Let's be rational

Peter Jäckel[*]

Abstract

Improving on [Jäc06] and [Vog07], we show how Black's volatility can be implied from option prices with as little as two iterations to maximum attainable precision on standard (64 bit floating point) hardware for all possible inputs. The method is based on four rational function branches for the initial guess adapted to the log-moneyness x, two of which are combined with non-linear transformations of the input price, and the use of the convergence order four Householder method which comprises a rational function of the residual. Despite sounding difficult, the method is simple in practice, and a reference implementation is provided at [Jäc13]. As was perhaps previously underestimated, of crucial importance for the precision of the implied volatility is a highly accurate Black function that minimizes round-off errors and numerical truncations in the various parameter limits. We implement the Black call option price by the aid of Cody's [Cod69, Cod90] rational approximation for the complementary error function $\mathrm{erfc}(\cdot)$ and its little known cousin, the scaled complementary error function $\mathrm{erfcx}(\cdot)$. The source code of the reference implementation is available at www.jaeckel.org/LetsBeRational.7z.

1 Introduction

In [Jäc06], we provided a robust and comparatively efficient method to imply the volatility $\hat{\sigma}$ from the undiscounted price $\tilde{p}$ of an option via the Black formula

$$B(F, K, \hat{\sigma}, T, \theta)$$
$$= \theta \cdot \left[F \cdot \Phi \left(\theta \cdot \left[\frac{\ln(F/K)}{\hat{\sigma} \cdot \sqrt{T}} + \frac{\hat{\sigma} \cdot \sqrt{T}}{2} \right] \right) \right. \quad (1.1)$$
$$\left. - K \cdot \Phi \left(\theta \cdot \left[\frac{\ln(F/K)}{\hat{\sigma} \cdot \sqrt{T}} - \frac{\hat{\sigma} \cdot \sqrt{T}}{2} \right] \right) \right]$$

where $\theta = 1$ for call options and $\theta = -1$ for put options. The starting point was gaining a fundamental understanding of the difficulties involved, which lie largely with the fact that the Black function permits no Taylor expansion around $\hat{\sigma} = 0$ when $F \neq K$. Based on the asymptotics of the Black function for small and large values of $\hat{\sigma}$, the key components of the method published in "By Implication" were:-

- Separate the input price domain into a lower and an upper half at the point of inflexion of B over $\hat{\sigma}$.

- In the upper half, estimate an initial guess on a functional form that is essentially a linear rescaling of the asymptotics for large $\hat{\sigma}$.

- In the lower half, estimate an initial guess based on a geometric interpolation between the initial guess function for the upper half, and a functional form that, in a certain sense, dominates the asymptotics for small $\hat{\sigma}$.

- Define an objective function based on the price error in the upper half, and based on the reciprocal of the logarithm of the price in the lower half.

- Invoke an iteration procedure known as *Halley's method* of convergence order three on the respective objective function.

It was clear in [Jäc06] that this can, for very low input prices $\tilde{p}$, still lead to the requirement for a significant number of iterations[1]. It has since been pointed out [Vog07] that this can be improved upon by the aid of a different functional form for the asymptotics for low $\hat{\sigma}$. In this article, we will review and refine some of the choices made in [Jäc06] to arrive at an industrial solution that for standard IEEE 754 (53 bit mantissa) floating point hardware converges to the maximum attainable accuracy within two iterations for all possible inputs. In a nutshell, the new method can be summarized as follows:-

- Define four segments for the initial guess function, all of which use rational approximations.

- The highest and lowest segments are defined via non-linear transformations that ensure the *correct* asymptotic behaviour of the initial guess function to first order, not just *dominance*.

- Define three branches for the objective function, based on the reciprocal of the logarithm of the price for the lower branch, the price itself in the middle branch, and the logarithm of the distance of the price from its limit value for infinite volatility for the upper branch.

---

[*]OTC Analytics

[1]E.g., see that in figure 6 in [Jäc06] the number of iterations for a relative accuracy of 10^{-8} in implied volatility goes as high as 10.

- Use two iterations of the third order Householder iteration method which is a rational function of the objective residual and has convergence order four [Hou70].

In order to avoid any misunderstandings, we state at this point the purpose of this communication. The aim of the method presented here is *not* to provide 15 digits of accuracy of implied volatility for trading purposes, or for the sake of gaining an intellectual understanding of the relationship between volatility and option price. This would of course be ridiculous. For the latter, the publications by Brenner and Subrahmanyam [BS88] and Corrado and Miller [CM96, CM04] are excellent resources. Indeed, Corrado and Miller themselves emphasize that that is the aim of their own publication, and that for industrial applications, numerical solutions should be employed. The purpose of this communication, instead, is to satisfy that industrial need. One of the main reasons is that the Black formula has become an integral part of many analytical representations of other models and approximations, and is part of a range of analytical transformations. In those applications, the Black formula can end up being used with input parameters that, per se, in the context of a trading desk's purposes, would *never* be encountered, and, all this may yet be combined with numerical calibration routines which may end up exploring even more extreme input parameters. And of course it isn't just about the mapping from volatility to option prices, but the reverse, too, is needed. For the sake of brevity, we name but three such analytical mapping situations: the representation of 1) CEV volatility, 2) displaced diffusion volatility, and 3) dividend model process volatility as a Black implied volatility smile, especially, for short maturities. Also, in some applications, *local volatility* is numerically computed from implied volatility and its derivatives up to second order[2] via finite differencing by the aid of their analytical relationship, e.g., [BBF02, equation 15]. It is in these applied analytical calculations when practitioners really should be able to use the Black formula and its inverse to reproduce inputs close to within machine accuracy, just as we would demand for the exponential function and the natural logarithm, or for the sine and cosine functions and their inverses. What's more, the calculation of implied volatility may be part of analytical computations that reside within modules that are executed a great many number of times (e.g., in local volatility precomputations on a refined grid), and for that reason, may need to be very fast, in addition to accurate.

For the sake of at least partial completeness, we include a brief literature review. Li [Li06] gave a rational approximation for $|x| \le 1/2$ and $\sigma > |x|/2$ (though we were only informed of his work after having conducted the research presented here). This range of parameters is not even wide enough for normal trading desk purposes, and the accuracy

---

[2]Numerical differentiation inherently loses accuracy, requiring the underlying function to be of significantly higher precision. As a rule of thumb, if a function f has relative accuracy ϵ, then its numerical second order derivative f'' can only attain $\sqrt{\epsilon}$, i.e., half on a logarithmic scale.

of the approximations is only about 10^{-2}. Vogt [Vog07], as already mentioned, points out that the method described in our previous publication on the subject suffers from an increased required number of iterations when the strike is close to the forward and the input price is very low. Vogt gave an improved asymptotic guess for this parameter region based on a transformation to Lambert's W function, which preserves the correct asymptotic behaviour as the input price goes to zero. This was indeed the original starting point of the work presented here, only that we avoid the Lambert W function and instead express all transformations (used in aid of correct asymptotics) in terms of the cumulative normal function $\Phi(\cdot)$ and its inverse since we already have those as part of our standard financial analytics library. Grunspan [Gru11] demonstrates impressive stamina and gives higher order asymptotic expansions derived by the aid of the formal *transseries* framework, but we make no use of those results here.

2 Preliminaries

Instead of the standard Black function (1.1), we prefer to work with the normalization

$$x := \ln(F/K) \tag{2.1}$$

$$\sigma := \hat{\sigma}\sqrt{T} \tag{2.2}$$

$$b(x, \sigma, \theta) := B(F, K, \hat{\sigma}, T, \theta)/\sqrt{FK} \tag{2.3}$$

$$= \theta \cdot \left[e^{x/2} \cdot \Phi(\theta[x/\sigma + \sigma/2]) \right. \tag{2.4}$$

$$\left. - e^{-x/2} \cdot \Phi(\theta[x/\sigma - \sigma/2]) \right]$$

The normalized Black function (2.4) satisfies the "reciprocal-strike-put-call invariance"

$$b(x, \sigma, \theta) = b(-x, \sigma, -\theta) \tag{2.5}$$

and the "time-value-put-call invariance"

$$b(x, \sigma, \theta) - \iota(x, \theta) = b(x, \sigma, -\theta) - \iota(x, -\theta) \tag{2.6}$$

with

$$\iota(x, \theta) := \left(b_{max} - b_{max}^{-1} \right)_+ \tag{2.7}$$

and

$$b_{max} := e^{\theta x/2} . \tag{2.8}$$

From here on, we shall only deal with out-of-the-money call options, i.e., the case $\theta = +1$ and $x \le 0$ which is without loss of generality by virtue of the invariances (2.5) and (2.6). With this restriction, we have the bounds

$$0 \le b \le b_{max} \le 1 \tag{2.9}$$

with

$$b_{max}\Big|_{\theta=1} = e^{x/2} . \tag{2.10}$$

3 Asymptotics

In [Jäc06], we gave the asymptotic behaviour of b for small and large σ, which (for $\theta = +1$ and $x \leq 0$) are:-

$$\lim_{\sigma \to 0} b \approx x \cdot \varphi(x/\sigma) \cdot \left(\frac{\sigma}{x}\right)^3 \tag{3.1}$$

$$\lim_{\sigma \to \infty} b \approx b_{\max} - 4/\sigma \cdot \varphi(\sigma/2) \tag{3.2}$$

which can be derived using [AS84, (26.2.12)]. With the same formula, we can convert (3.1) and (3.2) to

$$\lim_{\sigma \to 0} b \approx \frac{2\pi|x|}{3\sqrt{3}} \cdot \Phi\left(\frac{-|x|}{\sqrt{3}\,\sigma}\right)^3 \tag{3.3}$$

$$\lim_{\sigma \to \infty} b \approx b_{\max} - 2\Phi(-\sigma/2) \tag{3.4}$$

which is accurate to asymptotic first order in σ and $1/\sigma$, respectively. Note that both of these expressions can be solved for σ, i.e., these are invertible asymptotics. Expression (3.4) was of course already given in [Jäc06]. Expression (3.3), however, is new. We mention that it is in spirit similar to the non-linear transformation to Lambert's W function in [Vog07] which we will not use here. The reason for this is our application: our aim is to have an industrially accurate and fast implementation. Since we already have a precise implementation of the cumulative normal and its inverse, and since the Lambert W function requires a good deal of attention in its own right if we wish to achieve high accuracy [Veb09], we prefer not to introduce yet another special function. What's more, the implementation of the Lambert W function, apart from being typically iterative itself, involves more exponentials or logarithms than the cumulative normal and its inverse, and that is something we definitely wish to avoid as we shall explain later. We point out that in [Jäc06] we only used a functional form for the limit $\sigma \to 0$ that would dominate the correct solution in a way that enabled us to obtain a viable initial guess for the subsequent root finding. In fact, this was precisely the reason that, for very low volatilities, the method there required an ever increasing number of iterations. Here, in contrast, formula (3.3) is asymptotically correct to first order, and hence, the limit of $\sigma \to 0$ is a case where our initial guess will become increasingly accurate, as we will see soon.

4 The initial guess in four branches

The normalized Black function (2.4) has a single point of inflexion at (σ_c, b_c) given by

$$\sigma_c = \sqrt{2|x|} \tag{4.1}$$
$$b_c = b(x, \sigma_c) \tag{4.2}$$

where we have dropped θ since we are only dealing with out-of-the-money call options, i.e., $\theta \equiv 1$. For $\sigma < \sigma_c$, $b(x < 0, \sigma)$ is convex, and for $\sigma > \sigma_c$ it is concave,

whence, having zero slope at both ends of its range, it is of sigmoid shape. In its central region, near σ_c, with that point being a turning point, it is comparatively linear. To take advantage of this near-linearity in the central section for an initial guess, we need to identify a lower and an upper limit of this as yet only vaguely defined central region. An obvious and easy choice is to draw a tangent through the point (σ_c, b_c), and let the location of the intersections of this tangent with the limit levels of b be the lower limit σ_l and the upper limit s_u, i.e.,

$$\sigma_l := \sigma_c - \frac{b_c}{b'(\sigma_c)} \tag{4.3}$$

$$\sigma_u := \sigma_c + \frac{(b_{\max} - b_c)}{b'(\sigma_c)} \tag{4.4}$$

with

$$b'(\sigma) := \frac{\mathrm{d}}{\mathrm{d}\sigma} b(x, \sigma) \tag{4.5}$$

$$= \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left[\left(\frac{x}{\sigma}\right)^2 + \left(\frac{\sigma}{2}\right)^2\right]} . \tag{4.6}$$

We denote the points on the curve $b(x, \sigma)$ over σ at the locations σ_l and σ_u as (σ_l, b_l) and (σ_u, b_u), i.e.,

$$b_l = b(x, \sigma_l) \tag{4.7}$$
$$b_u = b(x, \sigma_u) , \tag{4.8}$$

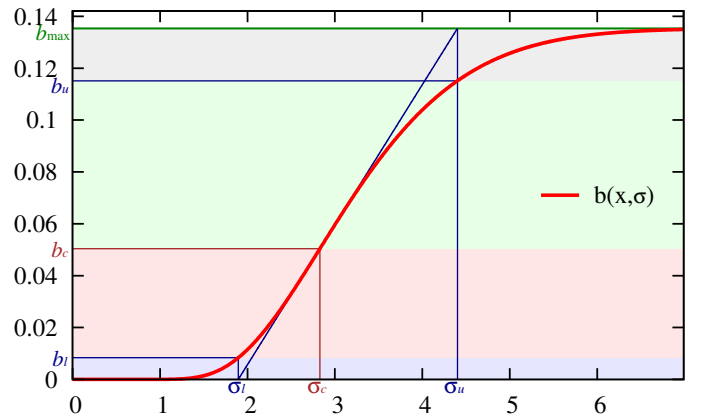as is shown in figure 1 for $|x| = 4$. We use the locations

Figure 1: The construction of the two branch locations σ_l and σ_u from the tangent at the inflexion point (σ_c, b_c) for $|x| = 4$.

σ_l, σ_c, and σ_u as three branching points to divide the range of $\sigma \in [0, \infty)$ into four segments. This translates into a segmentation for the initial guess function into the zones $[0, b_l)$, $[b_l, b_c]$, $(b_c, b_u]$, and $(b_u, b_{\max})$ which we show in figure 2 as a function of $|x|$ (as usual for $x \leq 0, \theta = +1$).

Having defined four distinct interpolation zones for the initial guess, we now proceed to its specification. We use the notation $\sigma(\beta)$ to represent the exact implied volatility that solves

$$b(x, \sigma) = \beta \tag{4.9}$$

for σ from a given normalized price β. Obviously, any input β must be in the range $[0, \beta_{\max})$ with $\beta_{\max} \equiv b_{\max}$. We denote $\sigma_0(\beta)$ as the initial guess function.
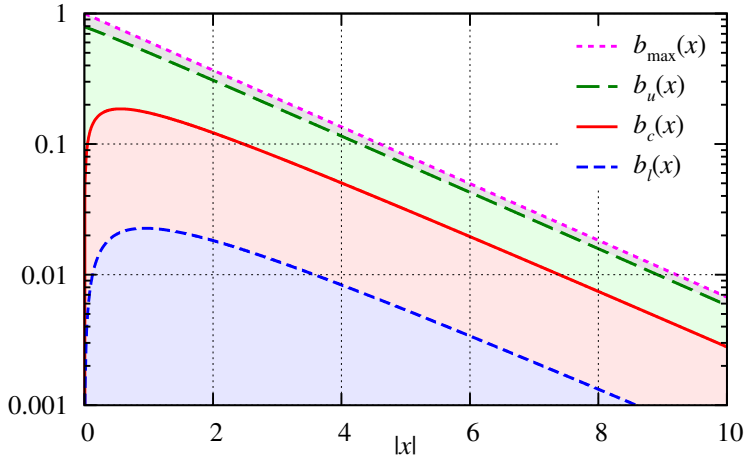
Figure 2: The four zones of the initial guess function.

In the two central regions $[b_l, b_c]$ and (b_c, b_u), we know that $\sigma(\beta)$ is linear to second order near b_c and only moderately curved at the outside ends of the two zones. Here, we take note of the extensive literature on the subject of function approximation. By far the most commonly used approach, especially for high-efficiency implementations, is to approximate the target function as a rational function, i.e., as the ratio of two polynomials, on carefully selected regions. Often, this is combined with non-linear transformations that are specific to each interpolation zone in order to match certain asymptotic features of the target function. In practical applications, this approach is behind the implementation of virtually any special function. In this context here, we mention three examples of particular relevance, namely, the cumulative normal function and its cousin the error function [Mar04, Cod69, Cod90, Mic93b], the inverse cumulative normal function [Wic88], and of course the Lambert W function [Veb09], though there are entire libraries of special functions based on rational approximations, e.g., [Mic93a]. Also, Halley's iteration method is ultimately based on a rational form as is its generalisation to higher order, the Householder method [Hou70]. For univariate functions, extensions of the Remez algorithm can be used to find rational approximations that are numerically effectively optimal in the sense of the *minimax* solution, and this is how most of the above mentioned rational approximations were computed. It is in principle possible to extend this to two dimensions when there is an extra dependency (as is the case with $\sigma(\beta)$ which also depends on x), though, the resulting formulae can readily involve a significant number of coefficients, rendering it more efficient to use a rational approximation for an initial guess and combine this with a very small number of iterations. We therefore make the rational choice (pun intended), and use a low order rational approximation to $\sigma(\beta)$ as our initial guess $\sigma_0(\beta)$ in the central sections. For this purpose, we use the rational cubic interpolation method of Delbourgo and Gregory [DG85] which we have grown to like as something of a supercharged swiss army knife when it comes to general purpose interpolation techniques.

Given an interval $[x_l, x_r]$, function values $f_l = f(x_l)$ and $f_r = f(x_r)$, and slope values $f_l' = f'(x_l)$ and $f_r' = f'(x_r)$,

the rational cubic interpolation $f^{rc}(x)$ reads

$$f^{rc}(x; x_l, x_r, f_l, f_r, f_l', f_r', r) = \qquad (4.10)$$

$$\frac{f_r s^3 + (r f_r - h f_r') s^2 (1-s) + (r f_l + h f_l') s (1-s)^2 + f_l (1-s)^3}{1 + (r-3) s (1-s)}$$

with

$$h := x_r - x_l, \quad \text{and} \quad s := (x - x_l)/h . \qquad (4.11)$$

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 [DG85] also provide simple conditions for r such that the interpolation preserves monotonicity [their equation (3.8)] and convexity [their equation (3.18)], when the input data permit it. Conveniently, it is easy to configure r to meet a given second derivative of $f(\cdot)$ at either the left hand side edge of the interpolation bracket as

$$r_l(x_l, x_r, f_l, f_r, f_l', f_r', f_l'') = \frac{\frac{1}{2} h f_l'' + (f_r' - f_l')}{\Delta - f_l'} \qquad (4.12)$$

or, respectively, at the right hand side edge via

$$r_r(x_l, x_r, f_l, f_r, f_l', f_r', f_r'') = \frac{\frac{1}{2} h f_r'' + (f_r' - f_l')}{f_r' - \Delta} \qquad (4.13)$$

with $\Delta := (f_r - f_l)/h$. We choose the parameter r such that, on both the centre left and the centre right segment, respectively, we obtain a rational interpolation form that matches the second derivative of $\sigma(\beta)$ in the inflexion point b_c, subject to the aforementioned monotonicity and convexity restrictions. We compute the second derivative $\sigma''(\beta)$ from

$$\frac{\mathrm{d}}{\mathrm{d}\beta} \sigma(\beta) = \frac{1}{b'} \qquad (4.14)$$

$$\frac{\mathrm{d}^2}{\mathrm{d}\beta^2} \sigma(\beta) = \frac{\mathrm{d}}{\mathrm{d}\beta} \left(\frac{1}{b'} \right)$$

$$= \frac{\mathrm{d}}{\mathrm{d}\sigma} \left(\frac{1}{b'} \right) \cdot \frac{\mathrm{d}}{\mathrm{d}\beta} \sigma(\beta)$$

$$= -\frac{b''}{b'^3} \qquad (4.15)$$

whence

$$\sigma''(\beta)\big|_{\beta=b_c} = -\frac{b''(\sigma_c)}{b'(\sigma_c)^3} = 0 \qquad (4.16)$$

due to $b''(\sigma_c) \equiv 0$. This gives us the initial guess function in the centre left region

$$\sigma_0(\beta)\big|_{\beta \in [b_l, b_c]} = f_{cl}^{rc}(\beta) \qquad (4.17)$$

with

$$f_{cl}^{rc}(\beta) = f^{rc}(\beta; b_l, b_c, \sigma_l, \sigma_r, 1/b_l', 1/b_c', r_{[b_l, b_c]}) \qquad (4.18)$$

and

$$r_{[b_l, b_c]} = r_r(b_l, b_c, \sigma_l, \sigma_c, 1/b_l', 1/b_c', 0) , \qquad (4.19)$$

and in the centre right region

$$\sigma_0(\beta)\big|_{\beta \in (b_c, b_u]} = f_{cr}^{rc}(\beta) \qquad (4.20)$$

4

with

$$f_{cr}^{\text{rc}}(\beta) = f^{\text{rc}}(\beta; b_c, b_u, \sigma_c, \sigma_u, 1/b_c', 1/b_u', r_{(b_c, b_u]}) \quad (4.21)$$

and

$$r_{(b_c, b_u]} = r_l(b_c, b_u, \sigma_c, \sigma_u, 1/b_c', 1/b_u', 0) . \quad (4.22)$$

We note that the extra calculation effort required for the evaluation of the respective rational cubic interpolation formulae over and above what already has been computed is only

$$\begin{aligned} b'(\sigma_l) &\quad \text{when} \quad \beta \in [b_l, b_c] \\ b'(\sigma_u) &\quad \text{when} \quad \beta \in (b_c, b_u] . \end{aligned}$$

The evaluation of the rational cubic form itself is very little effort. Depending on the hardware and compiler, the CPU effort is little more than, or possibly the same as, the evaluation of a single vega expression $b'(\sigma)$.

In the upper region $\beta \in (b_u, b_{\max})$, we use the asymptotic formula (3.4) to define a non-linear transformation $f_u(\beta)$ of $\sigma(\beta)$ that is asymptotically linear in β when $\beta \to b_{\max}$:

$$f_u(\beta) := \Phi(-\sigma(\beta)/2) \quad (4.23)$$

We approximate this function by a rational cubic interpolation that matches its level, slope, and second derivative at the left edge of the interval, and its level and slope at the right edge, using

$$f_u'(\beta) = -\frac{1}{2} \cdot e^{\frac{1}{2}\frac{x^2}{\sigma^2}} \quad (4.24)$$

$$f_u''(\beta) = \sqrt{\frac{\pi}{2}} \cdot \frac{x^2}{\sigma^3} \cdot e^{\frac{x^2}{\sigma^2}+\frac{\sigma^2}{8}} \quad (4.25)$$

and

$$\lim_{\beta \to b_{\max}} f_u(\beta) = 0 \quad (4.26)$$

$$\lim_{\beta \to b_{\max}} f_u'(\beta) = -\frac{1}{2} \quad (4.27)$$

wherein $\sigma = \sigma(\beta)$. This gives us

$$f_u^{\text{rc}}(\beta) := f^{\text{rc}}(\beta; b_u, b_{\max}, f_u(b_u), 0, f_u'(b_u), -1/2, r_{(b_u, b_{\max})}) \quad (4.28)$$

with

$$r_{(b_u, b_{\max})} = r_l(b_u, b_{\max}, f_u(b_u), 0, f_u'(b_u), -1/2, f_u''(b_u)) . \quad (4.29)$$

The initial guess in the upper region is then composed by solving (4.23) for σ, and replacing f_u^{rc} for f_u:

$$\sigma_0(\beta)\Big|_{\beta \in (b_u, b_{\max})} = -2 \cdot \Phi^{-1}(f_u^{\text{rc}}(\beta)) . \quad (4.30)$$

This leaves us to define the initial guess function for the lower region $\beta \in [0, b_l)$. Here, we make use of the asymptotic form (3.3) to define the non-linear transformation

$$f_l(\beta) := \frac{2\pi|x|}{3\sqrt{3}} \cdot \Phi(z)^3 \quad \text{with} \quad z := \frac{-|x|}{\sqrt{3}\sigma(\beta)} \quad (4.31)$$

which is asymptotically linear in β when $\beta \to 0$. Continuing with our rational theme, we approximate this function, too, by the Delbourgo-Gregory interpolation, matching level and slope at either end of the region, and setting the control parameter r to match the second derivative at the right hand side edge. For this, we compute

$$f_l'(\beta) = 2\pi \cdot z^2 \cdot \Phi(z)^2 \cdot e^{z^2 + \frac{\sigma^2}{8}} \quad (4.32)$$

$$f_l''(\beta) = \frac{\pi}{6} \cdot \frac{z^2}{\sigma^3} \cdot \Phi(z) \cdot e^{2z^2 + \frac{\sigma^2}{4}} \cdot \quad (4.33)$$
$$\cdot \left(8\sqrt{3}\sigma|x| + (3\sigma^2(\sigma^2 - 8) - 8x^2) \cdot \frac{\Phi(z)}{\varphi(z)}\right)$$

and

$$\lim_{\beta \to 0} f_l(\beta) = 0 \quad (4.34)$$

$$\lim_{\beta \to 0} f_l'(\beta) = 1 \quad (4.35)$$

wherein, as before $z = -|x|/\sqrt{3}\sigma$ and $\sigma = \sigma(\beta)$, to obtain

$$f_l^{\text{rc}}(\beta) := f^{\text{rc}}(\beta; 0, b_l, 0, f_l(b_l), 1, f_l'(b_l), r_{[0, b_l)}) \quad (4.36)$$

with

$$r_{[0, b_l)} = r_r(0, b_l, 0, f_l(b_l), 1, f_l'(b_l)f_l''(b_l)) . \quad (4.37)$$

The initial guess in the lower region is then the result of solving (4.31) for σ, and replacing f_l^{rc} for f_l:

$$\sigma_0(\beta)\Big|_{\beta \in [0, b_l)} = \left| \frac{x}{\sqrt{3}} \left[\Phi^{-1}\left(\sqrt{3} \cdot \sqrt[3]{f_l^{\text{rc}}(\beta)/2\pi|x|}\right) \right]^{-1} \right| . \quad (4.38)$$

The net function $\sigma_0(\beta)$ divides into four branches:

$$\sigma_0(\beta) = \begin{cases} \text{see expression (4.38)} & \text{for} \quad \beta \in [0, b_l) \\ f_{cl}^{\text{rc}}(\beta) & \text{for} \quad \beta \in [b_l, b_c] \\ f_{cr}^{\text{rc}}(\beta) & \text{for} \quad \beta \in (b_l, b_u] \\ -2 \cdot \Phi^{-1}(f_u^{\text{rc}}(\beta)) & \text{for} \quad \beta \in (b_u, b_{\max}) \end{cases} \quad (4.39)$$

Overall, it is of class C^1 with $\sigma_0''(\beta)$ being discontinuous at $\beta = b_l$ and $\beta = b_u$, and $\sigma_0'''(\beta)$ being discontinuous at $\beta = b_c$. We show examples of $\sigma_0(\beta)$ for four different values of x in figure 3. The quality of the approximation speaks for itself.

5 Rational iteration

Having established the initial guess function $\sigma_0(\beta)$, with β being the normalized input price, we now determine the iteration procedure to obtain an accurate implied volatility figure. To specify the iteration, we need to choose a) an objective function, and b) the iteration functional.
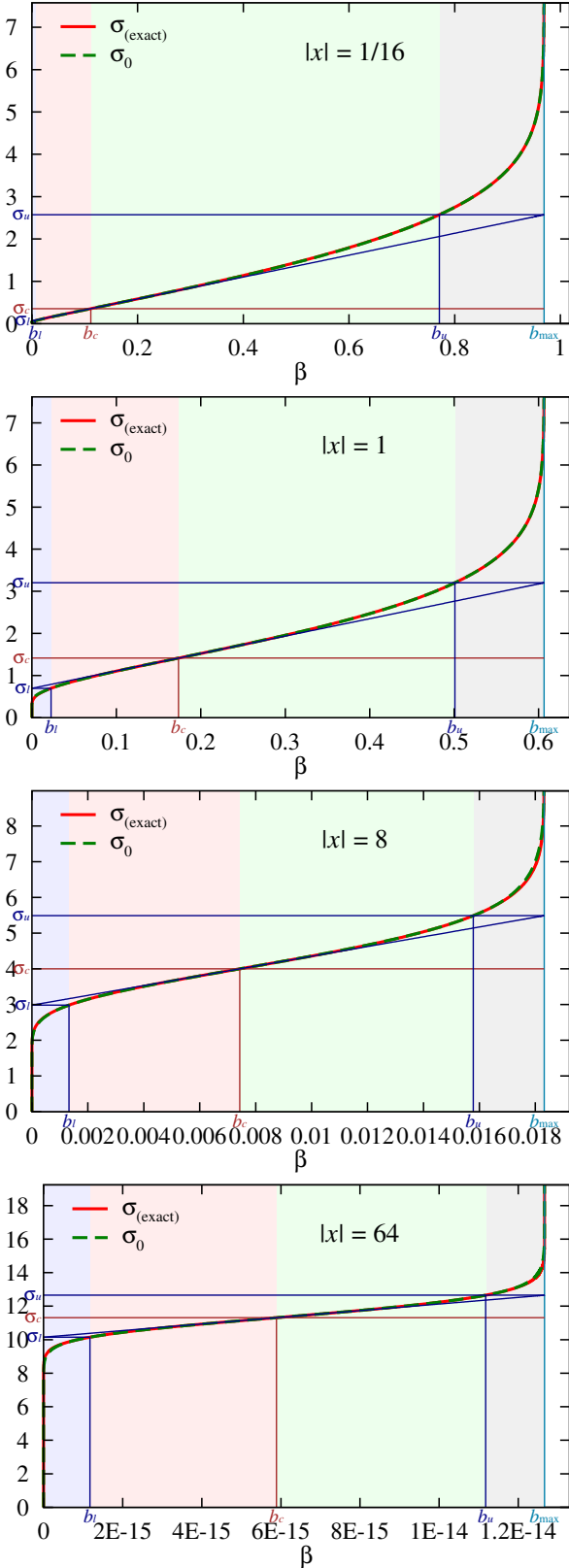
Figure 3: Four examples of the initial guess function $\sigma_0(\beta)$.

Assuming as before that we have used the invariances (2.5) and (2.6) to transform to the case of $x \leq 0$ and $\theta = 1$, we define the objective function in three branches according to

$$
g(\sigma) = \begin{cases} \frac{1}{\ln(b(\sigma))} - \frac{1}{\ln(\beta)} & \text{for} \quad \beta \in [0, b_l) \\ b(\sigma) - \beta & \text{for} \quad \beta \in [b_l, \tilde{b}_u] \\ \ln\left(\frac{b_{\max} - \beta}{b_{\max} - b(\sigma)}\right) & \text{for} \quad \beta \in (\tilde{b}_u, b_{\max}) \end{cases} \quad (5.1)
$$

with

$$
\tilde{b}_u := \max(b_u, {}^{b_{\max}}\!/_2) \quad (5.2)
$$

where we have again suppressed the dependence of the normalized Black function b on x. The respective transformations in (5.1) have been chosen to improve the convergence of the respective Lagrange inversion series of $g(\sigma)$. In common parlance, this means that we chose the objective function branches in order to make the inverse of the objective function well approximated by a low order local rational approximation.

In order to find the sought implied volatility, we need to locate the the root of $g(\sigma)$. For this, we use an iterative procedure. Whilst most practitioners are familiar primarily with the Newton-Raphson method, there are in fact quite a few generic techniques for this purpose in the literature. In [Jäc06], we used Halley's method which consists of a rational function of order $(1, 1)$ of the residual $g(\sigma)$, i.e., it can be written as the ratio of a polynomial of first order in g divided by another polynomial of first order in g.

Other authors have suggested the use of the Chebyshev method which is a second order polynomial in g, which means the iteration would have the form

$$
\sigma_{n+1}^{\text{Cheby}} = \alpha_n^{\text{Cheby}} + \gamma_n^{\text{Cheby}} g_n + \delta_n^{\text{Cheby}} g_n^2 \quad (5.3)
$$

with $g_n := g(\sigma_n)$ and all coefficients being functions of σ_n that, generically, do not become small or infinite as $g \to 0$ (whence they do not affect convergence considerations). In comparison, Halley's method takes on the form

$$
\sigma_{n+1}^{\text{Halley}} = \frac{\alpha_n^{\text{Halley}} + \gamma_n^{\text{Halley}} g_n}{1 + \delta_n^{\text{Halley}} g_n} \ . \quad (5.4)
$$

In aid of clarification, we mention that both Halley's and Chebyshev's method are of the same convergence order, i.e., order three, and that, in fact, Chebyshev's method is identical to a second order Taylor expansion of Halley's method, and, in turn, Halley's method is identical to the Padé(1,1) approximant (which is a kind of rational function expansion of the same convergence order) of Chebyshev's method. The reason we chose Halley's method, and not Chebyshev's method in [Jäc06] was that, in general, rational function approximations tend to be more flexible and are overall preferred[3], though as for the convergence order there is of course no difference whatsoever.

When it comes to the choice of an iteration procedure for the purpose of high accuracy solutions, we have the choice between either going for a higher convergence order, hoping to need fewer iterations, or to save the effort to compute

---

[3]To highlight this point we quote from [PTVF92] on the subject of (rational) Padé approximants: "It is sometimes quite mysterious how well this can work" and "Padé approximation has the uncanny knack of picking the function *you had in mind* from among all the possibilities.". This is followed by some caveats that we need not worry about here since we choose our objective function to be amenable to rational approximation, and since we are guaranteed to be close to the solution by the excellent quality of our initial guess, thus receiving the full benefit of the local rational approximation of the inverse function.

the extra coefficients, and carry out more iterations. As a rule of thumb, when the effort to compute derivatives of the objective function is higher than the evaluation of the objective function itself, it is advisable to use a lower order method and iterate more, else, use a higher order method with fewer iterations. In our context, the higher order derivatives of the objective function are all easier to compute than the objective function itself. This is ultimately because $\Phi(\cdot)$ is more effort to evaluate than $\varphi(\cdot)$. A generic iteration procedure of arbitrary order d is *Householder's method* [Hou70] given by

$$\sigma_{n+1} \;=\; \sigma_n + d \cdot \frac{(1/g)^{(d-1)}(\sigma_n)}{(1/g)^{(d)}(\sigma_n)}\,. \qquad (5.5)$$

The first and second order versions are identical to the Newton-Raphson and Halley's method, respectively. We have chosen to use the third order method

$$\sigma_{n+1} \;=\; \sigma_n + \nu_n \cdot \frac{1 + \frac{1}{2}\gamma_n\nu_n}{1 + \nu_n\left(\gamma_n + \frac{1}{6}\delta_n\nu_n\right)} \qquad (5.6)$$

with

$$\nu_n := -\frac{g(\sigma_n)}{g'(\sigma_n)}\,,\quad \gamma_n := \frac{g''(\sigma_n)}{g'(\sigma_n)}\,,\quad \delta_n := \frac{g'''(\sigma_n)}{g'(\sigma_n)}\,, \quad (5.7)$$

which, somewhat confusingly, whilst being the *third* order Householder method, is of *fourth* order convergence in the residual error. We remark that the third order Householder method is a rational function of order $(2,2)$ in the residual g. We spare the reader the listing of all the involved terms of the third order Householder method for all three branches of $g(\sigma)$ but mention that they are explicitly given in the source code comments of the reference implementation available at www.jaeckel.org/LetsBeRational.7z [Jäc13]. Preempting our numerical results later on somewhat, we mention that the combination of our four-branch initial guess function, with our three-branch objective function, and the third order Householder method enables us to attain the maximum achievable accuracy on standard IEEE 754 (53 bit mantissa) floating point hardware *with exactly two iterations* for all possible input values. The subtle point here is the *maximum achievable accuracy* which surprisingly strongly depends on the implementation of the (normalized) Black function that we use in our iteration, as we shall discuss in the next section.

Remark 5.1. The reason for the choice of the third order method is the balancing of comparative efforts. With a second order method (e.g., Halley's), we would often need three iterations to reach maximum accuracy. On the other hand, to obtain full attainable precision with a single iteration we would either need to go to at least 14th order (i.e., 15th order of convergence), or improve our initial guess by at least two decimal orders of magnitude in its weakest points which invariably would be numerically more effort than an additional iteration of the third order Householder method. As a compromise, we have settled for the initial guess function presented in section 4 and combined it with two iterations of the third order Householder method.

6 The Black function

Irrespective of any transformations we may choose in our target objective function whose root will be our sought implied volatility number, such as those given in (5.1), we inevitably need to evaluate the Black function which is conventionally implemented directly in the form in which it is written. In our case, for the normalized Black function (2.4) with $x \leq 0$ and $\theta = 1$, this means we take the numerical difference of two exponentially weighted cumulative normal functions:

$$b \;=\; \Phi(h+t) \cdot \phi \;-\; \Phi(h-t)/\phi\,, \qquad (6.1)$$

with

$$\phi = \mathrm{e}^{x/2}\,,\quad h := x/\sigma\,,\ \text{and}\ t := \sigma/2\,. \qquad (6.2)$$

When both $0 < |x| \ll 1$ and $\sigma \ll 1$, as is the case for almost all options that are near the money, this means that we have $\phi \approx 1$, and the numerical value of b is dominated by the result of the subtraction of two cumulative normal function values of nearby arguments, centred around $\Phi(h) \approx 1/2$. This poses one of the most common and standard problems of error propagation in numerical analysis: the divergence of the relative error of a function defined as a difference, also known as *Subtractive Cancellation*. The error analysis of this case, to first order, is as follows. First, denote by ε_i a real-valued number that is randomly[4] somewhere in the range $[-\epsilon, \epsilon]$ where ϵ is defined as the IEEE 64 bit constant DBL_EPSILON. Since all numerical evaluations are only accurate to within ϵ on a relative scale, when numerically evaluated, the normalized Black function actually returns

$$b \;\approx\; \Phi(h+t)(1+\varepsilon_1) \;-\; \Phi(h-t)(1+\varepsilon_2) \qquad (6.3)$$

where we have dropped ϕ since it is a number near 1 and irrelevant for our analysis. By expansion, this becomes

$$b \;\approx\; \Phi(h)(1+\varepsilon_1) + \varphi(h)t(1+\varepsilon_1) \qquad (6.4)$$
$$-\; \Phi(h)(1+\varepsilon_2) + \varphi(h)t(1+\varepsilon_2)$$

$$\approx\; 2\Phi(h)\varepsilon_3 + 2\varphi(h)t(1+\varepsilon_4) \qquad (6.5)$$

$$\approx\; 2\varphi(h)t + 2\left[\Phi(h)\varepsilon_3 + \varphi(h)t\varepsilon_4\right] \qquad (6.6)$$

where we have consolidated $\varepsilon_1 - \varepsilon_2 \approx 2\varepsilon_3$ and $\varepsilon_1 + \varepsilon_2 \approx 2\varepsilon_4$. This makes the relative numerical evaluation error

$$\frac{b^{\text{numerical}}}{b^{\text{exact}}} - 1 \;\approx\; \frac{\Phi(h)\varepsilon_3 + \varphi(h)t\varepsilon_4}{\varphi(h)t} \qquad (6.7)$$

$$\approx\; \frac{1}{t} \cdot \frac{\Phi(h)}{\varphi(h)} \cdot \varepsilon_3 \qquad (6.8)$$

for small t. As $t \to 0$, the relative error grows like the inverse of t, *and there is nothing we can do about it*. Unless, that is, we don't carry out the subtraction in (6.1) in the first

---

[4]The shape of the distribution is irrelevant here: we only need to know the attainable range.

place! The obvious thing to do is to use a Taylor expansion in t around zero. *Don't!* Remember that the main reason that computing implied volatilities is so difficult is the fact that the Black function does not permit a Taylor expansion around $\sigma = 0$ (unless $x \equiv 0$) since all of its derivatives in $\sigma = 0$ are identically zero! Unfortunately, this dilemma is not resolved by viewing the Black function as weighted differences of $\Phi(h \pm t)$, with h and t as defined in (6.2), keeping h constant, and expanding in t. If you try this, you will find that some of the coefficients still diverge such that your numerical results are spoiled when σ is very small unless you keep increasing the expansion order to ludicrous levels. So, instead, we reformulate the normalized Black function according to

$$b = \Phi(h+t)e^{ht} - \Phi(h-t)e^{-ht} \qquad (6.9)$$

$$= \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(h^2+t^2)} \cdot [Y(h+t) - Y(h-t)] \qquad (6.10)$$

with

$$Y(z) := \frac{\Phi(z)}{\varphi(z)} \qquad (6.11)$$

which we show in figure 4. The advantage of casting the
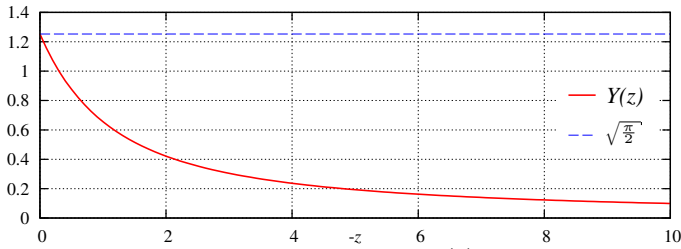
Figure 4: The function $Y(z)$.

normalized Black function in this form is that the expression

$$[Y(h+t) - Y(h-t)] \qquad (6.12)$$

permits a perfectly usable Taylor expansion in t for $h \leq 0$, even when h is exactly zero, and that is how we do it. We skip the details of the actual expansion and refer the reader to the code comments in the reference implementation in [Jäc13], though we mention that we use it when $t < \tau_{\text{small}}$ with $\tau_{\text{small}} := 2\sqrt[16]{\epsilon} \approx 0.21$ (and $|h|$ not too large). We show two examples as to how noisy the Black function $b(x, \sigma)$ can be as a function of σ, on a relative scale, in figure 5, in comparison with the results we obtain when using an expansion of expression (6.12) in t. Note that the
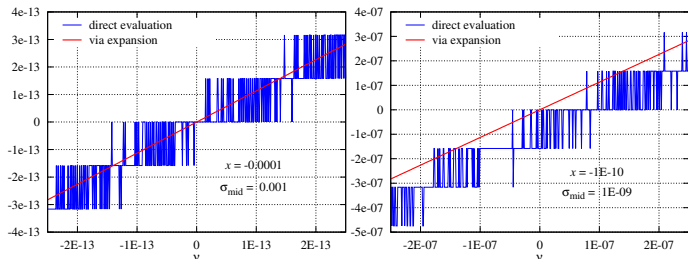

Figure 5: Two examples for the noise on the Black function when evaluated directly, in comparison to the use of an expansion of expression (6.12). The abscissa ν is the relative distance to an arbitrarily chosen mid value σ_{mid}. The ordinate is the function value's relative distance from its value in the centre, i.e., $\frac{b(x, \sigma_{\text{mid}} \cdot (1+\nu))}{b(x, \sigma_{\text{mid}})} - 1$.

abscissa ν in the shown examples is directly the scale of relative accuracy of the implied volatility σ. It is clear that any root-finding procedure cannot resolve a root σ^* below a relative resolution of $\Delta\nu$ if the objective function appears to have multiple roots within $\sigma^* \pm \Delta\sigma$ with $\Delta\sigma = \Delta\nu \cdot \sigma^*$. This is what we alluded to earlier when we referred to the *maximum attainable accuracy*: in order to be able to compute implied volatility to a relative accuracy of, say, 10^{-15}, we first need to have a Black function that near the solution is smooth down to the same relative accuracy.

Unfortunately, the previously handled region is not the only area where the conventional implementation of the Black function suffers catastrophic loss of accuracy. Another such region is the limit of large but negative $h \equiv x/\sigma$ even when σ itself is not small at all. In this case, the Black function may not incur any significant *subtractive cancellation* of cumulative normal function values since it is very possible that the two evaluations $\Phi(h \pm t)$ are of largely different orders of magnitude. The problem here is different in nature, and comes down to the inevitable loss of accuracy of the cumulative normal function itself as is explained in the excellent article by George Marsaglia [Mar04]. The cause of this is that all implementations of the cumulative normal function of a large negative argument in some way or another involve an evaluation of the exponential function with a large and negative argument. Unless we implement our own exponential function, we are therefore at the mercy of the platform's built-in exponential function which tends to be a low-level assembler function call, and, typically, only gives us about 14–15 decimal digits of relative accuracy for large negative arguments. This teaches us two lessons. The first is that we should avoid (whenever affordable on balance) computing the Black function as the difference of terms involving individual exponential terms since this exacerbates the loss of accuracy due to subtractive cancellation. The second is that we may prefer formulations that have fewer exponential function evaluations when available. In the region of large negative h, we can realize these preferences by the aid of the formulation (6.10) for the normalized Black function, and make use once again of the asymptotic expansion [AS84, (26.2.12)] to write $Y(z)$ as the rational function

$$Y(z) \approx \frac{1}{z} - \frac{1}{z^3} + \frac{1 \cdot 3}{z^5} + \ldots + \frac{(-1)^n 1 \cdot 3 \ldots (2n-1)}{z^{(2n+1)}} \qquad (6.13)$$

when $z \ll 0$. Note that this is a divergent series which means that for any value of z, there is a critical level for n beyond which the approximation series worsens as you increase n. In other words, there is some optimal level n where the relative error of the approximation series compared to the exact value of $Y(z)$ is minimal. Obviously, the larger $|z|$, the larger the optimal level n at which the relative error is minimal. We found that for $n = 17$, the approximation series has a maximum relative error of $1.64 \cdot 10^{-16}$ for all $z \leq -10$, which makes it accurate to within the best attainable limit on 64 bit floating point hardware. In order to avoid any subtractive cancellation in the normal-

ized Black function when $h < h_{\text{large}}$, with $h_{\text{large}} := -10$ (and t somewhat smaller than $|h|$), in addition to minimizing the exponential noise, we proceed as follows. We use the asymptotic series (6.13) with $n = 17$ and substitute it into $[Y(h + t) - Y(h - t)]$ in (6.10), and *analytically evaluate and simplify* the resulting expression (which is of considerable length), in order to take advantage of all possible analytical cancellations of terms such as $+t$ and $-t$. The resulting asymptotic expression, whilst somewhat lengthy, turns out to give us a reliable and smooth normalized Black function that enables us to compute implied volatilities even when x approaches the absolute lower limit[5] on 64 bit hardware, which is about -707.

When neither $t \ll 1$, nor $h < h_{\text{large}}$, as we mentioned above as the lessons we learned from George Marsaglia's excellent article, we should still avoid computing the Black function as the difference of terms involving individual exponential terms, or at least minimize the number of exponentials, and so we stick with the formulation (6.10). Here, we take advantage of the fact that $Y(z)$ is related to the little known special function called the *scaled complementary error function* $\mathrm{erfcx}()$ via the simple relationship

$$Y(z) = \tfrac{1}{2} \cdot \sqrt{2\pi} \cdot \mathrm{erfcx}(-z/\sqrt{2}) . \qquad (6.14)$$

Since there is a highly accurate and efficient numerical implementation for $\mathrm{erfcx}()$ based on rational approximations involving at most one exponential function evaluation, we at least halve the noise level. Otherwise, if we go the conventional route to evaluate $b()$, we incur at least one exponential inside the implementation of the cumulative normal function, and another one for each of the scaling terms $\mathrm{e}^{-x/2}$ and $\mathrm{e}^{-x/2}$, resulting in the subtraction of two terms that each involved two exponentials. What's more, for $x \geq 0.46875$, Cody's implementation in [Cod69, Cod90] of $\mathrm{erfcx}(x)$, is given *only* as a rational function approximation, which means that we obtain $Y(z)$ represented by a pure rational approximation, without any exponentials, when $z \leq -0.66291260736239$.

Having emphasized the benefits of the formulation (6.10), we must, alas, make an exception when $b()$ is dominated by the first of the two terms in the Black formula when expressed as (6.9). We then retain more relative accuracy by not attempting to combine the two terms in any way, and sticking with the formulation (6.9). As a rule of thumb, we do this when $t > 0.85 + |h|$ (with $h \leq 0$ and $\theta \equiv 1$ as before). We then use the equality

$$\Phi(z) = \tfrac{1}{2} \cdot \mathrm{erfc}(-z/\sqrt{2}) \qquad (6.15)$$

and evaluate Cody's [Cod69, Cod90] implementation of the complementary error function $\mathrm{erfc}()$ which contains a round-off limiting technique specifically aimed at the inaccuracy of the exponential function for large negative argument mentioned in George Marsaglia's article. Whenever

---

[5]The limit for x is determined by the minimum attainable ratio of F/K, which is about 10^{-307}, making $x_{\min} \approx \ln(10^{-308}) \approx -707$.

e^{-y^2} for sizeable $y > 0$ is required, instead, the product of *two* exponential evaluations is computed, one aiming at the magnitude of the result, and one aiming at the fine resolution according to

$$\mathrm{e}^{-y^2} = \mathrm{e}^{-\tilde{y}^2} \cdot \mathrm{e}^{-(y-\tilde{y}) \cdot (y+\tilde{y})} \qquad (6.16)$$

where $\tilde{y}$ is chosen to give the overall magnitude (down to one 16th) as

$$\tilde{y} := \frac{\lfloor y \cdot 16 \rfloor}{16} . \qquad (6.17)$$

This does not completely solve the issue, but it helps a long way. It is worth mentioning that this technique is also used in other implementations, e.g., the one given in [Mic93b].

We show in figure 6 the four different evaluation zones for the normalized Black function for $h \leq 0$ and $\theta \equiv 1$. In
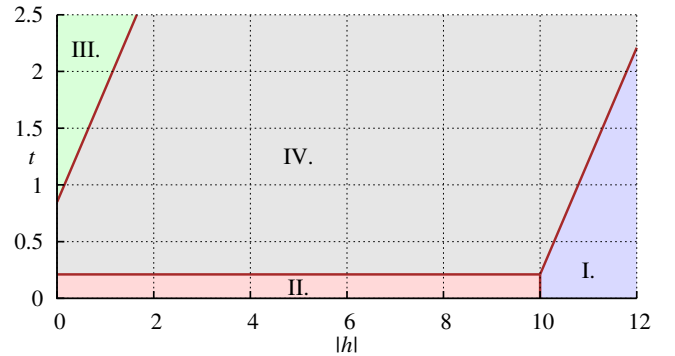
Figure 6: The four different evaluation regimes of the normalized Black function in the (h, t)-plane with $h = x/\sigma$, $t = \sigma/2$, $x \leq 0$, and $\theta = 1$.

summary, these are, in order of precedence:-

I. $(|h| > |h_{\text{large}}|) \wedge (t < |h| - |h_{\text{large}}| + \tau_{\text{small}})$ with $\tau_{\text{small}} = 2 \sqrt[16]{\epsilon} \approx 0.21$: substitute the series approximation (6.13) of order $n = 17$ into (6.10). Analytically simplify the sub-expression (6.12)

$$[Y(h + t) - Y(h - t)]$$

after the substitution (6.13) to take advantage of exact cancellation of terms such as $+t$ and $-t$. The net result gives the sub-expression (6.12) as a rational function of h, multiplied with one exponential.

II. $t < \tau_{\text{small}}$: substitute a twelvth order Taylor expansion of the sub-expression (6.12)

$$[Y(h + t) - Y(h - t)]$$

in t around zero in the normalized Black function formulation (6.10).

III. $t > 0.85 + |h|$: evaluate $b()$ as the exponentially-weighted difference of cumulative normals as given in (6.9). Use a highly accurate version of the cumulative normal such as mapping it to Cody's $\mathrm{erfc}()$ [Cod69, Cod90] via (6.15).

IV. everywhere else: evaluate $b()$ in the formulation (6.10) with $Y()$ being computed via (6.14) by the aid of Cody's $\mathrm{erfcx}()$ [Cod69, Cod90].

Finally, we mention that there is yet another reason for loss of accuracy, though this one is directly for the implied volatility, and not for the Black function. This happens when the input price is near the maximum: $\beta \lesssim b_{\max}$. In this limit, the relevant quantity of information content is the difference from the maximum, namely $(b_{\max} - \beta)$, and the relative accuracy of the output implied volatility can only be expected to be as good as

$$\frac{b_{\max}}{b_{\max} - \beta} \cdot \epsilon \qquad (6.18)$$

with ϵ being as before the relative hardware accuracy. The problem here is unsurmountable. It is caused by the fact that the input number β, when it is, say, within 10^{-m} (relative) of $b_{\max}$, only contains approximately $(N_\epsilon - m)$ decimal digits of relevant information, with $N_\epsilon := |\log_{10}(\epsilon)|$, and thus we cannot produce a result that has the full N_ϵ digits of accuracy. This, limit case, however, is in practice of no concern since this is the situation of volatilities and prices being so high that prices have no discernible vega. What's more, whilst we do in practical calculations encounter low volatility scenarios of any imaginable magnitude, the ultra-high volatilities just don't arise. We will, however, in our numerical charts see the gradual increase of the residual noise level in the limit of $\beta \to b_{\max}$.

7 Numerical results

We now show a number of graphs with numerical results. First, in figures 7 and 8, we show two diagrams depicting
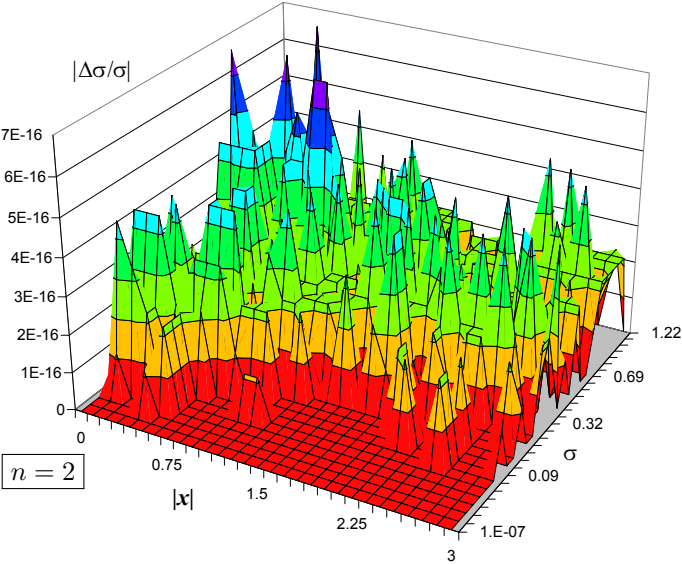
Figure 7: Relative accuracy $|\frac{\Delta\sigma}{\sigma}|$ for $(|x|, \sigma) \in [0, 3] \times [10^{-7}, 1.22]$.

relative residual errors of implied volatilties for a range of (x, σ) pairs, very similar to figures 8 and 9 in [Jäc06]. Note that these results here, whilst being of significantly smaller relative error, have been computed with exactly two Householder(3) iterations as described in section 5. The residuals are against an original value of σ from which a normalized Black value was computed that in turn was the input price to the implied volatility calculation.
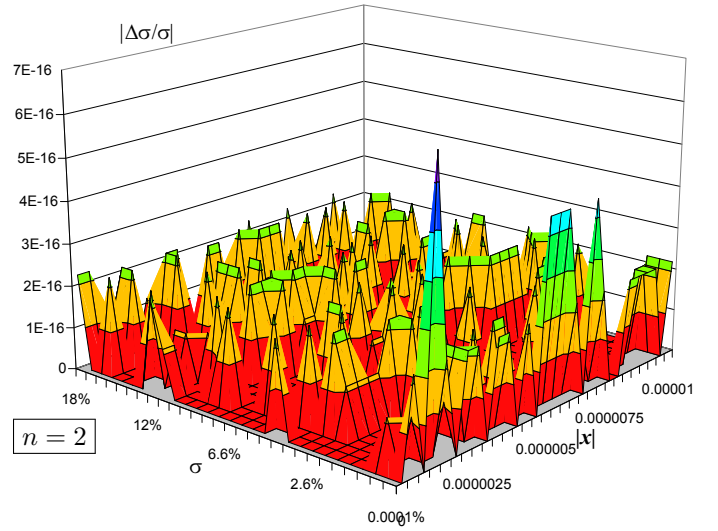

Figure 8: Relative error $|\frac{\Delta\sigma}{\sigma}|$ for $(|x|, \sigma) \in [0, 10^{-5}] \times [10^{-5}, 0.18]$.
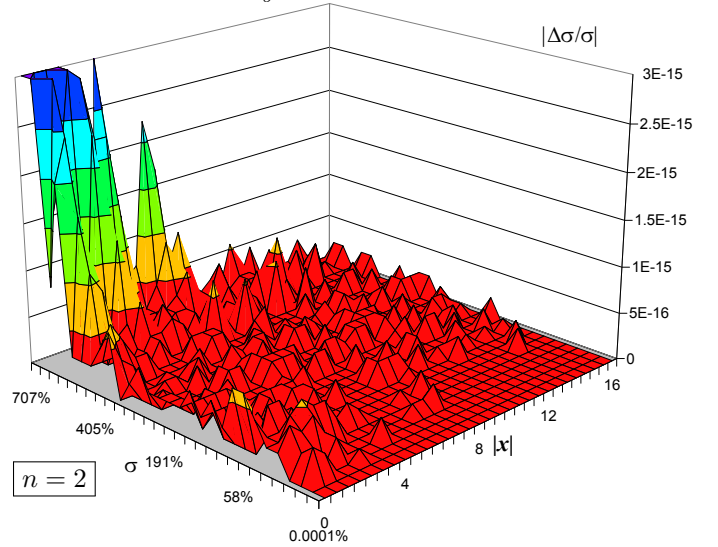

Figure 9: Relative accuracy $|\frac{\Delta\sigma}{\sigma}|$ for $(|x|, \sigma) \in [0, 16] \times [10^{-5}, 7.07]$.

In figure 9, we show the same relative error after two iterations for a much wider range in both $|x|$ and σ. Here, we can see the onset of the loss of accuracy mentioned at the end of section 6 due to the Black function values approaching its maximum. Specifically in this figure, on the line $x = 0$, the input β is above 90% of the maximum $b_{\max}(x = 0) = 1$ on the leftmost ten out of a total of 29 plot grid nodes, explaining why the residual error begins to grow in the left hand side corner of the diagram.

In figure 10, we show the relative error of the initial guess, and after one, two, and three Householder(3) iterations on a linear scale of $\beta/b_{\max}$ to the left, and on a logarithmic scale to the right, for $|x| = 1/2$. The mentioned "pin locations" are the interpolation nodes of the initial guess at $\beta = b_l$, $\beta = b_c$, and $\beta = b_u$. We compare this with the residual errors when we iterate with Halley's method in figure 11, and with Newton's method in figure 12. We show the same set of data in figures 13, 14, and 15 for $|x| = 32$. In that latter set of data, specifically in figure 14 for $|x| = 32$, we see on the logarithmic scale an example where two iterations of Halley's method fail to give us full attainable precision, which is symptomatic for large $|x|$. This is why we overall chose to use two iterations of the Householder(3) method whose extra effort over Hal-
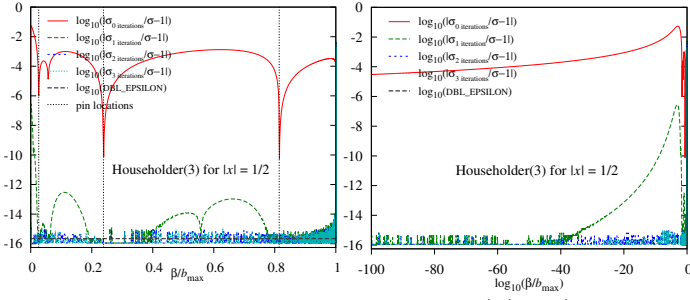
Figure 10: Residual relative errors for $|x| = \frac{1}{2}$.
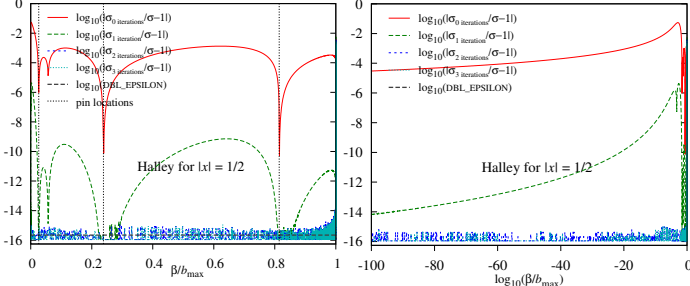

Figure 11: Residual relative errors for $|x| = \frac{1}{2}$ with Halley's method.
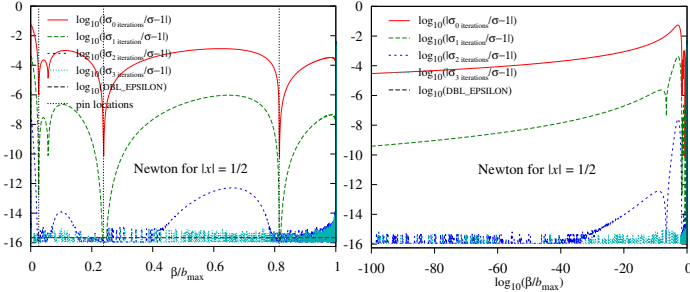

Figure 12: Residual relative errors for $|x| = \frac{1}{2}$ with Newton's method.

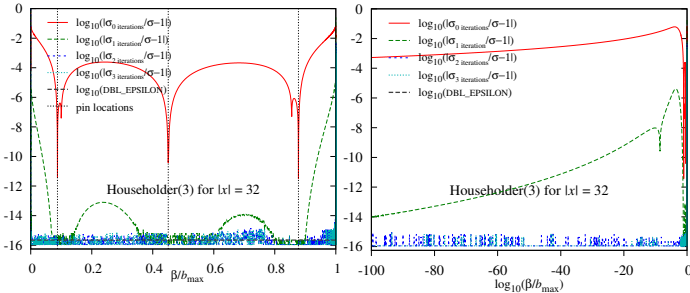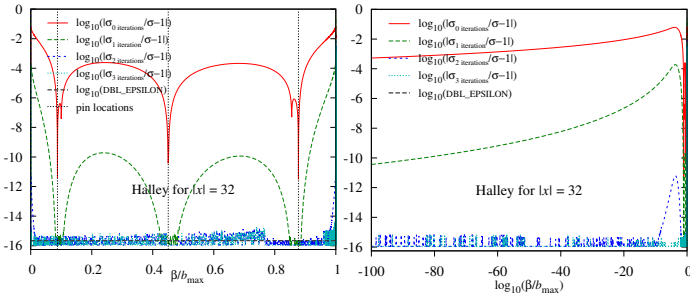

Figure 13: Residual relative errors for $|x| = 32$.

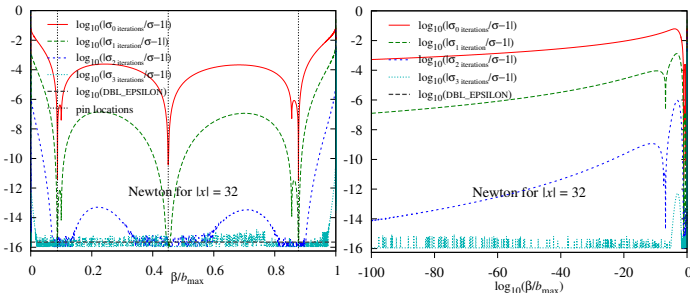Figure 14: Residual relative errors for $|x| = 32$ with Halley's method.

Figure 15: Residual relative errors for $|x| = 32$ with Newton's method.

ley's method is negligible in comparison to the evaluation of the normalized Black function required in each iteration.

As for its execution speed, the presented method evaluates a single implied volatility with two iterations on a standard computer in just under one microsecond most of which is spent in the normalized Black function. This is more than 5 times faster than the algorithm of [Jäc06] which takes about 5 microseconds on the same hardware when configured to have comparable accuracy, specifically, 1E-15. The speed advantage goes down to only about 30% when the algorithm of [Jäc06] is evaluated with a target precision of 2E-12 (though at that level we could here get away with just one iteration most of the time, and perhaps increase the Householder order by one). It is clear that the algorithm of "By Implication" converges relatively easily, on average, to that lesser accuracy (albeit that it needs more iterations), but requires significantly more effort to then home in on the higher precision. The difficulty of convergence to high accuracy is caused by the much noisier normalized Black function used there. The net effect is that each single iteration of "By Implication" is faster than here, which is due to the simpler, but much noisier, normalized Black function, but high accuracy is very difficult to attain for the very same reason. On a like for like comparison, i.e., when using the same normalized Black function, the algorithm of "By Implication" will be significantly slower overall since it will almost everywhere need more iterations, and will be *much* slower in all those low volatility regions where it was previously identified to need significant numbers of iterations.

8 Conclusion

We have introduced an algorithm for the calculation of Black implied volatility that can for all intents and purposes be considered to be *within attainable machine accuracy* where the latter is defined to mean within what can be supported by the used normalized Black function. This has been accomplished by combining a four-branched initial guess function based on two asymptotically correct transformations and rational function approximations, with two iterations of the Householder(3) root finding algorithm. The objective function is separated into three branches, the top and bottom of which involve non-linear transformations. Crucially, the objective function is based on a high-accuracy and low-noise implementation of the normalized Black function, which turns out to be a problem as difficult as implied volatility calculation in its own right.

We mention that the mere two steps of our Householder(3) procedure can, instead of viewing them as numerical iteration, also be seen as an analytical approximation based on the recursive definition

$$\sigma(\beta) = \sigma^{\mathrm{HH3}}(\sigma^{\mathrm{HH3}}(\sigma_0(\beta))) \tag{8.1}$$

where $\varsigma \to \sigma^{\mathrm{HH3}}(\varsigma)$ is defined to be the Householder(3) propagation step $\sigma_n \to \sigma_{n+1}$ given in (5.6), and $\sigma_o()$ is our initial guess function (4.39). The twice-recursive formulation (8.1) of the solution presented here might help assuage

the concerns of those who intrinsically dislike numerical solutions.

A reference implementation of the discussed new method for implied Black volatility is available at www.jaeckel.org/LetsBeRational.7z [Jäc13], including a total of 187 figures demonstrating the accuracy in various parameter regions.

References

[AS84]   M. Abramowitz and I.A. Stegun. *Pocketbook of Mathematical Functions*. Harri Deutsch, 1984. ISBN 3-87144-818-4.

[BBF02]   H. Berestycki, J. Busca, and I. Florent. Asymptotics and calibration of local volatility models. *Quantitative Finance*, 2:61–69, 2002. www.iop.org/EJ/article/1469-7688/2/1/305/qf2105.pdf.

[BS88]   M. Brenner and M. Subrahmanyam. A simple formula to compute the implied standard deviation. *Financial Analyst Journal*, 44(5):80–83, 1988.

[CM96]   C.J. Corrado and T.W. Miller. Volatility without tears. *Risk*, 9(7):49–52, 1996.

[CM04]   C.J. Corrado and T.W. Miller. Tweaking Implied Volatility. Technical report, Deakin University and Mississippi State University, September 2004. ssrn.com/abstract=584982.

[Cod69]   W. J. Cody. Rational Chebyshev approximations for the error function. *Mathematics of Computation*, pages 631–638, 1969. www.ams.org/journals/mcom/1969-23-107/S0025-5718-1969-0247736-4/S0025-5718-1969-0247736-4.pdf.

[Cod90]   W. J. Cody. Fortran implementation of rational Chebyshev approximations for the error function. www.netlib.org/specfun/erf, 1969–1990.

[DG85]   R. Delbourgo and J.A. Gregory. Shape preserving piecewise rational interpolation. *SIAM Journal of Scientific and Statistical Computing*, 6(4):967–976, 1985. dspace.brunel.ac.uk/bitstream/2438/2200/1/TR_10_83.pdf.

[Gru11]   C. Grunspan. Asymptotic Expansions of the Lognormal Implied Volatility : A Model Free Approach. Technical report, ESILV, Department of Financial Engineering, December 2011. arxiv.org/pdf/1112.1652v1.pdf.

[Hou70]   A.S. Householder. *The Numerical Treatment of a Single Nonlinear Equation*. McGraw-Hill, New York, 1970. LCCN 79103908, en.wikipedia.org/wiki/Householder's_method.

[Jäc06]   P. Jäckel. By Implication. *Wilmott Magazine*, pages 60–66, November 2006. www.jaeckel.org/ByImplication.pdf.

[Jäc13]   P. Jäckel. Implementing "Let's be rational". www.jaeckel.org/LetsBeRational.7z, November 2013.

[Li06]   M. Li. You Dont Have to Bother Newton for Implied Volatility. Technical report, Georgia Institute of Technology, November 2006. ssrn.com/abstract=952727.

[Mar04]   G. Marsaglia. Evaluating the Normal Distribution. *Journal of Statistical Software*, 1:1–11, 2004. www.jstatsoft.org/v11/a05/paper.

[Mic93a] Sun Microsystems. Freely Distributable LIBM, 1993. www.netlib.org/fdlibm.

[Mic93b] Sun Microsystems. Implementation of the error function erf() and the complementary error function erfc(). www.netlib.org/fdlibm/s_erf.c, 1993.

[PTVF92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992. www.nrbook.com/a/bookcpdf.php.

[Veb09]   D. Veberic. Having Fun with Lambert W(x) Function. *ArXiv Mathematics e-prints*, June 2009. arxiv.org/pdf/1003.1628.

[Vog07]   A. Vogt. Initial Estimating and Refining volatility, 2007. www.axelvogt.de/maplekram/initialVolatility_refined.pdf.

[Wic88]   M.J. Wichura. Algorithm AS 241: The Percentage Points of the Normal Distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 37(3):477–484, 1988. lib.stat.cmu.edu/apstat/241.