomes:

$$\text{Call}_{\text{quanto}}(K,T) - \text{Put}_{\text{quanto}}(K,T) = e^{-r_d T} \times (S_0 \times e^{(r_f - q + \rho\sigma_S\sigma_Q)T} - K) \tag{8}$$

Numerical Verification

We tested the parity across multiple strikes (80-120) and maturities (0.5-2.0 years):

```
=== Maturity (T) - 1.0 year(s) ===
Strike      LHS                 RHS                 Abs Error      Rel Error
80.00       22.58912284         22.58912284         0.00e+00       0.00e+00
90.00       12.78713611         12.78713611         7.11e-15       5.56e-16
100.00      2.98514938          2.98514938          0.00e+00       0.00e+00
110.00      -6.81683736         -6.81683736         0.00e+00       0.00e+00
120.00      -16.61882409        -16.61882409        0.00e+00       0.00e+00
```

The maximum relative error across all tests was approximately 10^{-15}, confirming the theoretical consistency of our implementation.

Effect of Asset/FX Correlation on Option Prices

Experiment Design

We analyzed the effect of correlation (ρ) on quanto option prices by varying ρ from -1.0 to 1.0 while keeping other parameters constant.

Results and Analysis

The key findings include:

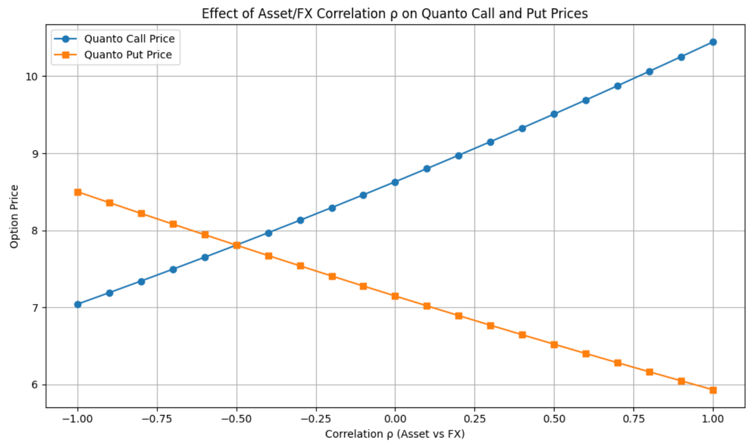

Figure 4: Effect of Correlation on Quanto Option Price

Quanto Calls

- Call prices increase with correlation
- When $\rho > 0$: As asset price rises, FX strengthens, increasing domestic payoff
- When $\rho < 0$: As asset price rises, FX weakens, reducing domestic payoff

Quanto Puts

- Put prices decrease with correlation
- When $\rho > 0$: As asset price falls, FX weakens, reducing domestic payoff
- When $\rho < 0$: As asset price falls, FX strengthens, increasing domestic payoff

Mathematical Interpretation

- The effective drift adjustment is increased by $\rho\sigma_S\sigma_Q$
- A positive ρ shifts the forward price upward, favoring calls
- A negative ρ shifts the forward price downward, favoring puts

Economic Explanation

The correlation between the asset and FX rates has significant economic implications for quanto option pricing:

- **Positive Correlation** ($\rho > 0$): When the foreign asset performs well, the foreign currency also strengthens against the domestic currency. In a standard (non-quanto) foreign investment, this would amplify returns when converted to domestic currency. However, in a quanto product, the exchange rate is fixed, so this natural "double benefit" is lost. The quanto call option must therefore be priced higher to compensate, while the quanto put becomes less valuable.

- **Negative Correlation** ($\rho < 0$): When the foreign asset performs well, the foreign currency weakens against the domestic currency. In a standard foreign investment, this would reduce returns when converted to domestic currency. A quanto structure removes this natural hedge, making quanto call options less valuable and quanto puts more valuable.

Summary and Conclusions

We have successfully implemented and verified three methods for pricing European quanto options within the qflib library:

1. The analytical Black-Scholes solution provides an efficient and accurate baseline for pricing.
2. The Monte Carlo method offers flexibility for complex scenarios and demonstrates excellent convergence properties.
3. The PDE approach provides both pricing and the full solution grid for detailed analysis.

Our numerical tests confirm that all three methods converge to the same price, with relative errors below 5×10^{-5}. We also verified the theoretical quanto call-put parity relationship and analyzed how asset-FX correlation affects option prices.

The correlation parameter (ρ) plays a crucial role in quanto option pricing, with quanto calls benefiting from positive correlation and quanto puts benefiting from negative correlation. This effect is explained by how correlation influences the effective growth rate of the foreign asset when viewed in domestic currency terms.

This implementation provides a robust framework for pricing and analyzing quanto options, supporting both practical trading applications and theoretical research.

References

1. Hull, J. C. (2018). *Options, Futures, and Other Derivatives.* Pearson.
2. Wilmott, P. (2006). *Paul Wilmott on Quantitative Finance.* Wiley.
3. Shreve, S. E. (2004). *Stochastic Calculus for Finance II: Continuous-Time Models.* Springer.