By Implication

Peter Jäckel*

First version: 9th July 2006
This version: 24th November 2010

Abstract

Probably the most frequently executed numerical task in practical financial mathematics is the calculation of the *implied volatility* number consistent with a given forward, strike, time to expiry, and observable market price for European plain vanilla call and put options. At the same time, this task is probably also the least documented one in applied financial mathematics. In this document, it is explained why it is not as easy as one might think to implement an implied volatility function that is both efficient and robust, and a possible solution to the difficulty is suggested.

1 Introduction

A plain vanilla call or put option price p in the Black-Scholes-Merton [BS73, Mer73] framework is given by

$$p \;=\; \delta\cdot\theta\cdot\Big[F\cdot\Phi\Big(\theta\cdot\Big[\tfrac{\ln(F/K)}{\sigma}+\tfrac{\sigma}{2}\Big]\Big) - K\cdot\Phi\Big(\theta\cdot\Big[\tfrac{\ln(F/K)}{\sigma}-\tfrac{\sigma}{2}\Big]\Big)\Big] \tag{1.1}$$

where $\theta = 1$ for call options and $\theta = -1$ for put options, $F := S\mathrm{e}^{(r-d)T}$, S is the current spot, K is the strike, r is a flat continuously compounded interest rate to maturity, d is a flat continuously compounded dividend rate, $\sigma = \hat\sigma\cdot\sqrt{T}$, $\hat\sigma$ is the root-mean-square lognormal volatility, T is the time to expiry, and $\Phi(\cdot)$ is the standard cumulative normal distribution function. In the Black-Scholes-Merton framework, the quantitity δ represents a discount factor to time T, and in general, might be referred to as the *deflater* of the option price.

Black [Bla76] extended the applicability of the geometric Brownian motion framework to (what we might call today) an arbitrary *numéraire* by ingeniously separating the price deflation from the calculation of an option value relative to today's value of the numéraire, thus making (1.1) applicable to almost all areas of financial option valuation theory. To value an interest rate swaption, for instance, we evaluate (1.1) with F representing the forward swap rate, and set the price deflater δ to be today's net present value of the forward starting annuity.

Since market quoting conventions for many asset classes are such that option prices are compared for their relative value in terms of the root-mean-square lognormal volatility $\hat\sigma$, it is important for any derivatives library to be able to convert actual option prices into the equivalent *implied Black volatility* figure. In addition, the implied volatility function is often also used either explicitly or implicitly in exotic pricing models or analytical approximations. Particularly in the latter application, it is often important that the implied volatility computed by the derivatives analytics library is of high accuracy, and is robust even for parameter combinations that may, at first sight, seem not to be relevant. For instance, in the displaced diffusion model [Rub83] governed by the stochastic differential equation

$$\mathrm{d}S \;=\; \varsigma\,[qS + (1-q)\,S_0]\,\mathrm{d}W\;, \tag{1.2}$$

the vanilla option price for strike κ can also be computed using the Black formula (1.1) with adjusted input parameters

$$\begin{aligned} F &\to S_0/q \\ K &\to \kappa + {}^{(1-q)}\!/_{q}\cdot S_0 \\ \sigma &\to \varsigma\cdot q\cdot\sqrt{T}\;. \end{aligned}$$

In order to attain a strongly pronounced negative implied volatility skew such as it is observed in the equity and interest rate market for low strikes, q often has to take on values as small as 10^{-4}. Conversely, for some high strikes in the FX or commodity markets, q may need to exceed 2. All of this means that the effective standard deviation number σ in the Black formula (1.1) can easily be in the range $[10^{-4}\%, 1000\%]$, or possibly even outside. As a consequence, any implied volatility solver should be able to produce a comparatively, i.e. relatively, accurate figure even for parameter combinations that mean that σ is a very small or moderately large number, since it shouldn't assume that the returned number is used straightaway: it may yet, for instance, undergo division by $q\cdot\sqrt{T}$ for small or large q and small or large T. This clearly requires any solver termination criterion to be based on relative accuracy in σ, not in function value.

*Global head of credit, hybrid, inflation, and commodity derivative analytics, ABN AMRO, 250 Bishopsgate, London EC2M 4AA, UK

2 Implying volatility

Given an option price p, the task at hand is to find the log-standard deviation number σ that makes p equal to expression (1.1) for the forward F and strike K. Once we know σ, the implied volatility figure is given by $\hat{\sigma} = \sigma/\sqrt{T}$. So far, it all seems easy.

Using the definitions

$$x := \ln\left({}^{F}\!/\!{}_{K}\right) \qquad b := \frac{p}{\delta\sqrt{FK}}\ , \tag{2.1}$$

the equation we need to solve for σ becomes

$$b = \theta e^{x/2}\cdot\Phi\left(\theta\cdot[x/\sigma + \sigma/2]\right) - \theta e^{-x/2}\cdot\Phi\left(\theta\cdot[x/\sigma - \sigma/2]\right)\ . \tag{2.2}$$

The special but very important case $F = K$ reduces to

$$b|_{x=0} = 1 - 2\Phi(-\sigma/2) \tag{2.3}$$

which allows for the exact solution[1]

$$\sigma = -2\cdot\Phi^{-1}\left(\tfrac{1}{2}(1-\beta)\right) \tag{2.4}$$

wherein β is the normalised option price that is to be matched.

2.1 Limits

The normalised option price b is a positively monotonic function in $\sigma \in [0, \infty)$ with the limits

$$h(\theta x)\cdot\theta\cdot\left(e^{x/2} - e^{-x/2}\right) \ \le\ b\ <\ e^{\theta x/2} \tag{2.5}$$

wherein $h(\cdot)$ is the Heaviside function.

In order to understand the asymptotic behaviour of (2.2) from a purely technical point of view, let us recall [AS84, (26.2.12)]

$$\Phi(z) = h(z) - \frac{\varphi(z)}{z}\left[1 - \frac{1}{z^2} + \mathcal{O}\left(\frac{1}{z^4}\right)\right] \ \text{for } |z| \to \infty \tag{2.6}$$

with $\varphi(z) = e^{-z^2/2}\Big/\sqrt{2\pi}$. Equation (2.6) highlights a common practical issue with the cumulative normal distribution function: when its argument z is significantly positive, as is the case here for deeply in the money options[2], $\Phi(z)$ becomes indistinguishable from 1, or has only very few digits in its numerical representation that separates it from 1. The best way to overcome this problem is to use an implementation of $\Phi(z)$ that is highly accurate for negative z, and to only ever use out-of-the-money options when implying Black volatility[3].

[1] assuming that one has a highly accurate inverse cumulative normal function available, such as the one published by Peter Acklam [Ack00]

[2] A one-week-to-expiry call option struck at 50% of the spot for $\hat{\sigma} = 50\%$ corresponds to $z \approx 10$.

[3] This is the reason why put-call parity should never be used in applications: it is a nice theoretical result but useless when you rely on it in your option pricing analytics.

Equation (2.6) enables us to derive the asymptotics

$$\lim_{\sigma\to\infty} b = e^{\theta x/2} - {}^{4}\!/\!{}_{\sigma}\cdot\varphi(\sigma/2) \tag{2.7}$$

$$\lim_{\sigma\to 0} b = \iota + x\cdot\varphi(x/\sigma)\cdot\left(\frac{\sigma}{x}\right)^3 \tag{2.8}$$

using the definition of the normalised intrinsic value

$$\iota := h(\theta x)\cdot\theta\cdot\left(e^{x/2} - e^{-x/2}\right)\ . \tag{2.9}$$

From equation (2.8) we can see what happens as volatility approaches zero (for $x \neq 0$): since $\varphi(y)$ decays more rapidly than y^{-n} for any positive integer n as $y \to \infty$, the Black option price does not permit for any regular expansion for small volatilities. The extremely flat functional form of b for small σ for $x \neq 0$ can also be seen in figure 1, and this is where the trouble starts.

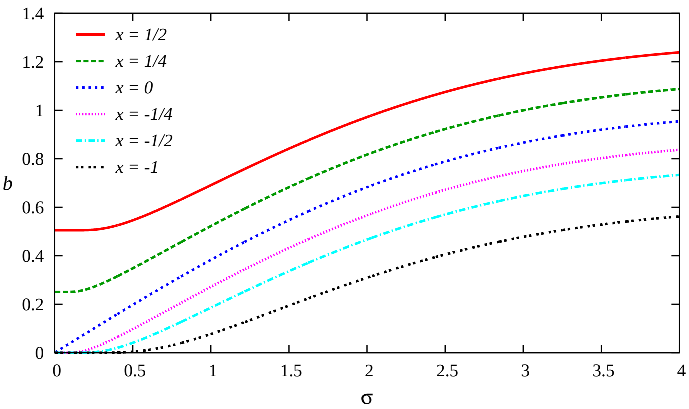

Figure 1: Normalised call ($\theta = 1$) option prices given by (2.2).

Conventional wisdom has it that the best all-round method of choice for the root-finding of smooth functions is Newton's algorithm. Alas, this is not always so for functions that do not permit regular expansions on a point in the domain of iteration, or on its boundary. This unpleasant feature of the Black option formula is made worse by the fact that it is convex for low volatilities and concave for higher volatilities. This means that, given an arbitrary initial guess, a Newton iteration may, if the initial guess is too low and in the convex domain, be fast forwarded to very high volatilities, and not rarely to levels where the numerical implementation of our (normalised or conventional) Black function no longer distinguishes the result from the limit value for high volatilities. When the latter happens, the iteration ceases and fails. Conversely, when the arbitrary initial guess is too high and in the concave domain, the first step may attempt to propel volatility to the negative domain. Even when this doesn't happen, the Newton algorithm can enter a non-convergent, or near-chaotic, cycle as discussed in [PTVF92, section 9.4]. Both of these unfortunate accidents can be prevented by the addition of safety controls in the iteration step, e.g. [PTVF92, routine `rtsafe`]. Still, even with a safety feature, the Newton method can still take many more iterations than one would want it to in any time-critical derivatives valuation and risk management system where it is invoked literally billions or even trillions of times every day [4]. Ideally, when the correct implied volatility is in the convex domain of the function, we would want to converge from above (in function value), when it is in the concave domain, we would want to converge from below, since we are then guaranteed to never leave the respective domain. Fortunately, the normalised Black option formula (2.2) allows for the solution for the point of inflection. It is at

$$\sigma_c \; = \; \sqrt{2|x|} \; . \tag{2.10}$$

The temptation is now to simply start at σ_c and Newton-iterate until we are converged. This works fine for all $\sigma > \sigma_c$, but fails for many $\sigma < \sigma_c$. The reason for this failure is the near-flat shape of the normalised Black function for small volatilities. If you try it, you will find that the iteration almost grinds to a halt since the update step for the next guess converges practically as rapidly to zero as the current guess to the correct solution (for low volatilities and $x \neq 0$) as shown in figures[5] 2 and 3. Clearly, we would prefer to

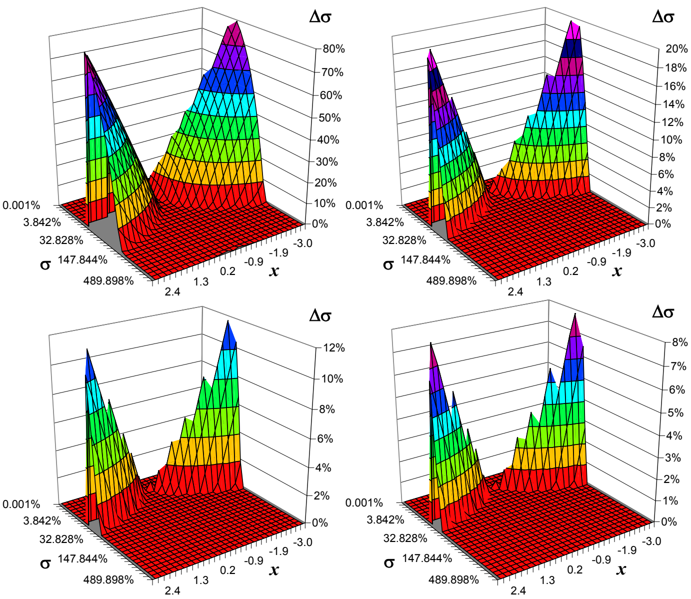

Figure 2: The absolute difference between the correct implied volatility and the number attained after n iterations when starting at σ_c and simply Newton-iterating with start value σ_c given in (2.10) for different n on $(\sigma, x) \in [10^{-5}, 2\sqrt{2 \cdot 3}] \times [-3, 3]$. Top left: $n = 5$, top right: $n = 50$, bottom left: $n = 100$, bottom right: $n = 150$. Note that the σ-axis has been scaled non-linearly to highlight the region of interest. The calculations for $x = 0$ have been done without iteration using equation (2.4).

have a method that doesn't need hundreds and hundreds of iterations to converge to an acceptable accuracy in σ for some perfectly reasonable parameter values.

[4] The implied volatility function is not only used for the representation of market prices but often also implicitly in exotic pricing models or analytical approximations.

[5] Note that the zero levels at the back of all shown diagrams for nonzero x and very small σ represent outright calculation failures since for those parameter combinations the normalised Black function value is smaller than the smallest representable floating point number (whence it was rounded down to zero). In other words, those areas in the parameter plane are not attainable in practice.

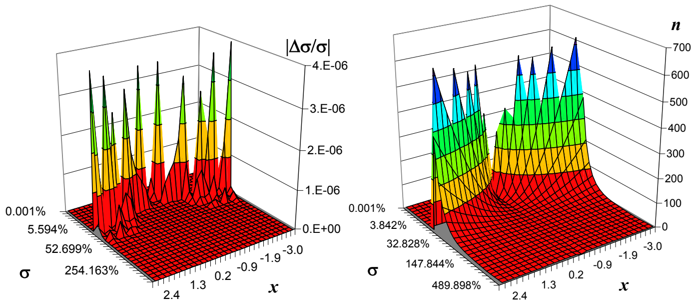

Figure 3: Left: residual relative difference between the correct implied volatility and the number attained after Newton-iterating with start value σ_c given in (2.10) until the current relative step size is less than 10^{-4} of the attained volatility level. Right: the number of iterations associated with the residual relative differences on the left. The calculations for $x = 0$ have been done without iteration using equation (2.4).

3 Where to start and what to aim for

Solving for roots of near-flat functions can be a nightmare to tackle. Again, though, we are lucky: the normalised Black function is amenable to straightforward root-finding even in its near-flat region by simply switching to solving for the value of σ that makes the *logarithm* of a given normalised option value equal to the logarithm of (2.2) ! In the concave domain, i.e. for $\sigma > \sigma_c$, this transformation is not helpful, and we stick with solving for the original normalised option value match. This means, given a target normalised option value β, the normalised moneyness x, and the call-put flag θ, we define the objective function for our root-finding algorithm as

$$\tilde{f}(\sigma) = \begin{cases} \ln\left(\frac{b-\iota}{\beta-\iota}\right) & \text{if } \beta < b_c \\ b - \beta & \text{else} \end{cases} \tag{3.1}$$

using

$$b_c \; = \; b_c(x, \theta) \; := \; b(x, \sigma_c, \theta) \; . \tag{3.2}$$

Alas, this leaves us with a new dilemma: our simplistic initial guess σ_c is no longer such a good idea for all $\beta < b_c$ since, with the function value on a logarithmic scale, the normalised Black function is no longer convex but concave, and the first step is likely to attempt overshooting into the domain of negative σ. We can overcome this issue by finding a better initial guess. In order to do this, we use a technique similar to *asymptotic matching* known in some sciences. Let us recapitulate: the functional form of the asymptotic expansion[6] for $\sigma \to 0$ given in (2.8) is $\varphi(x/\sigma) \cdot \sigma^3 / x^2$ which, alas, is not invertible in σ. The term $\varphi(x/\sigma)$, which is what gives rise to the near-flat behaviour, *is* invertible in σ, though! What's more, a function of the form $c(x) \cdot \varphi(x/\sigma)$ (for some $c(x)$ depending only on x) will, for small enough σ be always *larger* than (2.8), which means that, when inverted, it will give rise to an estimate for σ that is *lower* than the target root. This means, in the areas where it matters most, namely for small σ and thus

[6] assuming that we are only dealing with out-of-the-money option prices

small option prices, using an approximation for $(b-\iota)$ that is too high in value, but invertible, gives us precisely what we need in order to start off the Newton algorithm on a logarithmic scale (in value) as suggested in (3.1). The only open question is how to choose $c(x)$, but that is easily done: we set it such that the crude approximation for the asymptotics near zero match the value at σ_c. This gives us

$$b_{\text{low}} \;:=\; (b_c-\iota)\,\mathrm{e}^{\frac{|x|}{4}-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2}+\iota\;. \tag{3.3}$$

We can use a similar approach to improve the initial guess when the given option price indicates that $\sigma > \sigma_c$, i.e. when the given value β is larger than b_c. In that case, we can use a functional form which, for very large σ, resembles (2.7). We choose $\mathrm{e}^{\theta x/2} - c(x)2\Phi(-\sigma/2)$ since, as we can see from (2.6), for large σ, $2\Phi(-\sigma/2) \sim 4/\sigma \cdot \varphi(\sigma/2)$ which matches (2.7). Matching the function's value at σ_c to determine $c(x)$ results in the invertible function

$$b_{\text{high}} \;:=\mathrm{e}^{\frac{\theta x}{2}} - \frac{\mathrm{e}^{\frac{\theta x}{2}}-b_c}{\Phi\left(-\sqrt{\frac{|x|}{2}}\right)}\Phi\left(-\tfrac{\sigma}{2}\right)\;. \tag{3.4}$$

A deliberate side-effect of the specific choice (3.4) is that for $x \to 0$, it converges to the exact invertible form (2.3), which means that we no longer need to special-handle the case of $x = 0$ as we did previously for the calculations shown in figures 2 and 3. We show the approximations for different values of x in figure 4. Inverting the approxima-

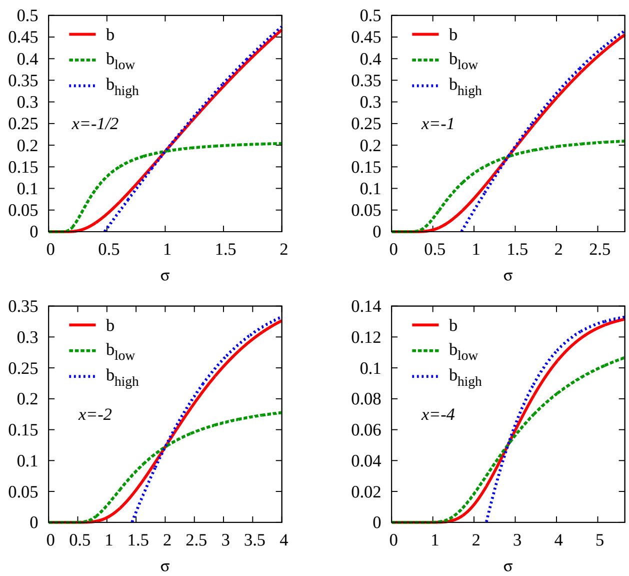

Figure 4: The invertible approximations b_{low} and b_{high} for the normalised Black function b given by equations (3.3) and (3.4).

tions (3.3) and (3.4) gives us the improved initial guess for any given normalised option value β:

$$\sigma_0(x,\beta,\theta) \;:=\; \begin{cases} \sigma_{\text{low}}(x,\beta,\theta) & \text{if } \beta < b_c \\ \sigma_{\text{high}}(x,\beta,\theta) & \text{else} \end{cases} \tag{3.5}$$

with

$$\sigma_{\text{low}}(x,\beta,\theta) \;:=\; \sqrt{\frac{2x^2}{|x|-4\ln\left(\frac{\beta-\iota}{b_c-\iota}\right)}} \tag{3.6}$$

and

$$\sigma_{\text{high}}(x,\beta,\theta) \;:=\; -2\cdot\Phi^{-1}\left(\frac{\mathrm{e}^{\frac{\theta x}{2}}-\beta}{\mathrm{e}^{\frac{\theta x}{2}}-b_c}\Phi\left(-\sqrt{\tfrac{|x|}{2}}\right)\right)\;. \tag{3.7}$$

Starting at (3.5), we thus iterate

$$\sigma_{n+1} \;=\; \sigma_n + \tilde{\nu}_n \tag{3.8}$$

with the Newton step $\tilde{\nu}_n = \tilde{\nu}(x,\sigma_n,\theta)$ given by

$$\tilde{\nu}(x,\sigma,\theta) = \begin{cases} \ln\left(\frac{\beta-\iota}{b-\iota}\right)\cdot\frac{b-\iota}{b'} & \text{if } \beta < b_c \\ \frac{\beta-b}{b'} & \text{else} \end{cases} \tag{3.9}$$

with

$$b' \;=\; \mathrm{e}^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2-\frac{1}{2}\left(\frac{\sigma}{2}\right)^2}\Big/\sqrt{2\pi} \tag{3.10}$$

until $|\Delta\sigma/\sigma| \approx |\sigma_{n+1}/\sigma_n - 1| < \epsilon$ for some given relative tolerance level ϵ. We show some results in figures 5 and 6.

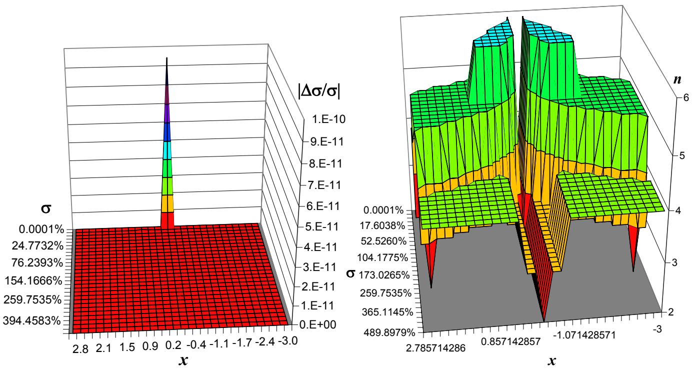

Figure 5: Left: residual relative error $|\Delta\sigma/\sigma| \approx |\sigma_{n+1}/\sigma_n - 1|$ after iterating (3.8) starting with σ_0 given in (3.5) until $|\sigma_{n+1}/\sigma_n - 1| < \epsilon$ with $\epsilon = 10^{-8}$ on $(\sigma,x) \in [10^{-6}, 2\sqrt{2\cdot 3}]\times[-3,3]$. Right: the number of iterations associated with the relative errors on the left.

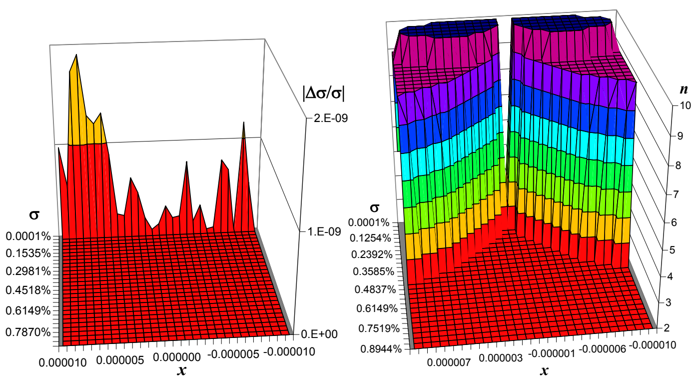

Figure 6: Left: residual relative error $|\Delta\sigma/\sigma| \approx |\sigma_{n+1}/\sigma_n - 1|$ after iterating (3.8) starting with σ_0 given in (3.5) until $|\sigma_{n+1}/\sigma_n - 1| < \epsilon$ with $\epsilon = 10^{-8}$ on $(\sigma,x) \in [10^{-6}, 2\sqrt{2\cdot 10^{-5}}] \times [-10^{-5}, 10^{-5}]$. Right: the number of iterations associated with the relative errors on the left.

4 Shaving off a few iterations

We can see in figures 5 and 6 that for small x, and $\beta < b_c$, the number of iterations required for convergence increases somewhat. Comparing this with the top left diagram in figure 4 tells us that this is caused by the initial guess (3.5) not being that good for those parameter combinations: for small x, and $\beta < b_c$, the initial guess σ_0 is given by inversion of b_{low} given in equation (3.3), which in turn is not that brilliant an approximation for small x and moderate $\beta < b_c$. However, for those parameter combinations, the approximation b_{high} seems to be a better approximation for the normalised Black function b, at least as long as $\beta > b(x, \sigma_{\text{high}}(x, 0, \theta), \theta)$ since $\sigma_{\text{high}}(x, 0, \theta)$ is the point below which $b_{\text{high}}(\cdot, \sigma, \cdot)$ as a function of σ gives negative values and thus is definitely inappropriate as an approximation for b. Wouldn't it be nice if we could somehow find a mixed form of σ_{low} and σ_{high} to provide a better initial guess than (3.5) when $\beta \in [0, b_c(x, \theta)]$? It turns out we can. The crucial idea here is *not* to try to find a better *invertible* approximation for $b(x, \sigma, \theta)$ on the interval $\sigma \in [0, \sigma_c]$ with $\sigma_c = \sqrt{2|x|}$, but to interpolate the two inversions σ_{low} and σ_{high} *directly*! We choose a functional form for interpolation in $\xi := {}^{\beta}/_{b_c}$ that transfers full weight from σ_{low} at $\xi = 0$ to σ_{high} at $\xi = 1$ in a fashion that some readers may recognize as *Gamma*-correction or *Gamma*-interpolation, namely

$$\sigma_{\text{interpolated}} \;:=\; \left(1 - w\!\left(\tfrac{\beta}{b_c}\right)\right) \cdot \sigma_{\text{low}} + w\!\left(\tfrac{\beta}{b_c}\right) \cdot \sigma_{\text{high}} \tag{4.1}$$

with $w(\xi) := \min(\xi^\gamma, 1)$, wherein we have omitted the dependency on (x, β, θ) for clarity, and with σ_{low} and σ_{high} given by equations (3.6) and (3.7), respectively. We choose γ such that the interpolation is exact when $\beta = b(x, \sigma_{\text{high}}(x, 0, \theta), \theta)$ by setting:

$$\sigma^* \;:=\; \sigma_{\text{high}}(x, 0, \theta) \tag{4.2}$$

$$b^* \;:=\; b(x, \sigma^*, \theta) \tag{4.3}$$

$$\sigma^*_{\text{low}} \;:=\; \sigma_{\text{low}}(x, b^*, \theta) \tag{4.4}$$

$$\sigma^*_{\text{high}} \;:=\; \sigma_{\text{high}}(x, b^*, \theta) \tag{4.5}$$

$$\gamma \;:=\; \ln\left(\frac{\sigma^* - \sigma^*_{\text{low}}}{\sigma^*_{\text{high}} - \sigma^*_{\text{low}}}\right) \Big/ \ln\left(\frac{b^*}{b_c}\right) \tag{4.6}$$

As it turns out, we cannot always use the above formulae as they stand. This is because, for very extreme ratios of forward and strike[7], it is possible that the functional form (3.3) for $b_{\text{low}}(x, \sigma, \theta)$ is not above, but below $b(x, \sigma, \theta)$, and as a consequence, here, we then may have $\sigma^* < \sigma^*_{\text{low}}$. That case, however, is benign since then σ_{low} is an excellent initial guess anyway, whence we use it when that happens. In summary, the initial guess can be written as

$$\sigma_{\text{interpolated}} \;=\; (1 - w^*) \cdot \sigma_{\text{low}} + w^* \cdot \sigma_{\text{high}} \tag{4.7}$$

$$w^* \;=\; \left(\min\left(\max\left(\frac{\sigma^* - \sigma^*_{\text{low}}}{\sigma^*_{\text{high}} - \sigma^*_{\text{low}}}, 0\right), 1\right)\right)^{\frac{\ln(b_c/\beta)}{\ln(b_c/b^*)}} . \tag{4.8}$$

How well this works is shown in figure 7. The improved initial guess formula (4.7) is the first, and most important, of three enhancements we present in this section that help to reduce the number of iterations required for a certain relative accuracy in implied Black volatility.

[7] Thanks to Chris Gardner for pointing this out when, for instance, $F = 1$ and $K = 10^6$.

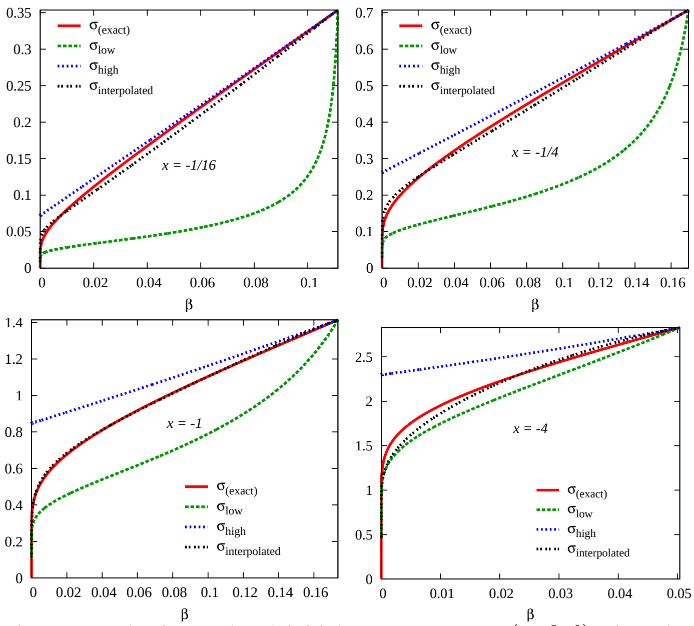

Figure 7: The interpolated initial guess $\sigma_{\text{interpolated}}(x, \beta, \theta)$ given by equation (4.7) for $\theta = 1$ and different x on $\beta \in [0, b_c(x, \theta)]$.

The second one is aimed at reducing the number of iterations needed for very small option values and very small values of x, i.e. near the money. It is motivated by looking at the asymptotic form (2.8) once again: the crucial term is $\mathrm{e}^{-({}^{x}/_{\sigma})^2}$. Taking the logarithm, gives it a hyperbolic form: $-({}^{x}/_{\sigma})^2$. At this point, we recall that most iterative algorithms are derived such that they work most efficiently when the objective function whose root is sought is well represented by a low order polynomial. Hyperbolic forms can be reasonably well represented. However, a hyperbolic form that is close to ${}^{1}/_{\sigma^2}$ is even better approximated by a low order polynomial if we take its reciprocal! This leads us to the objective function

$$f(\sigma) = \begin{cases} \frac{1}{\ln(b-\iota)} - \frac{1}{\ln(\beta-\iota)} & \text{if } \beta < b_c \\ b - \beta & \text{else} \end{cases} \tag{4.9}$$

and the Newton iteration becomes

$$\sigma_{n+1} \;=\; \sigma_n + \nu_n \tag{4.10}$$

with the Newton step $\nu_n = \nu(x, \sigma_n, \theta)$ given by

$$\nu(x, \sigma, \theta) = \begin{cases} \ln\left(\frac{\beta-\iota}{b-\iota}\right) \cdot \frac{\ln(b-\iota)}{\ln(\beta-\iota)} \cdot \frac{b-\iota}{b'} & \text{if } \beta < b_c \\ \frac{\beta-b}{b'} & \text{else} \end{cases} . \tag{4.11}$$

Finally, we pull our third and last trick which exploits the fact that the first and second derivative of $b(x, \sigma, \theta)$ with respect to σ have the comparatively simple relationship

$$\frac{b''}{b'} \;=\; \frac{x^2}{\sigma^3} - \frac{\sigma}{4} . \tag{4.12}$$

This means, that, whenever we have computed the value of the iteration's objective function and its Newton step, we can with comparatively little computational effort calculate the second order derivative terms required for the higher order iterative root finding algorithm known as *Halley's method* [Pic88]: no additional cumulative normals, no further inverse cumulative normals, not even exponentials are required! Halley's iterative method is in general given by adjusting the Newton step ν by a divisor of $1+\eta$ with $\eta = {}^{\nu}\!/\!_{2} \cdot {}^{f''}\!/\!_{f'}$. In our application to the calculation of the (normalised) Black implied volatility, we add some restrictions by capping and flooring the involved terms in order to avoid numerical round-off arising for extremely small values of x and σ sometimes leading to negative σ_n or excessive Halley steps:

$$\sigma_{n+1} = \sigma_n + \max\left(\frac{\hat{\nu}_n}{1+\hat{\eta}_n}, -\frac{\sigma_n}{2}\right) \tag{4.13}$$

$$\hat{\nu}_n := \max\left(\nu_n, -\frac{\sigma_n}{2}\right) \tag{4.14}$$

$$\hat{\eta}_n := \max\left(\frac{\hat{\nu}_n}{2}\frac{f''(x,\sigma_n,\theta)}{f'(x,\sigma_n,\theta)}, -\frac{3}{4}\right) \tag{4.15}$$

$$\frac{f''}{f'} = \frac{b''}{b'} - \frac{2+\ln(b-\iota)}{\ln(b-\iota)} \cdot \frac{b'}{b-\iota} \cdot \mathbf{1}_{\{\beta<b_c\}} \,. \tag{4.16}$$

All of these three enhancements together lead to the rapid convergence behaviour shown in figures 8 and 9.

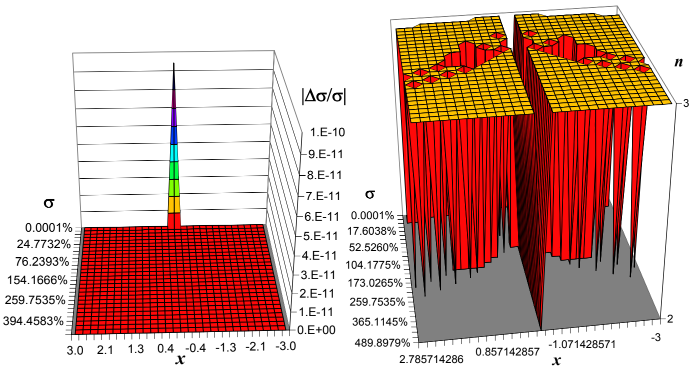

Figure 8: Left: residual relative error $|\Delta\sigma/\sigma| \approx |{}^{\sigma_{n+1}}\!/\!_{\sigma_n} - 1|$ after iterating (4.10) starting with initial guess given by (4.7) until $|{}^{\sigma_{n+1}}\!/\!_{\sigma_n} - 1| < \epsilon$ with $\epsilon = 10^{-8}$ on $(\sigma, x) \in [10^{-6}, 2\sqrt{2\cdot 3}] \times [-3, 3]$. Right: the number of iterations associated with the relative errors on the left.

5 Conclusion

For efficient and robust implied Black volatility calculation:-

1. Ensure you have a highly accurate cumulative and inverse cumulative normal function[8].
2. Transform input price p, forward F, and strike K to normalised coordinates $x = \ln {}^{F}\!/\!_{K}$ and $\beta = \frac{p}{\delta\sqrt{FK}}$ with δ being the discount factor to payment, annuity, or whichever other numéraire is used, respectively.

[8] such as the one published by Peter Acklam [Ack00]

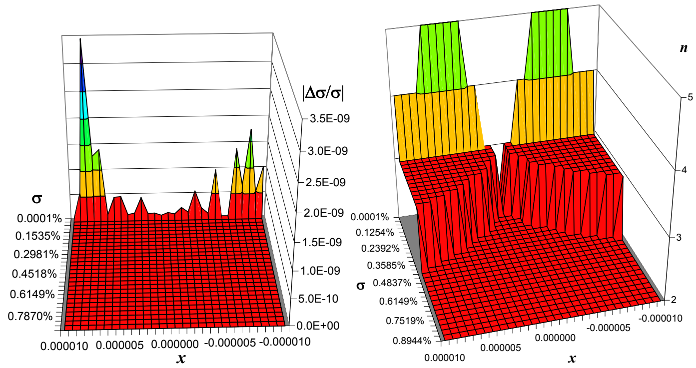

Figure 9: Left: residual relative error $|\Delta\sigma/\sigma| \approx |{}^{\sigma_{n+1}}\!/\!_{\sigma_n} - 1|$ after iterating (4.10) starting with initial guess given by (4.7) until $|{}^{\sigma_{n+1}}\!/\!_{\sigma_n} - 1| < \epsilon$ with $\epsilon = 10^{-8}$ on $(\sigma, x) \in [10^{-6}, 2\sqrt{2\cdot 10^{-5}}] \times [-10^{-5}, 10^{-5}]$. Right: the number of iterations associated with the relative errors on the left.

3. Ensure you only ever operate on out-of-the-money option prices, if necessary by subtracting the normalised intrinsic value $\iota = h(\theta x) \cdot \theta \cdot \left(\mathrm{e}^{x/2} - \mathrm{e}^{-x/2}\right)$ from β and switching $\theta \to 1 - 2h(x)$.
4. Use the initial guess formula (4.7) to start the iteration.
5. When the given normalised option price β is below the point of inflection $b_c(x,\theta)$ of the normalised Black formula, iterate to find the root of ${}^{1}\!/\!_{\ln b(x,\sigma,\theta)} - {}^{1}\!/\!_{\ln \beta}$, else of $b(x,\sigma,\theta) - \beta$, i.e. use the objective function (4.9).
6. Use Halley's method (4.13) with restricted stepsize.

References

[Ack00] P.J. Acklam. An algorithm for computing the inverse normal cumulative distribution function. `home.online.no/~pjacklam/notes/invnorm/index.html`, June 2000. University of Oslo, Statistics Division.

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

[Bla76] F. Black. The pricing of commodity contracts. *Journal of Financial Economics*, 3:167–179, 1976.

[BS73] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81:637–654, 1973.

[Mer73] R. C. Merton. Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science*, 4:141–183, Spring 1973.

[Pic88] C. A. Pickover. A note on chaos and Halley's method. *Communications of the ACM*, 31(11):1326–1329, November 1988.

[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`.

[Rub83] M. Rubinstein. Displaced diffusion option pricing. *Journal of Finance*, 38:213–217, March 1983.