Mike Staunton

# Close Encounters of the Third Order

## Implied volatility approximations atop the Devils Tower


The 1977 science fiction movie *Close Encounters of the Third Kind* was written and directed by Steven Spielberg. The title is derived from the classification by Hynek for close encounters with aliens, in which the third kind denotes human observations of animate beings. Richard Dreyfuss plays an electrical lineman from Indiana with an obsession with UFOs – Steve McQueen was Spielberg's first choice and he liked the script but turned it down because he felt that he would be unable to cry on cue.

Why is the third order relevant? And now I'm talking about third-order implied volatility approximations starring Matt Lorig and his two Italian co-stars, Stefano Pagliarani and Andrea Pascucci in my early 2014 adaptation of their original 2013 paper. I like to think that I possess a healthy skepticism when it comes to the use of analytic approximations in option pricing, but this working paper and their associated Mathematica code is well worth reading (even if you don't go as far as I did and actually buy a license for Mathematica, as you can read their notebook files for free).

Their paper has similar approximations for the CEV, SABR, Heston, and 3/2 stochastic volatility models, but I have chosen to concentrate on the quadratic local volatility model here. Leif Andersen suggests that, due to their tractability, quadratic volatility models might serve as a convenient alternative to more complicated local-stochastic volatility models. His 2008 working paper presents formulas for European option prices under quadratic volatility (for differing numbers of real roots as well as allowing for absorbing barriers) and pads out the remaining pages with a mere nine lemmas and eight propositions. His lemma 5 describes the case when the quadratic has two real roots and both are to the right of the initial asset value (replicated as equation 50 in Lorig, Pagliarani, and Pascucci) and forms the basis for my EuroCallQLV function.

Their third-order implied volatility approximation for the quadratic volatility model is given in equation 49 and, like their approximations for other volatility models, contains no numerical integration or, indeed, any special functions – you will see only cosh and sinh in my ImpliedVolQLV function code. The resulting implied volatility can then easily be used as an input into the standard Black–Scholes valuation formula. They claim a relative error of less than 1 percent for nearly all strikes when option maturity is less than four years, and a relative error below 3 percent when maturity extended to ten years.

Fortunately, their paper stops at third-order approximations and does not continue on to the corresponding fourth-order approximations – Hynek's scale has been extended to cover close encounters of the fourth kind that would involve the abduction of a human being by a UFO or its occupants!

### REFERENCES

Andersen, LBG. 2008. Option Pricing with Quadratic Volatility: A Revisit. http://ssrn.com/abstract=1118399 or http://dx.doi.org/10.2139/ssrn.1118399

Lorig, M, Pagliarani S, and Pascucci A. 2013. Implied Vol for Any Local-Stochastic Vol Model http://ssrn.com/abstract=2283874 or http://dx.doi.org/10.2139/ssrn.2283874

## THE VBA CODE

```
Option Base 0

Function vaEuroCallQLV(S#, K#, delta#, capL#,
capR#, Tyr#) As Double
Dim eL#, eR#, cR#, cL#
Dim K1#, K2#, S1#, S2#
Dim V#, SqrV#, d1p#, d1m#
Dim d2p#, d2m#

eL = Exp(capL)
eR = Exp(capR)
cR = (eR - S) / (eR - eL)
cL = (eL - S) / (eR - eL)
K1 = (eL - K) * cR
K2 = (eR - K) * cR
S1 = cL * (eR - K)
S2 = cL * (eL - K)

V = delta * delta * Tyr
SqrV = Sqr(V)
d1p = (Log(S1 / K1) + 0.5 * V) / SqrV
d1m = (Log(S1 / K1) - 0.5 * V) / SqrV
d2p = (Log(S2 / K2) + 0.5 * V) / SqrV
d2m = (Log(S2 / K2) - 0.5 * V) / SqrV

vaEuroCallQLV = K1 * vaCND(-d1m) - S2 * vaC-
ND(d2p) - S1 * vaCND(-d1p) + K2 * vaCND(d2m)
End Function

Function vaImpliedVolQLV(sk#, x#, delta#,
capL#, capR#, Tyr#) As Double
Dim kmx#, Lmx#, Rmx#, LmR#
Dim eL#, eR#, ex#, d2#
Dim a10#, a20#, a30#, sig0#
Dim sig02#, sig04#, sig1#, sig21#
Dim sig22#, sig2#, sig31#, sig32#
Dim sig3#

kmx = sk - x
Lmx = capL - x
Rmx = capR - x
LmR = capL - capR
eL = Exp(capL)
eR = Exp(capR)
ex = Exp(x)
d2 = delta * delta

a10 = d2 * (-Sinh(Lmx + Rmx) + Sinh(Lmx) +
Sinh(Rmx)) / (Cosh(LmR) - 1)
a20 = 0.25 * d2 * (2 * Cosh(Lmx + Rmx) -
Cosh(Lmx) - Cosh(Rmx)) * Csch(0.5 * LmR) *
Csch(0.5 * LmR)
a30 = eL * eR * d2 * (-4 * Sinh(Lmx + Rmx) +
Sinh(Lmx) + Sinh(Rmx)) / (3 * (eL - eR) * (eL -
eR))

sig0 = delta * (eR - ex) * (eL - ex) / (ex * (eR
- eL))
sig02 = sig0 * sig0
sig04 = sig02 * sig02

sig1 = 0.5 * a10 * kmx / sig0

sig21 = -Tyr * (12 + Tyr * sig02) * a10 * a10 /
(96 * sig0) + Tyr * sig0 * a20 / 6
sig22 = (-3 * a10 * a10 + 4 * sig02 * a20) * kmx
* kmx / (12 * sig0 * sig02)
sig2 = sig21 + sig22

sig31 = (-12 + Tyr * sig02) * a10 * a10 * a10 +
4 * sig02 * (8 + Tyr * sig02) * a10 * a20 - 48 *
sig04 * a30
sig32 = 3 * a10 * a10 * a10 - 5 * sig02 * a10 *
a20 + 3 * sig04 * a30
sig3 = -Tyr * sig31 * kmx / (192 * sig0 * sig02)
+ sig32 * kmx * kmx * kmx / (12 * sig0 * sig04)

vaImpliedVolQLV = sig0 + sig1 + sig2 + sig3
End Function
```