Quantitative Finance, 2020
Vol. 20, No. 8, 1227–1235, https://doi.org/10.1080/14697688.2020.1726440

Strike from volatility and delta-with-premium†

PETER JÄCKEL*

Quantitative Research, VTB Europe SE, Frankfurt, Germany

(*Received 24 September 2019; accepted 31 January 2020; published online 7 April 2020*)

Speedy-yet-full-accuracy evaluation motivates this research

1. Introduction

We give an efficient two-step procedure to compute the strike implied by the quote pair of a Black volatility and a delta-with-premium, thus effectively providing an analytical solution to this ubiquitous problem in FX markets.

The Black-Scholes-Merton (Black and Scholes 1973, Black 1976) *forward* option price formula, i.e. without consideration of any discounting to the payment date, has the form

$$v = \theta \cdot [F \cdot \Phi(\theta \cdot d_1) - K \cdot \Phi(\theta \cdot d_2)] \tag{1}$$

with

$$\theta := \pm 1 \tag{2}$$

for calls and respectively puts, and

$$z := \ln(K/F) \quad d_1 := -\frac{z}{\sigma} + \frac{\sigma}{2} \quad d_2 := -\frac{z}{\sigma} - \frac{\sigma}{2} \tag{3}$$

where σ is the product of Black implied volatility and the square root of time to expiry. The *forward delta* is

$$\Delta_F = \theta \cdot \Phi(\theta \cdot d_1) \tag{4}$$

and the *forward-delta-with-premium*, as used for many FX currency pairs (Reiswich and Wystup 2010), is

$$\begin{aligned} \Delta'_F &:= \Delta_F - v/F \\ &= \theta \cdot K/F \cdot \Phi(\theta \cdot d_2) \\ &= \theta \cdot \mathrm{e}^{z} \cdot \Phi(-\theta \cdot (z/\sigma + \sigma/2)) \\ &= \theta \cdot \mathrm{e}^{\alpha(y-\alpha/2)} \cdot \Phi(-y) \end{aligned} \tag{5, 6}$$

with

$$\alpha := \theta \cdot \sigma \quad y := z/\alpha + \alpha/2. \tag{7}$$

With discounting, the option value is $v \cdot P_T$ where P_T is the domestic (quotation currency) discount factor to the payment date. The spot relates to the forward via

$$F = S \cdot \frac{P_T^{\mathrm{FOR}}}{P_T} \tag{8}$$

which makes

$$\frac{\partial F}{\partial S} = \frac{P_T^{\mathrm{FOR}}}{P_T} \tag{9}$$

where P_T^{FOR} is of course the foreign discount factor. The *spot delta* of $v \cdot P_T$ is

$$\Delta_S = \Delta_F \cdot \frac{\partial F}{\partial S} \cdot P_T. \tag{10}$$

As a result, the *spot-delta-with-premium*, is

$$\begin{aligned} \Delta'_S &:= \Delta_S - (v \cdot P_T)/S \\ &= \Delta_S - \left(v \cdot P_T \cdot \frac{\partial F}{\partial S}\right) \Big/ F \\ &= \frac{\partial F}{\partial S} \cdot P_T \cdot [\Delta_F - v/F] \\ &= \frac{\partial F}{\partial S} \cdot P_T \cdot \Delta'_F \end{aligned} \tag{11, 12}$$

which means that any spot-delta-with-premium can be converted to a forward-delta-with-premium. Hence, we restrict

*Corresponding author. Email: p.jaeckel@vtb.eu
† This work was done while at VTB Capital London.

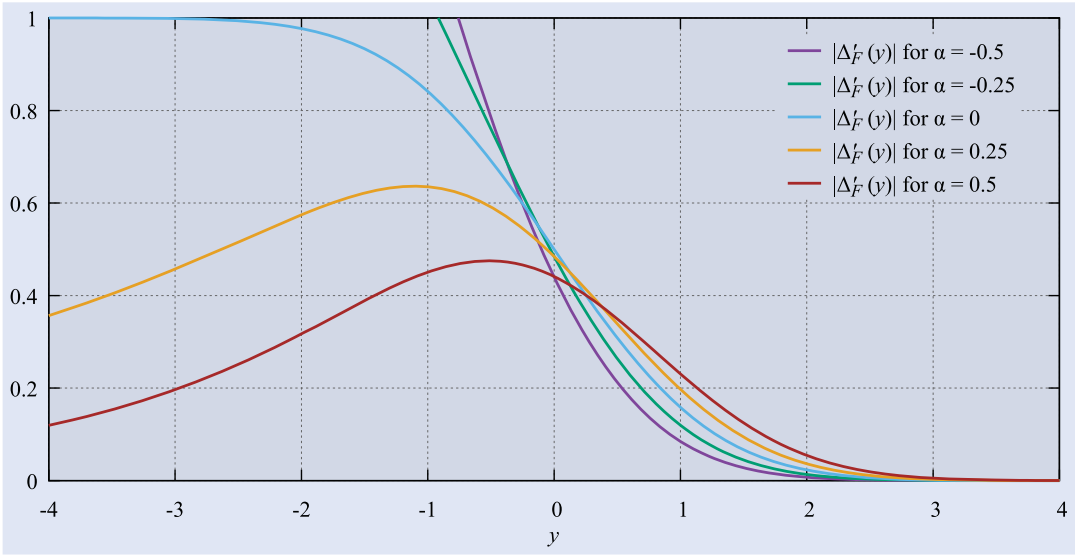

Figure 1. $|\Delta'_F(y)|$ as given by (6). Both α and Δ'_F are positive for calls and negative for puts.

ourselves from here on to the forward case without loss of generality. Some examples for $|\Delta'_F(y)|$ are shown in figure 1.

The objective of this note is to present a highly efficient method to compute K given a quote pair (σ, Δ'_F) which permits no closed form solution, unlike when the premium is not included. Since FX markets' practitioners tend to be particularly keen on high computational speed, it is somewhat surprising that the only literature on this problem suggests iterative root-finding algorithms (Reiswich and Wystup 2010), and in particular the Brent algorithm which, whilst superbly robust, only has a moderate convergence† order of about 1.84. In the following, we describe what is effectively an analytical two-step procedure that gives us, for all intents and purposes, *full attainable precision* (defined by rigorous error propagation analysis in Section 3.3). We mention that the speedy-yet-full-accuracy evaluation is the core motivation of this research. This is because it was borne out of a direct industrial need in a derivatives valuation setting: it was noticed that various financial valuation systems, which by their own technical requirements insisted on the calculation of market quote sensitivities by small perturbations (single or even sub-basis points shifts in implied volatilities and FX spot quotes, for instance), produced rather noisy Delta, Vega, and particularly Gamma figures, but only for FX rates for which market quotes included the premium in the delta. An investigation yielded that the noise was caused by the strike-from-delta analytics which had been designed with a Brent root finding algorithm and configured to return with moderate precision in aid of timely performance. For commercial reasons, it was considered not just prudent but necessary to have analytics that could produce the same accuracy for the strike-from-delta with or without the premium being included in the delta, but without the performance hit incurred by the standard (Brent root finding) methodology.

† Brent's convergence order is that of inverse quadratic interpolation given by the root of $\mu^3 - \mu^2 - \mu - 1$ (about 1.84). This is only marginally above that of the secant method (the golden ratio, about 1.62).

2. The modified delta-with-premium function

Dropping the subscript F from Δ'_F for brevity from here onwards, we re-write equation (6) as

$$\ln(2 \cdot \Phi(-y)) + \alpha \cdot y = \ln|2 \cdot \Delta'| + \frac{\alpha^2}{2}. \tag{13}$$

Since the left-hand side contains the unknown y, and the right-hand side only known terms, we define

$$f(y) := \ln(2 \cdot \Phi(-y)) + \alpha \cdot y \tag{14}$$

as the *modified delta-with-premium function* and find y from

$$f(y) = \ln|2 \cdot \Delta'| + \frac{\alpha^2}{2} \tag{15}$$

and obtain the strike via

$$K = F \cdot \mathrm{e}^{\alpha \cdot y - \alpha^2/2}. \tag{16}$$

Note that the seemingly redundant factor 2 inside the logarithm of the definition of $f(\cdot)$ is to anchor the function at the origin, i.e. to have

$$f(0) = 0. \tag{17}$$

This is done to facilitate the preservation of *relative* numerical accuracy of the 'loop' equation $y = f^{-1}(f(y))$ in regions where $|y|$ or $|f|$ are very small.

The function $f(y)$ intersects the origin, i.e. $f(0) = 0$, and has the asymptotic behaviour

$$\lim_{y \to -\infty} f(y) \approx \ln(2) + \alpha \cdot y + \frac{\varphi(y)}{|y|}\left[1 + \mathcal{O}\left(y^{-2}\right)\right] \tag{18}$$

$$\lim_{y \to +\infty} f(y) \approx \ln\left(\sqrt{\frac{2}{\pi}}\right) - \ln(y) + \alpha \cdot y - \frac{1}{2} \cdot y^2 + \mathcal{O}\left(y^{-2}\right). \tag{19}$$

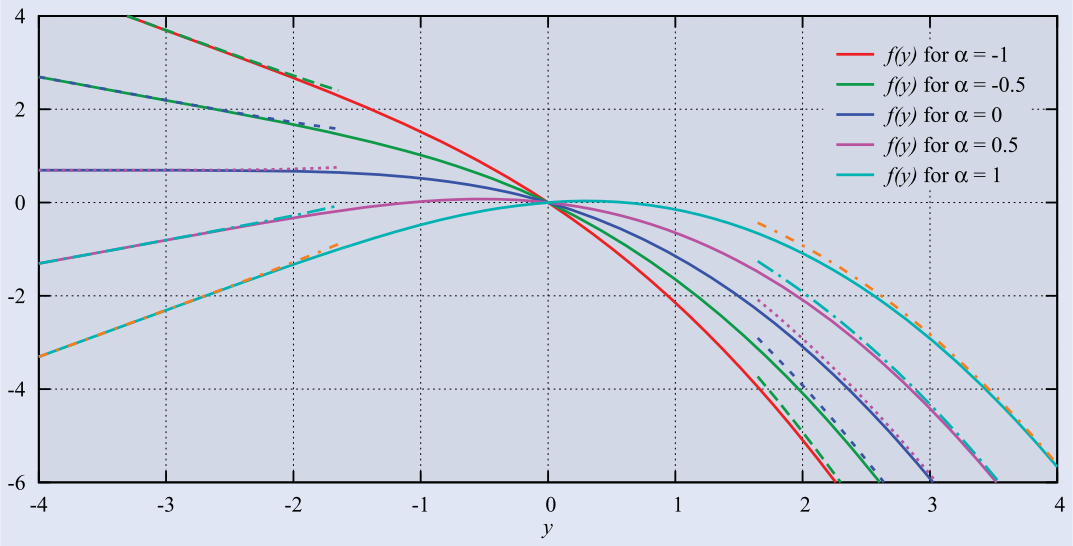

Figure 2. The modified delta-with-premium function $f(y)$ as given in (14) and its asymptotics (18) and (19) for $|y| > 1$ for five different values of α (which is positive for calls and negative for puts).

The function $f(y)$ is strictly decreasing when $\alpha \le 0$. For $\alpha \equiv 0$, it is asymptotically flat for $y \to -\infty$. When $\alpha > 0$, it has a maximum at y_{max} which is the solution of

$$\frac{\varphi(y_{max})}{\Phi(-y_{max})} = \alpha. \tag{20}$$

This can be expressed as

$$y_{max} = \sqrt{2} \cdot \mathrm{erfcx}^{-1}\left(\frac{\sqrt{\frac{2}{\pi}}}{\alpha}\right) \tag{21}$$

with erfcx(·) as defined in Cody (1969). An accurate and efficient two-step implementation of the inverse function $\mathrm{erfcx}^{-1}(\cdot)$ is given in Jäckel (2014, Appendix C). The location y_{max} is monotonic in α with

$$\begin{aligned} y_{max} < 0 \quad &\text{when} \quad \alpha < \sqrt{\frac{2}{\pi}}, \\ y_{max} = 0 \quad &\text{when} \quad \alpha = \sqrt{\frac{2}{\pi}}, \\ y_{max} > 0 \quad &\text{when} \quad \alpha > \sqrt{\frac{2}{\pi}}. \end{aligned} \tag{22}$$

We show examples for $f(y)$ in figure 2.

Since we will need them in the following, we also list here the derivatives of $f(y)$ up to sixth order

$$f'(y) = \alpha - q \tag{23}$$

$$f''(y) = q(y - q) \tag{24}$$

$$f^{(3)}(y) = q(q(3y - 2q) + 1 - y^2) \tag{25}$$

$$f^{(4)}(y) = q(q(q(12y - 6q) + 4 - 7y^2) + y(y^2 - 3)) \tag{26}$$

$$\begin{aligned} f^{(5)}(y) = q(&q(q(q(60y - 24q) + 20 - 50y^2) + y(15y^2 - 25)) \\ &+ y^2(6 - y^2) - 3) \end{aligned} \tag{27}$$

$$\begin{aligned} f^{(6)}(y) = q(&q(q(q(q(360y - 120q) + 120 - 390y^2) \\ &+ y(180y^2 - 210)) + y^2(101 - 31y^2) - 28) \\ &+ y(15 + y^2(y^2 - 10))) \end{aligned} \tag{28}$$

where we have used the abbreviation

$$q := \frac{\varphi(y)}{\Phi(-y)}. \tag{29}$$

Note that

$$f'(0) = \alpha - \sqrt{\frac{2}{\pi}} \tag{30}$$

$$f''(0) = -\frac{2}{\pi}. \tag{31}$$

3. The inverse modified delta-with-premium function

In order to impute y from a given function value f^* from (14), we combine a fairly accurate initial guess function with *exactly one* execution of the Householder(6) method (Householder 1970) which is of convergence order 7 (as opposed to Brent's method of order 1.84). Since we need no iterations whatsoever, we refer to this single polishing stage of the initial guess as a simple *improvement step* to avoid misunderstandings.

3.1. The initial guess function

We define an initial guess function $y_0(f^*)$ by subdivision into different regions in both α, and the target value f^*, for all of their respectively mathematically attainable values. This, notwithstanding the fact that for any real-world market data input, we only ever observed $f^* < 0$ (and thus $y > 0$), and $-\frac{1}{2} < \alpha < \frac{1}{2}$ with a standard deviation of α of about 0.115.

Without further dwelling on the methods that were used to select the respective regions' approximations, we give them below.

3.1.1. Non-positive alpha: $\alpha \le 0$. First, we discuss the case $f \ge 0$ (which had no practical usage in our market data tests), and thus $y < 0$. When

$$y < \Phi^{-1}\left(\frac{\ln(2)}{2} \cdot \epsilon\right) \approx -8.2535, \tag{32}$$

where $\epsilon \equiv$ DBL_EPSILON is the relative machine precision, we use the lowest order part of the asymptotic form (18). Thereafter, we patch together ten rational cubic approximations of y over f on the intervals between the points

$$\{\hat{y}_1, \ldots, \hat{y}_{11}\} = \{-8.2535, -7, -6\tfrac{1}{3}, -5\tfrac{2}{3}, -5, -4\tfrac{1}{3}, -3\tfrac{2}{3}, -3, -2, -1, 0\}. \tag{33}$$

In each of these segments, we use the rational cubic form of Delbourgo and Gregory (1985) matched to go through an interior point $\frac{1}{3}$ (in absolute terms) inwards from its upper edge (in y). In short, for $f \ge f(0) \equiv 0$, we use

$$y_0 = \begin{cases} \dfrac{(f^* - \ln(2))}{\alpha} & \text{if } f^* > f(-8.2535) \\ g_i^{\mathrm{rc}}\left(f^*; f(\hat{y}_i), f(\hat{y}_{i+1}),\right. & \text{if } f^* \ge f(\hat{y}_{i+1}) \\ \quad \hat{y}_i, \hat{y}_{i+1}, & \forall i = 1, \ldots, 10 \\ \quad \dfrac{1}{f'(\hat{y}_i)}, \dfrac{1}{f'(\hat{y}_{i+1})}, & \\ \quad \left. f(\hat{y}_{i+1} - \frac{1}{3}), \hat{y}_{i+1} - \frac{1}{3}\right) & \end{cases} \tag{34}$$

with g_i^{rc} as given in (A8), where we recalled that $\mathrm{d}y/\mathrm{d}f = 1/f'(y)$. Note that all of the involved function values in these segments are either constants or linear in α and therefore virtually no cost to evaluate, and the selection of the right segment can be done efficiently by binary nesting over the ordered sequence $f(\hat{y}_1) > \cdots > f(\hat{y}_{11})$.

When $f < 0$, we have $y > 0$ since $\alpha \le 0$ in this section, and $f(y)$ is strictly decreasing. Here, for increasing y, the function $f(y)$ follows the asymptotic form (19) which is essentially quadratic in y. We approximate the inverse as a square root with offset, matching the behaviour near zero to fifth order. This gives

$$y_0 = \sqrt{1 + 2 \cdot \nu \cdot \frac{6a_1 + \nu(6a_2 + \nu a_3)}{6a_1 + \nu(12b_1 + 3\nu b_2)}} - 1; \tag{35}$$

with

$$\begin{aligned} a_1 &= 90h_2^2(1-h_2)^2 - 60(2 - 3h_2 + h_2^2)h_3 \\ &\quad - 40h_3^2 + 30(h_2 - 1)h_4 \end{aligned} \tag{36}$$

$$\begin{aligned} a_2 &= 45h_2^2(1+3h_2)(1-h_2)^2 - 30(2 + 3h_2 - 10h_2^2 + 5h_2^3)h_3 \\ &\quad - 20(2 + h_2)h_3^2 + (15 - 60h_2 + 45h_2^2 + 10h_3)h_4 \\ &\quad + 6(1 - h_2)h_5 \end{aligned} \tag{37}$$

$$\begin{aligned} a_3 &= 135h_2^3(2 + h_2)(1-h_2)^2 - 90(4 - 7h_2 + 3h_2^3)h_2h_3 \\ &\quad + 180(h_2^2 - 1)h_3^2 - 80h_3^3 + (90 - 90h_2 + 60h_2h_3)h_4 \\ &\quad - 15h_4^2 + (18 - 18h_2^2 + 12h_3)h_5 \end{aligned} \tag{38}$$

$$\begin{aligned} b_1 &= 90h_2^3(1-h_2)^2 - 30(4 - 7h_2 + 3h_2^2)h_2h_3 \\ &\quad - 10(1 + 2h_2)h_3^2 + (15 - 45h_2 + 30h_2^2 + 5h_3)h_4 \\ &\quad + 3(1 - h_2)h_5 \end{aligned} \tag{39}$$

$$\begin{aligned} b_2 &= 135h_2^4(1-h_2)^2 - 60(2 - 5h_2 + 3h_2^2)h_2^2h_3 \\ &\quad - 40(2 - h_2 - h_2^2)h_3^2 - 40h_3^3 + 10(6h_2 - 9h_2^2 \\ &\quad + 3h_2^3 - 2(1 - 2h_2)h_3)h_4 - 5h_4^2 \\ &\quad + 4(3h_2 - 3h_2^2 + h_3)h_5 \end{aligned} \tag{40}$$

and

$$\nu = \frac{f^*}{f'(0)} \quad h_k = \frac{f^{(k)}(0)}{f'(0)}. \tag{41}$$

We mention that the case of $\alpha \equiv 0$ may require special error handling since then $f(y)$ does not have an inverse value for all possible inputs. In practice, however, the case of exactly zero volatility should never occur since we obviously will never need to associate a strike with a delta when volatility or time to expiry are exactly zero.

REMARK 3.1 As we will mention in the next section, the above inverse quadratic ('square root with offset') approximation (35) is also used when α is positive but below 0.1388 *and* f is negative. In the practical usage case of calibration to market-observable data, we found that this covers over 90% of all calculations.

3.1.2. Positive alpha: $0 < \alpha < 0.1388$. When $f < 0$, we use the same formula as for negative α, i.e. approximation (35), as was already mentioned in Remark 3.1.

Whilst we have not encountered the usage case for any practical market data, for $f \ge 0$, we proceed as follows. By virtue of (21), in this parameter region $\alpha \in (0, 0.1388]$, we obtain for the location of the maximum $\mathrm{y_{max}}$ of the maximum of $f(y)$ that $\mathrm{y_{max}} \in (-\infty, -1.5]$. For any value of α in this region, we compute $\mathrm{y_{max}}$ from (21), and then $\mathrm{f_{max}} = f(\mathrm{y_{max}})$, which satisfies $\mathrm{f_{max}} > 0$.

If the given input value f^* is greater than $\mathrm{f_{max}}$, then there is obviously no solution. Otherwise, we seek the solution on the right-hand-side branch, i.e. the larger of the two solutions. For this purpose, we define the index function that points to the nearest non-smaller element in the branch node array (33) for any value of y as

$$\hat{i}(y) := \arg\min_i(\hat{y}_i \ge y) \tag{42}$$

and set as follows:

$$\mathrm{i_{max}} := \hat{i}(\mathrm{y_{max}}) \tag{43}$$

$$\delta \mathrm{y_{max}} := \tfrac{1}{2}(\hat{y}_{\mathrm{i_{max}}+1} - \hat{y}_{\mathrm{i_{max}}}) \tag{44}$$

$$i_r := \hat{i}(\mathrm{y_{max}} + \delta \mathrm{y_{max}}) \tag{45}$$

$$y_r := \hat{y}_{i_r}. \tag{46}$$

If $f^* \le f(y_r)$, we reuse the segmented rational cubic interpolation mentioned in the second case of (34) explained in Section 3.1.1. Otherwise, i.e. when $f^* > f(y_r)$, we employ

a rational cubic approximation of the inverse of a quadratic transformation of $f(y)$ between the maximum and y_r. Specifically, we define

$$w := y - \mathrm{y}_{\max} \quad r_2 := \sqrt{\tfrac{1}{2}|\alpha(\mathrm{y}_{\max} - \alpha)|} \quad h := \sqrt{\mathrm{f}_{\max} - f}/r_2 \tag{47}$$

and consider the implicit function

$$w = w(h). \tag{48}$$

We interpolate between $\mathrm{y}_{\max}$ and y_r which translates in (h, w) coordinates to the interpolation of $w(h)$ between

$$h_l := 0 \quad \text{and} \quad h_r := \sqrt{\mathrm{f}_{\max} - f(y_r)}/r_2. \tag{49}$$

We have

$$w(h_l) = 0 \quad w'(h_l) = 1 \tag{50}$$

$$w(h_r) = y_r - \mathrm{y}_{\max} \quad w'(h_r) = -2r_2 \frac{\sqrt{\mathrm{f}_{\max} - f(y_r)}}{f'(y_r)} \tag{51}$$

and

$$w''(h_r) = -2\frac{r_2^2}{f'(y_r)}\left(1 + 2 \cdot [\mathrm{f}_{\max} - f(y_r)]\frac{f''(y_r)}{f'(y_r)^2}\right). \tag{52}$$

We then set from our Delbourgo–Gregory interpolation in $w(h)$ over h, configured to match the second derivative at h_r as given in (A7),

$$y_0 = \mathrm{y}_{\max} + g_r^{\mathrm{rc}}(h^*, h_l, h_r, w(h_l), w(h_r), w'(h_l), w'(h_r), w''(h_r)) \tag{53}$$

with

$$h^* := \sqrt{\mathrm{f}_{\max} - f^*}/r_2. \tag{54}$$

3.1.3. Positive alpha: $\alpha \geq 0.1388$. Here, we also use the transformation from (y,f) to (w,h) via (47). We expand $h(w)$ as a Taylor series to third order around $y = \mathrm{y}_{\max}$, i.e. $w = 0$. Since $f'(\mathrm{y}_{\max}) = 0$, this expansion is linear in w in leading order. Using the Lagrange inversion theorem, we invert this expansion as $w = w(h)$, now linear in h in leading order. Since the asymptotic behaviour of $f(y)$ for $y \to \infty$ is quadratic in y, which means that y is asymptotically linear in $\sqrt{\mathrm{f}_{\max} - f}$ for $f \to -\infty$, we recast the inverse (Taylor) expansion as a Padé(2,1) form that is asymptotically linear in h for $h \to \infty$. This results in the approximation

$$w_{(2,1)}(h) = h \cdot \frac{1 + h\left(\frac{3}{4}r_3 - \frac{r_4}{r_3}\right)}{1 + h\left(\frac{5}{4}r_3 - \frac{r_4}{r_3}\right)} \tag{55}$$

with

$$r_3 = \frac{1 + \alpha(3\mathrm{y}_{\max} - 2\alpha) - \mathrm{y}_{\max}^2}{6(\mathrm{y}_{\max} - \alpha)} \tag{56}$$

$$r_4 = \frac{\alpha(4 + \alpha(12\mathrm{y}_{\max} - 6\alpha) - 7\mathrm{y}_{\max}^2) + \mathrm{y}_{\max}(\mathrm{y}_{\max}^2 - 3)}{24(\mathrm{y}_{\max} - \alpha)}. \tag{57}$$

When $\mathrm{y}_{\max} \geq 0$, or when f^* is negative and significantly larger in absolute value than $\mathrm{f}_{\max}$, we use this directly and set

$$y_0(f^*) = \mathrm{y}_{\max} + w_{(2,1)}(h^*) \tag{58}$$

with

$$h^* := \sqrt{\mathrm{f}_{\max} - f^*}/r_2. \tag{59}$$

When $\mathrm{y}_{\max} < 0$, however, we would like the initial guess function $y_0(\cdot)$ to go *exactly* through the origin which is not satisfied by (58):

$$\mathrm{y}_{\max} + w_{(2,1)}(h_r) \neq 0 \tag{60}$$

where we have reused the definition (49), i.e., when $f^* = 0$, we have $h^* = h_r$. In order to compensate for this, we subtract the error term, but scaled quadratically in h/h^*, or linearly in $f/\mathrm{f}_{\max}$, such that it does not destroy the local lowest order behaviour near the maximum, and attenuated quadratically for large (negative) f^*, so that it mainly corrects the behaviour near $f^* \approx 0$. In a formula, we set

$$y_0(f^*) = \left[\mathrm{y}_{\max} + w_{(2,1)}(h^*)\right] - \left[\mathrm{y}_{\max} + w_{(2,1)}(h_r)\right] \cdot \frac{1 - \frac{f^*}{\mathrm{f}_{\max}}}{1 + \left(\frac{f^*}{\mathrm{f}_{\max}}\right)^2} \tag{61}$$

when $\mathrm{y}_{\max} < 0$ and $f^* > -10\mathrm{f}_{\max}$ (recall that $\mathrm{f}_{\max} > 0$). We can see on the right-hand side of (61) that the subtracted term is zero at the maximum, i.e. when $f^* = \mathrm{f}_{\max}$, and is equal to the first term when $f^* = 0$, thus forcing the initial guess function exactly through the origin. In order to avoid subtractive cancellation errors, the implementation must, however, use the slightly more opaque contracted version

$$y_0(f^*) = \frac{f^*[\mathrm{f}_{\max}(\mathrm{y}_{\max} + w_r) + f^*(\mathrm{y}_{\max} + w^*)] + \mathrm{f}_{\max}^2(w^* - w_r)}{\mathrm{f}_{\max}^2 + f^{*2}} \tag{62}$$

with

$$w^* := w_{(2,1)}(h^*) \quad w_r := w_{(2,1)}(h_r). \tag{63}$$

REMARK 3.2 We emphasize that the practical usage statistics of this branch where the calculation of $\mathrm{y}_{\max}$ via (21) is actually invoked is for fewer than 10% of all cases whence, whilst it is possible to find even more efficient numerical implementations for $\mathrm{erfcx}^{-1}(\cdot)$ than given in Jäckel (2014, Appendix C), we shall not dwell on this possible optimization in this context since it would net provide only negligible, if any, performance improvements.

3.2. Asymptotic quadratic pre-polishing when $f < 0$ and $y_0 > 5$

When the target value of y is positive and of at least moderate magnitude, say $y > 5$, we can make use of the asymptotic behaviour of $f(\cdot)$ for $y \to +\infty$. In practice, of course, we don't know the target value of y yet. However, given the result y_0 of the initial guess calculation in the previous section, we can compute an expansion of $f(y_0 \cdot (1 + \varepsilon))$ to second order

in ε with $f(\cdot)$ given by (19). This second-order expansion is a quadratic form which enables us to solve

$$f(y_0 \cdot (1+\varepsilon)) \approx f^* \tag{64}$$

for ε to second order to obtain

$$\varepsilon = -b/(2a) + \sqrt{b^2/(4a^2) - c/a} \tag{65}$$

with

$$a = \frac{1}{2}(1-y_0^2), \quad b = \alpha \cdot y_0 - y_0^2 - 1, \quad \text{and}$$
$$c = \alpha \cdot y_0 - \frac{1}{2} \cdot y_0^2 - \ln(y_0) + \ln\left(\sqrt{\frac{2}{\pi}}\right) - f^*. \tag{66}$$

In short, when $f < 0$ and $y_0 > 5$, we improve y_0 by replacing it according to

$$y_0 \leftarrow y_0 \cdot (1+\varepsilon). \tag{67}$$

3.3. *Householder(6) improvement*

The sixth-order Householder method, which is of seventh-order convergence, to find the value for y that solves $f(y) = f^*$ can be expressed as

$$y_{n+1} = y_n + \frac{A_6}{B_6} \tag{68}$$

with

$$\nu = \frac{f^* - f(y_n)}{f'(y_n)} \quad h_k = \frac{f^{(k)}(y_n)}{f'(y_n)}. \tag{69}$$

and

$$A_6 = \nu\left(1 + \nu\left(2h_2 + \nu\left(\frac{3h_2^2}{4} + \frac{h_3}{2} + \nu\left(\frac{h_2h_3}{6} + \frac{h_4}{12} + \frac{h_5\nu}{120}\right)\right)\right)\right) \tag{70}$$

$$B_6 = 1 + \nu\left(\frac{5h_2}{2} + \nu\left(\frac{3h_2^2}{2} + \frac{2h_3}{3} + \nu\left(\frac{h_2^2}{8} + \frac{h_2h_3}{2} + \frac{h_4}{8} + \nu\left(\frac{h_3^2}{36} + \frac{h_2h_4}{24} + \frac{h_5}{60} + \frac{h_6\nu}{720}\right)\right)\right)\right). \tag{71}$$

Given the initial guess function y_0 from the previous section (including any quadratic pre-polishing where applicable), we found that *just a single Householder(6) improvement step* already gives us effectively the full attainable precision on standard IEEE 64-bit floating point hardware. Note that the actual accuracy attained depends on the accuracy of the functions $\varphi(y)$ and $\Phi(y)$, the round-off settings of the floating point unit, etc. In addition, straightforward error propagation analysis tells us that a relative uncertainty† of ϵ on a given input number x, when mapped through some function $g(x)$, translates into the relative uncertainty

$$\max\left(\left|\frac{g(x+\frac{\epsilon}{2}) - g(x-\frac{\epsilon}{2})}{g(x)}\right|, \epsilon\right) = \max\left(\left|\frac{xg'(x)}{g(x)}\right|, 1\right) \cdot \epsilon. \tag{72}$$

When $g(\cdot)$ is the inverse of some other function, say $g(y) = f^{-1}(y)$, then the relative (theoretically) attainable accuracy limit becomes

$$\max\left(\left|\frac{f(y)}{yf'(y)}\right|, 1\right) \cdot \epsilon. \tag{73}$$

4. Numerical examples

We show in figures 3–8 the resulting relative accuracy (on a logarithmic scale) for a range of values for α, with 2048 samples for each line or point set. Note that the meaningful range for y is arguably really only $y \in [-8.2535, 8.2535]$ since below that $\Phi(-y)$ can numerically not be distinguished from 1 because, then,

$$1 - \Phi(y) < \frac{\ln(2)}{2} \cdot \text{DBL\_EPSILON},$$

and above that

$$\Phi(-y) < \frac{\ln(2)}{2} \cdot \text{DBL\_EPSILON}$$

and thus the associated out-of-the-money deltas are of a similarly tiny magnitude. For the practical application of computing a strike from a volatility and a delta-with premium, however, the range $y \in [-10, 10]$ should definitely suffice.

It is clear from the shown diagrams that, for all intents and purposes, *a single Householder(6) improvement step* suffices to achieve full attainable accuracy. Either, the first HH(6)-improved value is already below the theoretical accuracy limit (70),‡ or further iteration gives no additional systematic improvement for reasons beyond our lowest order error propagation analysis, but even then, the attained accuracy is always very close to the theoretical limit. Hence, a single Householder(6) improvement step is always enough.

To test the speed of our analytics, we implemented the methodology in standalone reference code in C++. We extracted a set of (F, σ, Δ'_F) data from a real valuation system batch run, resulting in 5918 real-world input numbers. For good timing accuracy, we re-evaluated the calculation of the strike from delta + volatility just over 10 000 times, giving us about 60 million evaluations of our reference implementation of the analytical solution, and of the standard approach using Brent's root finding method (routine 'zbrent' in Press *et al.* (1992), modified to aim for DBL_EPSILON accuracy). We first confirmed that we get virtually the same relative error distribution from both our analytical and the iterative methodology: most residuals were identical, and for a nearly equal number of instances our analytics returned 0 relative error whereas a small error was attained with the Brent method, and vice versa. The timing was done on an Intel Xeon W-2123 CPU [nominal 3.60 GHz] running Windows 10, and the C++ code was compiled with Microsoft Visual Studio 9.0 (2008). We show in table 1 the resulting average single evaluation times both for 32-bit and 64-bit binary code.§ We note that

† Recall that any number x on a machine with relative precision $\epsilon \equiv$ DBL_EPSILON actually represents all *mathematical* numbers in the range $x \cdot (1 \pm \epsilon/2)$.

‡ Which, incidentally, also captures the case of loss of significant digits when $\alpha \approx 0$ and $f^* \approx \ln(2)$.

§ Note that the compilation bitness only affects the CPU instruction set (32-bit set or 64-bit set). Both versions then compute values in 'double precision' (i.e. 64 bit) floating point number representation.

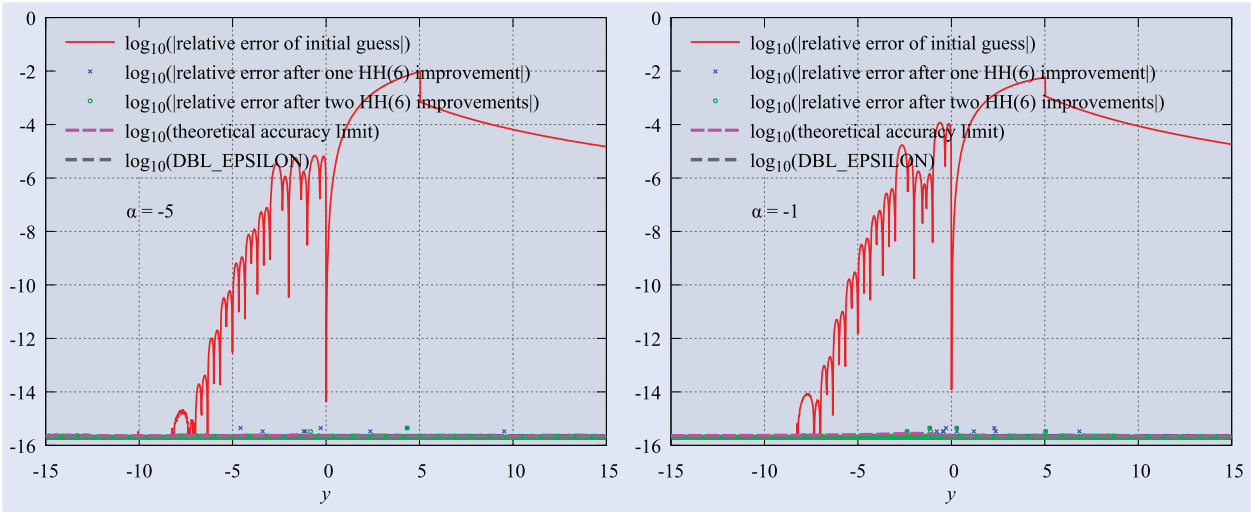

Figure 3. Relative accuracy for $\alpha = -5$ and $\alpha = -1$.

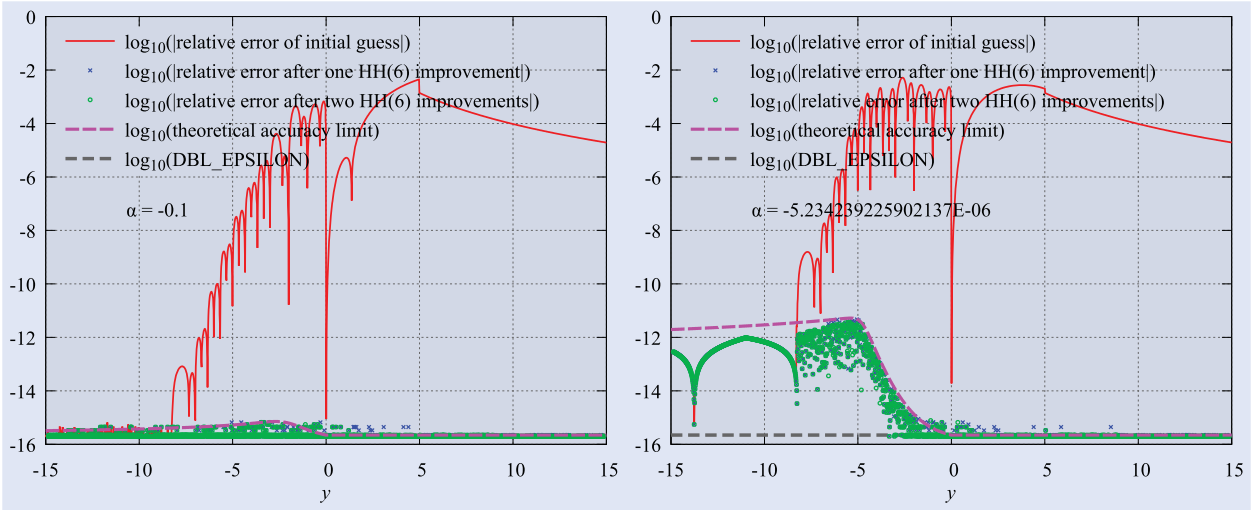

Figure 4. Relative accuracy for $\alpha = -0.1$ and $\alpha = -0.0001 \cdot \sqrt{\frac{1}{365}} = -5.234239225902137 \cdot 10^{-6}$.

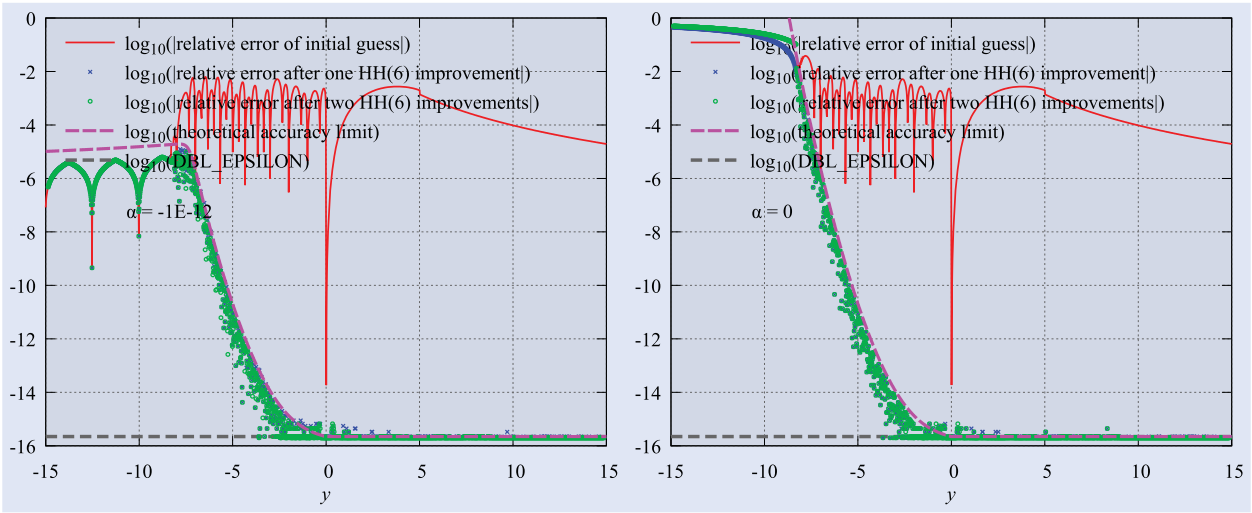

Figure 5. Relative accuracy for $\alpha = -10^{-12}$ and $\alpha = 0$.

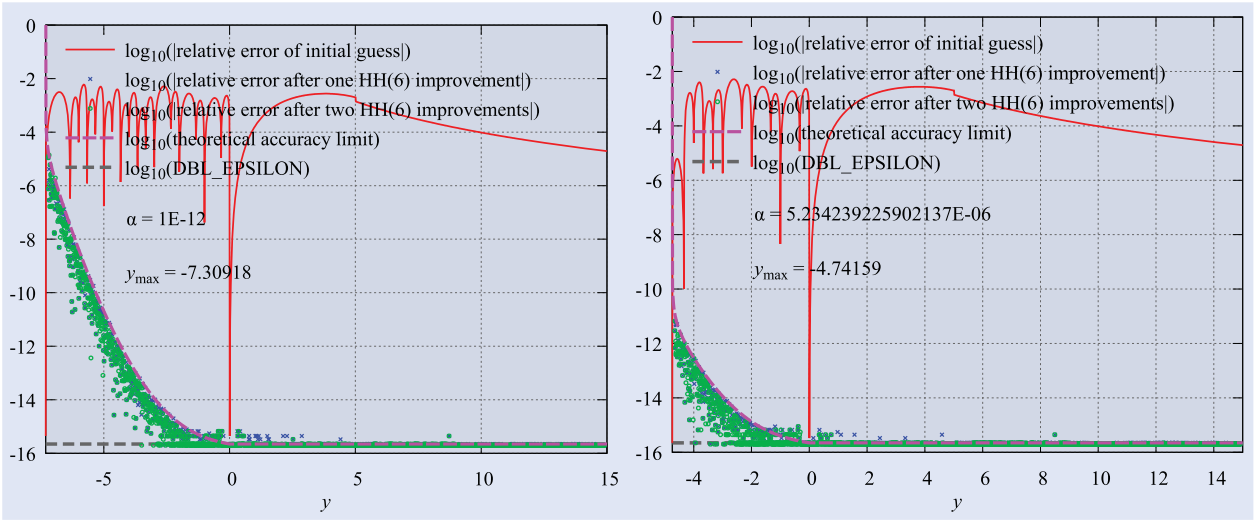

Figure 6. Relative accuracy for $\alpha = 10^{-8}$ and $\alpha = 0.0001 \cdot \sqrt{\frac{1}{365}} = 5.234239225902137 \cdot 10^{-6}$.

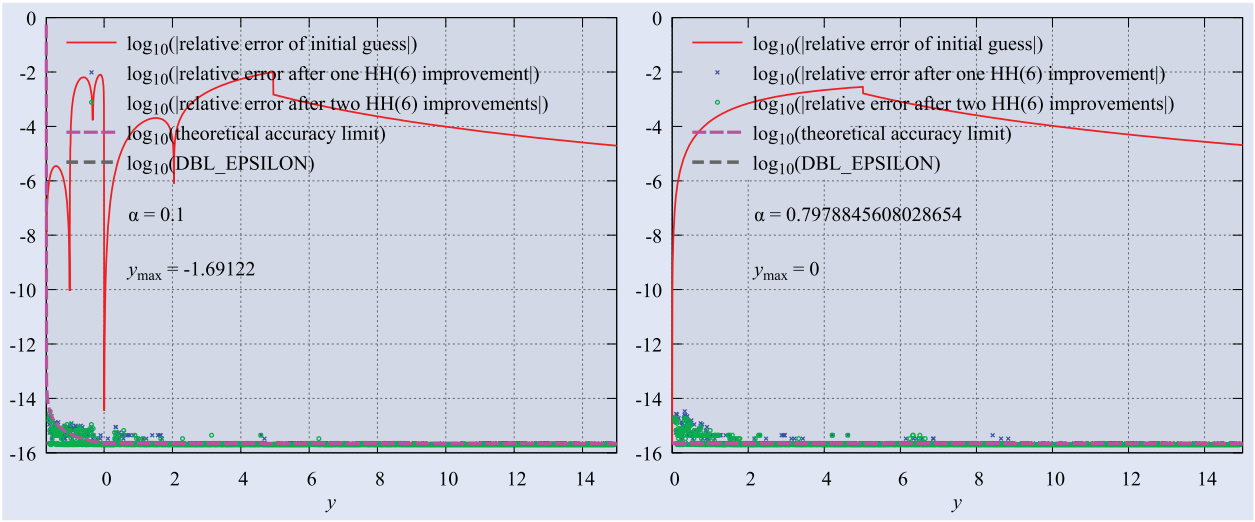

Figure 7. Relative accuracy for $\alpha = 0.1$ and $\alpha = \sqrt{2/\pi} = 0.7978845608028654$.

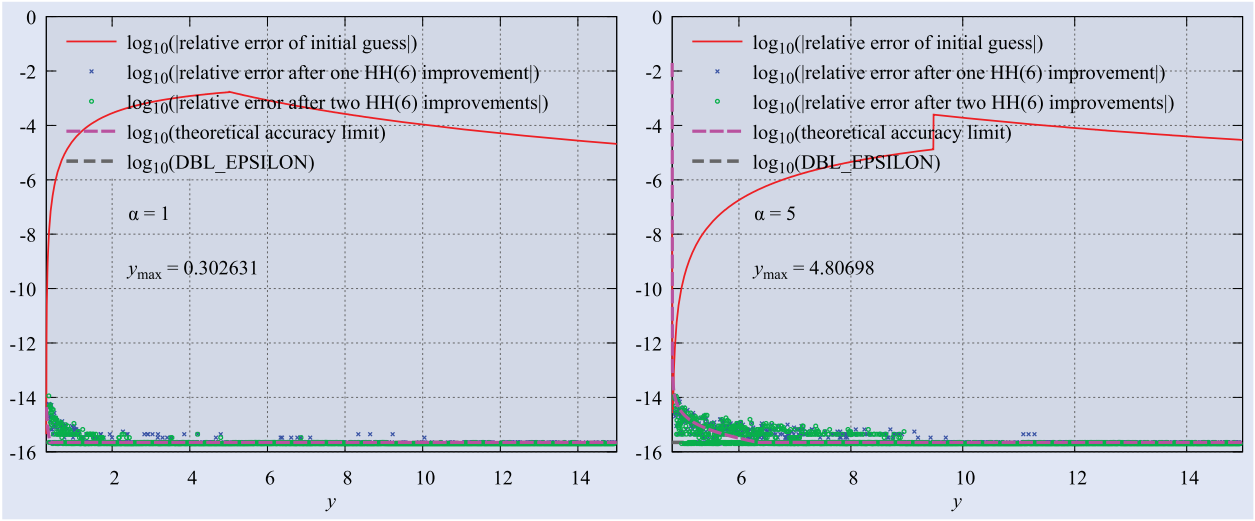

Figure 8. Relative accuracy for $\alpha = 1$ and $\alpha = 5$.

Table 1. Average calculation time in microseconds for a single strike-from-delta evaluation.

	32-bit binary code	64-bit binary code
Analytical inversion (µs)	0.30 ± 0.01	0.21 ± 0.01
Iterative (Brent) (µs)	2.16 ± 0.01	1.04 ± 0.01
Ratio	7.2 ± 0.2	4.95 ± 0.2

the analytical method is 5 to 7 times faster than the standard root finding approach. If the speed of our seemingly complicated analytics appears surprising, please consider that the computation of *any one* strike-from-delta only ever involves *one* of the various initial guess function branches, and all of those branches, *individually*, are not very complex at all. To put these figures into perspective, we submit to the reader the comparison that the calculation of *one* implied volatility from a given market price, which is a much harder problem, requires only about 1.1 µs in 32 bit and 0.6 µs in 64-bit code.† Our presented analytical method is about three times faster than the computation of a single implied volatility, which by complexity of the underlying problem, feels about right.

Acknowledgments

The author is grateful to Charles-Henri Roubinet, Head of Quantitative Research at VTB Capital, for authorizing the release of the presented material (originally from December 2013, with some improvements incorporated in September 2019) into the public domain.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Black, F., The pricing of commodity contracts. *J. Financ. Econ.*, 1976, **3**, 167–179.

Black, F. and Scholes, M., The pricing of options and corporate liabilities. *J. Politic. Econ.*, 1973, **81**, 637–654.

Cody, W.J., Rational Chebyshev approximations for the error function. *Math. Comput.*, 1969, **23**, 631–638.

Delbourgo, R. and Gregory, J.A., Shape preserving piecewise rational interpolation. *SIAM J. Sci. Stat. Comput.*, 1985, **6**(4), 967–976. Available online at: https://bura.brunel.ac.uk/bitstream/2438/2200/1/TR_10_83.pdf.

Householder, A.S., *The Numerical Treatment of a Single Nonlinear Equation*, 1970 (McGraw-Hill: New York).

Jäckel, P., Clamping down on arbitrage. *Wilmott Magazine*, May, 2014, pp. 54–69. First published in December 2013 at www.jaeckel.org.

Jäckel, P., Let's be rational. *Wilmott Magazine*, January, 2015a, pp. 40–53. First published in November 2013 at www.jaeckel.org.

Jäckel, P., Let's be rational reference source code, 2015b. Available online at: www.jaeckel.org.

† using the 'Let's Be Rational' (Jäckel 2015a, 2015b) algorithm on the same platform, etc.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., *Numerical Recipes in C*, 1992 (Cambridge University Press: Cambridge). Available online at: www.nrbook.com/a/bookcpdf.php.

Reiswich, D. and Wystup, U., A guide to FX options quoting conventions. *J. Deriv.*, November, 2010, **18**, 58–68. doi:10.3905/jod.2010.18.2.058.

Appendix. Delbourgo and Gregory's rational cubic form

Given an interval $[x_l, x_r]$, function values $g_l = g(x_l)$ and $g_r = g(x_r)$, and slope values $g'_l = g'(x_l)$ and $g'_r = g'(x_r)$, the rational cubic interpolation $g^{\mathrm{rc}}(x)$ reads

$$g^{\mathrm{rc}}(x; x_l, x_r, g_l, g_r, g'_l, g'_r, r) = \frac{g_r s^3 + (r g_r - h g'_r) s^2 (1-s) + (r g_l + h g'_l) s (1-s)^2 + g_l (1-s)^3}{1 + (r-3) s (1-s)} \tag{A1}$$

with

$$h := x_r - x_l, \quad \text{and} \quad s := (x - x_l)/h. \tag{A2}$$

The parameter r is a control parameter that can be chosen freely subject to $r > -1$, else the interpolation would incur a pole inside $[x_l, x_r]$. In the limit of $r \to \infty$, the rational cubic interpolation converges to a linear form. Delbourgo and Gregory (1985) also provide simple conditions for r such that the interpolation preserves monotonicity [their equation (39)] and convexity [their equation (49)], when the input data permit it. Conveniently, it is easy to configure r to meet a given second derivative of $g(\cdot)$ at either the left-hand side edge of the interpolation bracket as

$$r_l(x_l, x_r, g_l, g_r, g'_l, g'_r, g''_l) = \frac{\frac{1}{2} h g''_l + (g'_r - g'_l)}{\Delta_g - g'_l} \tag{A3}$$

or, respectively, at the right-hand side edge via

$$r_r(x_l, x_r, g_l, g_r, g'_l, g'_r, g''_r) = \frac{\frac{1}{2} h g''_r + (g'_r - g'_l)}{g'_r - \Delta_g} \tag{A4}$$

with $\Delta_g := (g_r - g_l)/h$. If we want to choose r in order to match an interior point (x, g), equation (A1) can be solved for r to obtain

$$r_i(x_l, x_r, g_l, g_r, g'_l, g'_r, x, g) = \frac{g \cdot (1 - 3s(1-s)) - g_r s^3 + h g'_r s^2 (1-s) - h g'_l s (1-s)^2 - g_l (1-s)^3}{(g_r s + g_l (1-s) - g) \cdot s \cdot (1-s)}. \tag{A5}$$

For the sake of brevity in the main text, we define

$$g^{\mathrm{rc}}_l(x; x_l, x_r, g_l, g_r, g'_l, g'_r, g''_l) := g^{\mathrm{rc}}(x; x_l, x_r, g_l, g_r, g'_l, g'_r, r_l(x_l, x_r, g_l, g_r, g'_l, g'_r, g''_l)) \tag{A6}$$

as the left-hand-side second-order matched interpolation,

$$g^{\mathrm{rc}}_r(x; x_l, x_r, g_l, g_r, g'_l, g'_r, g''_r) := g^{\mathrm{rc}}(x; x_l, x_r, g_l, g_r, g'_l, g'_r, r_r(x_l, x_r, g_l, g_r, g'_l, g'_r, g''_r)) \tag{A7}$$

as the right-hand-side second-order matched interpolation, and

$$g^{\mathrm{rc}}_i(x; x_l, x_r, g_l, g_r, g'_l, g'_r, x_i, g_i) := g^{\mathrm{rc}}(x; x_l, x_r, g_l, g_r, g'_l, g'_r, r_i(x_l, x_r, g_l, g_r, g'_l, g'_r, x_i, g_i)) \tag{A8}$$

as the interior-point matched rational cubic interpolation.