On building FX vol surface - Vanna Volga method

May 29, 2017

1 Smile pricing

1.1 The vol smile

The FX options market is one of the largest and most liquid OTC derivatives market in the world. In the OTC market, FX options are quoted in terms of volatilities.

The classic Black-Scholes assumption that all options trade at *single flat volatility* is wrong. There is a skew or assymetry in the volatilities. For example, if the underlying were trading at 100, the 90 put and 110 calls trade at different volatilies in the market.

1.2 Reasons for the vol smile

Before the stock market crash of 1987, pretty much all moneyness levels traded at the same volatility. The assymetry first appeared in the options markets after the crash of 19th October, 1987, Black Monday, where the Dow Jones Industrial Average fell by 22% and the world's stock markets crashed. Volatilities are entirely supply and demand driven. For the first time since 1929, it was discovered that a giant market could drop by 20% or more in a day. After the crash, traders found that downside protection was priced way too cheaply. Sell-side trading desks who wrote out-of-the-money puts and floor traders who made markets in such options suffered steep losses and found out that they were not being compensated enough for writing options such deep out of the money. The market underestimated the probability of such extreme events happening. The return distribution has a negative skew. Clearly, low-strike puts should be worth more than high-strike calls, when you think about the higher probabilities of a downside move.

Also, the return volatility far away from current price levels tends to be higher than the current return volatility. For example, if the market makes a large downside move, the underlying is much more volatile.

Indexes tend to jump down more than up. So, equity markets show a volatility skew. In the FX markets, a large gain of the fixed currency, means a giant loss in the variable currency. So, a smile is present in the FX markets. The smiles are more symmetric for equally *powerful* currencies, and less so for *unequal* ones. In the Gold markets, since gold is a safe haven, that appreciates when stock prices fall, a positive volatility skew is present.

The conclusion is, options markets have become more experienced about the kind of shocks that can occur, have have started to display more sophisticated volatility smiles. Professional FX traders construct a *vol surface*, where implied volatilies are plotted as a function of moneyness and maturities.

1.3 Bisection method to find the implied vol

The analytical Black-Scholes formula for European calls and puts is:

$$Call(\sigma) = DF_d(t)[F\Phi(d_+) - K\Phi(d_-)]$$

$$Put(\sigma) = DF_d(t)[F\Phi(-d_+) - K\Phi(-d_-)]$$

where

$$d_\pm = \frac{\ln F/K \pm \frac{\sigma^2}{2}}{\sigma\sqrt{t}}$$

The implied volatility $\sigma^{implied}$ is such a value of the volatility parameter σ that the Black Scholes price *matches* the observed market price:

$$C_{BS}(\sigma^{implied}) = C_{market}$$

Define $f(\sigma) = C_{BS}(\sigma) - C_{market}$. Any root finding algorithm e.g. the *bisection method* can be used to find the roots of the equation $f(\sigma) = 0$ is as follows:

1. Choose a subinterval $[\sigma_L, \sigma_H]$ for example, $[0.01, 1.00]$, and calculate the two option prices $C_{BS}(\sigma_L), C_{BS}(\sigma_H)$, such that $C_{BS}(\sigma_L) < C_{market}$ and $C_{BS}(\sigma_H) > C_{market}$.

2. Within this sub-interval, we know that the function $f(\sigma)$ is monotonically increasing, and at the end points $f(\sigma_L) < 0$ and $f(\sigma_H) > 0$. Since the function $f(\sigma)$ is continuous on the interval $[\sigma_L, \sigma_H]$, its graph will cut the x-axis in some one point between σ_L and σ_H.

3. Draw a chord AB connecting the end points of the curve $y = f(\sigma)$, which corresponds to the abscissas σ_L and σ_H. Then the abscissa σ_1 of the point of intersection of this chord with the x-axis will be the approximate root. In order to find this approximate value, let us write the equation of the straight line AB that passes through the given points $A[\sigma_L, f(\sigma_L)]$ and $B[\sigma_H, f(\sigma_H)]$:

$$\frac{y - f(\sigma_L)}{x - \sigma_L} = \frac{f(\sigma_H) - f(\sigma_L)}{\sigma_H - \sigma L}$$

Since, $y = 0$ at $x = \sigma_1$, we get:

$$\frac{-f(\sigma_L)}{\sigma_1 - \sigma_L} = \frac{f(\sigma_H) - f(\sigma_L)}{\sigma_H - \sigma L}$$

so,

$$\begin{aligned}\sigma_1 &= \sigma L - \frac{(\sigma_H - \sigma_L) f(\sigma_L)}{f(\sigma_H) - f(\sigma_L)} \\ &= \sigma_L - (C_{BS}(\sigma_L) - C_{market}) \frac{(\sigma_H - \sigma_L)}{C_{BS}(\sigma_H) - C_{BS}(\sigma_L)}\end{aligned}$$

4. To obtain a more exact value of the root, we determine $f(\sigma_1)$. If $f(\sigma_1) < 0$, then repeat the same procedure applying the bisection formula to the interval $[\sigma_1, \sigma_H]$. If $f(\sigma_1) > 0$, then apply this formula to the interval $[\sigma_L, \sigma_1]$. By repeating this procedure several times more, we obvious obtain more and more previse values of the root σ_2, σ_3 etc. This is continued until the desired accuracy is achieved:

$$C_{market} - C_{BS}(\sigma_i) < \epsilon$$

1.4 Implementation of the bisection method

```
In [6]: import numpy as np
        import math
        import scipy.stats as stats

        def BS(S0,K,r,q,T,sigma,optType):
            dplus = (math.log(S0/K)+((r - q) + (sigma ** 2)/2)*T) \
            /(sigma * math.sqrt(T))

            dminus = (math.log(S0/K)+((r - q) - (sigma ** 2)/2)*T) \
            /(sigma * math.sqrt(T))

            discD = math.exp(-r*T)
            discF = math.exp(-q*T)

            if optType == 'C':
                price = (S0 * discF * stats.norm(0,1).cdf(dplus)) \
                - (K * discD * stats.norm(0,1).cdf(dminus))
            else:
                price = -((S0 * discF * stats.norm(0,1).cdf(-dplus)) \
                          -(K * discD * stats.norm(0,1).cdf(-dminus)))

            return price

        def findImpliedVol(S0,K,r,q,T,optType,Call_market):

            # Parameters
            sigma_Lo = 0.01
            sigma_Hi = 1.00
            epsilon = 0.001

            Call_BS_Lo = BS(S0,K,r,q,T,sigma_Lo,optType)
            Call_BS_Hi = BS(S0,K,r,q,T,sigma_Hi,optType)
            Call_BS_i = Call_BS_Lo

            # Iteratively find a root using bisection method
            while abs(Call_BS_i - Call_market) > epsilon:
                sigma_i = sigma_Lo \
```

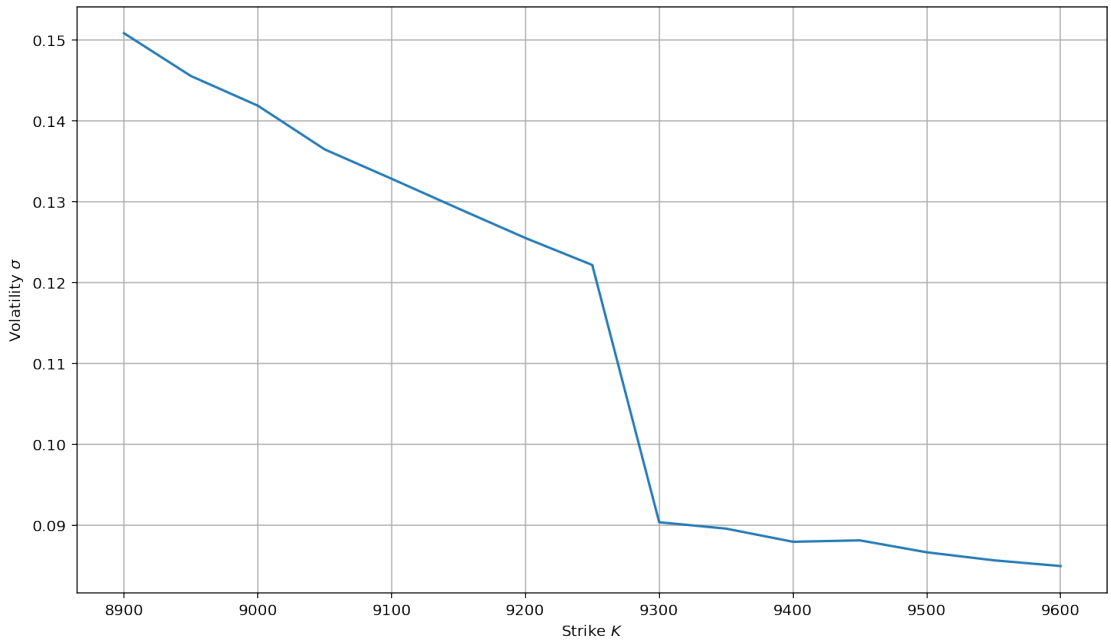
```
            -((Call_BS_Lo - Call_market)\
              *((sigma_Hi - sigma_Lo)/(Call_BS_Hi - Call_BS_Lo)))

            Call_BS_i = BS(S0,K,r,q,T,sigma_i,optType)

            if Call_BS_i > Call_market:
                Call_BS_Hi = Call_BS_i
                sigma_Hi = sigma_i
            else:
                Call_BS_Lo = Call_BS_i
                sigma_Lo = sigma_i

        return sigma_i
```

Equity indices have a volatility skew - puts have higher higher volatility then calls. The following code snippet plots the vol skew of May NIFTY options. NIFTY spot was at 9285.30 on 05th May, 2017.

```
In [7]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd

        d = {'call': pd.Series([420.875,379.45,328.325,277.775,240.475,\
                                199.3,162.425,128.325,97.575,71.1,48.975, \
                                33.2, 20.625, 12.2, 6.85],
                               index=[8900,8950,9000,9050,9100,9150,9200, \
                                      9250,9300,9350,9400,9450,9500,9550,9600]),
             'put':pd.Series([12.95,16,20.6,25.525,32.825,42,53.525,68.05,\
                              86.925,107.625,134.95,167.1,203.95,250.65,289.75],
                             index=[8900,8950,9000,9050,9100,9150,9200,\
                                    9250,9300,9350,9400,9450,9500,9550,9600])}

        optionPrices = pd.DataFrame(d)
        optionPrices

In [8]: iv = []

        # Find the IV of OTM puts
        for strike in range(8900,9300,50):
            vol = findImpliedVol(9285.30,strike,
                                 0.10,0,0.05479,'P',
                                 optionPrices['put'][strike])
            iv.append(vol)

        # Find the IV of OTM calls
        for strike in range(9300,9650,50):
            vol = findImpliedVol(9285.30,strike,
                                 0.10,0,0.05479,'C',
```

```
                                optionPrices['call'][strike])
        iv.append(vol)

    # Plot the volatility skew as function of strikes
    strikes = list(range(8900,9650,50))
    plt.grid(True)
    plt.xlabel('Strike $K$')
    plt.ylabel('Volatility $\sigma$')
    plt.plot(strikes,iv)
    plt.show()
```

Out[8]:



1.5 The Vanna Volga Method

As stochastic volatility models are computationally demanding, FX brokers and traders use a fast method - the *Vanna-Volga* (VV) also known as the *trader's rule of thumb* to construct the whole smile for a given maturity.

In a nutshell, the vanna-volga(VV) method is based on adding an analytically derived correction to the Black-Scholes price of the instrument. The method constructs a hedging portfolio that zeroes out the Black-Scholes greeks that measure the option's sensitivity with respect to the volatility i.e. the vega, vanna and volga of the option. We find the *market price* of the volatility risk. In this way, VV method produces smile-consistent values.

1.5.1 Strikes corresponding to standard ATM, 25Δ call and 25Δ put volatilities

We start with 3 moneyness levels : ATM, 25Δ call and 25Δ put(75Δ call) volatilities, and the objective is to interpolate this curve.

The strike corresponding to ATM volatility can be obtained as:

$$\begin{aligned} D_f\Phi(d_+) + D_f(\Phi(d_+) - 1) &= 0 \\ \Phi(d_+) &= 0.50 \\ d_+ &= 0 \\ \frac{\log\left(\frac{F}{K_{atm}}\right) + \sigma_{atm}^2 t}{\sigma_{atm}\sqrt{t}} &= 0 \\ K_{atm} &= F\exp\frac{1}{2}\sigma_{atm}^2 t \end{aligned}$$

The strikes corresponding to 25Δ call and 25Δ put volatilities are obtained as:

$$\begin{aligned} \Delta_C = D_f\Phi(d_+) &= 0.25 \\ d_+ &= -\alpha \\ \frac{\log\left(\frac{F}{K_{25\Delta C}}\right) + \frac{\sigma_{25\Delta C}^2}{2} t}{\sigma_{25\Delta C}\sqrt{t}} &= -\alpha \\ -\log\left(\frac{K_{25\Delta C}}{F}\right) + \frac{\sigma_{25\Delta C}^2}{2} t &= -\alpha\sigma_{25\Delta C}\sqrt{t} \\ \log\left(\frac{K_{25\Delta C}}{F}\right) &= \alpha\sigma_{25\Delta C}\sqrt{t} + \frac{\sigma_{25\Delta C}^2}{2} t \\ K_{25\Delta C} &= F\exp\left(\alpha\sigma_{25\Delta C}\sqrt{t} + \frac{\sigma_{25\Delta C}^2}{2} t\right) \end{aligned}$$

and

$$\begin{aligned} \Delta_P = -D_f\Phi(-d_+) &= -0.25 \\ d_+ &= \alpha \\ K_{25\Delta P} &= F\exp\left(-\alpha\sigma_{25\Delta P}\sqrt{t} + \frac{\sigma_{25\Delta P}^2}{2} t\right) \end{aligned}$$

1.5.2 Vega, Volga and Vanna of calls and puts

Call and put vega Vega is the first derivative of the option price with respect to the volatility σ. The derivatives of d_+ and d_- with respect to σ are:

$$\begin{aligned} \frac{\partial d_+}{\partial\sigma} &= \frac{\partial}{\partial\sigma}\left(\frac{1}{\sigma\sqrt{t}}(\log(F/K) + \frac{\sigma}{2}\sqrt{t})\right) \\ &= \frac{-1}{\sigma^2\sqrt{t}}\log(F/K) + \frac{\sqrt{t}}{2} \\ &= -\frac{1}{\sigma^2\sqrt{t}}\log(F/K) - \frac{\sigma^2 t}{\sigma^2\sqrt{t}} + \frac{\sqrt{t}}{2} + \frac{\sqrt{t}}{2} \\ &= -\frac{d_+}{\sigma} + \sqrt{t} \\ &= -\frac{d_-}{\sigma} \end{aligned}$$

$$\begin{aligned}
\frac{\partial d_-}{\partial \sigma} &= \frac{\partial}{\partial \sigma}(d_+ - \sigma\sqrt{t}) \\
&= \frac{\partial d_+}{\partial \sigma} - \sqrt{t} \\
&= -\frac{d_+}{\sigma} + \sqrt{t} - \sqrt{t} \\
&= -\frac{d_+}{\sigma}
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
\frac{\partial C}{\partial \sigma} &= \frac{\partial}{\partial \sigma}\left(SD_f\Phi(d_+) - KD_d\Phi(d_-)\right) \\
&= -SD_f\phi(d_+)\frac{d_-}{\sigma} + KD_d\phi(d_-)\frac{d_+}{\sigma} \\
&= -SD_f\phi(d_+)\frac{d_-}{\sigma} + KD_d \cdot \frac{F}{K}\phi(d_+)\frac{d_+}{\sigma} \\
&= -SD_f\phi(d_+)\frac{d_-}{\sigma} + SD_f\phi(d_+)\frac{d_+}{\sigma} \\
&= SD_f\phi(d_+)\frac{d_+ - d_-}{\sigma} \\
\nu_C &= SD_f\phi(d_+)\sqrt{t} \\
\frac{\partial P}{\partial \sigma} &= -\frac{\partial}{\partial \sigma}\left(SD_f\Phi(-d_+) - KD_d\Phi(-d_-)\right) \\
&= -\left(-SD_f\phi(d_+)\frac{\partial d_+}{\partial \sigma} + KD_d\phi(d_-)\frac{\partial d_-}{\partial \sigma}\right) \\
\nu_P &= \nu_C
\end{aligned}$$

Vanna of call and puts Vanna is the cross-derivative of the option price with respect to underling price S and the volatility σ. The vanna can be derived as,

$$\begin{aligned}
\text{Vanna}_C = \frac{\partial^2 C}{\partial S \partial \sigma} &= \frac{\partial \Delta_C}{\partial \sigma} \\
&= \frac{\partial}{\partial \sigma}(D_f\Phi(d_+)) \\
&= D_f\phi(d_+)\frac{\partial d_+}{\sigma} \\
&= -D_f\phi(d_+)\frac{d_-}{\sigma} \\
&= -\frac{\nu_C}{S\sqrt{t}}\frac{d_-}{\sigma} \\
&= -\frac{d_-}{S\sigma\sqrt{t}}\nu_C
\end{aligned}$$

$$\begin{aligned}
\text{Vanna}_P &= \frac{\partial^2 P}{\partial S \partial \sigma} = \frac{\partial \Delta_P}{\partial \sigma} \\
&= \frac{\partial}{\partial \sigma}(-D_f \Phi(-d_+)) \\
&= D_f \phi(d_+) \frac{\partial d_+}{\sigma} \\
&= -\frac{d_-}{S\sigma\sqrt{t}} \nu_P \\
\text{Vanna}_P &= \text{Vanna}_C
\end{aligned}$$

Volga of calls and puts Volga is the second derivative of the option price with respect to the volatility σ. Let's first derive the following equation:

$$\begin{aligned}
\frac{\partial \phi(d_+)}{\partial d_+} &= \frac{\partial \left(\frac{1}{\sqrt{2\pi}} \exp(-\frac{d_+^2}{2}) \right)}{\partial d_+} \\
&= -\frac{1}{\sqrt{2\pi}} d_+ \exp\left(-\frac{d_+^2}{2} \right) \\
&= -d_+ \phi(d_+)
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
\text{Volga}_C &= \frac{\partial^2 C}{\partial \sigma^2} = \frac{\partial \nu_C}{\partial \sigma} \\
&= D_f S \sqrt{t} \frac{\partial \phi(d_+)}{\partial \sigma} \\
&= -S D_f \sqrt{t} d_+ \phi(d_+) \frac{\partial d_+}{\partial \sigma} \\
&= S D_f \sqrt{t} \phi(d_+) \frac{d_- d_+}{\sigma} \\
&= \frac{\nu_C d_+ d_-}{\sigma}
\end{aligned}$$

and

$$\begin{aligned}
\text{Volga}_P &= \frac{\partial^2 P}{\partial \sigma^2} = \frac{\partial \nu_P}{\partial \sigma} \\
\text{Volga}_P &= \text{Volga}_C
\end{aligned}$$

1.5.3 Ito's lemma standard results

Univariate case Consider a continuous and differentiable function f of two variables, time t and space x. The Taylor's series expansion for the differential of the function, df is expressed as:

$$\begin{aligned}
df = &\frac{\partial f}{\partial x}(t,x)dx + \frac{\partial f}{\partial t}(t,x)dt \\
&+ \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t,x)(dx)^2 + \frac{\partial^2 f}{\partial x \partial t}(t,x)(dxdt) + \frac{1}{2}\frac{\partial^2 f}{\partial t^2}(t,x)(dt)^2
\end{aligned}$$

Consider an Ito process of the form:

$dX_t = v_t dt + u_t dB_t$

Intuitively, dX_t can be thought of as a $Bernoulli(1/2)$ r.v. with values $\{u_t\sqrt{dt}, -u_t\sqrt{dt}\}$. The random variable $dX_t^2 = u_t^2 dt$ is of the same order as dt and cannot be neglected. Formally, consider the stochastic increment $(dX_t)^2$.

$$E(dX_t^2) = E[v_t^2(dt)^2 + 2u_t v_t(dtdB_t) + u_t^2 dB_t^2]$$
$$= u_t^2 E(dB_t^2)$$
$$= u_t^2 dt$$

Thus, $dX_t^2 = u_t^2 dt$ where the equality holds in the mean square sense. Therefore in the Taylor's series expansion, if x were a stochastic variable, $dx^2 = u_t^2 dt$ cannot be ignored, all the other terms $dt^2, dxdt$ can be neglected as compared to dt. This is because in the limit as $dt \to 0$, the terms $(dt)^2$ and $dx \cdot dt = (v_t dt + u_t dB_t)dt$ approach zero faster than dt.

In the single-variable case, the Ito's lemma is therefore

$$df(t, X_t) = \frac{\partial f}{\partial t}(t, X_t)dt + \frac{\partial f}{\partial x}(t, X_t)dX_t + u_t^2 \frac{\partial^2 f}{\partial x^2}(t, X_t)dt$$

Generalization to the multivariate case Suppose that a function depends on n stochastic variables $X_1, X_2, \ldots, X_n$. Suppose further that X_i follows an Ito process with mean a_i and instantaneous variance b_i^2, that is:

$dX_i = a_i dt + b_i dB_t^i$

where B_t^i is a brownian motion. A Taylor's series expansion of f yields,

$$df = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} dx_i + \frac{\partial f}{\partial t} dt$$
$$+ \frac{1}{2} \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i x_j}(dx_i dx_j) + \frac{1}{2} \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i \partial t} dx_i dt + \frac{1}{2} \frac{\partial^2 f}{\partial t^2}(dt)^2$$

The terms $(dx_i dt)$ and $(dt)^2$ can be neglected compared to dt. What does $(dx_i dx_j)$ converge to in the mean square limit?

In the single-variable case, we know that $E(dX_t^2) = b^2 dt$. Let's find the $E(dX_i dX_j)$.

$$E(dX_t^i dX_t^j) = E[(a_i dt + b_i dB_t^i)(a_j dt + b_j dB_t^j)]$$
$$= E[a_i a_j dt^2 + a_i b_j dt dB_t^j + b_i a_j dt dB_t^i + b_i b_j dB_t^i dB_t^j]$$
$$= b_i b_j E(dB_t^i dB_t^j)$$

Passing to the limit as $dt \to 0$, all of the terms approach zero relatively faster than dt, except $dB_t^i dB_t^j$. Suppose, the correlation of $dB_t^i dB_t^j$ is ρ_{ij}. Intuitively, dB_t^i and dB_t^j are non-independent $Bernoulli(1/2)$ random variables that can take values $\{\sqrt{dt}, -\sqrt{dt}\}$ and $dB_t^j = \rho_{ij} dB_t^i$.

$$E(dB_i dB_j) = (1/2)(\sqrt{dt})(\rho_{ij}\sqrt{dt}) + (1/2)(-\sqrt{dt})(-\rho_{ij}\sqrt{dt})$$
$$= \rho_{ij} dt$$
$$E(dX_t^i dX_t^j) = b_i b_j \rho_{ij} dt$$

and therefore $dx_i dx_j = b_i b_j \rho_{ij} dt$ where equality holds in the mean square sense. Thus, the generalized version of the Ito's Lemma is:

$$df = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(t, X_1, \ldots, X_n)dX_i + \frac{\partial f}{\partial t}(t, X_1, \ldots, X_n)dt$$
$$+ \frac{1}{2} \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}(t, X_1, \ldots, X_n) b_i b_j \rho_{ij}$$

1.5.4 Replicating Portfolio

Our aim is to price an arbitrary option C with strike K, that is vega-neutral in the Black-Scholes(flat-smile) world. We assume that the option price can be described by the Black-Scholes

PDE, with a *flat but stochastic implied volatility*. Our risk-neutral replicating portfolio Π^{BS} consists of a long position in the option K, a short position in Δ units of the underlying S_t and short x_i units of three pivot European vanilla calls C_i (or puts) maturing at T, that are quoted in the market 25Δ Put(75Δ call), ATM call (or ATM put), 25Δ call(75Δ put). The corresponding strikes are further denoted by $K_i, i = 1, 2, 3$, $K_1 < K_2 < K_3$ (which is equivalent to writing $K_1 = K_{25\Delta P}$, $K_2 = K_{ATM}$, $K_3 = K_{25\Delta C}$) and the market implied volatilities associated with K_i are denoted by σ_i.

The change in the $d\Pi^{BS}$ of the value of the portfolio Π^{BS} in a small time-interval dt and in the Black-Scholes world is given by the equation:

$d\Pi^{BS} = dC^{BS} - \Delta dX_t - \sum_{i=1}^{3} x_i dC_i^{BS}$

Applying Ito's lemma, we have:

$$\begin{aligned} d\Pi^{BS} &= \frac{\partial C}{\partial t}dt + \frac{\partial C}{\partial x}dx + \frac{\partial C}{\partial \sigma}d\sigma \\ &+ \frac{1}{2}\left[\frac{\partial^2 C}{\partial x^2}(dx)^2 + \frac{\partial^2 C}{\partial x \partial \sigma}(dx d\sigma) + \frac{\partial^2 C}{\partial \sigma^2}(d\sigma)^2\right] \\ &- \Delta dx - \sum_{i=1}^{3} x_i \left[\frac{\partial C_i}{\partial t}dt + \frac{\partial C_i}{\partial x}dx + \frac{\partial C_i}{\partial \sigma}d\sigma\right] \\ &- \frac{1}{2}\sum_{i=1}^{3} x_i \left[\frac{\partial^2 C_i}{\partial x^2}(dx)^2 + \frac{\partial^2 C_i}{\partial x \partial \sigma}(dx d\sigma) + \frac{\partial^2 C_i}{\partial \sigma^2}(d\sigma)^2\right] \end{aligned}$$

Gathering all the terms, we have:

$$\begin{aligned} d\Pi^{BS} &= \left(\frac{\partial C}{\partial t} - \sum_{i=1}^{3} x_i \frac{\partial C_i}{\partial t}\right)dt + \left(\frac{\partial C}{\partial x} - \Delta - \sum_{i=1}^{3} x_i \frac{\partial C_i}{\partial x}\right)dx + \left(\frac{\partial C}{\partial \sigma} - \sum_{i=1}^{3} x_i \frac{\partial C_i}{\partial \sigma}\right)d\sigma \\ &+ \frac{1}{2}\left[\left(\frac{\partial^2 C}{\partial x^2} - \sum_{i=1}^{3} x_i \frac{\partial^2 C_i}{\partial x^2}\right)(dx)^2 + \left(\frac{\partial^2 C}{\partial x \partial \sigma} - \sum_{i=1}^{3} x_i \frac{\partial^2 C_i}{\partial x \partial \sigma}\right)(dx d\sigma) + \left(\frac{\partial^2 C}{\partial \sigma^2} - \sum_{i=1}^{3} x_i \frac{\partial^2 C_i}{\partial \sigma^2}\right)(d\sigma)^2\right] \end{aligned}$$

We choose the Δ and the weights x_i, so as to zero out the coefficients of dx, $d\sigma$, $dxd\sigma$ and $d\sigma^2$(given that other higher order risks like gamma can be ignored). The replicating portfolio will be locally risk-free at time t and must earn a risk-free rate of return $d\Pi = r\Pi dt$. Therefore, when the flat volatility is *stochastic* and the options are valued with the Black-Scholes formula, we can still have a *locally* perfect hedge.

1.5.5 Determining the weights x_1, x_2, x_3

The weights can be determined by making the replicating porfolio vega-, volga- and vanna-neutral i.e. fully hedged with respect to stochastic flat volatility risk. Given a flat volatility $\sigma = \sigma_2 = \sigma_{atm}$, we must therefore have:

Vega : $\frac{\partial C}{\partial \sigma} = \sum_{i=1}^{3} x_i \frac{\partial C_i}{\partial \sigma}$

Volga : $\frac{\partial^2 C}{\partial \sigma^2} = \sum_{i=1}^{3} x_i \frac{\partial^2 C_i}{\partial \sigma^2}$

Vanna : $\frac{\partial^2 C}{\partial x \partial \sigma} = \sum_{i=1}^{3} x_i \frac{\partial^2 C_i}{\partial x \partial \sigma}$

We know from the previous derivation that:

$\text{Volga}_C = \frac{d_+ d_-}{\sigma}\nu_C$

$\text{Vanna}_C = -\frac{d_-}{S\sigma\sqrt{t}}\nu_C$

The above three equations can therefore be written as,

$\nu_C = x_1\nu_1 + x_2\nu_2 + x_3\nu_3$

$\frac{d_C^+ d_C^-}{\sigma}\nu_C = x_1\frac{d_1^+ d_1^-}{\sigma}\nu_1 + x_2\frac{d_2^+ d_2^-}{\sigma}\nu_2 + x_3\frac{d_3^+ d_3^-}{\sigma}\nu_3$

$\frac{d_C^-}{S\sigma\sqrt{t}}\nu_C = x_1\frac{d_1^-}{S\sigma\sqrt{t}}\nu_1 + x_2\frac{d_2^-}{S\sigma\sqrt{t}}\nu_2 + x_3\frac{d_3^-}{S\sigma\sqrt{t}}\nu_3$

In matrix form, the above system of algebraic equations can be represented as:

$$\begin{pmatrix} \nu_1 & \nu_2 & \nu_3 \\ d_1^+ d_1^- \nu_1 & d_2^+ d_2^- \nu_2 & d_3^+ d_3^- \nu_3 \\ d_1^- \nu_1 & d_2^- \nu_2 & d_3^- \nu_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \nu_C \\ d_C^+ d_C^- \nu_C \\ d_C^- \nu_C \end{pmatrix}$$

The solution to this system of linear equations for the option C is,

$$x_1 = \frac{\nu_C}{\nu_1} \cdot \frac{\log \frac{K_2}{K} \log \frac{K_3}{K}}{\log \frac{K_2}{K_1} \log \frac{K_3}{K_1}}, x_2 = \frac{\nu_C}{\nu_2} \cdot \frac{\log \frac{K}{K_1} \log \frac{K_3}{K}}{\log \frac{K_2}{K_1} \log \frac{K_3}{K_2}}, x_3 = \frac{\nu_C}{\nu_3} \cdot \frac{\log \frac{K}{K_1} \log \frac{K}{K_2}}{\log \frac{K_3}{K_1} \log \frac{K_3}{K_2}}$$

A smile consistent price C^{mkt} for the call option C with strike K is obtained by adding an analytical correction to the Black-Scholes price.

$C^{mkt} = C^{BS}(\sigma_{atm}) + \sum_{i=1}^{3} x_i (C_i^{mkt}(\sigma_i) - C_i^{BS}(\sigma_{atm}))$

The new option price is thus defined by adding to the flat smile Black-Scholes price, the vanna-volga correction, adjustment .

1.5.6 A first-order approximation for the implied volatility of the option C

The Taylor's series approximation polynomial for the term $C^{mkt}(\sigma)$ is:

$$C^{mkt}(\sigma) = C^{BS}(\sigma_{atm}) + (\sigma - \sigma_{atm}) \frac{\partial C}{\partial \sigma}$$

$C^{mkt}(\sigma) - C^{BS}(\sigma_{atm}) = \nu(\sigma - \sigma_{atm})$

And thus,

$\sum_{i=1}^{3} x_i (C_i^{mkt}(\sigma_i) - C^{BS}(\sigma_{atm})) = \sum_{i=1}^{3} x_i \nu_i (\sigma_i - \sigma_{atm})$

Let $\nu_C = \sum_{i=1}^{3} x_i \nu_i$, we have:

$$\begin{aligned} C^{mkt} &= C^{BS}(\sigma_{atm}) + \sum_{i=1}^{3} x_i \nu_i (\sigma_i - \sigma_{atm}) \\ &= C^{BS}(\sigma_{atm}) + \sum_{i=1}^{3} (x_i) \nu_i \sigma_i - \sum_{i=1}^{3} x_i \nu_i \sigma_{atm} \\ &= C^{BS}(\sigma_{atm}) + \sum_{i=1}^{3} (\frac{\nu_C}{\nu_i} y_i) \nu_i \sigma_i - \nu_C \sigma_{atm} \\ &= C^{BS}(\sigma_{atm}) + \nu_C \left[\sum_{i=1}^{3} (y_i \sigma_i) - \sigma_{atm} \right] \\ &= C^{BS}(\sigma_{atm}) + \nu_C (\xi - \sigma_{atm}) \end{aligned}$$

where $\xi = \sum_{i=1}^{3} (y_i \sigma_i)$ would become the first order approximation of the implied volatility for strike K, and the coefficients y_i are given by:

$$y_1 = \frac{\log \frac{K_2}{K} \log \frac{K_3}{K}}{\log \frac{K_2}{K_1} \log \frac{K_3}{K_1}}, y_2 = \frac{\log \frac{K}{K_1} \log \frac{K_3}{K}}{\log \frac{K_2}{K_1} \log \frac{K_3}{K_2}}, y_3 = \frac{\log \frac{K}{K_1} \log \frac{K}{K_2}}{\log \frac{K_3}{K_1} \log \frac{K_3}{K_2}}$$

1.5.7 A second-order approximation for the implied volatility of the option C

A second-order Taylor's series approximation for the $C^{mkt}(\sigma)$ would be:

$$\begin{aligned} C^{mkt}(\sigma) &= C^{BS}(\sigma_{atm}) + (\sigma - \sigma_{atm}) \frac{\partial C}{\partial \sigma} + \frac{1}{2} (\sigma - \sigma_{atm})^2 \frac{\partial^2 C}{\partial \sigma^2} \\ &= C^{BS}(\sigma_{atm}) + \nu(\sigma - \sigma_{atm}) + \frac{d_+ d_- \nu}{2\sigma} (\sigma - \sigma_{atm})^2 \end{aligned}$$

The expression for the market price of the option is then:

$$C^{mkt} = C^{BS}(\sigma_{atm}) + \sum_{i=1}^{3} x_i \left[\nu_i(\sigma_i - \sigma_{atm}) + \frac{d_i^+ d_i^- \nu_i}{2\sigma}(\sigma_i - \sigma_{atm})^2 \right]$$

$$= C^{BS}(\sigma_{atm}) + \nu_C \left[\sum_{i=1}^{3} y_i \sigma_i - \sigma_{atm} \right] + \frac{\nu_C}{2\sigma} \sum_{i=1}^{3} \left[d_i^+ d_i^- y_i (\sigma_i - \sigma_{atm})^2 \right]$$

Let's call the second order approximation ζ. We must have:

$\nu_C(\zeta - \sigma_{atm}) + \frac{d_C^+ d_C^- \nu_C}{2\sigma_{atm}}(\zeta - \sigma_{atm})^2 = \nu_C(\xi - \sigma_{atm}) + \nu_C \frac{1}{2\sigma} \sum_{i=1}^{3} d_i^+ d_i^- y_i (\sigma_i - \sigma_{atm})^2$

$\implies \frac{d_C^+ d_C^-}{2\sigma_{atm}}(\zeta - \sigma_{atm})^2 + (\zeta - \sigma_{atm}) = P + \frac{Q}{2\sigma}$

$\implies d_C^+ d_C^- (\zeta - \sigma_{atm})^2 + 2\sigma_{atm}(\zeta - \sigma_{atm}) - (2\sigma_{atm} P + Q) = 0$

where $P = (\xi - \sigma_{atm})$ and $Q = \sum_{i=1}^{3} d_i^+ d_i^- y_i (\sigma_i - \sigma_{atm})^2$

Solving the above quadratic equation, we must have:

$$\zeta - \sigma_{atm} = \frac{-2\sigma_{atm} + \sqrt{4\sigma_{atm}^2 + 4 d_C^+ d_C^- (2\sigma_{atm} P + Q)}}{2 d_C^+ d_C^-}$$

$$\zeta = \sigma_{atm} + \frac{-\sigma_{atm} + \sqrt{\sigma_{atm}^2 + d_C^+ d_C^- (2\sigma_{atm} P + Q)}}{d_C^+ d_C^-}$$

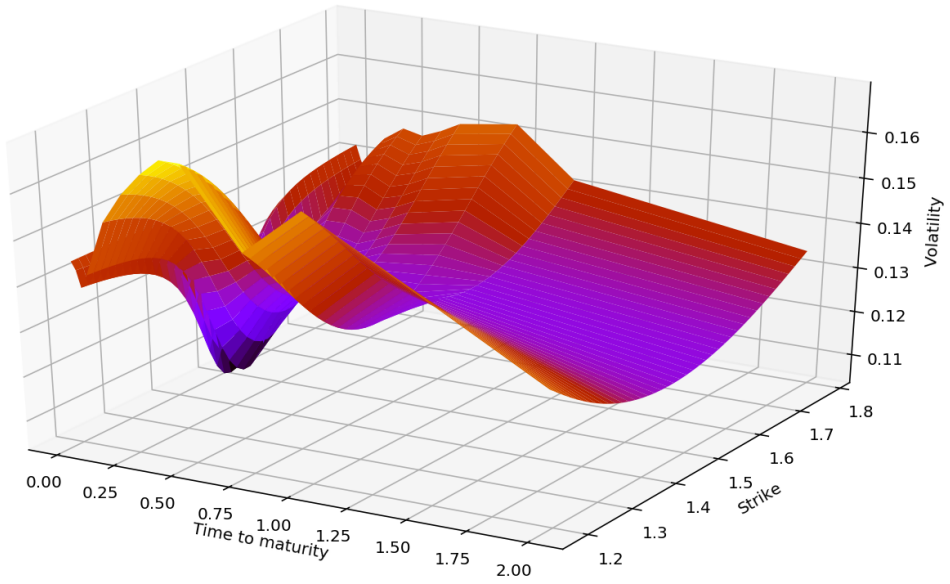
1.6 Sample code snippet to implement the VV method

```
In [9]: import pandas as pd
        import numpy as np
        import math
        import scipy.stats as sct
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D
        from matplotlib import cm

        # Market quotes for maturities ranging from 1 week to 2 years
        T = np.array([0.0192,0.0384,0.0575,0.0833,0.1667,0.2500,
                      0.3333,0.4167,0.5000,0.7500,1.0000,2.0000 ])
        n = len(T)
        sigma25DeltaPut=np.array([0.1170,0.1121,0.1068,0.1130,0.1210,0.1283,
                          0.1341,0.1372,0.1369,0.1263,0.1396,0.1234])
        sigmaATM=np.array([0.1105,0.1100,0.1048,0.1060,0.1130,0.1203,
                  0.1247,0.1270,0.1273,0.1238,0.1300,0.1213])
        sigma25DeltaCall=np.array([0.1090,0.1129,0.1088,0.1050,0.1130,0.1203,
                        0.1237,0.1265,0.1286,0.1328,0.1314,0.1291])
        r = np.array([0.0023,0.0023,0.0024,0.0024,0.0026,0.0028,
             0.0036,0.0044,0.0055,0.0089,0.0119,0.0124])
        q = np.array([0.0027,0.0027,0.0028,0.0032,0.0038,0.0042,
             0.0050,0.0060,0.0073,0.0110,0.0141,0.0139])

        Dd = np.exp(-np.multiply(r,T))
        Df = np.exp(-np.multiply(q,T))
        mu = r - q
```

```
S0 = 1.4844
delta = 0.25
F = S0*np.exp(np.multiply(mu,T))
alpha = -sct.norm.ppf(delta*np.reciprocal(Df))
K25DCall = np.array([])
KATM = np.array([])
K25DPut = np.array([])

for i in range(12):
    K1 = F[i]*math.exp(-(alpha[i]*sigma25DeltaPut[i]*math.sqrt(T[i]))\
                       +0.5*(sigma25DeltaPut[i]**2)*T[i])
    K25DPut = np.append(K25DPut,K1)
    K2 = F[i]*math.exp(0.5*(sigmaATM[i]**2)*T[i])
    KATM = np.append(KATM,K2)
    K3 = F[i]*math.exp((alpha[i]*sigma25DeltaCall[i]*math.sqrt(T[i]))\
                       +0.5*(sigma25DeltaCall[i]**2)*T[i])
    K25DCall = np.append(K25DCall,K3)

def d1(F,K,sigma,t):
    dplus = (math.log(F/K) + 0.5 *(sigma**2)*t)/(sigma*math.sqrt(t))
    return dplus

def d2(F,K,sigma,t):
    dminus = d1(F,K,sigma,t)-sigma*math.sqrt(t)
    return dminus

def VannaVolgaImpliedVol(F,K,t,K1,K2,K3,
                sigma1,sigma2,sigma3):

    y1 = (math.log(K2/K) * math.log(K3/K))\
    /(math.log(K2/K1) * math.log(K3/K1))
    y2 = (math.log(K/K1) * math.log(K3/K))\
    /(math.log(K2/K1) * math.log(K3/K2))
    y3 = (math.log(K/K1) * math.log(K/K2))\
    /(math.log(K3/K1) * math.log(K3/K2))

    P = (y1*sigma1 + y2*sigma2 + y3 *sigma3) - sigma2
    Q = y1 * d1(F,K1,sigma1,t) * d2(F,K1,sigma1,t) * ((sigma1-sigma2)**2) \
    + y2 * d1(F,K2,sigma2,t) * d2(F,K2,sigma2,t) * ((sigma2-sigma2)**2) \
    + y3 * d1(F,K3,sigma3,t) * d2(F,K3,sigma3,t) * ((sigma3-sigma2)**2)

    d1d2 = d1(F,K,sigma2,t) * d2(F,K,sigma2,t)

    sigma = sigma2 + (-sigma2 + math.sqrt(sigma2**2 + d1d2 *(2*sigma2*P+Q)))/(d1d2)

    return sigma

strike = (1+np.arange(-0.20,0.21,0.01))*S0
```

```
VVImpliedVol = np.zeros((41,12),dtype=float)

for i in range(41):
    for j in range(12):
        VVImpliedVol[i][j]=VannaVolgaImpliedVol(F[j],strike[i],T[j],
                                                K25DPut[j],KATM[j],
                                                K25DCall[j],sigma25DeltaPut[j],
                                                sigmaATM[j],sigma25DeltaCall[j])

X = T
Y = strike
X, Y = np.meshgrid(X,Y)
Z = VVImpliedVol

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.set_xlabel('Time to maturity')
ax.set_ylabel('Strike')
ax.set_zlabel('Volatility')
surf = ax.plot_surface(X, Y, Z, cmap=cm.gnuplot,linewidth=0)
plt.show()
```

Out[9]:



In [0]: