ESG Documentation Pack

SVJD

Market Consistent Calibration Method

Version 1.0
March 2013

barrie+hibbert
A Moody's Analytics Company

Document version history

Version	Date	Last updated by	Comments
1.0	8/3/13	Gioel Calabrese, Witold Gawlikowicz, Natacha Lord	

Contents

Introduction

This document describes the market-consistent (MC) calibration of the stochastic volatility jump diffusion (SVJD) model. The model, its parameters, its implementation and the pricing of European equity options are discussed in the model definition (MD) document.

Overview of SVJD MC calibration

The market consistent calibration of the SVJD model entails finding the parameter values which optimise the model's ability to reproduce equity option market data and the 25 year extrapolated forward at-the-money equity implied volatility. The effect of stochastic interest rates on equity option pricing is not taken into account, but is monitored via the ESG validation.

SVJD calibration methodology

The SVJD model, its parameters, its implementation and the equity option pricing are described in the MD document. Calibrating the model amounts to finding the values of Initial Variance (v_0), Reversion Level (θ), Reversion Speed (α), Vol Of Variance (ξ), Correlation (ρ), Jump Intensity (λ), Jump Mean (μ_J) and Jump Volatility (σ_J) such that the discrepancies between the model prices and market prices of European equity put options, including the extrapolated 25 year target, are minimised.

A step-by-step description of the SVJD calibration methodology is shown below. The subsections that follow describe the different components of the calibration methodology in more detail.

The calibration step-by-step

Once the SVJD calibration tool is populated with appropriate data and settings, the market consistent calibration of the model consists of the following automated steps:

Regressed implied volatility surface

Perform a local linear regression on the market implied volatility, including the 25 year target. Use the regressed data to perform a second iteration of local linear regression.

↓

Vega scaled weighting matrix

Calculate vega from the regressed surface at points having a non-zero weight and use vega to rescale/filter weights.

↓

Pseudo market prices

Generate European put prices based on regressed implied volatility using Black-Scholes. These are the prices we fit to.

↓

Levenberg-Marquardt optimiser

Search for locally optimum parameter values (within bounds) minimising the weighted squared norm of the vector of residuals, including Feller condition.

↓

Calculate all model prices and implied volatilities

Calculate model prices at optimum value of parameters for all maturities and strikes and convert to implied volatility for assessment of quality of fit.

↓

Display and assess quality of fit against original market implied volatility

Market equity option implied volatilities

The calibration of the SVJD model requires market data for equity option implied volatilities for maturities ranging from 3 months to 10 years and forward-strike prices ranging from 0.6 to 1.4 (relative to initial stock price).

25 year at-the-money implied volatility target

Our standard quarterly calibration also uses a 25 years forward ATM equity implied volatility target. This target is calculated by extrapolating the ATM market implied volatilities using an exponential functional form, taking into account the real world unconditional volatility target for developed economies. The target ensures that we have a degree of control over the long term behaviour of the model and that we have consistency between different equity models (Fixed Volatility, TVDV and SVJD).

Regressed equity option implied volatilities

Including target, the implied volatility surface contains up to 222 points. However, the SVJD model only has 8 parameters and, in general, cannot be expected to achieve a perfect fit to all market data points. There is therefore scope to reduce the number of market data points to speed up the calibration process, without necessarily reducing the quality of fit.

The elimination of market data points from the fitting process must satisfy two requirements: (1) we want to avoid accidentally fitting to a particularly noisy point; (2) we want to maintain some dependency on all market data points. These requirements can be met by performing a local 3x3 linear regression to smooth the original market data surface.

The local linear regression is carried out as follows:

1. Let $\mathrm{T_i}, \mathrm{i} = 1, \dots 14$ and $\mathrm{K_j}, \mathrm{j} = 1, \dots 17$ be the maturities and the forward strikes.
2. The regressed market data point $IV_{i,j}^{\mathrm{reg}}$ is calculated by performing a linear regression on the 9 neighbouring market data points $IV_{i-1,j-1}^{\mathrm{market}}$, $IV_{i-1,j}^{\mathrm{market}}$, $IV_{i-1,j+1}^{\mathrm{market}}$, $IV_{i,j-1}^{\mathrm{market}}$, $IV_{i,j}^{\mathrm{market}}$, $IV_{i,j+1}^{\mathrm{market}}$, $IV_{i+1,j-1}^{\mathrm{market}}$, $IV_{i+1,j}^{\mathrm{market}}$, $IV_{i+1,j+1}^{\mathrm{market}}$.
3. In cases where the linear regression fails (e.g. there are fewer than 3 market data points), a one dimensional linear regression is performed on the 3 market data points $IV_{i-1,j}^{\mathrm{market}}$, $IV_{i,j}^{\mathrm{market}}$, $IV_{i+1,j}^{\mathrm{market}}$. If this still fails, then no regressed data point is produced and, if available, $IV_{i,j}^{\mathrm{market}}$ is returned.

Weighting matrix

The objective function used by the optimiser is a weighted sum of squares of residuals. The weights are specified in the weighting matrix for different maturities (T) and strikes (K). The weights used at EndMar2013 for all equity indices are based on the weighting matrix shown below. Although we do not anticipate making changes to this matrix, we may do so if market conditions change significantly.

The matrix is deliberately sparse to reduce number of points to price, ensuring rapid calibrations. The allocation of weights together with the rescaling described below ensures that fits of reasonably good quality are obtained.

Exhibit 1

User defined weighting matrix.

User defined weights

	Swap Tenor																
Maturity	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1	1.05	1.1	1.15	1.2	1.25	1.3	1.35	1.4
0.25	-	-	-	-	-	-	-	-	10.00	-	-	-	-	-	-	-	-
0.5	-	-	3.00	-	-	-	-	-	-	-	-	1.00	-	-	2.00	-	-
0.75	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	1.00	-	-	1.00	-	-	-	-	-	-	-	1.00	-	-	1.00	-
4	-	-	-	-	-	-	-	-	10.00	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	1.00	-	-	-	-	-	-	-	-	-	-	-	-	-	1.00	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	1.00	-	-	-	8.00	-	-	-	1.00	-	-	-	-
25	-	-	-	-	-	-	-	-	2.00	-	-	-	-	-	-	-	-

Rescaling of weights

The SVJD calibration does not fit to implied volatilities, but to prices. The reason for this is that the tool estimates European equity put option prices directly using the COS method [KB 2013-2532] and it would be computationally expensive to convert these prices into implied volatilities (this conversion would have to be carried out each time an option price is estimated, which can happen several thousands of times for a typical calibration).

Equity option prices vary considerably across different maturities and strikes. A uniform weighting matrix would place less emphasis in regions where prices are small (low maturities, deeply out-of-the-money options). To avoid this, at each point where the user defined weights are non-zero, the weight is rescaled according to the following algorithm:

1. Calculate Vega $\nu = \phi(d_1)\sqrt{T-t}$, where $T-t$ is the maturity, $d_1 = \frac{-\ln \tilde{K} + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}}$, $\phi(x) = \frac{1}{\sqrt{2\,\pi}}\, e^{-\frac{x^2}{2}}$, $\tilde{K}$ is the forward strike and σ is the regressed implied volatility.
2. If ν drops below a certain threshold, we set the corresponding weight to zero. The threshold value is currently set to 0.001. This assures that the optimisation ignores points for which there is little variation in prices when implied volatility changes[1].
3. Rescale the weight by dividing it by ν^2.

We note that if $\Delta\sigma$ is sufficiently small, then the mean squared implied volatility error can be approximated by the mean squared price error, where the weights have been rescaled by ν^2:

Equation 1

$$\sum_i w_i(\Delta\sigma_i)^2 \approx \sum_i \frac{w_i}{\nu_i^2}(\Delta P_i)^2$$

This relationship shows that the weight rescaling effectively achieves a fit to implied volatility. An example of a vega scaled weighting matrix is given below.

Exhibit 2

Vega scaled weighting matrix.

Vega Scaled Weights

	Swap Tenor																
Maturity	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1	1.05	1.1	1.15	1.2	1.25	1.3	1.35	1.4
0.25	-	-	-	-	-	-	-	-	252.25	-	-	-	-	-	-	-	-
0.5	-	-	312.90	-	-	-	-	-	-	-	-	27.65	-	-	711.17	-	-
0.75	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	5.96	-	-	3.35	-	-	-	-	-	-	-	2.23	-	-	3.03	-
4	-	-	-	-	-	-	-	-	16.87	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	1.86	-	-	-	-	-	-	-	-	-	-	-	-	-	0.79	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	1.04	-	-	-	6.07	-	-	-	0.65	-	-	-	-
25	-	-	-	-	-	-	-	-	0.82	-	-	-	-	-	-	-	-

The Feller condition

The variance process of the stochastic volatility component of the SVJD model is governed by a Cox-Ingersoll-Ross (CIR) model. To preclude vanishing variance and to avoid issues related to the numerical discretisation of the scheme within Monte Carlo simulations, we impose the following condition on the model parameters

$$\xi^2 \leq 2\alpha\theta$$

This condition is known as the *Feller condition* and it represents a non-linear constraint for the optimisation problem. To numerically enforce this condition we append a penalty term to the vector of residuals. The penalty term has the form

[1] *In these cases, implied volatilities are considered uncertain and we ignore these points in the fitting process.*

Equation 2

$$\Theta_{\text{Feller}}(\xi^2 - 2\alpha\theta),$$

with

$$\Theta^2_{\text{Feller}}(x) = \begin{cases} 0 & , \quad x < -\dfrac{B}{2} \\ \left(\dfrac{x + \frac{B}{2}}{B}\right)^S & , \quad x \geq -\dfrac{B}{2} \end{cases}$$

where B is a buffer beyond which the penalty starts to act and S controls the strength of the penalty. In practice we set $B = 0.001$ and $S = 4$ and find that the Feller condition is not violated.

The objective function

The calibration of the model is achieved by minimising an objective function. We use the Levenberg-Marquardt optimiser[2] to find parameter values corresponding to a local minimum of the objective function. This optimiser is designed for non-linear least squares problems and represents the objective function as the squared norm of a vector of residuals. Residuals represent the differences between model and (pseudo) market price, including the 25 year target point. Residuals are multiplied by the square root of the corresponding vega scaled weights.

We fix the jump arrival rate λ to 0.1 and optimise over the remaining parameters $v_0, \theta, \alpha, \xi, \rho, \mu_J, \sigma_J$. The first five parameters relate to the stochastic variance process. The remaining parameters control the jump component in the SVJD model. If we denote by $\boldsymbol{\varphi} = (v_0, \theta, \alpha, \xi, \rho, \mu_J, \sigma_J)$ the set of parameters over which the optimisation is taking place, the objective function $\Theta^2(\boldsymbol{\varphi})$ is given by

Equation 3

$$\Theta^2(\boldsymbol{\varphi}) = \Theta^2_{\text{reg}}(\boldsymbol{\varphi}) + \Theta^2_{\text{Feller}}(\xi^2 - 2\alpha\theta)$$

where Θ^2_{Feller} is the Feller condition penalty term and

Equation 4

$$\Theta^2_{\text{reg}}(\boldsymbol{\varphi}) = \sum_{T,K} w^{\nu}_{T,K}\left(S^{\text{reg}}_{T,K} - S^{\text{model}}_{T,K}(\boldsymbol{\varphi})\right)^2$$

where $S^{\text{reg}}_{T,K}$ is the put price corresponding to the regressed implied volatility market data, $S^{\text{model}}_{T,K}(\boldsymbol{\varphi})$ is the model put price with maturity T and strike K and $w^{\nu}_{T,K}$ are the vega scaled weights. Model prices are calculated using the COS method.

The optimal value of $\boldsymbol{\varphi}$ is sensitive to the choice of seeds and bounds and to the choice of optimiser settings (such as maximum number of iterations, minimum step size, minimum norm of gradient, etc.).

Parameter bounds and seeds

All parameters are bounded to ensure reasonableness and appropriateness with the approximations made in the equity option pricing algorithm. The default parameter seeds and upper bounds used in our quarterly calibrations from EndMar2013 are given in the table to the right. These seeds and bounds are used for all equity indices.

Parameter	*Seed*	*Lower bound*	*Upper bound*
v_0	0.05	0.001	0.25
θ	0.1	0.0025	0.25
α	0.5	0.01	3.0
ξ	0.15	0.1	1.0
ρ	-0.8	-0.98	-0.55
μ_J	-0.2	-0.4	-0.05
σ_J	0.2	0.1	0.3

[2] K. Madsen, H.B. Nielsen, O. Tingleff, *Methods for Non-Linear Least Squares Problems (2nd ed.)*, *Technical University of Denmark*, 2004, pp. 24 - 29

Although we do not anticipate altering these values in the future, we may do so if market conditions were to change significantly.

Other considerations

Nominal interest rates affect the expected value of the stock price at maturity, as well as the forward strike price. In the case of deterministic interest rates, an exact cancellation takes place and the equity option prices coincide with those obtained in the absence of interest rates. In the case of stochastic interest rates, on the other hand, equity option prices are affected. The SVJD calibration does not take this effect into account, but it is monitored via the ESG validations presented in our quarterly calibration reports.

Further reading

Readers interested in the technical details of the equity option pricing technique should refer to SVJD *Model Definition* and *COS Method Pricing* [KB 2013-2532]. Further publications may be found by browsing the Stochastic Volatility & Jump Diffusion Model in the Knowledge Base.

Disclaimer