**Lecture 6 (Exotics Markets) Assignment, MTH 9865**

Due start of class, October 21, 2015

**Question 1 (6 marks)**

In class we looked at Gaussian copulas for pricing two-asset derivatives. We talked about using that model to price cross-pair options based on the two USD-pair option markets, calibrating the Gaussian copula correlation parameter such that the model reproduces the ATM cross option price.

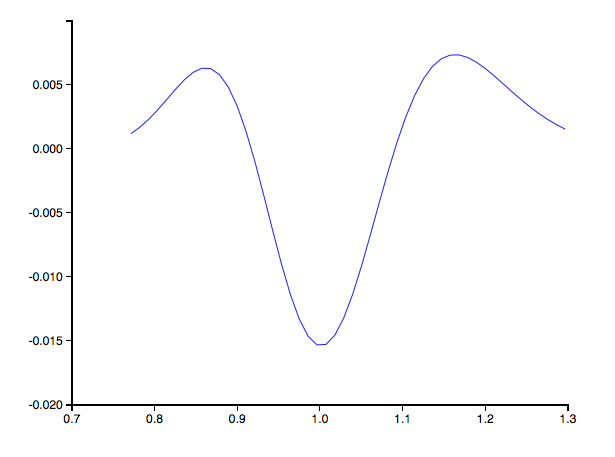
When we do that, the model tends to underprice the butterfly – ie the implied volatility smile that comes out of the model is lower than the market. We said that was due to the fact that the model does not include any premium for stochastic correlation, which it should because correlation is not constant.

A stochastic correlation only affects the value of a derivative if its exposure to that correlation is non-linear. Let’s consider a specific case of the Gaussian copula to examine that: one where the RR and BF of the USD pairs is zero. In that case, their pricing is just Black-Scholes pricing, and if we assume a constant correlation, the pricing of the cross option is Black-Scholes with an implied volatility

Calculate the “gamma” of the cross option price with respect to the correlation parameter . Assume a market where the two USD-pair spots are 1, interest rates are zero, time to expiration is 0.5y, the USD-pair volatilities are both equal to 10%, and the correlation is +25%, and plot the correlation gamma as a function of strike for the cross-pair options.

Discuss the qualitative impact stochastic correlation should have on the cross-pair implied volatilities based on that plot.

We know how to price the cross option given the correlation: just use the Black-Scholes formula with cross spot=spot1/spot2 and implied volatility equal to . With that, we can numerically calculate the second partial derivative of option price to the correlation parameter and plot it out:



This shows the correlation gamma on the y-axis vs the cross option strike on the x-axis.

The correlation gamma is negative for ATM cross options and positive for OTM cross options. This means that any volatility in the correlation parameter will tend to push down the prices of ATM cross options, and push up the prices of OTM cross options, thereby increasing the cross butterfly.

The intuition here: I can eg buy an ATM cross option, then trade enough of a slightly-OTM cross option to make the correlation sensitivity (the first-order derivative) zero. Then I’ve got a payoff vs correlation that always loses money as correlation moves around – just like with a delta-hedged vanilla option and regular gamma, or the vega gamma/vega dspot stuff we were talking about as the source of skew and smile. Because I’ll always lose money on correlation moves, I’m going to tend to sell that position.

This is implemented in wst/baruch/assig6.py, run\_corr\_gamma function.

**Question 2 (3 marks)**

Describe the market dynamic that is most important for knockout out pricing, and compare that to the market dynamic that is the most important for volatility swap pricing.

For each, explain why that market dynamic is important to the pricing.

Knockout pricing is most sensitive to the risk reversal beta, and volatility swap pricing is most sensitive to realized volatility of implied volatility.

For knockouts, as we saw in class, you can construct a hedge portfolio of vanilla options that is a “semi-static” hedge for the knockout. If I’m long a down-and-out knockout call, I can sell the regular call (same strike as the knockout) and buy a put whose strike is the knockout strike “reflected” through the barrier: B2/K. That two-vanilla portfolio hedges almost all my market risk unless the barrier is hit – then I need to unwind it. The cost of unwinding that two-vanilla portfolio when spot is at the barrier is mostly a function of risk reversal, because at that point the call and put are (roughly) equally out-of-the-money. That means their net sensitivity to ATM vol is small, as is their net sensitivity to butterfly. That portfolio is sensitive only to the level of risk reversal. The expected level of risk reversal is determined by how much we think risk reversal will have moved while spot moved down to the barrier, which is basically our measure of risk reversal beta: the regression coefficient of moves in risk reversal with moves in spot.

For vol swaps, the natural hedge for a long vol swap is a short variance swap, since we can replicate the variance swap with vanilla options in a model-free way (usual caveats apply: doesn’t work if there are jumps, and gives exposure to implied vol extrapolation behavior). The vol swap then looks like a square-root payoff against the variance swap, which is non-linear dependence. That means you need to keep rebalancing the notional of the short variance swap, much like you’d need to keep rebalancing the notional of a delta hedge against a short option position. And much the same way, you expect to lose money doing so, as it’s a short gamma position (in the case of a vol swap it’s short gamma to the variance swap fair strike; in the case of a vanilla option it’s short gamma to the underlying spot). You then expect to lose money over time running a long vol swap position, and you get compensated for that by being able to enter the vol swap at a fair strike that’s less than the square root of the variance swap fair strike. The discount in fair strike depends on the realized volatility of the variance swap fair strike, which is roughly the same as realized volatility of implied volatility.

**Question 3 (4 marks)**

Consider a dual digital option that pays $1 if EURUSD is above a strike K1 **and** GBPUSD is above a strike K2. All discount rates are zero. The price of the EURUSD European digital option (paying $1 if EURUSD is above K1) is 65% and the price of the GBPUSD European digital option (paying $1 if GBPUSD is above K2) is 30%.

Plot the price of the dual digital option priced under a Gaussian copula model, for correlation parameter ranging from -100% to +100%. Qualitatively explain the behavior of the price sensitivity to correlation.

We’ll use a Gaussian copula, which means that the first step is translating both the two strikes into standard normal variables. This transformation is governed by matching the cumulative distributions:

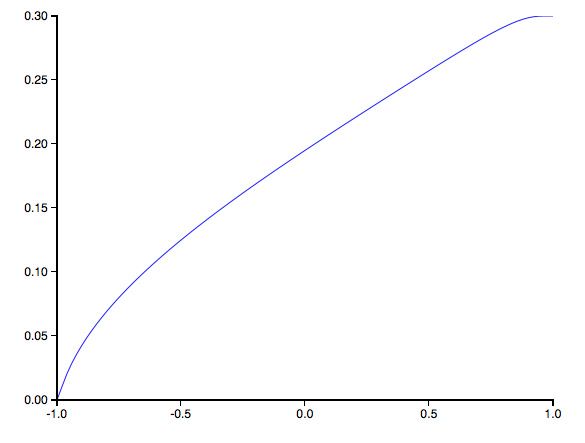
is the standard cumulative normal distribution function: the probability that a standard normal variable is less than or equal to some level . is the (risk neutral) probability that the spot is less than or equal to some level .

For the EURUSD piece, we know the price of a digital call, which is (since we stated that all the interest rates are zero). So we can calculate

Similarly for the GBPUSD piece we can calculate

Then we can calculate the joint probability of the first standard normal being less than or equal to AND the second standard normal being less than or equal to by using the bivariate standard normal cumulative distribution function.

We can then plot out that dual digital call price as a function of correlation:



Note that the price increases with correlation, which is what you’d expect: if the first asset is above the strike, when the correlation is positive, it’s more likely that the second asset is above the barrier as well.

We can say a bit more, actually. When correlation=100%, the price should be the minimum of the two European digital prices, which we see in the example. When correlation=0%, the price should be the product of the two European digital prices (25%), which we also see. When the correlation is -100% the price should be zero, as if one asset is above its strike, the second must be below; we also see that in the plot.

**Question 4 (10 marks)**

Investigate knockout pricing under the LV/SV approximation model.

Let’s imagine a market environment with spot=1, zero denominated and asset discount rates, and implied volatilities by benchmark expiration time defined as

|  |  |  |  |
| --- | --- | --- | --- |
| Expiration Time (yr) | ATM Vol (%) | 25d Risk Reversal (%) | 25d Butterfly (%) |
| 1/12 | 7.50 | 0.25 | 0.25 |
| 1/4 | 7.80 | 0.00 | 0.30 |
| 1/2 | 8.00 | -0.50 | 0.32 |
| 3/4 | 8.05 | -1.00 | 0.30 |
| 1 | 8.20 | -1.25 | 0.30 |

The 10d risk reversal is equal to the 25d risk reversal multiplied by 1.8 for all expirations; the 10d butterfly is equal to the 25d butterfly multiplied by 3.0 for all expirations.

You should use a mean reversion parameter  equal to 1 in all cases. You should try eleven different values for : 0 through 1.0 in steps of 0.1. For numerical parameters, use nu=500, nt=150, and nsd=5.

The goal of this question is to investigate how sensitive a knockout price is to our choice of mixture parameter, which in this model is the value of . The knockout you’re pricing is a call option with one year to expiration, a strike price of 1.03, and a down-and-out barrier at 0.95.

For each value of , you should:

1. Calibrate the local volatilities in the model such that the model reproduces the market prices for all the benchmark options (for each of the five expiration dates and each of the five points in the strike direction). You’ll of course need to calculate the five benchmark implied volatilities from the ATM, RR, and BF marks; and you’ll need to calculate the strikes those benchmark deltas correspond to.
2. Print out a table showing the calibrated local volatilities.
3. Price the knockout option under the model, calculating a basis point price (knockout price divided by spot, multiplied by 104).
4. Price the vanilla option underlying the barrier option, again as a basis point price.

Plot the knockout price as a function of , and plot the vanilla option price as a function of . (The vanilla price should be only a weak function of .)

Remembering that the main dynamic impacting the knockout price is risk reversal beta, qualitatively explain the behavior of the knockout price with .

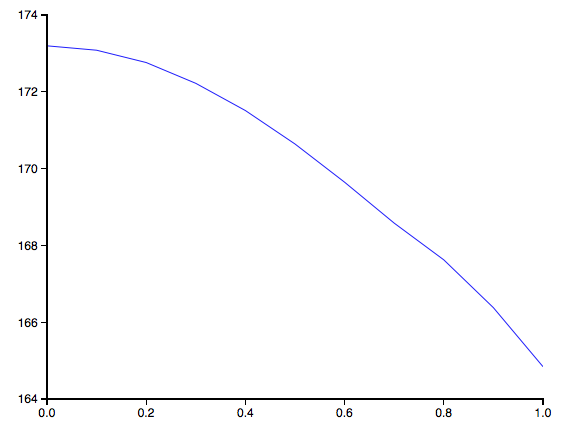
For this question we’ll use the backward and forward induction engines in the WST environment that we went over in the class.

The first step is calibration. For this we’ll use the forward induction engine since it lets us price a bunch of vanillas in one numerical pass, so is more computationally efficient than using backward induction.

We’ll also “bootstrap”: generate the five local vol parameters for the first time period, calibrated to the first period’s five implied vols; then use those plus the five implied vols for the second period to get the five local vol parameters for the second period; and so on until everything’s calibrated.

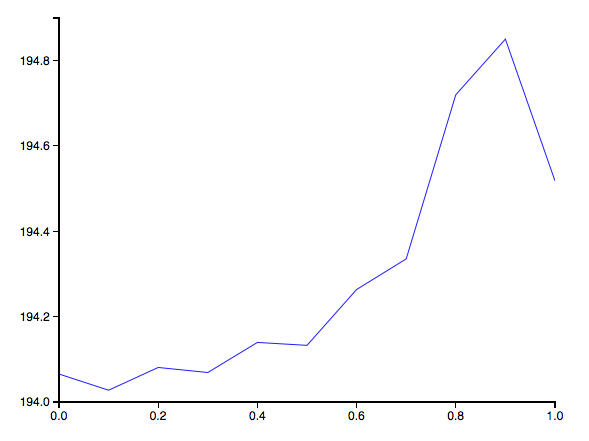
Once we have the calibrated local vols we can use a backward induction engine to price the knockout option, and a separate backward induction to price the underlying vanilla option.

We’ll do that for the specified range of values of the  parameter. The chart of knockout price vs  looks like:



The chart shows the vol of vol parameter  on the x-axis, and the price of the knockout in basis points on the y-axis. The typical bid/ask spread for knockouts is only a few basis points, so the impact of changing the model parameter is very significant.

Here’s the same chart but for the vanilla price:



As expected the vanilla option price hardly varies at all, because for each different value of  we’re recalibrating the model to hit vanilla prices again.

This is implemented in wst/baruch/assig6.py, run\_lvs function.