Finite Difference Techniques for Arbitrage Free SABR

Fabien Le Floc'h* and Gary Kennedy†

* *Calypso Technology, 106 rue de La Boétie, 75008 Paris*
† *Clarus Financial Technology, London*

(v1.3 released June 2013)

Abstract *This paper presents various finite difference schemes applied to the SABR arbitrage free density problem with a focus on stability and speed.*

Key Words: stochastic volatility, SABR, TR-BDF2, Crank-Nicolson, finite difference, finance

1. Introduction

It is now well known that original SABR analytic formula from (Hagan *et al.*, 2002) used to compute option implied volatility is not arbitrage free as the probability density can become negative for low strikes and long maturities. Given the current low rates environment, many authors have proposed various improvements to the original formula (Oblój, 2008; Johnson and Nonas, 2009; Paulot, 2009; Benaim *et al.*, 2008). A single step finite difference method is proposed in (Andreasen and Huge, 2011) which leads to an arbitrage free 'SABR-like' model. Whilst the approach from Andreasen and Huge converges for short maturities to the original SABR analytic formula, it is (deliberately) different for longer expiries, even at the money.

Hagan recently proposed a new arbitrage free SABR solution, based on a finite difference discretisation of the probability density in (Hagan *et al.*, 2013). This approach provides a solution very close to the original SABR analytic formula, well known and widely used, while being arbitrage free, and thus allowing pricing with low rates. The authors use a Crank-Nicolson time-stepping scheme, which is known to have oscillation issues (Duffy, 2004; Giles and Carter, 2006) as it is only A-stable but not L-stable (LeVeque, 2007). We will show that this issue arises in the context of SABR pricing, and propose alternative schemes that are not very well known in computational finance, and yet are effective on this problem. One such scheme is TR-BDF2, used in the semiconductor industry as well as more generally in computational physics (Bank *et al.*, 1985; Bathe and Baig, 2005; Edwards *et al.*, 2011; Flavell *et al.*, 2013). Another scheme is due to Lawson and Swayne; whilst somewhat obscure, it is simple and effective on this problem (Lawson and Swayne, 1976).

Speed and accuracy were key ingredients in popularising the original SABR formula. Given that for a 30 year cap on a 3M LIBOR, there are potentially 119 PDEs to solve, we will focus our attention on the performance of the proposed schemes, as well as to what extent the discretisation grid can be reduced in size.

Correspondence Address: Calypso Technology, 106 rue de La Boétie, 75008 Paris. Email: fabien_lefloch@calypso.com

ISSN: Print/ISSN Online/YY/issnum1stp–ppcnt
DOI:

2. Arbitrage Free SABR

In (Hagan *et al.*, 2013), pricing with SABR parameters α, β, ρ, ν and forward f at τ_{ex} relies on the solution of a PDE on the probability density Q:

$$\frac{\partial Q}{\partial T}(T,F) = \frac{\partial^2 M(T,F)Q(T,F)}{\partial F^2} \text{ and } \begin{cases} \frac{\partial Q_L}{\partial T}(T) = \lim_{F\to F_{min}} \frac{\partial M(T,F)Q(T,F)}{\partial F} \\ \frac{\partial Q_R}{\partial T}(T) = \lim_{F\to F_{max}} \frac{\partial M(T,F)Q(T,F)}{\partial F} \end{cases} \tag{1}$$

with

$$M(T,F) = \frac{1}{2}\alpha^2(1+2\rho\nu z+\nu^2 z^2)e^{\rho\nu\alpha\Gamma T}C^2(F)\,,\, \Gamma(F) = \frac{C(F)-C(f)}{F-f} \tag{2}$$

$$z(F) = \frac{1}{\alpha(1-\beta)}(F^{1-\beta}-f^{1-\beta}) \qquad ,\, C(F) = F^\beta \tag{3}$$

It is suggested that the lower boundary F_{min} for the standard SABR model is placed at or near zero. However, the finite difference grid described in Appendix C of their paper starts at $F_0 = F_{min} - \frac{h}{2}$, where, h is the asset forward discretisation step size, potentially requiring the evaluation of functions not well-defined for negative values of F_0. Fortunately, only the product M_0Q_0 is used in the discretisation of equation (1) and it is entirely defined by M_1Q_1 because of the mirror-like boundary condition (imposed at the fictitious point F_0):

$$M_0Q_0 + M_1Q_1 = 0 \tag{4}$$

As long as $M_0 \neq 0$, M_0Q_0 will take the value $-M_1Q_1$. For example, we can use $|F_0|$ to compute M_0 and this will result in a symmetry around F_{min}.

Another alternative would be to place the grid so that $F_0 = F_{min}$ and use boundary condition $Q_0 = 0$ there. The probability of absorption Q_L could then be evaluated using an forward finite difference first derivative estimate. This would result in the exact same equation as (C.10a) of their paper and the scheme would still be moment preserving. However this comes at a cost of a slight loss of accuracy as, effectively, the derivative will be estimated using $Q_1 = Q(h)$ instead of $Q_1 = Q(\frac{h}{2})$.

The formula for Γ is also undefined for $F = f$, in which case we just use $\Gamma(f) = \frac{\partial C}{\partial F}(f)$.

3. Crank-Nicolson Oscillations with SABR

We use the same parameters as the example of negative density with the standard SABR formula in (Hagan *et al.*, 2013): $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%$ and forward $f = 1$ at $\tau_{ex} = 1$, a relatively fine discretisation in the rate dimension (500 points, that is $h = 0.01005$) and large time-steps (40 steps, that is $\delta = 0.025$). Hagan *et al.* (2013) recommend between 200 and 500 points and 30 to 100 time-steps.

Figure 1 shows strong oscillations around the forward. To guarantee the absence of oscillations, the *Courant number* should be small enough $\Psi = M\frac{\delta}{h^2} \leq 1$ (Theorem 2.2 in Morton and Mayers (2005)). This corresponds directly to the stability of the explicit Euler part of Crank-Nicolson. In practice, a higher value is acceptable because of a slight damping in Crank-Nicolson (Lawson and Morris, 1978). Although, M depends on F, we can use the at-the-money value at f, as the spike is located there; that is,

$$\Psi = \frac{1}{2}\alpha^2(1+2\rho\nu z+\nu^2 z^2)f^{2\beta}\frac{\delta}{h^2} \tag{5}$$

In our example, in Figures 1 and 2(a) $\Psi \approx 15$ while Figure 2(b) shows that indeed when $\Psi < 1$ there are no oscillations. In the next sections it is shown that a much smaller number

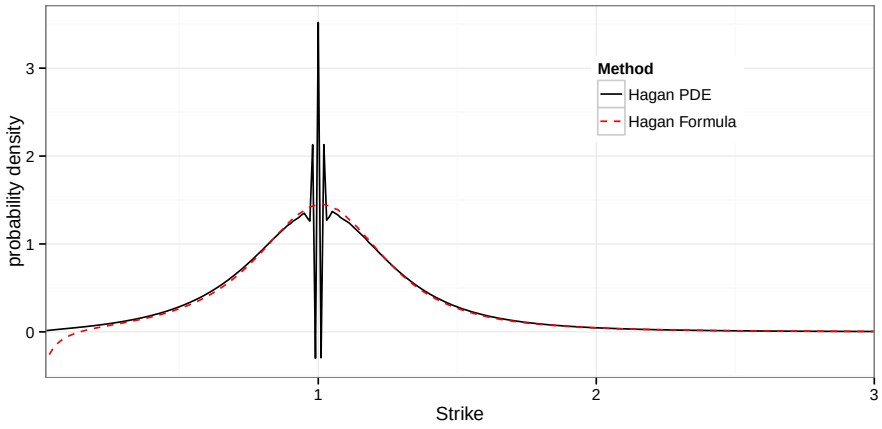
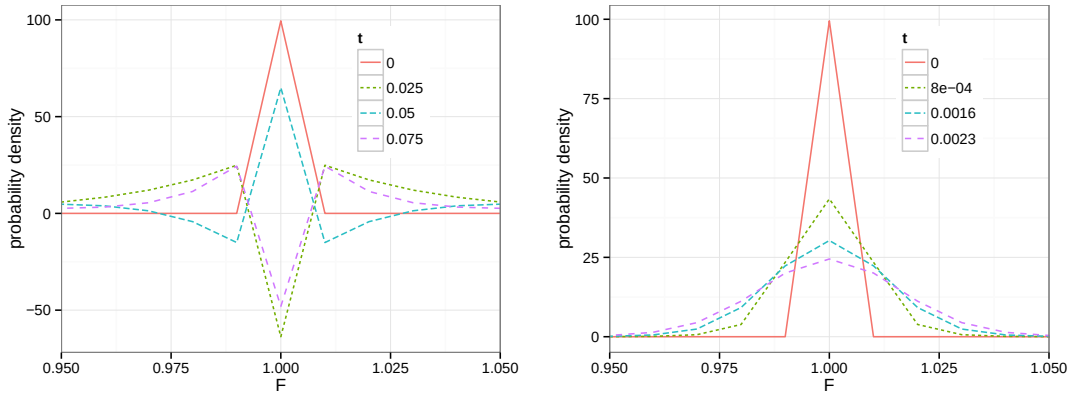


Figure 1.: Probability density in Hagan PDE discretised with Crank-Nicolson with 500 points and 40 time-steps. $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%, \tau_{ex} = 1$



(a) with 40 time steps $\Psi = 15.16$

(b) with 1280 time steps $\Psi = 0.47$

Figure 2.: First 4 time steps of the probability density in Hagan PDE discretised with Crank-Nicolson

of time steps can be used with other finite difference time-stepping techniques whilst still preserving good accuracy.

4. Alternative Schemes

The boundary condition described by equation (4) is applicable to all schemes considered in this section as it is independent of the time-stepping.

4.1 Rannacher

A common fix for Crank-Nicolson oscillations related to non smooth initial data is Rannacher time-stepping (Rannacher, 1984; Giles and Carter, 2006). It consists of introducing two half time-steps of implicit Euler time-stepping before applying Crank-Nicolson, because implicit Euler has much stronger damping properties. This comes at a cost in accuracy as implicit Euler is an order-1 scheme in time, especially when only a few time-steps are needed. The

SABR density discretisation will still be moment preserving if we discretise the Euler half steps as:

$$Q_j^{n+\frac{1}{2}} - Q_j^n = \frac{\delta}{2h^2}\left(M_{j+1}^{n+\frac{1}{2}}Q_{j+1}^{n+\frac{1}{2}} - 2M_j^{n+\frac{1}{2}}Q_j^{n+\frac{1}{2}} + M_{j-1}^{n+\frac{1}{2}}Q_{j-1}^{n+\frac{1}{2}}\right) \tag{6}$$

$$Q_L^{n+\frac{1}{2}} - Q_L^n = \frac{\delta}{2h}\left(M_1^{n+\frac{1}{2}}Q_1^{n+\frac{1}{2}} - M_0^{n+\frac{1}{2}}Q_0^{n+\frac{1}{2}}\right) \tag{7}$$

$$Q_R^{n+\frac{1}{2}} - Q_R^n = -\frac{\delta}{2h}\left(M_{J+1}^{n+\frac{1}{2}}Q_{J+1}^{n+\frac{1}{2}} - M_J^{n+\frac{1}{2}}Q_J^{n+\frac{1}{2}}\right) \tag{8}$$

for $j = 1, ..., J$ and $n = 0, \frac{1}{2}, 1, \frac{3}{2}$.

4.2 Implicit Richardson Extrapolation

A simple Richardson extrapolation in time (Richardson, 1911) on implicit Euler will also provide a nearly order-2 scheme in time, keeping strong damping properties of the implicit Euler scheme at the cost of increased computational load: the implicit Euler scheme (Equations 6, 7) is evaluated with $\frac{\delta}{2}$ and δ. In practice it is around twice as slow as Crank-Nicolson. At $T = N\delta = \tau_{ex}$, we apply:

$$Q(F) = 2\hat{Q}^{\frac{\delta}{2}}(F) - \hat{Q}^{\delta}(F) \tag{9}$$

$$Q_L = 2\hat{Q}_L^{\frac{\delta}{2}} - \hat{Q}_L^{\delta} \tag{10}$$

$$Q_R = 2\hat{Q}_R^{\frac{\delta}{2}} - \hat{Q}_R^{\delta} \tag{11}$$

where $\hat{Q}^{\delta}$ is Q computed by implicit Euler with a time step of δ.

4.3 Lawson-Morris-Gourlay

A local Richardson extrapolation in time of second and third order is proposed in (Lawson and Morris, 1978) and (Gourlay and Morris, 1980). In practice, it is a faster alternative to the standard Richardson extrapolation because the tridiagonal matrix stemming out of the finite difference discretisation can be reused, while keeping L-stability and thus strong damping properties.

For the second order scheme, at each time-step, equation (9) is applied. For the third order scheme, at each time-step we apply:

$$Q(F) = 4.5\hat{Q}^{\frac{\delta}{3}}(F) - 4.5\hat{Q}^{\frac{2\delta}{3}}(F) + \hat{Q}^{\delta}(F) \tag{12}$$

$$Q_L = 4.5\hat{Q}_L^{\frac{\delta}{3}} - 4.5\hat{Q}_L^{\frac{2\delta}{3}} + \hat{Q}_L^{\delta} \tag{13}$$

$$Q_R = 4.5\hat{Q}_R^{\frac{\delta}{3}} - 4.5\hat{Q}_R^{\frac{2\delta}{3}} + \hat{Q}_R^{\delta} \tag{14}$$

where $\hat{Q}^{\frac{\delta}{3}}$ is Q computed by implicit Euler with 3 time steps of $\frac{\delta}{3}$ and $\hat{Q}^{\frac{2\delta}{3}}$ is Q computed by implicit Euler with a time step of $\frac{2\delta}{3}$ and $\frac{\delta}{3}$. Being linear combinations of implicit Euler, those schemes are moment preserving.

4.4 Lawson-Swayne

A slightly faster second order unconditionally stable scheme is presented as a remedy to Crank-Nicolson in (Lawson and Swayne, 1976; Lawson and Morris, 1978). Let $b = 1 - \frac{\sqrt{2}}{2}$, it consists in applying two implicit Euler steps with time-step of $b\delta$ and an extrapolation on

the values at those two steps.

First stage:

$$\begin{aligned} Q_j^{n+b} - Q_j^n &= \frac{b\delta}{h^2}\left(M_{j+1}^{n+b}Q_{j+1}^{n+b} - 2M_j^{n+b}Q_j^{n+b} + M_{j-1}^{n+b}Q_{j-1}^{n+b}\right) \\ Q_L^{n+b} - Q_L^n &= \frac{b\delta}{h}\left(M_1^{n+b}Q_1^{n+b} - M_0^{n+b}Q_0^{n+b}\right) \\ Q_R^{n+b} - Q_R^n &= -\frac{b\delta}{h}\left(M_{J+1}^{n+b}Q_{J+1}^{n+b} - M_J^{n+b}Q_J^{n+b}\right) \end{aligned} \tag{15a}$$

Second stage:

$$\begin{aligned} Q_j^{n+2b} - Q_j^{n+b} &= \frac{b\delta}{h^2}\left(M_{j+1}^{n+2b}Q_{j+1}^{n+2b} - 2M_j^{n+2b}Q_j^{n+2b} + M_{j-1}^{n+2b}Q_{j-1}^{n+2b}\right) \\ Q_L^{n+2b} - Q_L^{n+b} &= \frac{b\delta}{h}\left(M_1^{n+2b}Q_1^{n+2b} - M_0^{n+2b}Q_0^{n+2b}\right) \\ Q_R^{n+2b} - Q_R^{n+b} &= -\frac{b\delta}{h}\left(M_{J+1}^{n+2b}Q_{J+1}^{n+2b} - M_J^{n+2b}Q_J^{n+2b}\right) \end{aligned} \tag{15b}$$

And finally:

$$\begin{aligned} Q_j^{n+1} &= (\sqrt{2}+1)Q_j^{n+2b} - \sqrt{2}Q_j^{n+b} \\ Q_L^{n+1} &= (\sqrt{2}+1)Q_L^{n+2b} - \sqrt{2}Q_L^{n+b} \\ Q_R^{n+1} &= (\sqrt{2}+1)Q_R^{n+2b} - \sqrt{2}Q_R^{n+b} \end{aligned} \tag{15c}$$

for $j = 1, ..., J$ and $n = 0, ..., N-1$.

The scheme is moment preserving as it can also be seen as a linear combination of implicit Euler schemes.

4.5 TR-BDF2

TR-BDF2 is a two-stage method where the first stage consists in applying the (weighted) trapezoidal rule (Crank-Nicolson) and the second stage consists in applying the second order backward difference scheme (BDF2) on the first stage result and the first stage initial input (Bank *et al.*, 1985; LeVeque, 2007). It is second order accurate in time and L-stable. It is not to be confused with the simpler multistep method BDF2: the full step only depends on the previous full step while BDF2 depends on the two previous timesteps and can lose its accuracy (Windcliff *et al.*, 2001) with variable timesteps and linear complimentary problems. This scheme does not suffer from such drawbacks. The scheme has been applied to finance in the context of American option pricing (Le Floc'h, 2013).

$$Q^{n+\alpha} = Q^n + \frac{\alpha\delta}{2}\left(\frac{\partial^2 M^n Q^n}{\partial F^2} + \frac{\partial^2 M^{n+\alpha}Q^{n+\alpha}}{\partial F^2}\right) \tag{16a}$$

$$Q^{n+1} = \frac{1}{2-\alpha}\left(\frac{1}{\alpha}Q^{n+\alpha} - \frac{(1-\alpha)^2}{\alpha}Q^n + \delta(1-\alpha)\frac{\partial^2 M^{n+1}Q^{n+1}}{\partial F^2}\right) \tag{16b}$$

The weight α can be chosen to match Crank-Nicolson ($\alpha = \frac{1}{2}$) or to have proportional Jacobians ($\alpha = 2 - \sqrt{2}$). The later provides optimal stability (Dharmaraja *et al.*, 2009).

This can be extended to three-stages, with two stages of the trapezoidal rule and one stage of third order backward difference scheme (BDF3) as in (Bathe and Baig, 2005), resulting in

a method with even stronger damping properties:

$$Q^{n+\frac{1}{3}} = Q^n + \frac{\delta}{6}\left(\frac{\partial^2 M^n Q^n}{\partial F^2} + \frac{\partial^2 M^{n+\frac{1}{3}} Q^{n+\frac{1}{3}}}{\partial F^2}\right) \tag{17a}$$

$$Q^{n+\frac{2}{3}} = Q^n + \frac{\delta}{6}\left(\frac{\partial^2 M^{n+\frac{1}{3}} Q^{n+\frac{1}{3}}}{\partial F^2} + \frac{\partial^2 M^{n+\frac{2}{3}} Q^{n+\frac{2}{3}}}{\partial F^2}\right) \tag{17b}$$

$$Q^{n+1} = \frac{1}{11}\left(18Q^{n+\frac{2}{3}} - 9Q^{n+\frac{1}{3}} + 2Q^n + 2\delta\frac{\partial^2 M^{n+1} Q^{n+1}}{\partial F^2}\right) \tag{17c}$$

4.6 Smoothing before Crank-Nicolson

Similarly to Rannacher idea, we could use Lawson-Morris-Gourlay for the first few timesteps in order to dampen the oscillations before applying Crank-Nicolson. This ensures an overall order-2 accuracy, even with a low number of time-steps. TR-BDF2 and TR-BDF3 fit particularly well as the first stages are a weighted Crank-Nicolson. So one could stop doing the BDF2 (respectively BDF3) stages when the solution is sufficiently smooth.

We found by experimentation that Crank-Nicolson, dampened by 4 initial steps of TR-BDF2 or Lawson-Swayne was very efficient, but we are less comfortable recommending such a hybrid scheme without further analysis.

4.7 Optimising for Performance

The function $M(F,T)$ needs to be computed for every grid point (F_j, t_n). The performance of the overall algorithm can be greatly improved by minimising the calls to the `pow` and `exp` functions as those are expensive. The quantities $\frac{1}{2}\alpha^2(1+2\rho\nu z+\nu^2 z^2)C^2(F)$ and $\rho\nu\alpha\Gamma$ are constant in time and can be thus be cached between time-steps. A further improvement is to decompose t_{n+1} as $t_n + \delta$, then

$$e^{\rho\nu\alpha\Gamma t_{n+1}} = e^{\rho\nu\alpha\Gamma t_n} e^{\rho\nu\alpha\Gamma\delta} \tag{18}$$

We can therefore just compute the initial value $\frac{1}{2}\alpha^2(1+2\rho\nu z+\nu^2 z^2)C^2(F)$ together with $e_j = e^{\rho\nu\alpha\Gamma(F_j)\delta}$ for $j = 0, ..., J+1$ once, and at each step simply update M as:

$$M_j^{n+1} = e_j M_j^n \tag{19}$$

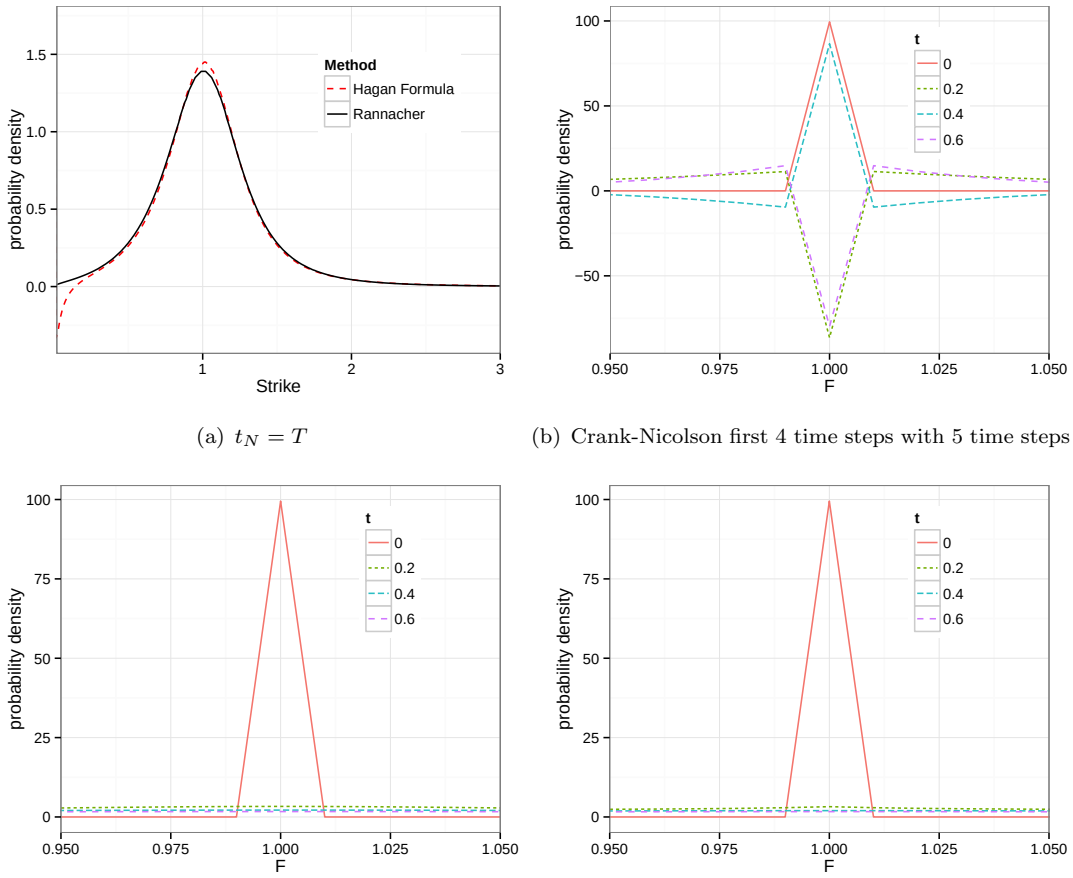
This can be easily extended to multiple time-step sizes used in multi-stage schemes.

For multi-stage schemes, it is also possible to consider M as piecewise constant between full time-steps and thus to avoid its computation for fractions of time-steps. In our tests, this led to a slightly decreased accuracy and little performance gain. The increase in error was particularly visible for long term options and large time-steps. We did not make that approximation for the tests presented in the next section.

5. Numerical Results

5.1 Oscillations

With the same parameters as in section 3, Figure 3(a) shows a smooth positive probability density using only a total of 5 time-steps when Rannacher smoothing is applied to Crank-Nicolson. The density computed using second or third order Lawson-Morris-Gourlay (LMG2, LMG3), Lawson-Swayne (LS), TR-BDF2 or Richardson extrapolation on implicit Euler would



(a) $t_N = T$

(b) Crank-Nicolson first 4 time steps with 5 time steps

(c) Rannacher first 4 time steps with 5 time steps

(d) LMG3 first 4 time steps with 5 time steps

Figure 3.: Probability density in Hagan PDE using a total of 5 time-steps

look very similar. Figures 3(c) and 3(d) show no apparent oscillations in the first steps. In contrast, Crank-Nicolson had strong oscillations visible at $T = \tau_{ex}$ with 40 time steps.

5.2 Performance

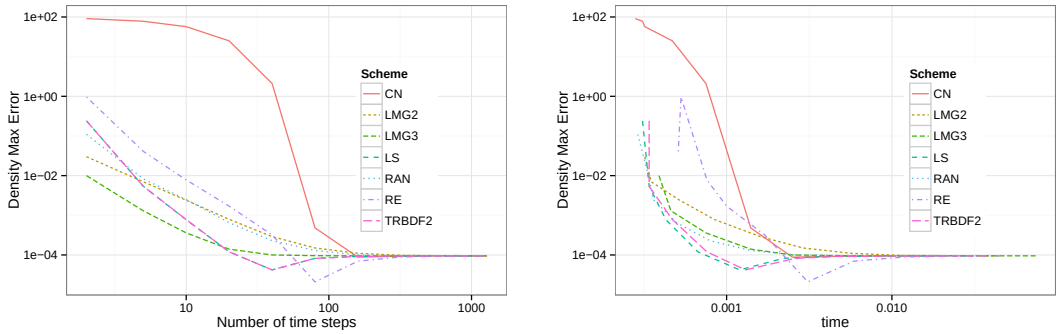
5.2.1 Hagan Example

With the same parameters as in section 3, we look at the maximum error in the probability density with a varying number of time-steps compared to a Crank-Nicolson scheme with 5120 points for the rate dimension and 81920 time-steps.

Other tests we performed indicate that the implied volatility maximum error or even the at-the-money implied volatility error would lead to similar conclusions. Furthermore, a Black implied volatility with an absolute error under 0.1% was achieved with only 2 time steps for LMG3, Lawson-Swayne and TR-BDF2, 5 time steps for LMG2, and 10 steps for RAN. Lawson-Swayne is the most efficient scheme on this problem, closely followed by TR-BDF2, Rannacher and LMG3.

5.2.2 Andreasen-Huge Example

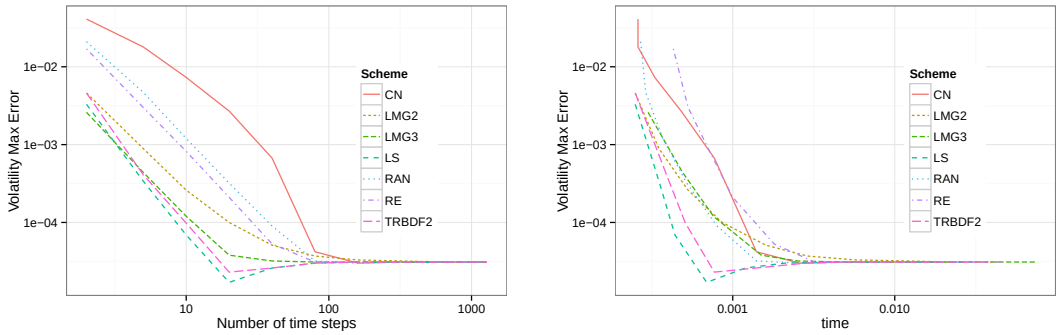
We consider the SABR parameters used in (Andreasen and Huge, 2011): $\alpha = 0.0873, \beta = 0.7, \rho = -0.48, \nu = 0.47$ with a forward of $f = 0.025$ and a maturity $\tau_{ex} = 10.0$ and look at



(a) Accuracy vs. number of time steps (b) Accuracy vs. time

Figure 4.: Performance on Hagan example

the maximum error in implied volatility between $0.2f$ and $2f$.



(a) Accuracy vs. number of time steps (b) Accuracy vs. time

Figure 5.: Performance on Andreasen-Huge example

Only 2 time-steps are enough with LMG3 for good accuracy, and 5 for LMG2, LS, RAN and TR-BDF2.

6. Conclusion

It is possible to accurately compute option prices under the arbitrage free SABR approach with very few time-steps, even for long maturities. The Rannacher smoothing is a particularly simple and efficient way to improve accuracy significantly compared to Crank-Nicolson on this problem. Other schemes such as TR-BDF2 or Lawson-Swayne can be even more efficient. In our experiments TR-BDF2 and Lawson-Swayne were robust and had similar stability and convergence properties.

Thus, with a careful choice of finite difference scheme, (Hagan *et al.*, 2013) is particularly competitive to the one step finite difference approach of (Andreasen and Huge, 2011).

References

Andreasen, J. and Huge, B. (2011) ZABR–Expansions for the Masses, *Available at SSRN 1980726*.

Bank, R., Coughran, W., Fichtner, W., Grosse, E., Rose, D. and Smith, R. (1985) Transient simulation of silicon devices and circuits, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 4(4), pp. 436–451.

Bathe, K. J. and Baig, M. M. I. (2005) On a composite implicit time integration procedure for nonlinear dynamics, *Computers & Structures*, 83(31), pp. 2513–2524.

Benaim, S., Dodgson, M. and Kainth, D., An arbitrage-free method for smile extrapolation. (2008) , Technical report, Working Paper, QuaRC, Royal Bank of Scotland.

Dharmaraja, S., Wang, Y. and Strang, G. (2009) Optimal stability for trapezoidal-backward difference split-steps, *IMA Journal of Numerical Analysis*.

Duffy, D. (2004) A Critique of the Crank Nicolson Scheme Strengths and Weaknesses for Financial Instrument Pricing, *Wilmott Magazine*.

Edwards, J. D., Morel, J. E. and Knoll, D. A. (2011) Nonlinear variants of the TR/BDF2 method for thermal radiative diffusion, *Journal of Computational Physics*, 230(4), pp. 1198–1214.

Flavell, A., Machen, M., Eisenberg, B., Liu, C. and Li, X. (2013) A Conservative Finite Difference Scheme for Poisson-Nernst-Planck Equations, *arXiv preprint arXiv:1303.3769*.

Giles, M. and Carter, R. (2006) Convergence analysis of Crank-Nicolson and Rannacher time-marching, *Journal of Computational Finance*, 9(4), p. 89.

Gourlay, A. and Morris, J. L. (1980) The extrapolation of first order methods for parabolic partial differential equations, II, *SIAM Journal on Numerical Analysis*, 17(5), pp. 641–655.

Hagan, P. S., Kumar, D., Lesniewski, A. S. and Woodward, D. E. (2002) Managing smile risk, *Wilmott magazine*.

Hagan, P. S., Kumar, D., Lesniewski, A. S. and Woodward, D. E. (2013) Arbitrage free SABR, *Not published*.

Johnson, S. and Nonas, B. (2009) Arbitrage-free construction of the swaption cube, *Wilmott Journal*, 1(3), pp. 137–143.

Lawson, J. D. and Morris, J. L. (1978) The extrapolation of first order methods for parabolic partial differential equations. I, *SIAM Journal on Numerical Analysis*, 15(6), pp. 1212–1224.

Lawson, J. and Swayne, D. (1976) A simple efficient algorithm for the solution of heat conduction problems. In: *Proc. 6th Manitoba Conf. Numer. Math*, pp. 239–250.

Le Floc'h, F. (2013) TR-BDF2 for Stable American Option Pricing, *Journal of Computational Finance (to appear)*.

LeVeque, R. J. (2007) *Finite Difference Methods for Ordinary and Partial Differential Equations*, (Society for Industrial and Applied Mathematics (SIAM)).

Morton, K. W. and Mayers, D. F. (2005) *Numerical solution of partial differential equations: an introduction*, (Cambridge university press).

Oblój, J. (2008) Fine-tune your smile: Correction to Hagan et al, *Wilmott Magazine*.

Paulot, L. (2009) Asymptotic implied volatility at the second order with application to the SABR model, *Available at SSRN 1413649*.

Rannacher, R. (1984) Finite element solution of diffusion problems with irregular data, *Numerische Mathematik*, 43(2), pp. 309–327.

Richardson, L. F. (1911) The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210, pp. 307–357.

Windcliff, H., Forsyth, P. and Vetzal, K. (2001) Shout options: A framework for pricing contracts which can be modified by the investor, *Journal of Computational and Applied Mathematics*, 134(1-2), pp. 213–241.

Appendix A. Moment conservation in Implicit Euler

We have for implicit Euler:

$$Q_j^{n+1} - \frac{\delta}{h^2}\left(M_{j+1}^{n+1}Q_{j+1}^{n+1} - 2M_j^{n+1}Q_j^{n+1} + M_{j-1}^{n+1}Q_{j-1}^{n+1}\right) = Q_j^n \tag{A1}$$

$$Q_L^{n+1} - Q_L^n = \frac{\delta}{h}\left(M_1^{n+1}Q_1^{n+1} - M_0^{n+1}Q_0^{n+1}\right) \tag{A2}$$

$$Q_R^{n+1} - Q_R^n = -\frac{\delta}{h}\left(M_{J+1}^{n+1}Q_{J+1}^{n+1} - M_J^{n+1}Q_J^{n+1}\right) \tag{A3}$$

with absorbing boundary conditions

$$M_0^{n+1}Q_0^{n+1} + M_1^{n+1}Q_1^{n+1} = 0$$

$$M_{J+1}^{n+1}Q_{J+1}^{n+1} + M_J^{n+1}Q_J^{n+1} = 0$$

Summing over j yields:

$$\sum_{j=1}^{J} h(Q_j^{n+1} - Q_j^n) = \frac{\delta}{h}\left(M_{J+1}^{n+1}Q_{J+1}^{n+1} - M_J^{n+1}Q_J^{n+1} - M_1^{n+1}Q_1^{n+1} + M_0^{n+1}Q_0^{n+1}\right) \tag{A4}$$

$$= -(Q_L^{n+1} - Q_L^n) - (Q_R^{n+1} - Q_R^n) \tag{A5}$$

The total probability follows:

$$Q_L^{n+1} + \sum_{j=1}^{J} hQ_j^{n+1} + Q_R^{n+1} = Q_L^n + \sum_{j=1}^{J} hQ_j^n + Q_R^n \tag{A6}$$

The total probability is conserved from one time-step to another.

Let's look now at the conservation of the forward. We multiply equation by F_j and sum, using the fact that the forward obeys $F_{j+1} - 2F_j + F_{j-1} = 0$:

$$\sum_{j=1}^{J} hF_j(Q_j^{n+1} - Q_j^n) = \tag{A7}$$

$$\frac{\delta}{h}\left(F_J M_{J+1}^{n+1}Q_{J+1}^{n+1} - F_{J+1}M_J^{n+1}Q_J^{n+1} - F_0 M_1^{n+1}Q_1^{n+1} + F_1 M_0^{n+1}Q_0^{n+1}\right) \tag{A8}$$

$$F_{min}(Q_L^{n+1} - Q_L^n) + \sum_{j=1}^{J} hF_j(Q_j^{n+1} - Q_j^n) + F_{max}(Q_R^{n+1} - Q_R^n) = \tag{A9}$$

$$\frac{\delta}{2}\left(M_1^{n+1}Q_1^{n+1} + M_0^{n+1}Q_0^{n+1}\right) - \frac{\delta}{2}\left(M_{J+1}^{n+1}Q_{J+1}^{n+1} + M_J^{n+1}Q_J^{n+1}\right) \tag{A10}$$

The boundary conditions ensures that the right hand side is 0, and therefore the forward is conserved between time-steps.

Appendix B. Moment conservation in TR-BDF2

As the first stage is just Crank-Nicolson applied to a fraction of time-step, (Hagan *et al.*, 2013) already proved the first stage is moment preserving. We will now prove that the BDF2 stage is also moment preserving. We use $n + \alpha$ to represent the value at $t_n + \alpha\delta$. We have for the second stage:

$$(2-\alpha)Q_j^{n+1} - \frac{1}{\alpha}Q_j^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_j^n = \delta(1-\alpha)\frac{\partial^2}{\partial F^2}M^{n+1}Q^{n+1}$$

$$= (1-\alpha)\frac{\delta}{h^2}\left(M_{j+1}^{n+1}Q_{j+1}^{n+1} - 2M_j^{n+1}Q_j^{n+1} + M_{j-1}^{n+1}Q_{j-1}^{n+1}\right)$$

with absorbing boundary conditions

$$M_0^{n+1}Q_0^{n+1} + M_1^{n+1}Q_1^{n+1} = 0$$

$$M_{J+1}^{n+1}Q_{J+1}^{n+1} + M_J^{n+1}Q_J^{n+1} = 0$$

After Q_j^{n+1} is updated, one can update Q_L^{n+1} and Q_R^{n+1}

$$(2-\alpha)Q_L^{n+1} - \frac{1}{\alpha}Q_L^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_L^n = (1-\alpha)\frac{\delta}{h}\left(M_1^{n+1}Q_1^{n+1} - M_0^{n+1}Q_0^{n+1}\right)$$

$$(2-\alpha)Q_R^{n+1} - \frac{1}{\alpha}Q_R^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_R^n = -(1-\alpha)\frac{\delta}{h}\left(M_{J+1}^{n+1}Q_{J+1}^{n+1} - M_J^{n+1}Q_J^{n+1}\right)$$

Let's look at the total probability conservation:

$$h\sum_{j=1}^{J}\left[(2-\alpha)Q_j^{n+1} - \frac{1}{\alpha}Q_j^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_j^n\right] \tag{B1}$$

$$= (1-\alpha)\frac{\delta}{h}\left(M_{J+1}^nQ_{J+1}^n - M_J^nQ_J^n - M_1^nQ_1^n + M_0^nQ_0^n\right) \tag{B2}$$

$$= -\left[(2-\alpha)Q_L^{n+1} - \frac{1}{\alpha}Q_L^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_L^n\right] \tag{B3}$$

$$-\left[(2-\alpha)Q_R^{n+1} - \frac{1}{\alpha}Q_R^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_R^n\right] \tag{B4}$$

in which case, since we already confirmed that total probability was preserved at $Q^{n+\alpha}$, we can see that the BDF2 stage will also preserve total probability to Q^n.

$$(2-\alpha)\left[Q_L^{n+1} + h\sum_{j=1}^{J}Q_j^{n+1} + Q_R^{n+1}\right] \tag{B5}$$

$$= \frac{1}{\alpha}\left[Q_L^{n+\alpha} + h\sum_{j=1}^{J}Q_j^{n+\alpha} + Q_R^{n+\alpha}\right] \tag{B6}$$

$$-\frac{(1-\alpha)^2}{\alpha}\left[Q_L^n + h\sum_{j=1}^{J}Q_j^n + Q_R^n\right] \tag{B7}$$

$$= (2-\alpha) \tag{B8}$$

Now recall that our grid has even spacing, and so $F_{j+1} - 2F_j + F_{j-1} = 0$. Then

$$h\sum_{j=1}^{J} F_j\left[(2-\alpha)Q_j^{n+1} - \frac{1}{\alpha}Q_j^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_j^n\right] \tag{B9}$$

$$= (1-\alpha)\frac{\delta}{h}\left(F_J M_{J+1}^{n+1}Q_{J+1}^{n+1} - F_{J+1}M_J^{n+1}Q_J^{n+1}\right) \tag{B10}$$

$$+(1-\alpha)\frac{\delta}{h}\left(F_1 M_0^{n+1}Q_0^{n+1} - F_0 M_1^{n+1}Q_1^{n+1}\right) \tag{B11}$$

Furthermore, recalling that $F_{min} = \frac{1}{2}(F_0 + F_1)$ and $F_{max} = \frac{1}{2}(F_J + F_{J+1})$ and $F_1 - F_0 = F_{J+1} - F_J = h$,

$$F_{min}\left[(2-\alpha)Q_L^{n+1} - \frac{1}{\alpha}Q_L^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_L^n\right] \tag{B12}$$

$$+ h\sum_{j=1}^{J} F_j\left[(2-\alpha)Q_j^{n+1} - \frac{1}{\alpha}Q_j^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_j^n\right] \tag{B13}$$

$$+ F_{max}\left[(2-\alpha)Q_R^{n+1} - \frac{1}{\alpha}Q_R^{n+\alpha} + \frac{(1-\alpha)^2}{\alpha}Q_R^n\right] \tag{B14}$$

$$= (1-\alpha)\frac{\delta}{2}\left(M_1^{n+1}Q_1^{n+1} + M_0^{n+1}Q_0^{n+1}\right) \tag{B15}$$

$$+(1-\alpha)\frac{\delta}{2}\left(M_{J+1}^{n+1}Q_{J+1}^{n+1} + M_J^{n+1}Q_J^{n+1}\right) \tag{B16}$$

$$= 0 \tag{B17}$$

Collecting terms, we see that

$$(2-\alpha)\left[F_{min}Q_L^{n+1} + h\sum_{j=1}^{J} F_j Q_j^{n+1} + F_{max}Q_R^{n+1}\right] = \frac{1}{\alpha}f - \frac{(1-\alpha)^2}{\alpha}f$$

and so the forward is preserved by BDF2 at t_{n+1}.

Appendix C. Reference Implementation Values

Scheme	ATM price	Q(f)	QL	QR
CN	0.155491886707	-76.222597308083	0.036145997780	0.000811969902
RAN	0.149165623132	1.390318228263	0.037030534101	0.001026159943
RE	0.150061501089	1.342391047522	0.036966009503	0.000850746756
LMG2	0.149448704254	1.390737156096	0.037351038244	0.000808345304
LMG3	0.149595211756	1.385108845032	0.036878097804	0.000775853690
LS	0.149701563313	1.378405046490	0.036466946406	0.000797983056
TRBDF2	0.149703134940	1.378343390764	0.036463543893	0.000797557279
TRBDF3	0.149630615131	1.390034574220	0.036719878912	0.000785705142

Table C1.: Sample values using 500 points and 5 time-steps, $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%, T = 1, f = 1, F_{min} = 0.0, F_{max} \approx 5, \delta = 0.2, h = 0.010050251256$

Checking the moments at every step is very useful way to validate a scheme's implementation in code. To support further validation we recorded specific values from our implementations as reference. The TR-BDF2 is using a value of $\alpha = 2 - \sqrt{2}$ whilst the RAN scheme is using Rannacher time-stepping for the first two full time-steps.

Appendix D. Example Code

For illustration purpose, we detail here Matlab code (also working with Octave) for pricing vanilla options under the Arbitrage Free SABR model using Lawson-Swayne method. The performance numbers in this paper come from an optimized Scala implementation, not from this Scilab code.

Listing 1: makeDensityLawsonSwayne.m - Matlab/Octave code computing the arbitrage free SABR density with Lawson-Swayne

```
function [Q,QL,QR, F, Fmin, Fmax, h] = makeDensityLawsonSwayne(alpha, beta, nu, rho
    , forward, T, N, timesteps, Fmin, Fmax)
  %init h, F and Q
  h0 = (Fmax-Fmin)/N;
  j0 = (forward-Fmin)/h0;
  h = (forward-Fmin)/(j0-0.5);
  F = Fmin-0.5*h:h:Fmin+(N-1.5)*h;
  Fmax = Fmin + (N-2)*h;
  Q = zeros(N,1); Q(j0+1,1)=1.0/h;
  dt = T/timesteps;
  %Lawson Swayne params
  b = 1 - sqrt(2)/2;
  dt1 = dt*b; dt2 = dt*(1-2*b);
  %compute cache
  forwardonebeta = forward.^(1-beta);
  C0 = forward.^beta;
  f = abs(F);
  C = f.^beta;
  fonebeta = f./C;
  z = (fonebeta - forwardonebeta)/(alpha*(1-beta));
  M = C.*C*0.5*alpha*alpha.*(1+2*rho*nu*z+nu.^2*z.^2);
  gam = (C - C0) ./ (f - forward); gam(j0+1) = beta/forwardonebeta;
  M1cache = exp(rho*nu*alpha*gam*dt1); M2cache = exp(rho*nu*alpha*gam*dt2);
  %solve
  QL = 0.0; QR = 0.0;
  for t = 1:timesteps
    [M, Q1, QL1, QR1] = solveStep(M, M1cache, dt1, h, Q, QL, QR);
    [M, Q2, QL2, QR2] = solveStep(M, M1cache, dt1, h, Q1, QL1, QR1);
    Q=(sqrt(2)+1)*Q2-sqrt(2)*Q1;
    QL=(sqrt(2)+1)*QL2-sqrt(2)*QL1;
    QR=(sqrt(2)+1)*QR2-sqrt(2)*QR1;
    M = M.*M2cache;
  end
end
function [M, Q1, QL1, QR1] = solveStep(M, M1cache, dt1, h, Q, QL, QR)
  M = M .* M1cache;
  frac = dt1/(h.^2); N = length(Q);
  tri = diag(sparse(1+2*frac*M))+diag(sparse(-frac*M(1:N-1)),-1)+diag(sparse(-frac*
      M(2:N)),1);
  tri(1,:) = zeros(1,N); tri(1,1:2)= [M(1) M(2)];
  tri(N,:)= zeros(1,N); tri(N,N-1:N)=[M(end-1) M(end)];
  Q(1) = 0; Q(N) = 0;
  Q1 = tri\Q;
  QL1 = QL + dt1/h*(M(2)*Q1(2)-M(1)*Q1(1));
  QR1 = QR - dt1/h*(M(N)*Q1(N)-M(N-1)*Q1(N-1));
end
```

Listing 2: priceCallSABRDensity.m - price a call option using Arbitrage Free SABR density

```
function p = priceCallSABRDensity(strike, Q, QL,QR, Fmin, Fmax, h)
    if (strike < Fmin)
      p = forward-strike;
    else
      if (strike > Fmax)
        p = 0
      else
        k0 = ceil((strike-Fmin)/h);
        term = Fmin+k0*h-strike;
        p = 0.5 * term * term * Q(k0+1) + (Fmax - strike) * QR;
        k=k0+1:length(Q)-2;
        p = p + sum((Fmin + (k - 0.5) * h - strike) * h .* Q(k+1));
      end
    end
end
```