Implying Volatility Surfaces from Market Option Prices

Misha Fomytskyi, Wachovia Equity Quantitative Analysis

May 18, 2007

Abstract

Version 4

This report describes the algorithm for implying smooth volatility surfaces from market option prices used in EQAL model library.

1 Notation

- r - discount rate
- q - continuous dividend yield
- s - spread, the cost of borrowing the stock
- S_0 - spot price
- D_i - at time t_i, the stock drops by this amount (t_i is the ex-date, $D_i \approx$ cash dividend paid)
- proportional forward factor
$$f_p(t) = e^{(r(t)-q(t)-s(t))t} \tag{1}$$
- forward factor:
$$f(t) = f_p(t)\left(1 - \sum_{i:t_i \le t} \frac{D_i}{S_0 f_p(t_i)}\right) \tag{2}$$
- forward price at time t: $F(t) = S_0 f(t)$
- effective dividend yield: $q_e(t) = r(t) - \ln\left[f(t)\right]/t$

2 Introduction

The process of implying smooth parametric implied volatility surfaces consists of several steps. First, we will outline a general calculation flow and then describe every step of the calculation in detail.

One of distinct properties of the algorithm is that most of the values used in the calculation are assigned error bars. These error bars are estimated and carried through different stages of the algorithm. The primary purpose of this is to keep track of relative reliability of various inputs and intermediate results. E.g. instead of throwing away a less reliable point from a fit, we will often just assign a relatively large error bar (or small weight) to it. The advantage of this approach is that the algorithm becomes less sensitive to small perturbations in the input data.

Initially the error bars for option prices are set to the maximum of 0.01 and half of the bid-ask spread. All other quantities are assumed to have no errors.

The algorithm consists of the following steps:

1. Filter out bad input data and increase error bars for unreliable input data

2. Adjust unreasonably small error bars
3. Imply funding spread
4. Imply volatility for individual options
5. Combining call and put vols
6. Fit volatilities in the matrix form (one point per strike and maturity) to a smooth parametric surface

The output of the algorithm consists of thee objects:

- Implied funding spreads for each listed maturity with the estimate of the error bars. The funding spread enters the equations as an additional dividend yield. Addition of this term takes into account the cost of borrowing a stock and the change in the market expectations of the actual dividends.
- Implied volatility surface in the matrix form. This object contains volatilities implied for listed strikes and maturities.
- Implied volatility surface in parametric JW3 form. This parametrization represents implied volatility at a given maturity as an explicit analytical function of normalized strike

$$NS = \frac{\ln(K/F)}{\sigma_0\sqrt{T}},$$

  where K is the option strike price, F is the forward price at maturity T, σ_0 is the volatility at strike equal to the forward. The shape of the curve is determined by 3 parameters: at-the-money-forward volatility σ_0, skew, and smile. The analytical form is chosen in such a way that the curve can fit the market data well in most cases and, at the same time, the curve does not violate no-arbitrage conditions described in [1]. For more details on JW3 curve see [2].

The algorithm was implemented in May 2006 and have been running and tested since then. Presently it is running on a distributed computational grid and calculates implied volatility surfaces for hundreds of names intra-day and at the end of each business day. We would like to emphasize that the algorithm is fully automated, which provides transparent, reproducible and scalable way of generating implied volatility surfaces.

3 Filtering

3.1 Initial filter

In this filter we remove from the input data options with incorrect (rather than less accurate) information. We use the following filtering criteria:

- options with 5 or less days to maturity
- options with zero or negative Ask price
- options with irregular contract size (contract size not equal to 100 shares)

3.2 Second filter

- If option Bid price is smaller than the *minimum bid price* (currently set to 10^{-6}) and both neighboring options (with the next largest and smallest strikes) also have Bid price below the *minimum bid price*, then for both call and put options with this strike the error bar for the option price is rescaled by a factor of 10^9. Effectively, these options will not contribute to calculations of any fits or averages, but the implied vols for these options still will be calculated.

- For American options, if option Bid price is smaller than the intrinsic value of the option, the error bar for the price of this option is rescaled by a factor of 10^9. The intrinsic value of the option is defined as option price with zero volatility.

3.3 Monotonicity filter

Put prices have to increase with strike. Likewise, call prices have to decrease with strike. This condition is used in monotonicity filter. The error bar for a put/call is rescaled by a factor of 10^9 if its mid price is larger/smaller than both mid prices of the next two puts/calls with larger strikes or smaller/larger than both of the two previous puts/calls with smaller strikes.

This filter is not applied to the options with the two smallest and two largest strikes of each maturity.

3.4 Adjustment of unreasonably small error bars

It may happen that the bid-ask spread for a particular option is unusually small. Using it as an error bar would give very large weight to this option in the fit. Therefore, the error bars are floored with the average bid-ask spread of the options with 10% or less bid-ask spread.

4 Implied Spread Calculation

4.1 Calculation of the implied spread for given strike and maturity

The problem is formulated as follows. Given market prices C_0 of a call and P_0 of a put options with the same strike and maturity, calculate the spread implied by these market prices. In order to do this we solve simultaneously the following equations:

$$\left\{ \begin{array}{lcl} C(\sigma, s) & = & C_0, \\ P(\sigma, s) & = & P_0, \end{array} \right. \tag{3}$$

where $C(\sigma, s)$ and $P(\sigma, s)$ are the call and put prices calculated using one of the dividend models, σ is the implied volatility for this model. The description of the binomial vanilla pricer can be found in [3].

4.1.1 European exercise

For European options we use call-put parity to calculate the forward price:

$$F_0 = e^{rT}(C_0 - P_0) + K \tag{4}$$

Then we solve equation

$$F_0/S_0 = f(T, s') \tag{5}$$

numerically for implied spread s'.

Note that for divAsRate model, the dividends are converted into yield in the very beginning of the calculation, so the calculation of the spread is equivalent to

$$s' = r - q_e - \ln(F_0/S_0)/T \tag{6}$$

4.1.2 American exercise

For American options the algorithm consists of the following steps:

1. Find maximum $s_{\max}$ and minimum $s_{\min}$ values for the spread such that $CI(s_{\max}) < C_0$ and $PI(s_{\min}) < P_0$, where CI and PI are the intrinsic values of the call and the put. Intrinsic value for an American option is defined as the value calculated with zero volatility.

2. Starting with the interval $(s_{\min}, s_{\max})$, we search for an initial guess s_0 for the spread, which satisfies $CI(s_0) < C_0$ and $PI(s_0) < P_0$ using binomial search. If such s_0 can not be found within 10 iterations, we conclude that the inputs are not consistent and the algorithm returns an error for this particular strike and maturity.

3. For a given initial guess for the spread s_0, calculate implied volatilities σ_c and σ_p of the call and the put options by solving $C(\sigma_c, s_0) = C_0$ and $P(\sigma_p, s_0) = P_0$.

4. Using these volatilities, calculate the prices of the corresponding European call and put options.

5. Using the prices of the European options, calculate the new value s' for implied spread as described in section 4.1.1.

6. If $|s_0 - s'| < \epsilon$ and $|\sigma_c - \sigma_p| < \epsilon$, where ϵ is the precision level, or if the maximum number of iterations is reached, then return s', otherwise set $s_0 = s'$ and go to step 3.

4.2 Averaging implied spread from different strikes

After implied spreads have been calculated for each strike K_j of a given maturity, we calculate the average spread. First, we estimate the error bar for each strike:

$$\delta s_j \approx \left|\frac{\partial s}{\partial F}\right| \delta F = \frac{1}{F \cdot T}\sqrt{\delta C_j^2 + \delta P_j^2 - 2\rho\delta C_j \delta P_j}, \tag{7}$$

where δC_j and δP_j are the error bars for the call and put prices (estimated as bid-ask spread), ρ is the estimate of the correlation between the errors in call and put prices. The error in forward δF is estimated using the expression for call-put parity for European options.

Additional weights favoring at-the-money options are applied in the averaging algorithm:

$$w_j = \exp\left(-\frac{1}{2}\left(\frac{NS_j}{\text{widthOTMSpread}}\right)^2\right), \tag{8}$$

where widthOTMSpread is a parameter, $NS_j = \ln(K_j/F)/\sigma_j\sqrt{T}$ is the normalized strike.

The averaging is performed in several iterations:

1. Initialize the error bar for the average spread: $\delta s' = 0$

2. Calculate the estimate of the average:

$$s' = \sum_j s_j \frac{w_j}{\delta s_j^2 + \epsilon\delta s'^2}\left(\sum_j \frac{w_j}{\delta s_j^2 + \epsilon\delta s'^2}\right)^{-1}$$

3. Calculate the effective number of strikes (strikes with large error bars do not count)

$$n_{\text{eff}} = \frac{\sum_j \frac{w_j}{\delta s_j^2}}{\max_j\left[\frac{w_j}{\delta s_j^2}\right]} \tag{9}$$

4. Update the estimate for the error bar

$$\delta s' = \sqrt{\sum_j \left[\left(\delta s_j^2/n_{\text{eff}} + (s_j - s')^2\right)\frac{w_j}{\left(\delta s_j^2 + \epsilon\delta s'^2\right)^2}\right]\left(\sum_j \frac{w_j}{\left(\delta s_j^2 + \epsilon\delta s'^2\right)^2}\right)^{-1}} \tag{10}$$

This expression takes into account the errors of the values at different strikes ($\delta s_j^2/n_{\rm eff}$ term) and the dispersion of the these values with respect to their mean ($(s_j - s')^2$ term). If the dispersion is consistent[1] with the error bars of the spreads at individual strikes, the error bar for the mean spread is inversely proportional to the square root of the effective number of strikes (strikes with not very large error bars). Otherwise, the error bars of the spreads at individual strikes could not be relied upon, so the error of the mean is estimated as the square root of the second central moment of the sample. Note that in this case the estimate of the error will not go to zero as the number of strikes increases. This is because sample values δs_j are not independent.

5. If less than 2 iterations, go to step 2, otherwise return average spread s' with error bar $\delta s'$.

4.3 Recovering term structure of spreads

For each maturity T_i we have solved the problem of calculating the constant implied spread s_i for fixed interest rate $r_i = r(T_i)$, and continuous dividend yield $q_i = q(T_i)$.

$$f_p^i(t) = \exp\left[(r_i - q_i - s_i)t\right], \tag{11}$$

$$f_i = f_p^i(T_i)\left(1 - \sum_{i':t_i' \le T_i} \frac{D_i'}{S_0 f_p^i(t_{i'})}\right), \tag{12}$$

where $f_i = F(T_i)/S_0$ is the forward factor implied from the market.

Note that calculated s_i do *not* make a term structure of funding spreads.[2] Consider example with two maturities t_1 and t_2. When we calculated s_2, we assumed that interest rates were equal to r_2, dividend yield was equal to q_2 and spread was constant for all times. At this step of the algorithm we would like to calculate value $s(t_2)$ satisfying the following assumptions:

- for times before t_1 the interest rate is r_1, the dividend yield is q_1 and the spread is s_1,
- for times from t_1 to t_2 the interest rate is $r_{12} = (r_2 t_2 - r_1 t_1)/(t_2 - t_1)$, the dividend yield is $q_{12} = (q_2 t_2 - q_1 t_1)/(t_2 - t_1)$ and the spread is $s_{12} = (s(t_2)t_2 - s_1 t_1)/(t_2 - t_1)$,
- the forward factor at t_2 is f_2.

Obviously, the calculated value $s(t_2)$ not necessarily will be equal to s_2. In practice the difference is quite small. Similarly, for calculation of $s(t_3)$ we will use term structure of interest rates, dividend yields and the calculated $s(t_1)$ and $s(t_2)$.

Mathematically the calculation of the term structure of spreads $s(t_i)$, which is consistent with forward factors f_i and the term structures of rates and dividend yields is equivalent to solving the equations:

$$f_i = f(T_i) \tag{13}$$

for $s(T_i)$, where $f(T_i)$ is given by eqs. (1)-(2). We start with $i = 0$ and use bootstrap method to calculate the spread at the next maturity using previously calculated spreads at earlier maturities.

5 Imply Volatility for Individual Options

This steps requires solution of a nonlinear equation with respect to unknown volatility. The implementation of the algorithm contains a number of optimization techniques. For example, in European case implying vols is reduced to a two-dimensional interpolation. It is easy to show that undiscounted value of an European option over forward $v = \hat{V}/F$ is a function of $x = \ln(K/F)$ and $\Sigma = \sigma\sqrt{T}$. Likewise Σ is an implicit function

[1] In other words, if the dispersion term is smaller.

[2] Fixed income analogy of this statement would be: yields to maturity for bond of different maturities do note make a yield curve. One has to apply bootstrap algorithm to calculate a zero curve.

of v and x. Using precomputed table of values for Σ for a wide range of x and v we can very accurately imply volatilities with a few simple arithmetic operations.

The error bar for the calculated volatility is estimated as follows:

$$\Delta\sigma = \frac{\sqrt{(\Delta P/\text{Vega})^2 + (2\epsilon)^2}}{(\text{Vega}/\text{Vega}_{\text{max}})^{\text{vegaPower}-1}}, \tag{14}$$

where ϵ is the required accuracy in the rootfind algorithm for implied vol calculation, ΔP is option price error bar, Vega_{max} is the maximum Vega for a given maturity, vegaPower = 1.5 is an algorithmic parameter.

6 Combine Call and Put Vols

First we rescale the error bars of calls and puts in such a way that more weight is given to out-of-the-money options. The addition weight is equal to

$$w = \frac{1}{2}e^{-\frac{1}{2}(z/\text{widthOTM})^2} \quad \text{if } z < 0 \text{ for call or } z \geq 0 \text{ for put}, \tag{15}$$

$$w = 1 - \frac{1}{2}e^{-\frac{1}{2}(z/\text{widthOTM})^2} \quad \text{if } z \geq 0 \text{ for call or } z < 0 \text{ for put}, \tag{16}$$

where $z = \ln(K/S^*)/\sigma\sqrt{T}$ is the "normalized strike" for a given option, $S^* = S^{\text{spotPower}}F^{1-\text{spotPower}}$ is the reference spot, spotPower = 1, widthOTM = 0.3.

The error bar for volatility then is rescaled by a factor $1/\max(10^{-12}, \sqrt{w})$ for each option. The error bars rescaled to infinity (9×10^{99}) if the relative strike K/S^* is above/below the KSmax/KSmin. Currently KSmin=40%, KSmax=400%.

If both call and put vols are available for a given strike, their values are combined using the estimate (rescaled) error bars into one volatility for this particular strike. Otherwise only call or only put volatility is used with the corresponding error bar. The algorithm of estimating the combined vol and its error bar is similar to the one described in section 4.2.

7 Fitting

Once volatilities for each strike and maturity are calculated and the error bars estimated we perform a fit of the matrix data to a parametric JW3 (see [2]) curve.

7.1 First round of fitting

Before fitting, we select the best slice in the volatility matrix. The quality of a slice is assessed based on the number of strikes in a slice, the range of the strikes and the error bars.

The first round of fit involves fitting individual slices separately. First we fit the best slice. Then slices with smaller and larger times are fitted. Each time the results from the fit of the adjacent slice are used as initial guess for the next calculation.[3]

The surface is checked for calendar arbitrage. To do this we calculate forward variance

$$Var_{\text{fwd}} = \sigma(KF_j, T_i)^2 T_i - \sigma(KF_j, T_{i-1})^2 T_{i-1}, \tag{17}$$

where T_i is the maturity of slice i, $\sigma(K/F, T)$ is implied volatility as a function of strike-over-forward and maturity. Then we check this variance is positive for a wide range of relative strikes KF_j and all forward time periods. If all forward variances are positive, the algorithm returns the results of the first round of fit, otherwise we perform another round.

[3] nFitRounds=1, function fitFromVolData.

7.2 Second round of fitting

We sample the volatility matrix for a wide range of relative strikes KF_j, then adjust the volatilities in order to make forward var along a given value of strike over forward positive. We add the resulted volatilities to the input volatility matrix and refit it. If this solves the calendar arbitrage problem, we return the result of this fit. Otherwise we proceed with global optimization.

7.3 Third round of fitting

Finally, if simpler methods could not produce arbitrage-free fit, we solve global optimization problem. All slices are fitted at the same time. The objective function has a global penalty term, which becomes very large if the forward volatility (defined as $\sigma_{\text{fwd}} = \sqrt{Var_{\text{fwd}}/(T_i - T_{i-1})}$) becomes larger than $2 \times \sigma(KF_j, T_i)$ or smaller than $0.3 \times \sigma(KF_j, T_i)$. The penalty term is added for each forward time period in the volatility surface for a wide range of KF_j.

References

[1] J. Gatheral, “Case Studies in Financial Modeling Course Notes”, *Courant Institute of Mathematical Sciences*, (2005).

[2] T. Klassen, “Volatility Curve Parametrization”, *Equity Derivatives Analytics*, (2006).

[3] M. Fomytskyi, “Pricing American Vanilla Options”, *Equity Derivatives Analytics*, (2007).