Prepared for: MAFS 5220 Quantitative and statistical risk analysis
Prepared by: Group 2
17 May, 2015

# Computing CVA, DVA & FVA of Interest Rate Swap with a Non-collateral Counter-party Using C++ Programming Language

## EXECUTIVE SUMMARY

**INSIDE:**



### Background

After the financial crisis in 2007, risk from different aspects of counter-parties exploded. In order to find out the reasonable price of a specific financial instrument or a portfolio, taking into consideration the risk of defaulting events, many financial institutes started to implement methods and models to make adjustments to the price for the counter-party. Therefore we will increase the default-risky price by adding a supplementary amount to the default-free price of the contracts, which is often called Credit Valuation Adjustment, or CVA. Similarly, considering the adjustment in debit and funding issues, we would have DVA (Debit Valuation Adjustment) and FVA (Funding Valuation Adjustment).

Basel III requires CVA calculation in pricing an instrument or portfolio. Although DVA is not required in Basel III, Financial Accounting Standard (FAS) and International Accounting Standard (IAS) both require DVA calculation. FVA is not required now in any standard but raises debate among scholars, thus we calculate FVA as well.

### Project Outline

In our project, we calculate CVA, DVA and FVA of interest rate swap using QuantLib in C++ with the consideration of wrong way risk without collateral and netting issues.

| Main Function | Sub Function | Status |
|---|---|---|
| UCVA | AT1P | Pass |
| | CDS Calibration | Pass |
| | UCVA Calculation | Pass |
| | Intensity Model | Pass |
| WWR UCVA | CIR Calibration | Pass |
| | CIR | Pass |
| | UCVAwithWWR | Pass |
| BVA (CVA+DVA) | BVA Calculation | Pass |
| FVA | AMC Simulation | Pass |
| | FVA | Unfinished |

## ASSUMPTIONS

Considering one product with Wrong Way Risk in calculating CVA and DVA without collateral and netting.

We use two methods to calculate UCVA. In the first one, we use AT1P to model default process and use swaption to calculate UCVA. In the second one, we use CIR process to model short interest rate and hazard rate, and use Cholesky decomposition to make two processes correlated.

In the first one, we make following assumptions:

a) We have not considered WWR (Wrong Way Risk), which means that the interest rate is uncorrelated with the default process.

b) The underlying process for the value of the firm is a Geometric Brownian Motion (GBM) lognormal dynamics. The drift of the dynamics for the firm value and for the default barrier is the same.

c) The default threshold is a deterministic, known function of time, based on reliable accounting data.

d) We only consider one product rather than the portfolio with netting.

In the second one, we make following assumptions:

a) Default is not triggered by basic market observables but has an exogenous component, independent of all the default-free market information.

To simplify, we assume there is no collateral.

# MODELS & FORMULAS

| |
|---|
| AT1P model: |
| The survival probability is given analytically by<br>$$\mathbb{Q}(\tau > T) = \left[\Phi\left(\frac{\log\frac{V_0}{H} + \frac{2B-1}{2}\int_0^T \sigma_u^2 du}{\sqrt{\int_0^T \sigma_u^2 du}}\right) - \left(\frac{H}{V_0}\right)^{2B-1}\Phi\left(\frac{\log\frac{V_0}{H} + \frac{2B-1}{2}\int_0^T \sigma_u^2 du}{\sqrt{\int_0^T \sigma_u^2 du}}\right)\right]$$<br>Based on the above AT1P formula, we can fit the model parameters to the market data using CDS to calibrate the model. Then we can use all the parameters to calculate survival probability at every needed point.<br>Realized Class:<br>AT1Pmodel.cpp/ AT1Pmodel.hpp |
| CDS calibration: |
| Straightforward computations lead to the price at initial time of a CDS as<br>$$\mathrm{CDS}_{a,b}(0, \mathrm{R}, \mathrm{LGD}) := -\mathrm{R}\int_{T_a}^{T_b} P(0,t)(t - T_{\beta(t)-1})d\mathbb{Q}(\tau < t) - R\sum_{i=a+1}^{b} P(0,T_i)\,\alpha_i \mathbb{Q}(\tau \geq T_i) + LGD\int_{T_a}^{T_b} P(0,t)d\mathbb{Q}(\tau < t)$$<br>Then we can use the data from the Brigo's book of Leman Brother's default case to calibrate AT1P parameters: H, B, . In our calibration, we use the optimizer function in QuantLib.<br>Realized Class:<br>CDSCalibrateProblemFunction.cpp/CDSCalibrateProblemFunction.hpp<br>CDSCalibrator.cpp/CDSCalibrator.hpp |
| IRS UCVA calculation: |
| For the UCVA term considering only one instrument, we can use Monte Carlo simulation to simulate different paths and get the scenarios when default happens, then average the result to get our IRS UCVA. Thus we use the following formula:<br>$$U_{CVA}(t, T_b) = LGD\ \mathbb{E}_t\left[\mathbf{1}_{\{t \leq T_b\}} D(t,\tau)\left(NPV(\tau)\right)^+\right]$$<br>Realized Class:<br>IRSUCVAcalculator.cpp/ IRSUCVAcalculator.hpp |
| CIR process: |
| We use the Cox-Ingersoll-Ross Model to fit the short rate movement. The model formulation under the risk-neutral measure Q is:<br>$$dr(t) = k(\theta - r(t))dt + \sigma\sqrt{r(t)}dW(t), r(0) = r_0$$<br>***with*** $r_0$, $k$, $\theta$, $\sigma$ ***positive constants and the condition:*** $2k\theta > \sigma^2$.<br>In terms of the stochastic parameters, we use standard normal distribution.<br>Realized Class:<br>CIRprocess.cpp/CIRprocess.hpp |

CIR calibration:

There are many ways to calibrate CIR model. In QuantLib, there is a class related to the calibration work. But it is pretty difficult to rewrite and use in our own program. Thus we find an indirect way to calibrate the CIR process at this stage.

Parameters definition: $\boldsymbol{\phi}$: **discrete drift**; $\boldsymbol{\theta}$: **long run average**

Formula:

We need to optimize the residual terms:

$$\frac{(r_t - \phi r_{t-1})^2}{r_t + \theta}$$

by changing phi. RSS: residuals.

Then we can get: $$k = -\ln(\phi)$$

discrete volatility: $$\sigma_a = \sqrt{RSS/(N-1)}$$

Continuous volatility: $$\sigma = \sqrt{(2k\,\sigma_a{}^2)/(1 - e^{-2k})}$$

$$r_0 = \textbf{the first day short rate}$$

We use optimizer method in this class.

Realized Class:

CIRcalibration.cpp/CIRcalibration.hpp

IRS UCVA calculation with wrong way risk:

Using the intensity model, we can get default time by the following function:

$\tau = \Lambda^{-1}(\xi);\ \Lambda(\tau) := \int_0^t \lambda(u)du,$

where $\xi$ adopts the standard exponential random variable

Then we generate two independent series z1 and z2 (normal distribution), use Cholesky decomposition to get two dependent series and apply them into the integrating function of interest rate and hazard rate CIR process. If the hazard rate integrating result is greater than or equal to $\Lambda(t)$, default event happens.

If simulating result shows no default, the UCVA on this path is 0; else we use NPV function to get the UCVA value.

NPV formula for fixed payer:

$$NPV = Floating\ Leg\ NPV - Fixed\ Leg\ NPV$$

Realized Class: CVAwithWWR.cpp/ CVAwithWWR.hpp

IRS BVA calculation:

First we use CIR process to simulate two defaulting processes of the Investor and Counterparty simultaneously and then compare them to the simulated hazard intensity to get the first-to-default counterparty and default time.

The separate defaulting process is developed on the basis of WWR model because of better generalization.

Next we will calculate the BVA (DVA-CVA) using the NPV function. If simulating result shows no default, the UCVA on this path is 0; else if B defaults first, we use DVA formula to calculate; else if C defaults first, we use DVA formula to calculate.

$$\begin{aligned}
DVA(t,T) &= LGD_I\ \mathbb{E}_t\left[1_{\{\tau_I<\tau_C<T\}\cup\{\tau_I<T\le\tau_C\}} D(t,\tau_I)\big(-NPV(\tau_I)\big)^+\right]\\
CVA(t,T) &= LGD_C\ \mathbb{E}_t\left[1_{\{\tau_C<\tau_I<T\}\cup\{\tau_C<T\le\tau_I\}} D(t,\tau_C)\big(NPV(\tau_C)\big)^+\right]\\
BVA(t,T) &= DVA(t,T) - CVA(t,T)
\end{aligned}$$

Realized Class: IRSBVA.cpp/ IRSBVA.hpp

FVA Calculation:

We are considering the FVA Calculation for a non-collateral counterparty, thus a simplified model of that in the textbook. Similar with former sessions, we use CIR model to simulate the related stochastic processes. Since it involves highly non-linear and recursive computation, we cannot use normal Monte Carlo but American Monte Carlo to generate the Expectation Value of NPV. The formulas as followings:

$$\boldsymbol{F_t = \bar{V}_t - H_t}$$

$$\boldsymbol{E_{t_j}\left[\bar{\Phi}(t_j, t_{j+1}, F)\right] = -1_{\{\tau > t_j\}}\left[F_{t_j}^{-}\frac{P_{t_j}(t_{j+1})}{P_{t_j}^{f-}(t_{j+1})} + F_{t_j}^{+}\frac{P_{t_j}(t_{j+1})}{P_{t_j}^{f+}(t_{j+1})}\right]}$$

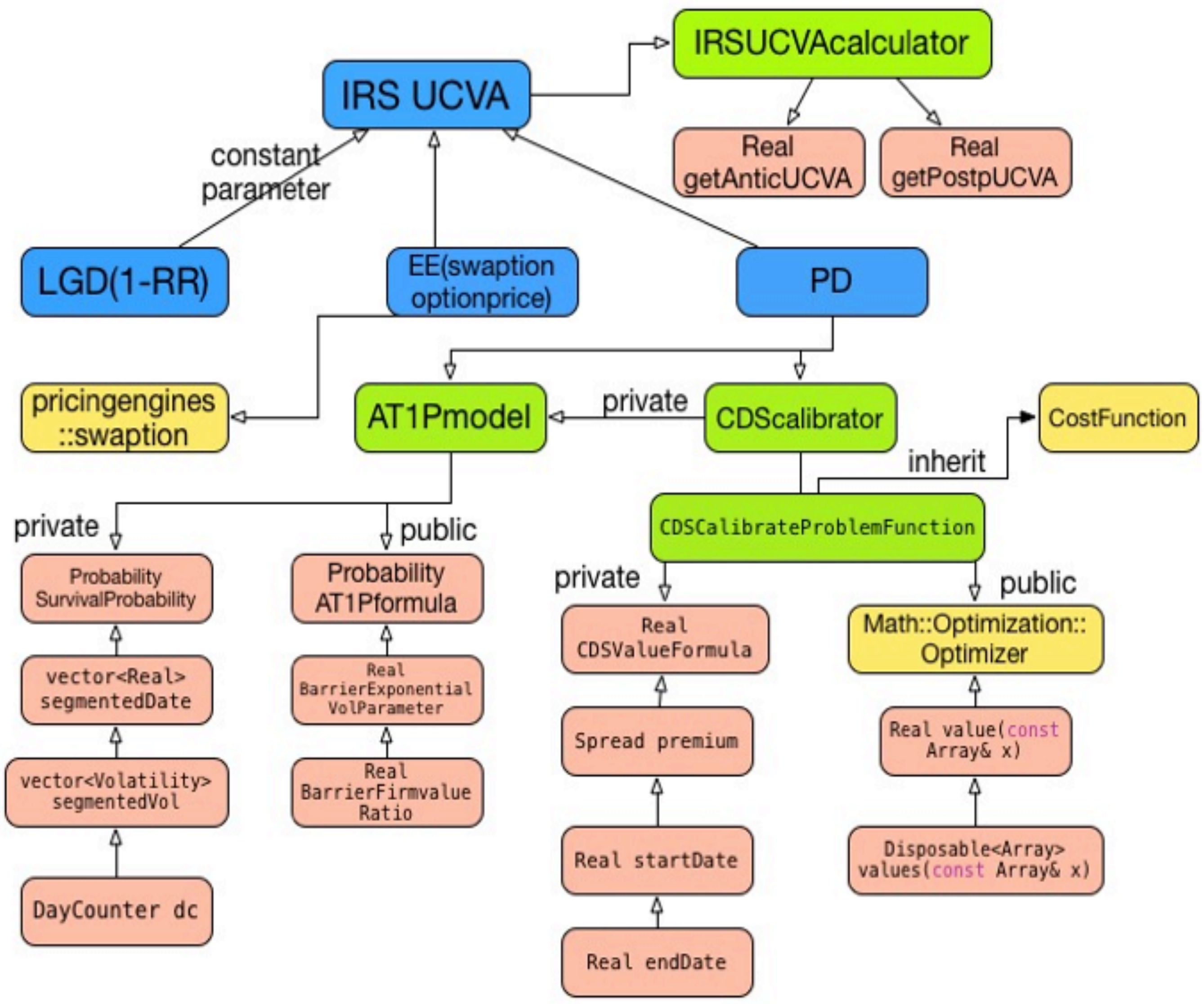
$$\boldsymbol{\phi(t, T\wedge\tau; F) = \sum_{j=1}^{m-1} 1_{\{t\le t_j < T\wedge\tau\}} D(t, t_j)\left(F_{t_j} + E_{t_j}\left[\bar{\Phi}(t_j, t_{j+1}, F)\right]\right)}$$

$$\boldsymbol{\bar{V}(F) = E\left[\bar{\Pi}(t, T) + \phi(t, T\wedge\tau; F)\right]}$$

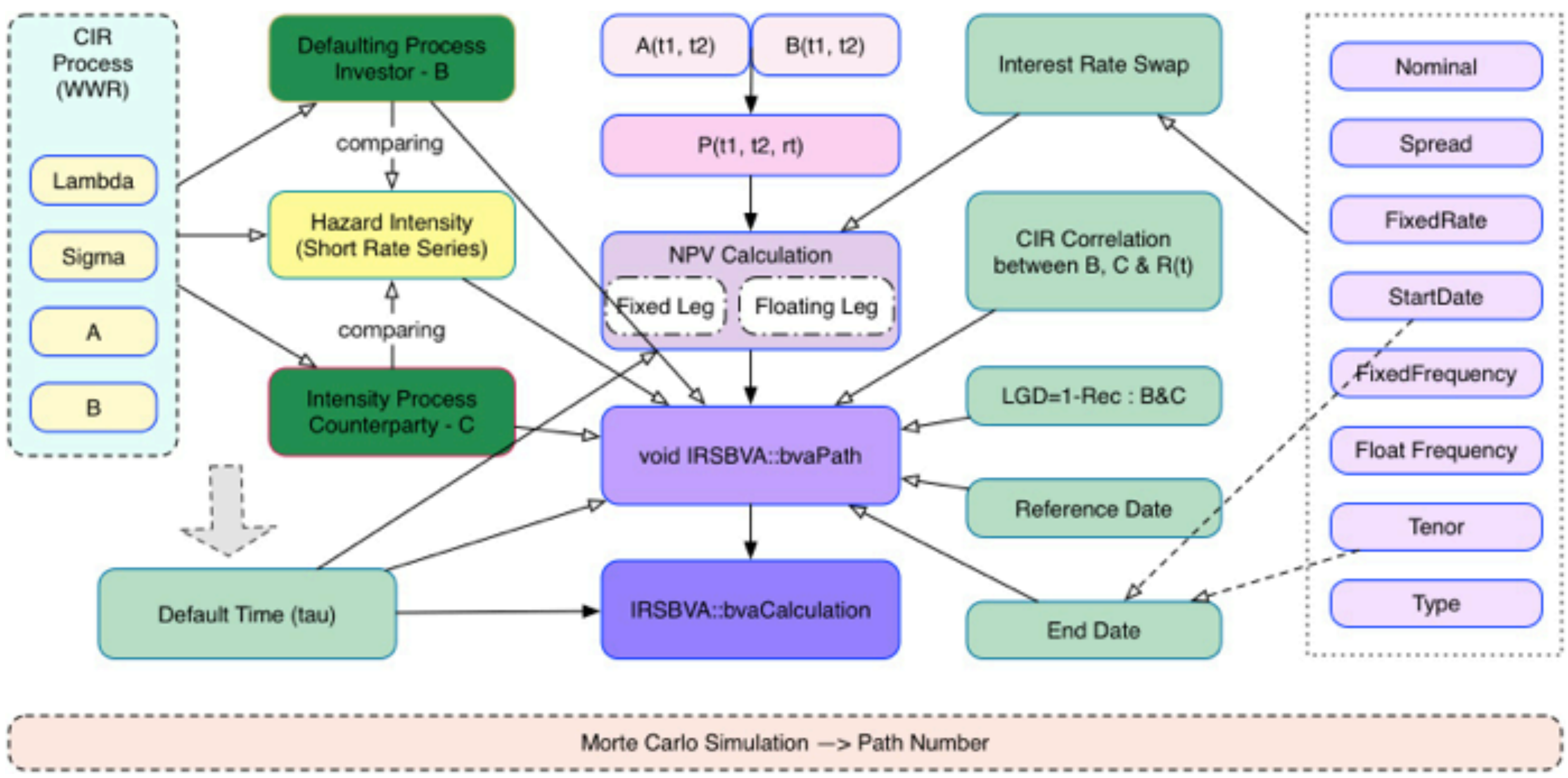
Due to unexpected workload, we didn’t realized the FVA class as planned.

Realized Class: AMC.cpp/ AMC.hpp

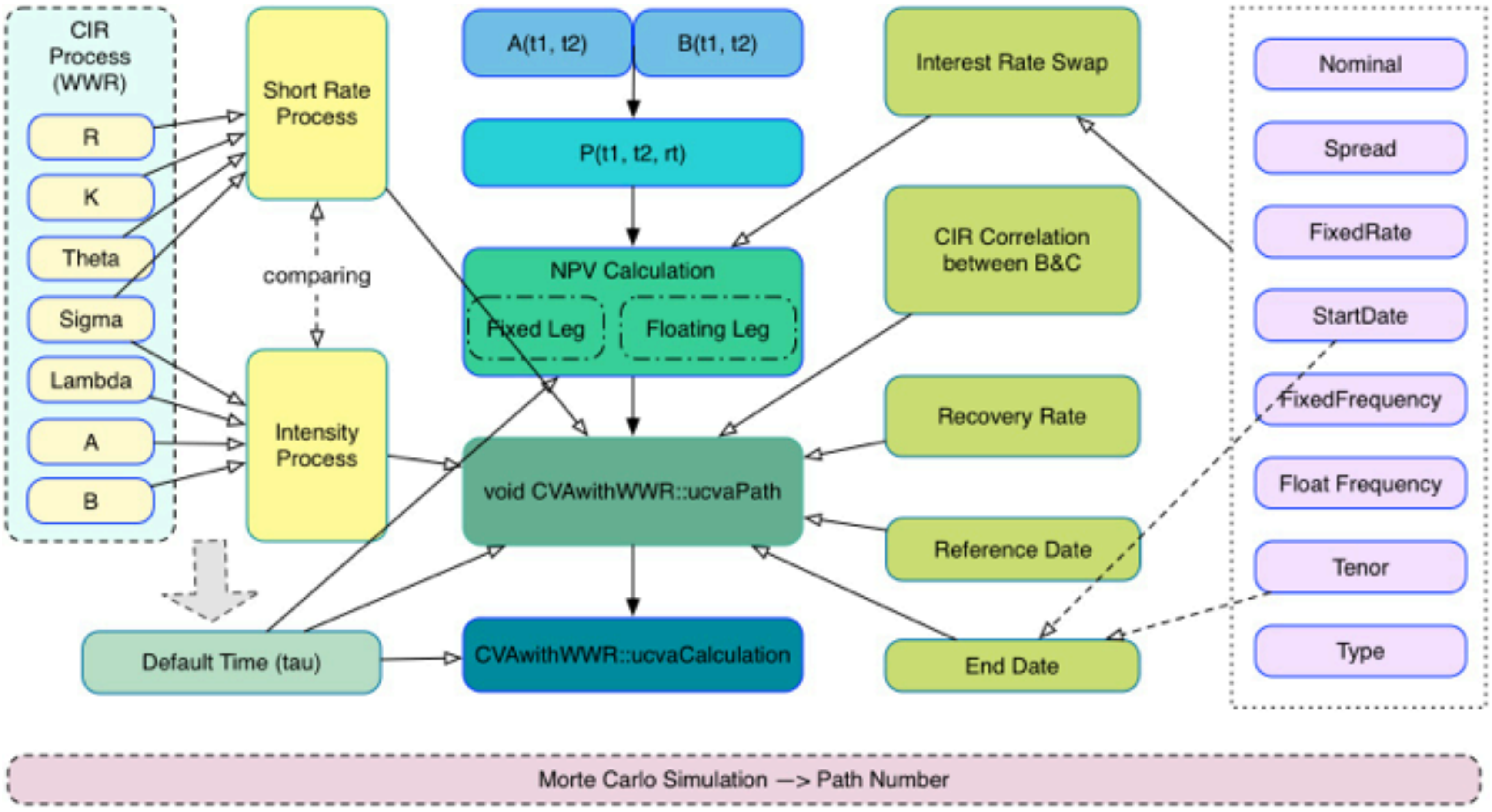
## SCHEMATICS



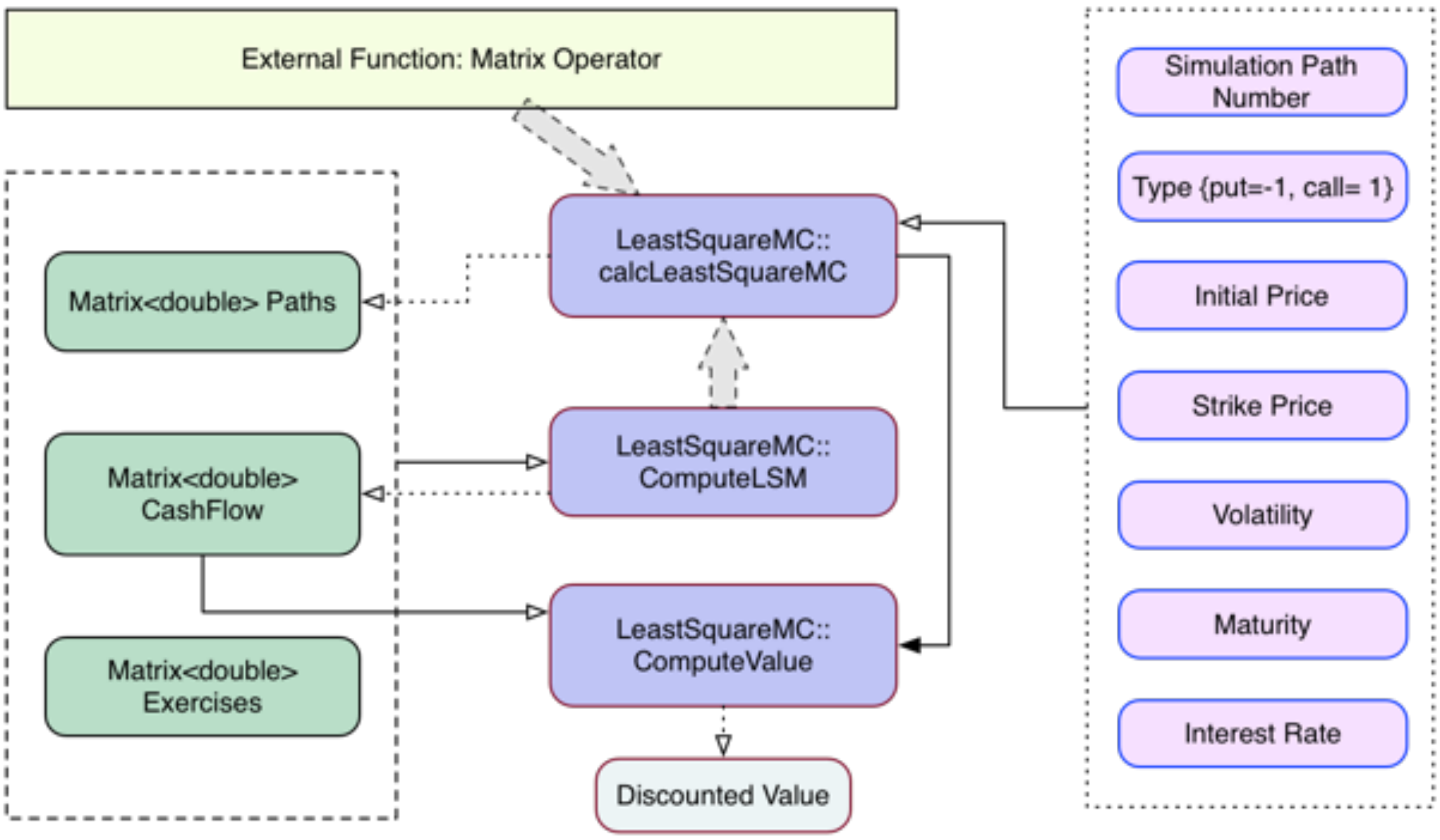

*Graph 1. IRS UCVA Schematic*




*Graph 2. BVA Schematic*




*Graph 3. IRS UCVA with WWR Schematic*




*Graph 4. American Monte Carlo Schematic in FVA*

# KEY POINTS

**In calculating CVA with WWR and BVA under CIR process**

1. How to deal with correlation?

To deal with the correlation between short rate and hazard rate in CVA calculation, and three correlation among short rate and two hazard rates of B and C. I firstly generate independent random number and then use Cholesky decomposition rather than copula function to make them correlate.

2. How to calculate NPV?

Another difficulty that I have encountered is to calculate NPV of remaining cash flow. To handle it, I use to vector of Date floating leg and fixed leg to store the cash flow exchange date. The NPV of Libor interest equals to P(tao,T1)-P(tao,TN), of which the T1 is the first floating leg exchange date and TN is the last one. P is the present price at t1 of 1 at t2

P(Date t1, Date t2, Real rt)= A(t1, t2)*exp(-B(t1,t2)*rt).

3. How to get default time?

Because the integral of hazard rate obeys exponential distribution, first I get a random number conforming exponential distribution, then when the integral equals to that number, I can get the default time.

**In Calibration CIR Model**

First when want to derive our own calibration model out of QuantLib's library. If we do so, we need three kinds of classes for calibration: 1. CIRprocess (as an instrument) 2. CIRhelper 3. CIR model. But we find it will be troublesome to do such calibration. Fortunately, we found a web site where mentioned a method using Excel and some functions to calibrate the model. Then we follow this framework. The reference website: *"http://financetrainingcourse.com/education/2012/06/cox-ingersoll-ross-cir-interest-rate-model-parameter-calibration-short-rates-simulation-and-modeling-of-longer-term-interest-rates-an-example/"*

**In programming of Least Square Monte Carlo Simulation**

1. In doing regression, we use paths that are in the money.

As Longstaff and Schwartz point out, using only in-the-money paths limits the region over which the conditional expectation must be estimated and greatly reduces the number of basis functions needed to obtain an accurate approximation of the conditional expectation function.

2. Matrix operation has been used heavily.

To get the regression beta coefficient, we use the function

std::vector<double> betavec(std::vector<std::vector<double> > X, std::vector<double> Y),

and to get the continuation value, we use the function

std::vector<double> MVecMult(std::vector<std::vector<double> > X, std::vector<double> Y);

In doing this, we use LU decomposition to get upper triangular matrix and lower triangular matrix first.

3. Basis Function

We use 1, S,  as basis function.

# NUMERICAL TESTS

Considering one product with Wrong Way Risk in calculating CVA and DVA without collateral and netting.

We use two methods to calculate UCVA. In the first one, we use AT1P to model default process and use swaption to calculate UCVA. In the second one, we use CIR process to model short interest rate and hazard rate, and use Cholesky decomposition to make two processes correlated.

In the first one, we make following assumptions:

a) We have not considered WWR (Wrong Way Risk), which means that the interest rate is uncorrelated with the default process.

b) The underlying process for the value of the firm is a Geometric Brownian Motion (GBM) lognormal dynamics. The drift of the dynamics for the firm value and for the default barrier is the same.

c) The default threshold is a deterministic, known function of time, based on reliable accounting data.

d) We only consider one product rather than the portfolio with netting.

In the second one, we make following assumptions:

a) Default is not triggered by basic market observables but has an exogenous component, independent of all the default-free market information.

| | |
|---|---|
| AT1P test: | Showing the same results as the Leman Brother's data in the book. |
| CDS calibration test: | Showing the same results as the Leman Brother's data in the book. |
| IRS UCVA test: | Example from: *Interest Rate Swap Credit Value Adjustment*. |
| CIR calibration test: | Excel example from following website: *http://financetrainingcourse.com/education/2012/06/cox-ingersoll-ross-cir-interest-rate-model-parameter-calibration-short-rates-simulation-and-modeling-of-longer-term-interest-rates-an-example/* |
| IRS CVA with WWR test: | Making trending analysis. |
| IRS DVA test: | Making trending analysis. |

### Trending Analysis

In this part, we take some numerical examples as test cases to analyze the movement of UCVA with WWR and DVA along with the movement of different parameters to see whether the results is accordance with our financial intuition. If the results meet our intuition, we can prove our program is right. If the results do not meet our intuition, there may be two reasons for these. Then we will change our program or consider whether there are other things that we did not take into consideration. Then we will fix the inconsistency.

**Trending Analysis for UCVA with wrong way risk**

1. The change of UCVA with the change of WWR (correlation coefficient)

| Parameter Used | |
|---|---|
| CIR For Interest Rate | |
| Initial R (R0) | 0.04 |
| Long Term Reversion(Theta) | 0.0404 |
| Reversing Speed (K) | 0.1 |
| Sigma1 | 0.2 |
| CIR For Hazard Rate | |
| Initial Lamda | 0.04 |
| Long Term Reversion(B) | 0.03 |
| Reversing Speed (A) | 0.1 |
| Sigma2 | 0.2 |
| Interest Rate Swap | |
| Recovery Rate | 0.4 |
| Start Date | 31/12/12 |
| Tenor | 5 |
| Nominal | 1 |
| Spread | 0.0001 |
| Fixed Rate | 0.0405 |
| Fixed Frequency | Annual |
| Floating Frequency | Semiannual |

** The reference date (the day to calculate UCVA) is the same as start date; the investor is the fixed payer*

The correlation in Brigo's book represents dependence between the instantaneous differences of the interest rates and default intensity, i.e. instantaneous correlation, which should be negative. In semi-analytical formula the correlation is between levels of the interest rates and the time of default, which should be positive.

Because we use CIR to model interest rate and default process, so the correlation represents the instantaneous correlation. For WWR problem, we only need to consider negative correlation coefficient. The more negative correlation represents the higher level of WWR, leading to higher UCVA for fixed payer.

We change the instantaneous correlation coefficient from -1.0 to 0.0, and the changing amount is 0.1 every time. For every correlation coefficient, we use 10000 paths to simulate. As shown in the following figure, more negative correlation coefficient represents higher level of WWR, leading to higher UCVA, which is in accordance with our intuition

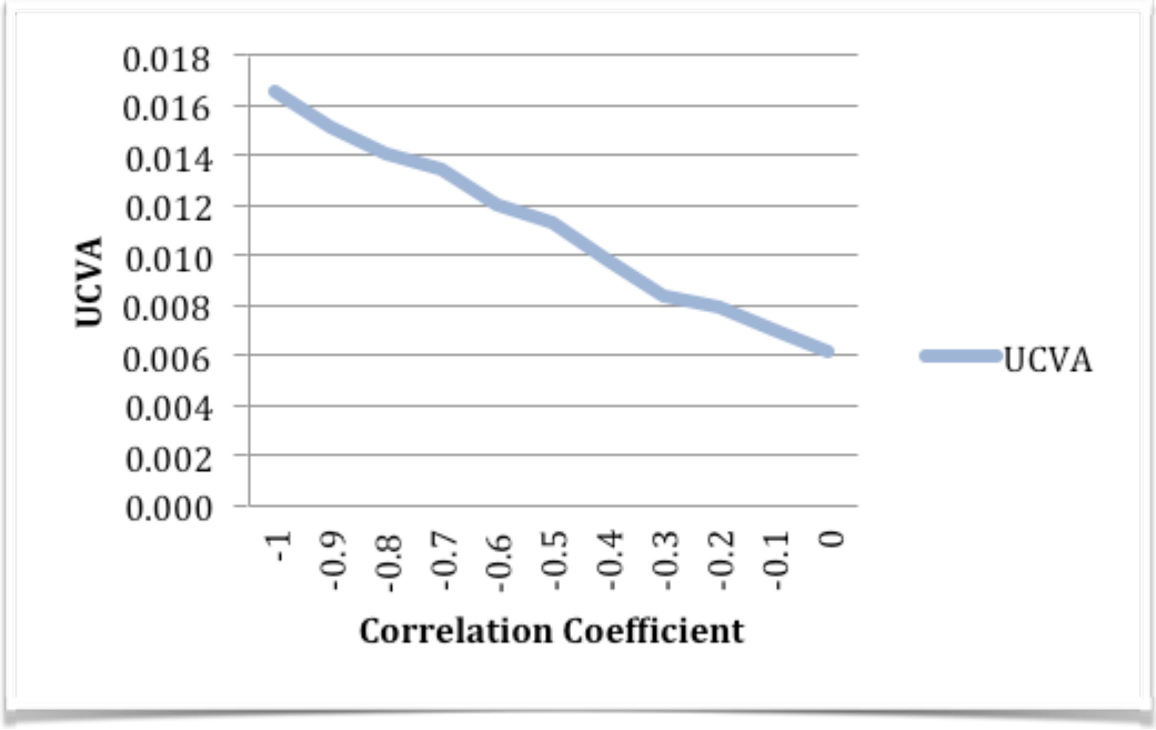



**Figure 1** Unilateral CVA with WWR as a function of the correlation between the instantaneous differences of exposure (interest rates) and default intensity

2. The change of UCVA with the change of tenor

| Parameter Used | |
|---|---|
| CIR For Interest Rate | |
| Initial R (R0) | 0.04 |
| Long Term Reversion(Theta) | 0.0404 |
| Reversing Speed (K) | 0.1 |
| Sigma1 | 0.2 |
| CIR For Hazard Rate | |
| Initial Lamda | 0.04 |
| Long Term Reversion(B) | 0.03 |
| Reversing Speed (A) | 0.1 |
| Sigma2 | 0.2 |
| Correlation Coefficient | -0.3 |
| Interest Rate Swap | |
| Recovery Rate | 0.4 |
| Start Date | 31/12/12 |
| Tenor | 5 |
| Nominal | 1 |
| Spread | 0.0001 |
| Fixed Rate | 0.0405 |
| Fixed Frequency | Annual |
| Floating Frequency | Semiannual |

** The reference date (the day to calculate UCVA) is the same as start date; the investor is the fixed payer*

We change the tenor from 1 to 10, and the changing amount is 1 every time. For every correlation coefficient, we use 10000 paths to simulate. As shown in the following figure, contract with longer tenor will have higher UCVA, which is in accordance with our intuition.

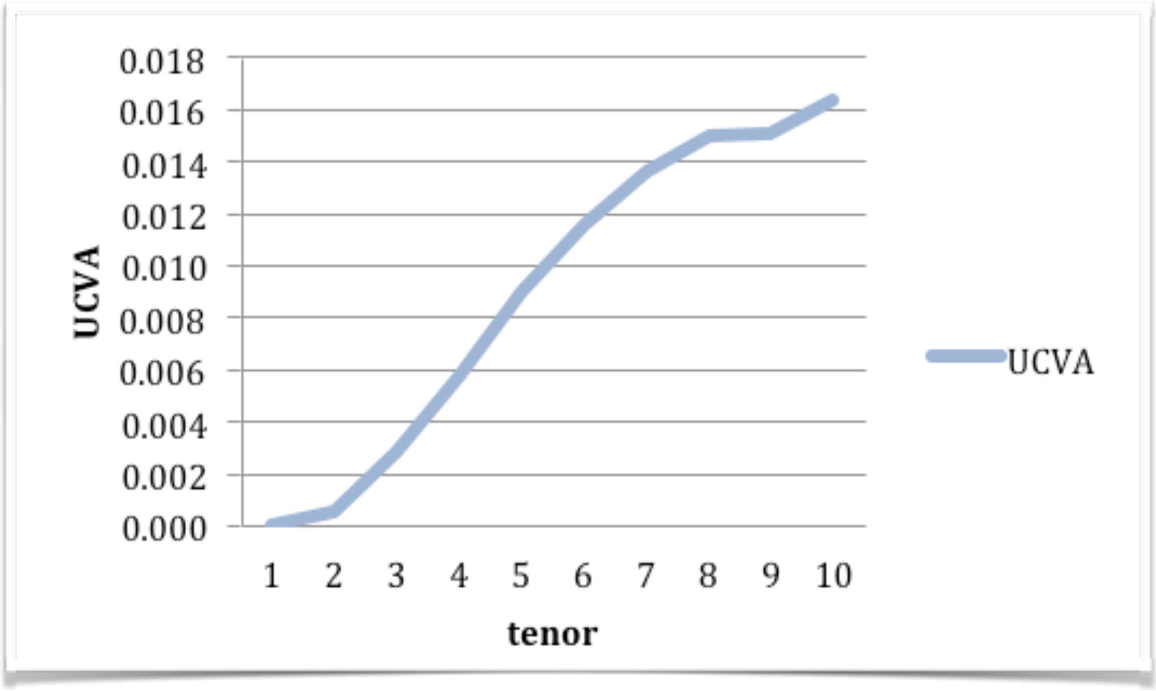



**Figure 2** Unilateral CVA with WWR as a function of the tenor

<u>3. The change of UCVA with the change of initial hazard rate</u>

| Parameter Used | |
|---|---|
| CIR For Interest Rate | |
| Initial R (R0) | 0.04 |
| Long Term Reversion(Theta) | 0.0404 |
| Reversing Speed (K) | 0.1 |
| Sigma1 | 0.2 |
| CIR For Hazard Rate | |
| Long Term Reversion(B) | 0.03 |
| Reversing Speed (A) | 0.1 |
| Sigma2 | 0.2 |
| Correlation Coefficient | -0.3 |
| Interest Rate Swap | |
| Recovery Rate | 0.4 |
| Start Date | 31/12/12 |
| Tenor | 5 |
| Nominal | 1 |
| Spread | 0.0001 |
| Fixed Rate | 0.0405 |
| Fixed Frequency | Annual |
| Floating Frequency | Semiannual |

** The reference date (the day to calculate UCVA) is the same as start date; the investor is the fixed payer*

We change the initial hazard rate from 0.02 to 0.2, and the changing amount is 0.02 every time. For every correlation coefficient, we use 10000 paths to simulate. As shown in the following figure, contract with higher initial hazard will have lower ucva. This result seems counterintuitive at first, so we refer to the book "Counterparty credit risk and credit value adjustment: a continuing challenge for global financial markets" written by Jon Gregory and find a reasonable explanation: It makes sense when one considers that for a better credit quality counterparty, default is a less probable event and therefore represents a bigger surprise when it comes.

Thus we note an important general conclusion: Wrong-way risk increases as the credit quality of the counterparty increases.

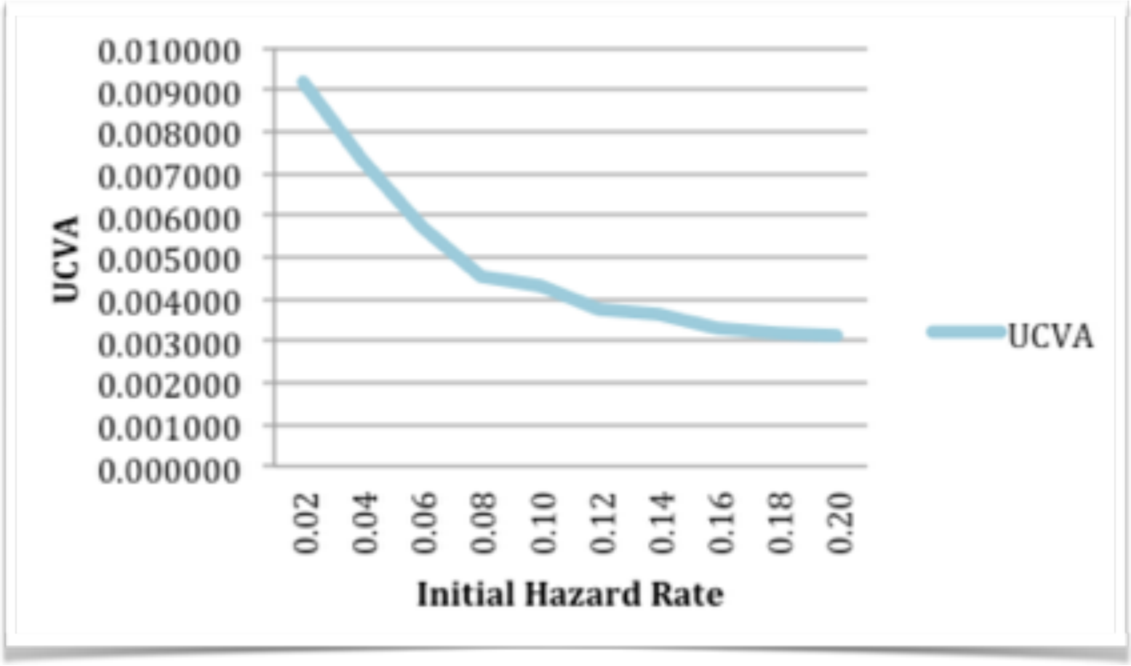



**Figure 3** Unilateral CVA with WWR as a function of the initial hazard rate

**Trending Analysis for BVA**

1. The change of BVA with the change of WWR (correlation coefficient)

In this part of analysis, we change the correlation coefficient from -0.8 to 0.9 to see whether the BVA is sensitive to the correlation coefficient between the investor's default rate and counterparty's default rate, between the investor's default rate and the interest rate, between the counterparty's default rate and the interest rate.

*Rho between B and C*

| Parameter Used | |
|---|---|
| CIR For Interest Rate | |
| Initial R (R0) | 0.04 |
| Long Term Reversion(Theta) | 0.0404 |
| Reversing Speed (K) | 0.1 |
| Sigma1 | 0.2 |
| CIR For Hazard Rate(B) | |
| Initial Lamda | 0.0165 |
| Long Term Reversion(B) | 0.026 |
| Reversing Speed (A) | 0.4 |
| Sigma2 | 0.14 |
| CIR For Hazard Rate© | |
| Initial Lamda | 0.0165 |
| Long Term Reversion(B) | 0.026 |
| Reversing Speed (A) | 0.4 |
| Sigma2 | 0.14 |
| Correlation Coefficient (br) | 0.2 |
| Correlation Coefficient (cr) | 0.3 |
| Interest Rate Swap | |
| Recovery Rate | 0.4 |
| Start Date | 31/12/12 |
| Tenor | 5 |
| Nominal | 1 |
| Spread | 0.0001 |
| Fixed Rate | 0.0405 |
| Fixed Frequency | Annual |
| Floating Frequency | Semiannual |

* *The reference date (the day to calculate BVA) is the same as start date; the investor is the fixed payer*

For the impact of wrong way risk, the symmetry of CVA and DVA implies that if one is affected by wrong- way risk then the other should show the effect of right-way risk. Then if the correlation between the investor and the counterparty is bigger, the BVA result will be easily clustered either on the negative part or the positive part. The "clustering effects" will lead to a increasing in absolute value of BVA on either positive side or negative side.

The following picture shows the moving trend of the BVA with the rho.

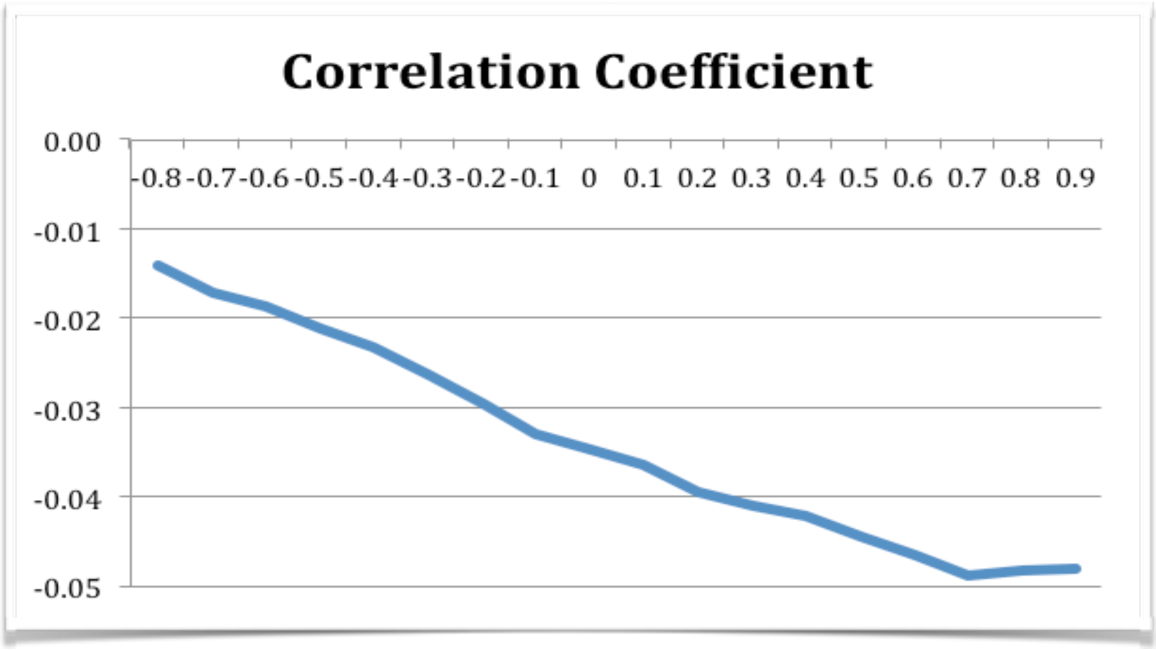



**Figure 4** BVA with WWR as a function of the correlation between two default intensities

*Rho between B, C and interest rate*

| Parameter Used | |
|---|---|
| CIR For Interest Rate | |
| Initial R (R0) | 0.04 |
| Long Term Reversion(Theta) | 0.0404 |
| Reversing Speed (K) | 0.1 |
| Sigma1 | 0.2 |
| CIR For Hazard Rate(B) | |
| Initial Lamda | 0.0165 |
| Long Term Reversion(B) | 0.026 |
| Reversing Speed (A) | 0.4 |
| Sigma2 | 0.14 |
| CIR For Hazard Rate(C) | |
| Initial Lamda | 0.0165 |
| Long Term Reversion(B) | 0.026 |
| Reversing Speed (A) | 0.4 |
| Sigma2 | 0.14 |
| Correlation Coefficient (bc) | 0.1 |
| Interest Rate Swap | |
| Recovery Rate | 0.4 |
| Start Date | 31/12/12 |
| Tenor | 5 |
| Nominal | 1 |
| Spread | 0.0001 |
| Fixed Rate | 0.0405 |
| Fixed Frequency | Annual |
| Floating Frequency | Semiannual |

** The reference date (the day to calculate BVA) is the same as start date; the investor is the fixed payer*

Just as we mentioned earlier, the BVA movement along correlation coefficient between B and C and interest rate is similar to the UCVA movement. We show the surface here and we can easily see the symmetry and monotonous decreasing trending here. The explanation is the same as UCVA's relationship.

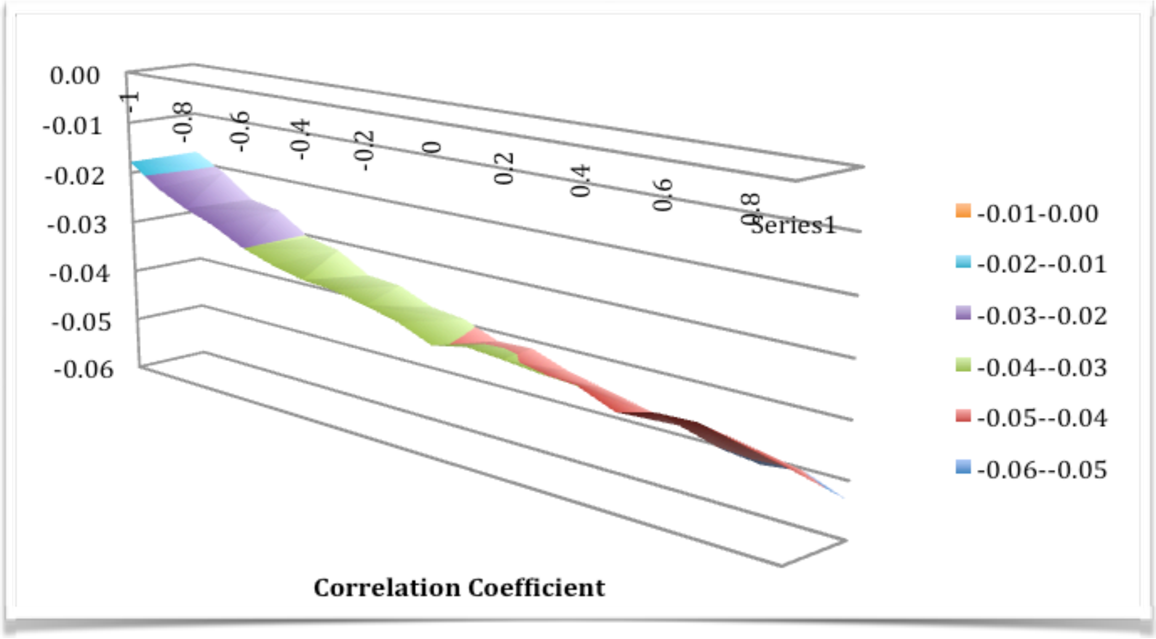



**Figure 5** BVA with WWR as a function of the correlation between the instantaneous differences of exposure (interest rates) and default intensity

<u>2. The change of BVA with the change of tenor</u>

| Parameter Used | |
|---|---|
| CIR For Interest Rate | |
| Initial R (R0) | 0.04 |
| Long Term Reversion(Theta) | 0.0404 |
| Reversing Speed (K) | 0.1 |
| Sigma1 | 0.2 |
| CIR For Hazard Rate(B) | |
| Initial Lamda | 0.0165 |
| Long Term Reversion(B) | 0.026 |
| Reversing Speed (A) | 0.4 |
| Sigma2 | 0.14 |
| CIR For Hazard Rate© | |
| Initial Lamda | 0.0165 |
| Long Term Reversion(B) | 0.026 |
| Reversing Speed (A) | 0.4 |
| Sigma2 | 0.14 |
| Correlation Coefficient (bc) | 0.1 |
| Correlation Coefficient (br) | 0.2 |
| Correlation Coefficient (cr) | 0.3 |
| Interest Rate Swap | |
| Recovery Rate | 0.4 |
| Start Date | 31/12/12 |
| Nominal | 1 |
| Spread | 0.0001 |
| Fixed Rate | 0.0405 |
| Fixed Frequency | Annual |
| Floating Frequency | Semiannual |

** The reference date (the day to calculate BVA) is the same as start date; the investor is the floating payer*

We change the tenor from 1 to 10, and the changing amount is 1 every time. For every correlation coefficient, we use 10000 paths to simulate. As shown in the following figure, contract with longer tenor will have higher BVA, which is in accordance with our intuition.


**Figure 6** BVA with WWR as a function of the tenor

**Least Square Monte Carlo Simulation**

We use our code to calculate the price of the American put option.

Parameters used:

| | |
|---|---|
| Path Number | 10000 |
| Number of Time Steps | 20 |
| Initial Stock Price | 100 |
| Strike Price | 95 |
| Time to Expiration in years | 1 |
| Interest Rate | 0.1 |
| Volatility per Year | 0.3 |

We contrast our result with the price calculated using Options Calculator in this website: http://www.math.columbia.edu/~smirnov/options13.html

Results show some differences perhaps due to different methods used and basis functions, which are not that accurate.

| Type | Least Square MC | Lattice Tree |
|---|---|---|
| Put | 5.66 | 6.28 |

# APPENDIX

## Interface

### a. AT1Pmodel Class

*Construction*

AT1Pmodel(Real B = 0.0, Real Ratio = 0.4):BarrierExponentialVolParameter(B), BarrierFirmvalueRatio(Ratio)

*Operations*

Probability SurvivalProbability(Date testDate)
Probability SurvivalProbability(Real testDate)
void Initialization(std::vector<Date> dateVector, std::vector<Volatility> volVector)
void Initialization(Date startDate, std::vector<Real> dateVector, std::vector<Volatility> volVector)
Probability at1pFormula(Volatility cumulativeVariance)

### b. CDSCalibrateProblemFunction Class

*Construction*

CDSCalibrateProblemFunction()

*Operations*

void Initialization(Real LGDRate, Real RiskFreeRate)
void Initialization(std::vector<Date> dateVector, std::vector<Real> quoteVector)
Real value(const Array& x) const
Disposable<Array> values(const Array& x) const
Real testCDSCalculation(std::vector<Volatility> volVector, int termNumber)
Real differentialization(Real testDate) const
Real IntergratedCallFunction(Spread premium, Real currDate) const
Real CDSValueFormula(Spread premium, Real startDate, Real endDate) const

### c. CDSCalibrator Class

*Construction*

CDSCalibrator(Real LGDRate, Real interestRate = 0.5)

*Operations*

void Initialization(std::vector<Date> dateVector, std::vector<Real> quoteVector)
void StartCalibrate()
Probability FinalizedSurvivalProbability(Date testDate)
Volatility FinalizedVolatility(int i)
Real testCDSCalculation(std::vector<Volatility> volVector, int termNumber)

### d. DefaultCurve Class

*Construction*

DefaultCurve( DefaultCurveType type, std::vector<Date> creditTerm, std::vector<Probability> survivalProb)
DefaultCurve( DefaultCurveType type, Date startDate, Rate ConstantIntensity)

*Operations*

DefaultCurveType getType()
Date getStartDate()
std::vector<Date> getDefaultTerm()
Probability getSurvivalProb(size_t index)
void buildCurve(Period maturity, Period interval)

### e. IntensityModel Class

*Construction*

IntensityModel()

*Operations*

Probability SurvivalProbability(Date date)
void Initialization(std::vector<Date> dateVector, std::vector<Rate> intensityVector)

### f. IRSUCVAcalculator Class

*Construction*

IRSUCVAcalculator()

*Operations*

void Initialization(std::vector<Date> creditTerm, std::vector<Probability> survivalProb)
Real getAnticUCVA(Rate swapRate, Real maturity)
Real getPostpUCVA(Rate swapRate, Real maturity)

### g. pch Class

#include <ql/quantlib.hpp>
#include <vector>
#include <math.h>

### h. CIRprocess Class

*Construction*

CIRprocess(Rate r01 = 0.05, Real theta1 = 0.1, Real k1 = 0.1, Real sigma1 = 0.1)

*Operations*

Real P(Date startDate, Date endDate, Rate rt) const
Real D(Date startDate, Date endDate) const
Real A(Date startDate, Date endDate) const
Real B(Date startDate, Date endDate) const

### i. CIRCalibration Class

*Construction*

CIRCalibration (Real r0)
void Initialization(std::vector<Date> dateVector, std::vector<Real> quoteVector)

*Operations*

Real value(const Array& x) const
Disposable<Array> values(const Array& x) const
Real Calculation(Real rt_, Real rt_1, Real phi1, Real gamma1) const
Real getGamma()

### j. CVAwithWWR Class

*Construction*

CVAwithWWR()
CVAwithWWR& WithShortRateCIRParam(Rate r0, Real theta, Real k, Real sigma1)
CVAwithWWR& WithIntensityCIRParam(Real lamda0, Real a, Real b, Real sigma2)
CVAwithWWR& WithCIRCorrelation(Real rho)
CVAwithWWR& WithRecoveryRate(Real recoveryRate)
CVAwithWWR& WithIRSSwap(Real nominal, Real spread, Real fixedRate, Date startDate, Frequency fixedFrequency, Frequency floatingFrequency, Period tenor, Type type)
CVAwithWWR& WithReferenceDate(Date referenceDate)

*Operations*

void initialization()
Real A(Date t1, Date t2)
Real B(Date t1, Date t2)
Real P(Date t1, Date t2, Real rt)
Real npv(Date tao)
Real ucvaPath(Date referenceDate, Date endDate)
Real cvaWithWWRCalculation(int pathNum)

## Interface (continued)

### k. IRSBVA Class

#### *Construction*

IRSBVA()
IRSBVA& WithShortRateCIRParam(Rate r0, Real ar, Real br, Real sigmar)
IRSBVA& WithBIntensityCIRParam(Real lamda0b, Real ab, Real bb, Real sigmab)
IRSBVA& WithCIntensityCIRParam(Real lamda0c, Real ac, Real bc, Real sigmac)
IRSBVA& WithCIRCorrelation(Real rhobc, Real rhobr, Real rhocr)
IRSBVA& WithRecoveryRate(Real recoveryRateb, Real recoveryRatec)
IRSBVA& WithIRSSwap(Real nominal, Real spread, Real fixedRate, Date startDate, Frequency fixedFrequency, Frequency floatingFrequency, Period tenor, Type type)
IRSBVA& WithReferenceDate(Date referenceDate)

#### *Operations*

void initialization ()
Real A(Date t1, Date t2)
Real B(Date t1, Date t2)
Real P(Date t1, Date t2, Real rt)
Real npv(Date tao)
Real bvaPath(Date referenceDate, Date endDate)
Real bvaCalculation(int pathNum)

### l. MatrixOperation Class

#### *Construction*

CMatrixOperations(void)

#### *Operations*

LUstruct LU(std::vector<std::vector<double> > A)
std::vector<std::vector<double> > MatUpTriangleInv(std::vector<std::vector<double> > U)
std::vector<std::vector<double> > MatLowTriangleInv(std::vector<std::vector<double> > L)
std::vector<std::vector<double> > MMult(std::vector<std::vector<double> >A, std::vector<std::vector<double> >B, int n, int k, int m)
std::vector<std::vector<double> > MInvLU(std::vector<std::vector<double> > A)
std::vector<std::vector<double> > MTrans(std::vector<std::vector<double> >A)
std::vector<double> MVecMult(std::vector<std::vector<double> > X, std::vector<double> Y) std::vector<double> betavec(std::vector<std::vector<double> > X, std::vector<double> Y)

### m. LeastSquareMC Class

#### *Construction*

LeastSquareMC(int M, int N, double initPrice, double T, double strike, double rate, double vol, Type type)

#### *Operations*

double computeLSM(int time, matrix<double>& Paths, matrix<double>& CashFlow, matrix<double>& Exercise)
double computeValue(matrix<double>& CashFlow)
double max(double x, double y)
double calcLeastSquareMC()