Fabien Le Floc'h
Financial engineer at Calypso Technology

Calypso Technology,
16-18 Rue du 4 Septembre,
75002 Paris,
France.

e-mail: fabien_lefloch@calypso.com
phone: +33 (0)1 44 50 01 84

Gary Kennedy
Founder and chief operating officer at Clarus Financial Technology

Clarus Financial Technology,
Suite 204, Russell Business Centre,
40-42 Lisburn Road,
Belfast BT9 6AA

e-mail: gary@clarusft.com

Finite Difference Techniques for Arbitrage-Free SABR

Special issue Lorentz workshop

Number of words: 7321,
number of figures: 10,
number of tables: 4.

Electronic copy available at: https://ssrn.com/abstract=2402001

Finite Difference Techniques for Arbitrage-Free SABR

Fabien Le Floc'h[⋆] and Gary Kennedy[†]

[⋆] *Calypso Technology, 16-18 Rue du 4 Septembre, 75002 Paris, France*
[†] *Clarus Financial Technology, Belfast*

(v1.0 released October 2015)

Abstract *In the current low rates environment, the classic SABR formula used to compute option implied volatilities lead to arbitrages. Hagan et al. recently proposed a new arbitrage-free SABR solution, based on a finite difference discretisation of an expansion of the probability density. They rely on a Crank-Nicolson discretisation, which can lead to undesirable oscillations in the option price. This paper applies a variety of second order finite difference schemes to the SABR arbitrage-free density problem and explores alternative formulations. It is found that the TR-BDF2 and Lawson-Swayne schemes stand out on this problem in terms of stability and speed. The probability density formulation is the most stable and benefits greatly from a variable transformation. A partial differential equation is also derived for the so called free-boundary SABR model, which allows for negative interest rates without any additional shift parameter, leading to a new arbitrage-free solution for this model. Finally the free-boundary model behavior is analyzed.*

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

1. Introduction

It is now well known that the 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. Through an expansion of the probability density, Doust (2012) describes another arbitrage-free SABR method, but the required absorption probability involves costly numerical computations.

Hagan *et al.* (2014) recently proposed a new arbitrage-free SABR solution, based on a finite difference discretisation of the probability density. This approach provides a solution very close to the original SABR analytic formula, well known and widely used, while being arbitrage-free by construction, 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

Correspondence Address: Calypso Technology, 16-18 Rue du 4 Septembre, 75002 Paris, France. Email: fabien_lefloch@calypso.com

Electronic copy available at: https://ssrn.com/abstract=2402001

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.

In order to handle negative interest rates without an additional shift parameter, while still keeping control over the backbone through the β SABR parameter, Antonov *et al.* (2015) propose an alternate SABR model they call free-boundary SABR. . The authors derive a semi-analytical formula, involving double numerical integration. While the formula is shown to be reasonably accurate in practice, it is not, a priori, arbitrage-free. To illustrate the flexibility of the arbitrage-free PDE approach, we will apply it to this model as well.

2. Arbitrage Free SABR

Hagan *et al.* (2014) use asympotic techniques to reduce the SABR model from two dimensions to one dimension. Pricing with SABR parameters α, β, ρ, ν and forward f at τ_{ex} then relies on the solution of the Fokker-Planck[1] 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

$$M(T,F) = \frac{1}{2}D^2(F)E(T,F),\ E(T,F) = e^{\rho\nu\alpha\Gamma(F)T},\ \Gamma(F) = \frac{F^\beta - f^\beta}{F-f}, \tag{2}$$

$$D(F) = \sqrt{\alpha^2 + 2\alpha\rho\nu y(F) + \nu^2 y(F)^2}F^\beta,\ y(F) = \frac{F^{1-\beta} - f^{1-\beta}}{1-\beta} \tag{3}$$

and initial condition

$$\lim_{T\to 0} Q(T,F) = \delta(F-f). \tag{4}$$

Vanilla option prices can then be computed through the Breeden-Litzenberger formula (Breeden and Litzenberger, 1978):

$$V_{call}(T,K) = \int_K^{F_{\max}} (F-K)Q(T,F)dF + (F_{\max} - K)Q_R(T), \tag{5}$$

$$V_{put}(T,K) = (K - F_{\min})Q_L(T) + \int_{F_{\min}}^K (K-F)Q(T,F)dF. \tag{6}$$

The term $M(T,F)$ represents the diffusion coefficient, and so, in financial terms, $D(F)\sqrt{E(T,F)}$ can be regarded as the normal local volatility. In (Andreasen and Huge, 2011), a slightly simpler (less accurate for long maturities) expansion of the local volatility is

[1] At a later time, the probability density will follow a different Fokker-Planck equation, as the forward f will have moved. The true underlying process can not be represented by a one dimensional diffusion, only its realization at a specific time can, which precisely what we care about numerically here. This is not to be confused with the 2D Fokker-Planck equation stemming from the classical SABR model stochastic differential equations

Electronic copy available at: https://ssrn.com/abstract=2402001

found and directly used in the normal Dupire forward PDE on the call prices V_{call}:

$$\frac{\partial V_{call}}{\partial T}(T,F) = \frac{1}{2}\vartheta^2(T,F)\frac{\partial^2 V_{call}}{\partial F^2}(T,F) \tag{7}$$

with initial condition $V_{call}(0,F) = (f-F)^+$ and

$$\vartheta(T,F) = D(F). \tag{8}$$

Both approaches are strikingly similar if the same local volatility approximation as well as the same PDE discretisation is used. One difference lies in the boundary conditions: Hagan uses an absorbing condition at $F_{\min}$ and $F_{\max}$, while Andreasen and Huge use the standard Hagan expansion formula at the boundaries. It is also possible to use the more classic linear boundaries condition in the Dupire forward PDE, which is even closer, then to an absorbing condition in the probability density.

While our will focus is mainly on the PDE in probability density, we will also have a quick look at the nearly equivalent Dupire representation in order to find out if one approach is more efficient or not.

3. Modeling negative interest rates

Negative interest rates are now a common feature of the market. As a consequence, dealers quote swaptions in terms of basis point volatilities (b.p. vols) or in terms of shifted lognormal volatility. In terms of modeling with the popular SABR model of (Hagan *et al.*, 2002), one possibility is to use the so called normal SABR model, with $\beta = 0$, that naturally allows negative forwards. Practitioners prefer to use a shifted SABR model that allows for negative forwards up to a shift b as in (Hagan *et al.*, 2014), while keeping β free. Both can be summarized in the following framework:

$$dF = AL(F)dW_F, \tag{9}$$

$$dA = \nu A dW_A \tag{10}$$

with W_F, W_A correlated Brownian motions with correlation ρ. In the normal case, we have:

$$L(F) = 1. \tag{11}$$

In the shifted lognormal case, we have:

$$L(F) = (F+b)^\beta. \tag{12}$$

Another alternative called free-boundary SABR is introduced in (Antonov *et al.*, 2015) and is defined with:

$$L(F) = |F|^\beta. \tag{13}$$

This naturally allows negative rates, without the need for a shift parameter.

4. Free-boundary PDE

The difference between the classic SABR PDE formulation and the free boundary SABR PDE formulation lies in the formulas related to $L(F)$ and in the boundary condition. The probability density Q is solution of:

$$\frac{\partial Q}{\partial T}(T,F) = \frac{\partial^2 M(T,F)Q(T,F)}{\partial F^2} \text{ and } \begin{cases} \frac{\partial Q_L}{\partial T}(T) = \lim_{F\to F_{\min}} \frac{\partial M(T,F)Q(T,F)}{\partial F} \\ \frac{\partial Q_R}{\partial T}(T) = \lim_{F\to F_{\max}} \frac{\partial M(T,F)Q(T,F)}{\partial F} \end{cases} \tag{14}$$

Electronic copy available at: https://ssrn.com/abstract=2402001

with

$$M(T,F) = \frac{1}{2}D^2(F)E(T,F),\ E(T,F) = e^{\rho\nu\alpha\Gamma(F)T}, \tag{15}$$

$$D(F) = \sqrt{\alpha^2 + 2\alpha\rho\nu y(F) + \nu^2 y(F)^2}|F|^\beta \tag{16}$$

using

$$\Gamma(F) = \frac{|F|^\beta - |f|^\beta}{F-f}, \tag{17}$$

$$y(F) = \int_f^F \frac{du}{L(u)} = \frac{\operatorname{sgn}(F)|F|^{1-\beta} - \operatorname{sgn}(f)|f|^{1-\beta}}{1-\beta} \tag{18}$$

and initial condition

$$\lim_{T\to 0} Q(T,F) = \delta(F-f). \tag{19}$$

In effect, only L, Γ, and y have changed compared to the classic SABR model with absorption at 0. Furthermore, instead of placing the left boundary at 0 (or at a shift $-b$), the left boundary should be located far enough on the negative side, a typical choice would be $F_{min} = -F_{max}$. It is important to keep the absorbing behavior at the boundaries, even if the absorption should be in practice quite small, as it allows to preserve the zeroth and first moments (corresponding to cumulative density and forward) exactly numerically.

5. Change of Variable

5.1 Transformation of the Fokker-Planck PDE

One practical difficulty that arises with the Arbitrage Free PDE described in equations (14) is the choice of F_{max}. Hagan *et al.* (2014) proposes a formula to estimate the required F_{max}, that, for some parameters is not suitable. It can be very large for long term deals and as a consequence the discretisation requires a very large number of points to obtain an acceptable accuracy. The technique becomes inefficient. However, this type of problem is not new and a common approach is to consider a change of variable as a remedy (Andersen and Piterbarg, 2010, p. 292 section 7.4). On this particular problem the following change of variable works well whilst still preserving the moments (Hagan, 2013):

$$z(F) = \int_f^F \frac{du}{D(u)}. \tag{20}$$

This leads to a PDE in $\theta(z) = Q(T,F(z))D(F(z)) = Q(T,F(z))C(z)$ with $C(z) = D(F(z))$:

$$\frac{\partial\theta}{\partial T}(T,z) = \frac{1}{2}\frac{\partial}{\partial z}\left\{\frac{1}{C(z)}\frac{\partial C(z)E(T,z)\theta(T,z)}{\partial z}\right\} \text{ and } \begin{cases}\theta(T,z) = 0 \text{ as } z \to z^- = z(F_{\min})\\ \theta(T,z) = 0 \text{ as } z \to z^+ = z(F_{\max})\end{cases} \tag{21}$$

with

$$y(z) = \frac{\alpha}{\nu}\left[\sinh(\nu z) + \rho(\cosh(\nu z) - 1)\right], \tag{22}$$

$$F(y) = \left[f^{1-\beta} + (1-\beta)y\right]^{\frac{1}{1-\beta}}. \tag{23}$$

Electronic copy available at: https://ssrn.com/abstract=2402001

The probability at the boundaries accumulates according to:

$$\frac{\partial P_L}{\partial T}(T) = \lim_{z \to z^-} \frac{1}{2}\frac{1}{C(z)}\frac{\partial C(z)E(T,z)\theta(T,z)}{\partial z}, \tag{24}$$

$$\frac{\partial P_R}{\partial T}(T) = \lim_{z \to z^+} -\frac{1}{2}\frac{1}{C(z)}\frac{\partial C(z)E(T,z)\theta(T,z)}{\partial z}. \tag{25}$$

As a result, the effect of the diffusion term D has almost been cancelled, and the probability density θ is closer to a Gaussian in z. The values z^+ and z^- are then naturally chosen to be $\pm n_{sd}\sqrt{\tau_{ex}}$, corresponding to n_{sd} standard deviations above and below the forward located at $z = 0$ (taking care of truncating at the barrier $F = 0$ if necessary). Figure 1 shows that the probability density will be computed with a high concentration of points around the forward, and a much lower one near the upper boundary.

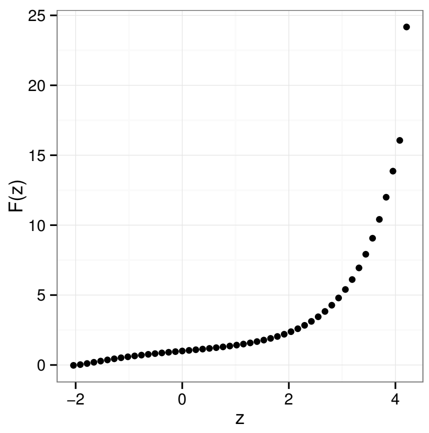


Figure 1.: $F(z)$ with $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%, \tau_{ex} = 1$

The call and put prices are obtained by integrating on the transformed density:

$$V_{call}(T,K) = \int_{z(K)}^{z^+} (F(z) - K)\theta(T,z)dz + (F_{\max} - K)P_R(T), \tag{26}$$

$$V_{put}(T,K) = (K - F_{\min})P_L(T) + \int_{z^-}^{z(K)} (K - F(z))\theta(T,z)dz. \tag{27}$$

For some extreme SABR parameters, the change of variable allows high accuracy with a small number of points. The uniform discretisation of Q in F can require approximately 1000 times more points to reach a similar accuracy (Table 1). The number of points used can not

Uniform discretisation of Q in F					Discretization of θ in z				
F_{max}	Points	Steps	Price	Vol	n_{sd}	Points	Steps	Price	Vol
5	10	5	0.65010	87.205	3	10	5	0.79848	198.504
50	100	10	0.78769	155.773	3	100	10	0.79853	199.148
500	1000	20	0.79782	191.658	4	100	20	0.79847	198.338
5000	10000	160	0.79835	196.930	10	10000	160	0.79845	198.134

Table 1.: Price by the Lawson-Swayne method without and with variable change for extreme SABR parameters: $\alpha = 100\%, \beta = 0.30, \rho = 90\%, \nu = 100\%, \tau_{ex} = 10, f = 1$

Electronic copy available at: https://ssrn.com/abstract=2402001

be too small: the forward should not be on the boundary. This restriction is much stricter for the uniform discretisation of Q than for the discretisation in the changed variable θ.

5.2 Coordinate Transformation for the Forward Dupire PDE

The same variable transformation (34) can be applied to the Dupire forward PDE with $\vartheta^2(T,F) = D^2(F)E(F,T)$ resulting in:

$$\frac{\partial V_{call}}{\partial T}(T,z) = \frac{1}{2}C(z)E(T,z)\frac{\partial}{\partial z}\left\{\frac{1}{C(z)}\frac{\partial V_{call}(T,z)}{\partial z}\right\} \tag{28}$$

with initial condition $V_{call}(0,z) = (f - F(z))^+$.

A slightly simpler alternative approach consists in using an equivalent non-uniform grid in F inside a finite difference discretisation of the forward Dupire PDE (7) as described in (Andersen and Piterbarg, 2010). The equivalent non uniform grid is defined by $F(z)$ where z is a uniform discretisation. Care must be taken to place the forward f in the middle of two nodes in order to decrease the numerical error (Tavella and Randall, 2000). Another technique to reduce the numerical error is to smooth out the payoff at maturity by averaging, but this would not preserve put-call parity (Le Floc'h, 2013).

Let's define a uniform grid in the coordinate z, for $j = 0, ..., J+1$:

$$z_j = z^- + jh \ , \ h = \frac{1}{J+1}(z^+ - z^-). \tag{29}$$

Using Equations (22) and (38), the corresponding grid in F would be:

$$F_j = F\left(y(z_j)\right) = F\left(y\left(z^- + \frac{j}{J+1}\left(z^+ - z^-\right)\right)\right). \tag{30}$$

To place the forward f in the middle of two consecutive points, we first locate the index j_0 so that $F_{j_0} \leq f < F_{j_0+1}$, that is:

$$j_0 = \lfloor\frac{z(y(f)) - z^-}{h}\rfloor. \tag{31}$$

And then we shift the grid by a distance d defined by:

$$d = z(y(f)) - z(y\left(\frac{1}{2}(F_{j_0} + F_{j_0+1})\right)). \tag{32}$$

The new grid is then just generated via

$$\tilde{F}_j = F(y(z_0+jh+d)) = F(y(z^-+\frac{j}{J+1}\left(z^+ - z^-\right)+z(y(f))-z(y\left(\frac{1}{2}(F_{j_0} + F_{j_0+1})\right)))). \tag{33}$$

In addition, we make sure to keep the original boundaries, this is especially important as z^- often corresponds to the absorbing barrier by forcing $\tilde{F}_0 = F(y(z^-))$ and $\tilde{F}_{J+1} = F(y(z^+))$.

Finally, to compute the price of an option in between two grid nodes, we interpolate the discrete prices using a natural cubic spline in order to preserve the second derivative continuity (and therefore the probability density continuity) everywhere.

5.3 Transformation of the free-boundary PDE

We rely on the same variable transform as in the case of the standard arbitrage-free SABR PDE (equation 34):

$$z(F) = \int_f^F \frac{du}{D(u)}. \tag{34}$$

Electronic copy available at: https://ssrn.com/abstract=2402001

This leads to a PDE in $\theta(z) = Q(T, F(z))D(F(z)) = Q(T, F(z))C(z)$ with $C(z) = D(F(z))$:

$$\frac{\partial \theta}{\partial T}(T,z) = \frac{1}{2}\frac{\partial}{\partial z}\left\{\frac{1}{C(z)}\frac{\partial C(z)E(T,z)\theta(T,z)}{\partial z}\right\} \text{ and } \begin{cases}\theta(T,z) = 0 \text{ as } z \to z^- = z(F_{\min})\\ \theta(T,z) = 0 \text{ as } z \to z^+ = z(F_{\max})\end{cases} \tag{35}$$

$z(y)$ is unchanged as in this coordinate, L is absent of its definition in terms of y:

$$z(y) = \int_{y(f)}^{y(F)} \frac{dy'}{\sqrt{\alpha^2 + 2\alpha\rho\nu y' + \nu^2 y'^2}}. \tag{36}$$

Therefore $y(z)$ is also unchanged:

$$y(z) = \frac{\alpha}{\nu}\left[\sinh(\nu z) + \rho(\cosh(\nu z) - 1)\right]. \tag{37}$$

Only $F(y)$ is modified by the free-boundary model. We just invert equation (18):

$$F(y) = \operatorname{sgn}(y - \bar{y})\left[(1-\beta)|y - \bar{y}|\right]^{\frac{1}{1-\beta}} \tag{38}$$

with $\bar{y} = y(F = 0) = -\frac{\operatorname{sgn}(f)|f|^{1-\beta}}{1-\beta}$.

Again, the boundary z^- should now be taken simply as $z^- = -z^+ = -n_{sd}\sqrt{\tau}$ where n_{sd} is a number of standard deviations. In practice $n_{sd} = 4$ is highly accurate.

6. Discretization of the the PDE in θ

Let's define for $j = 1, ..., J$:

$$z_j = z^- + jh \text{ , } \hat{y}_j = y(z_j - \frac{1}{2}h) \text{ , } \hat{F}_j = F(\hat{y}_j), \tag{39}$$

$$\hat{C}_j = D(\hat{F}_j) \text{ , } \hat{\Gamma}_j = \frac{\hat{F}_j^\beta - f^\beta}{\hat{F}_j - f} \text{ , } \hat{E}_j(T) = e^{\rho\nu\alpha\hat{\Gamma}_j T}. \tag{40}$$

We also define for $n = 0, ..., N-1$

$$t_n = n\delta \text{ with } \delta = \frac{\tau_{ex}}{N}, \tag{41}$$

$$\theta_j^n = \theta(z_j, t_n) \tag{42}$$

To preserve the zero-eth and first moment of F, the PDE (Equation 35) is discretized in z as (Hagan, 2013):

$$\frac{\partial \theta}{\partial T}(z, t_n) = \mathcal{L}_j^n \theta(z, t_n) \tag{43}$$

for $j = 1, ..., J$ with $\mathcal{L}_j^n$ the discrete operator defined by

$$\mathcal{L}_j^n \theta(z_j, t_n) = \frac{1}{h}\frac{\hat{C}_{j-1}}{\hat{F}_j - \hat{F}_{j-1}}\hat{E}_{j-1}(t_n)\theta(z_{j-1}, t_n) \tag{44}$$

$$- \frac{1}{h}\left(\frac{\hat{C}_j}{\hat{F}_{j+1} - \hat{F}_j} + \frac{\hat{C}_j}{\hat{F}_j - \hat{F}_{j-1}}\right)\hat{E}_j(t_n)\theta(z_j, t_n) \tag{45}$$

$$+ \frac{1}{h}\frac{\hat{C}_{j+1}}{\hat{F}_{j+1} - \hat{F}_j}\hat{E}_{j+1}(t_n)\theta(z_{j+1}, t_n) \tag{46}$$

Electronic copy available at: https://ssrn.com/abstract=2402001

and for the boundaries at $j = 0$ and $j = J + 1$:

$$\frac{\hat{C}_0}{\hat{F}_1 - \hat{F}_0}\hat{E}_0(T)\theta(z_0, T) = -\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(T)\theta(z_1, T) \tag{47a}$$

$$\frac{\hat{C}_{J+1}}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_{J+1}(T)\theta(z_{J+1}, T) = -\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(T)\theta(z_J, T) \tag{47b}$$

The boundary condition described by equations (47) is applicable to all schemes considered in this section as it is independent of the time-stepping.

The probability accumulated at the boundaries is discretized as:

$$\frac{\partial P_L}{\partial T}(T) = \frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(T)\theta(z_1, T) \tag{48}$$

$$\frac{\partial P_R}{\partial T}(T) = \frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(T)\theta(z_J, T) \tag{49}$$

The call and put prices are obtained by integrating with the mid-point method. Let $z^* = z(y(K))$. We first suppose that $z^- < z^* < z^+$. Let k be the index such that $z^- + (k-1)h < z^* \leq z^- + kh$ and $F_k = F(y(z^- + kh))$. Then

$$V_{call} = \frac{h}{4(F_k - \hat{F}_k)}(F_k - K)^2\theta_k + \sum_{j=k+1}^{J-1}(\hat{F}_j - K)h\theta_j + (F_{\max} - K)P_R \tag{50}$$

$$V_{put} = \frac{h}{4(\hat{F}_k - F_k)}(K - F_k)^2\theta_k + \sum_{j=1}^{k}(K - \hat{F}_j)h\theta_j + (K - F_{\min})P_L \tag{51}$$

For the call option case, the first term corresponds to

$$\int_{z^*}^{z_k}(F(z) - K)\theta(z)dz = \int_K^{F_k}(F - K)\frac{\theta(z(F))}{D(F)}dF. \tag{52}$$

We then assume θ constant between z_{k-1} and z_k, and make the approximation $D(F) = \frac{\partial F}{\partial z} \approx 2\frac{F_k - \hat{F}_k}{h}$ (we found that this choice preserved a smoother numerical density). The put option case is similar. Equations (50) and (51) preserve the put-call parity exactly.

When $z^* \leq z^-$, $V_{call} = f - K$ and $V_{put} = 0$. When $z^* \geq z^+$, $V_{call} = 0$ and $V_{put} = K - f$.

6.1 Moment Preserving Implicit Euler

$$\theta_j^{n+1} - \theta_j^n = \delta\mathcal{L}_j^{n+1}\theta_j^{n+1} \tag{53a}$$

$$P_L(t_{n+1}) - P_L(t_n) = \delta\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(t_{n+1})\theta_1^{n+1} \tag{53b}$$

$$P_R(t_{n+1}) - P_R(t_n) = \delta\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(t_{n+1})\theta_J^{n+1} \tag{53c}$$

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

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

Electronic copy available at: https://ssrn.com/abstract=2402001

product M_0Q_0 is used in the discretisation of equation (14) and it is entirely defined by M_1Q_1 because of the mirror-like boundary condition (imposed at the fictitious point F_0):

$$M_0Q_0 + M_1Q_1 = 0. \tag{54}$$

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 $\Gamma(f) = \frac{\partial C}{\partial F}(f)$.

6.2 Moment Preserving Crank-Nicolson

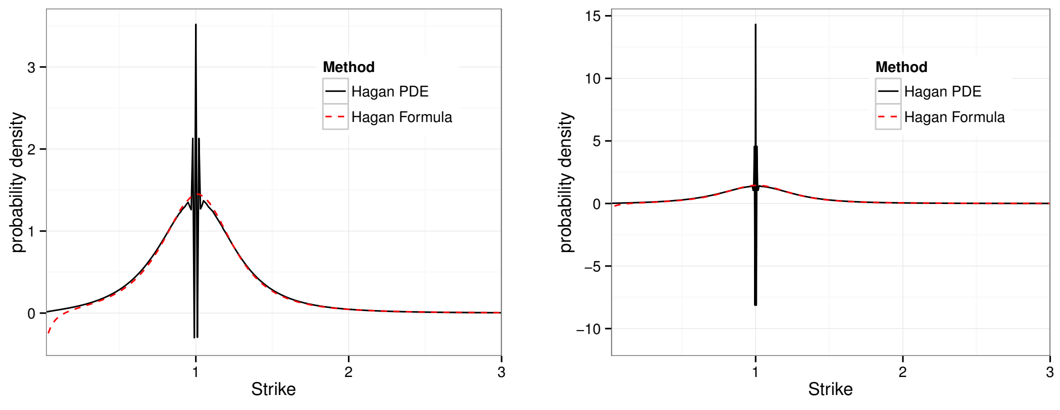
$$\theta_j^{n+1} - \theta_j^n = \frac{\delta}{2}\left(\mathcal{L}_j^{n+1}\theta_j^{n+1} + \mathcal{L}_j^n\theta_j^n\right) \tag{55a}$$

$$P_L(t_{n+1}) - P_L(t_n) = \frac{\delta}{2}\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\left(\hat{E}_1(t_{n+1})\theta_1^{n+1} + \hat{E}_1(t_n)\theta_1^n\right) \tag{55b}$$

$$P_R(t_{n+1}) - P_R(t_n) = \frac{\delta}{2}\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\left(\hat{E}_J(t_{n+1})\theta_J^{n+1} + \hat{E}_J(t_n)\theta_J^n\right) \tag{55c}$$

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

7. Crank-Nicolson Oscillations with SABR



(a) PDE in Q with 40 time steps and $F_{max} = 5$ (b) PDE in θ with 80 time steps and $n_{sd} = 4$

Figure 2.: Probability density in Hagan PDE discretised with Crank-Nicolson with 500 points. $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%, \tau_{ex} = 1$

We use the same parameters as the example of negative density with the standard SABR formula in (Hagan *et al.*, 2014): $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%$ and forward $f = 1$ at

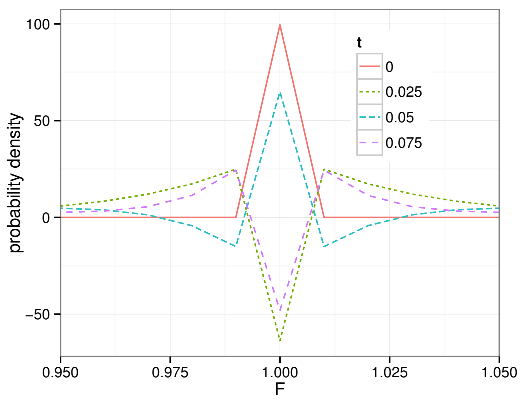
Electronic copy available at: https://ssrn.com/abstract=2402001

$\tau_{ex} = 1$, 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.* (2014) recommend between 200 and 500 points and 30 to 100 time-steps.

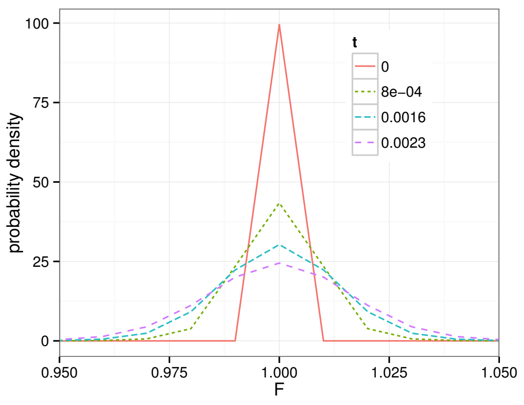
Figure 2 shows strong oscillations around the forward. To guarantee the absence of oscillations, the *Courant number* should be small enough $\Psi \leq 1$ (Theorem 2.2 in Morton and Mayers (2005)). For the uniform discretisation of $Q(F)$, $\Psi_Q = M\frac{\delta}{h^2}$. This corresponds directly to 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,

$$\Psi_Q = \frac{1}{2}\alpha^2 f^{2\beta}\frac{\delta}{h^2} \tag{56}$$

$$\Psi_\theta = f^\beta \frac{\delta}{h^2} \tag{57}$$



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



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

Figure 3.: First 4 time steps of the probability density in Hagan PDE discretised with Crank-Nicolson

In our example, in Figures 2 and 3(a) $\Psi_Q \approx 15$ while Figure 3(b) shows that indeed when $\Psi < 1$ there are no oscillations. The Crank-Nicolson oscillations are even stronger with the PDE in θ as for $\alpha \ll 1$, we have $\Psi_\theta \gg \Psi_Q$. Using the same SABR parameters and $n_{sd} = 3$ (corresponding to $F_{max} \approx 5$), $\Psi_\theta \approx 250$. In the next sections it is shown that a much smaller number of time steps can be used with other finite difference time-stepping techniques whilst still preserving good accuracy.

8. Alternative Schemes

We will focus our analysis on the PDE in θ, but similar conclusion can be drawn for the PDE in Q.

8.1 Rannacher

A common fix for Crank-Nicolson oscillations related to non smooth initial data is Rannacher time-stepping (Rannacher, 1984; Pooley *et al.*, 2003; Giles and Carter, 2006). It consists

Electronic copy available at: https://ssrn.com/abstract=2402001

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:

$$\theta_j^{n+\frac{1}{2}} - \theta_j^n = \frac{\delta}{2}\mathcal{L}_j^{n+\frac{1}{2}}\theta_j^{n+\frac{1}{2}} \tag{58a}$$

$$P_L(t_{n+\frac{1}{2}}) - P_L(t_n) = \frac{\delta}{2}\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(t_{n+\frac{1}{2}})\theta_1^{n+\frac{1}{2}} \tag{58b}$$

$$P_R(t_{n+\frac{1}{2}}) - P_R(t_n) = \frac{\delta}{2}\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(t_{n+\frac{1}{2}})\theta_J^{n+\frac{1}{2}} \tag{58c}$$

for $j = 1, ..., J$ and $n = 0, \frac{1}{2}, 1, \frac{3}{2}$. The next steps are Crank-Nicolson for $n = 2, ..., N-1$.

8.2 BDF2

The second order backward difference scheme (BDF2) is A-stable and L-stable multistep implicit scheme and will therefore damp any oscillation very quickly. Multi-steps schemes have however severe limitations: instabilities will occur for sudden changes in the system variables, and initialization by another method is needed for the first steps (Windcliff *et al.*, 2001). For example they can not be applied to linear complimentary problems like the pricing of American options through the discretisation of the Black-Scholes PDE (**?**). Those issues don't arise with the SABR density PDE. The first moments will be preserved with the following discretisation:

$$3\theta_j^{n+2} - 4\theta_j^{n+1} + \theta_j^n = 2\delta\mathcal{L}_j^{n+2}\theta_j^{n+2} \tag{59a}$$

$$3P_L(t_{n+2}) - 4P_L(t_{n+1}) + P_L(t_n) = 2\delta\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(t_{n+2})\theta_1^{n+2} \tag{59b}$$

$$3P_R(t_{n+2}) - 4P_R(t_{n+1}) + P_R(t_n) = 2\delta\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(t_{n+2})\theta_J^{n+2} \tag{59c}$$

for $j = 1, ..., J$ and $n = 0, ..., N-2$. Implicit Euler is used to compute θ_j^1, P_L^1, P_R^1 at $n = 0$.

8.3 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 58) 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:

$$\theta(z) = 2\bar{\theta}^{\frac{\delta}{2}}(z) - \bar{\theta}^{\delta}(z) \tag{60}$$

$$P_L = 2\bar{P}_L^{\frac{\delta}{2}} - \bar{P}_L^{\delta} \tag{61}$$

$$P_R = 2\bar{P}_R^{\frac{\delta}{2}} - \bar{P}_R^{\delta} \tag{62}$$

where $\bar{\theta}^{\delta}, \bar{P}_L^{\delta}, \bar{P}_R^{\delta}$ are θ, P_L, P_R computed by implicit Euler with a time step of δ.

Electronic copy available at: https://ssrn.com/abstract=2402001

8.4 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 (60) is applied. For the third order scheme, at each time-step we apply:

$$\theta(F) = 4.5\bar{\theta}^{\frac{\delta}{3}}(F) - 4.5\bar{\theta}^{\frac{2\delta}{3}}(F) + \bar{\theta}^{\delta}(F) \tag{63}$$

$$P_L = 4.5\bar{P}_L^{\frac{\delta}{3}} - 4.5\bar{P}_L^{\frac{2\delta}{3}} + \bar{P}_L^{\delta} \tag{64}$$

$$P_R = 4.5\bar{P}_R^{\frac{\delta}{3}} - 4.5\bar{P}_R^{\frac{2\delta}{3}} + \bar{P}_R^{\delta} \tag{65}$$

where $\bar{\theta}^{\frac{\delta}{3}}$ is θ computed by implicit Euler with 3 time steps of $\frac{\delta}{3}$ and $\bar{\theta}^{\frac{2\delta}{3}}$ is θ 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.

8.5 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:

$$\begin{aligned}\theta_j^{n+b} - \theta_j^{n} &= b\delta \mathcal{L}_j^{n+b}\theta_j^{n+b}\\ P_L(t_{n+b}) - P_L(t_n) &= b\delta\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(t_{n+b})\theta_1^{n+b}\\ P_R(t_{n+b}) - P_R(t_n) &= b\delta\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(t_{n+b})\theta_J^{n+b}\end{aligned} \tag{66a}$$

Second stage:

$$\begin{aligned}\theta_j^{n+2b} - \theta_j^{n+b} &= b\delta \mathcal{L}_j^{n+2b}\theta_j^{n+2b}\\ P_L(t_{n+2b}) - P_L(t_{n+b}) &= b\delta\frac{\hat{C}_1}{\hat{F}_1 - \hat{F}_0}\hat{E}_1(t_{n+2b})\theta_1^{n+2b}\\ P_R(t_{n+2b}) - P_R(t_{n+b}) &= b\delta\frac{\hat{C}_J}{\hat{F}_{J+1} - \hat{F}_J}\hat{E}_J(t_{n+2b})\theta_J^{n+2b}\end{aligned} \tag{66b}$$

And finally:

$$\begin{aligned}\theta_j^{n+1} &= (\sqrt{2}+1)\theta_j^{n+2b} - \sqrt{2}\theta_j^{n+b}\\ P_L(t_{n+1}) &= (\sqrt{2}+1)P_L(t_{n+2b}) - \sqrt{2}P_L(t_{n+b})\\ P_R(t_{n+1}) &= (\sqrt{2}+1)P_R(t_{n+2b}) - \sqrt{2}P_R(t_{n+b})\end{aligned} \tag{66c}$$

Electronic copy available at: https://ssrn.com/abstract=2402001

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.

8.6 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, 2014).

First stage:

$$\begin{aligned}
\theta_j^{n+\alpha} - \theta_j^n &= \frac{\alpha\delta}{2}\left(\mathcal{L}_j^{n+\alpha}\theta_j^{n+\alpha} + \mathcal{L}_j^n\theta_j^n\right)\\
P_L(t_{n+\alpha}) &= P_L(t_n) + b\delta\frac{\hat{C}_1}{\hat{F}_1-\hat{F}_0}\left(\hat{E}_1(t_{n+\alpha})\theta_1^{n+\alpha} + \hat{E}_1(t_n)\theta_1^n\right)\\
P_R(t_{n+\alpha}) &= P_R(t_n) + b\delta\frac{\hat{C}_J}{\hat{F}_{J+1}-\hat{F}_J}\left(\hat{E}_J(t_{n+\alpha})\theta_J^{n+\alpha} + \hat{E}_J(t_n)\theta_J^n\right)
\end{aligned} \tag{67a}$$

Second stage:

$$\begin{aligned}
\theta_j^{n+1} &= \frac{1}{2-\alpha}\left(\frac{1}{\alpha}\theta_j^{n+\alpha} - \frac{(1-\alpha)^2}{\alpha}\theta_j^n + \delta(1-\alpha)\mathcal{L}_j^{n+1}\theta_j^{n+1}\right)\\
P_L(t_{n+1}) &= \frac{1}{2-\alpha}\left(\frac{1}{\alpha}P_L(t_{n+\alpha}) - \frac{(1-\alpha)^2}{\alpha}P_L(t_n) + \delta(1-\alpha)\frac{\hat{C}_1}{\hat{F}_1-\hat{F}_0}\hat{E}_1(t_{n+1})\theta_1^{n+1})\right)\\
P_R(t_{n+1}) &= \frac{1}{2-\alpha}\left(\frac{1}{\alpha}P_R(t_{n+\alpha}) - \frac{(1-\alpha)^2}{\alpha}P_R(t_n) + \delta(1-\alpha)\frac{\hat{C}_J}{\hat{F}_{J+1}-\hat{F}_J}\hat{E}_J(t_{n+1})\theta_J^{n+1})\right)
\end{aligned} \tag{67b}$$

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 that we will name "Bathe":

$$\theta_j^{n+\frac{1}{3}} = \theta_j^n + \frac{\delta}{6}\left(\mathcal{L}_j^n\theta_j^n + \mathcal{L}_j^{n+\frac{1}{3}}\theta_j^{n+\frac{1}{3}}\right) \tag{68a}$$

$$\theta_j^{n+\frac{2}{3}} = \theta_j^n + \frac{\delta}{6}\left(\mathcal{L}_j^{n+\frac{1}{3}}\theta_j^{n+\frac{1}{3}} + \mathcal{L}_j^{n+\frac{2}{3}}\theta_j^{n+\frac{2}{3}}\right) \tag{68b}$$

$$\theta_j^{n+1} = \frac{1}{11}\left(18\theta_j^{n+\frac{2}{3}} - 9\theta_j^{n+\frac{1}{3}} + 2\theta_j^n + 2\delta\mathcal{L}_j^{n+1}\theta_j^{n+1}\right) \tag{68c}$$

Electronic copy available at: https://ssrn.com/abstract=2402001

8.7 Optimising for Performance

The function $E(T, F)$ 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 $D(F)$ and $\Gamma(F)$ are constant in time and can thus be cached between time-steps. A further improvement is to decompose t_{n+1} as $t_n + \delta$, then

$$e^{\rho\nu\alpha\Gamma t_{n+1}} = e^{\rho\nu\alpha\Gamma t_n} e^{\rho\nu\alpha\Gamma\delta} \tag{69}$$

We can therefore just compute $e_j = e^{\rho\nu\alpha\Gamma(F_j)\delta}$ for $j = 0, ..., J+1$ once, and at each step simply update E as:

$$E_j^{n+1} = e_j E_j^n \tag{70}$$

with initial value $E_j^0 = 1$. 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 E 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.

9. Numerical Results

9.1 Oscillations

With the same parameters as in section 7, Figure 4(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), BDF2, Lawson-Swayne (LS), TR-BDF2 or Richardson extrapolation on implicit Euler (RE) would look very similar. Figure 4(c) show no apparent oscillations in the first steps for the Rannacher scheme. BDF2 and LMG2 would look the same. Lawson-Swayne shows a nearly imperceptible oscillation at the first step, and no more afterwards (Figure 4(d)). TR-BDF2 and Bathe schemes behave similarly. In contrast, Crank-Nicolson had strong oscillations visible at $T = \tau_{ex}$ with 40 time steps for the PDE in Q and even stronger oscillations with 80 time steps for the PDE in θ.

9.2 Performance

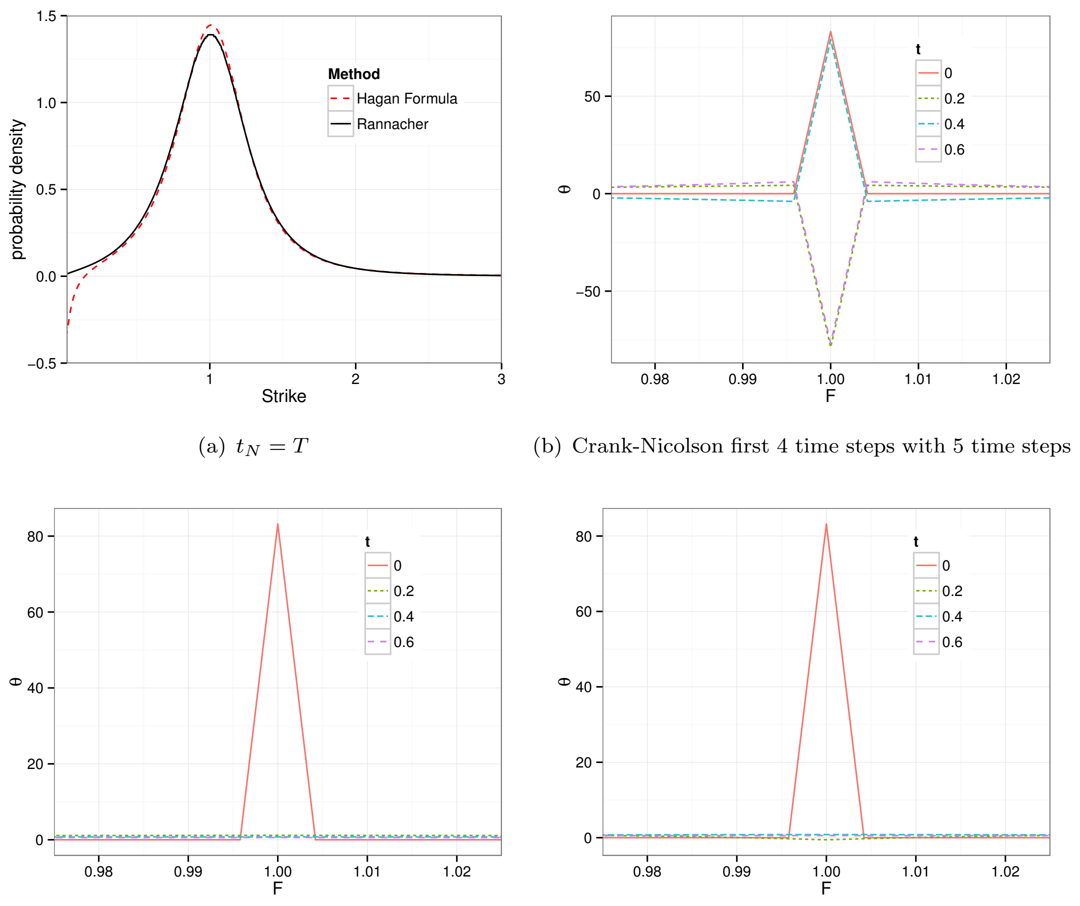
9.2.1 Hagan Example

With the same parameters as in section 7, 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 enough time-steps to ensure that $\Phi_\theta < 1$.

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 3 time-steps for Bathe, 6 for Lawson-Swayne and TR-BDF2, 10 for LMG2, and 12 for BDF2 and RAN. Lawson-Swayne is the most efficient scheme on this problem, closely followed by TR-BDF2, Rannacher and BDF2.

Higher order schemes like BDF3, LMG3 or a third order Richardson extrapolation were found to be no better performing than their second order variation on this problem.

Electronic copy available at: https://ssrn.com/abstract=2402001



(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) Lawson-Swayne first 4 time steps with 5 time steps

Figure 4.: Probability density in Hagan PDE using a total of 5 time-steps

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

Only 3 time-steps are enough to achieve a Black implied volatility accuracy better than 0.1% with Lawson-Swayne and Bathe schemes, 4 time-steps for TR-BDF2, 5 for LMG2, 12 for RAN and BDF2.

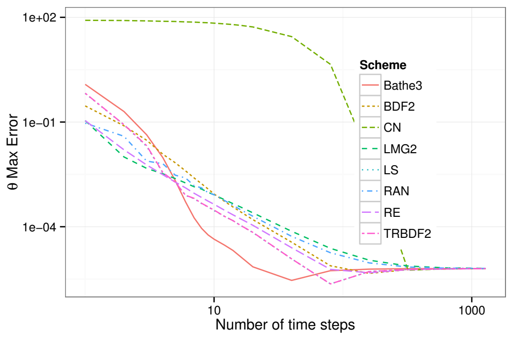
9.3 Dupire Forward PDE

The two different approaches result in the same smile, even with a relatively small number of time steps. With the Lawson Swayne finite difference method, the difference in implied volatility between the two approaches with 5 time-steps and $J = 50, n_{sd} = 4$ is always under 0.04%. Figure 7(a) displays the probability density computed by a numerical difference on the prices using the SABR parameters of Andreasen-Huge.

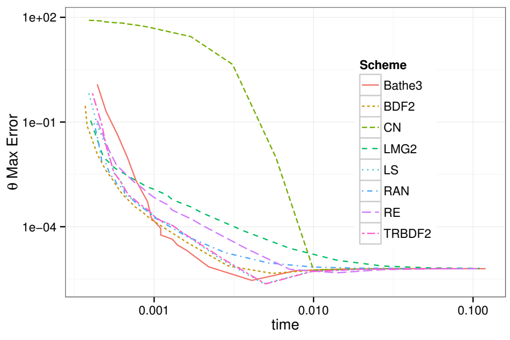
If one pushes the number of time steps smaller yet, then a difference appears in favour the the probability density approach (Figure 7(b) with 2 steps).

The probability density approach is more accurate with less time steps, and more stable. With only two time-steps, the Dupire approach leads to a large oscillation in probability

Electronic copy available at: https://ssrn.com/abstract=2402001

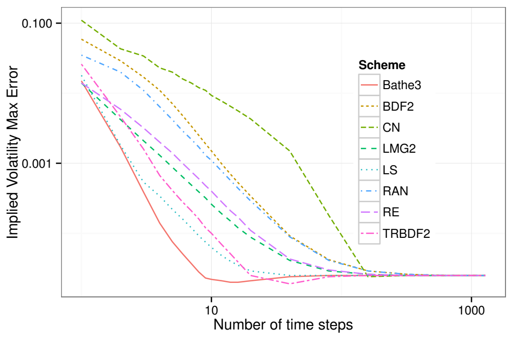


(a) Accuracy vs. number of time steps

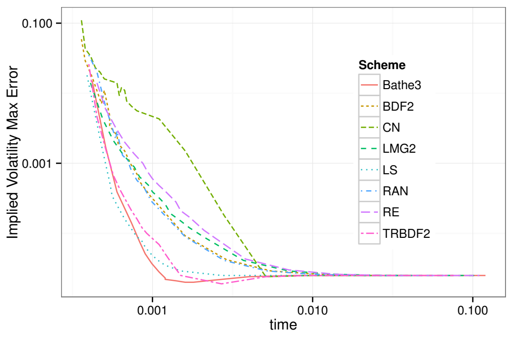


(b) Accuracy vs. time

Figure 5.: Performance on Hagan example

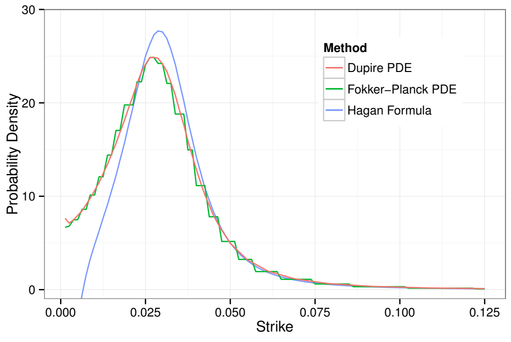


(a) Accuracy vs. number of time steps

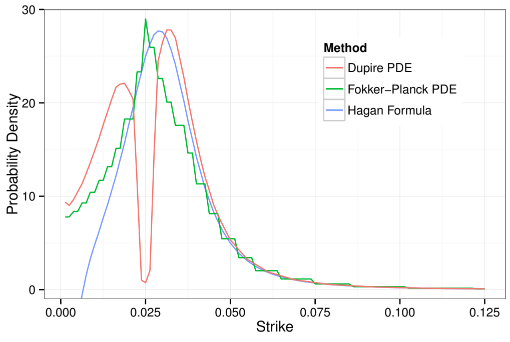


(b) Accuracy vs. time

Figure 6.: Performance on Andreasen-Huge example



(a) with 5 time-steps

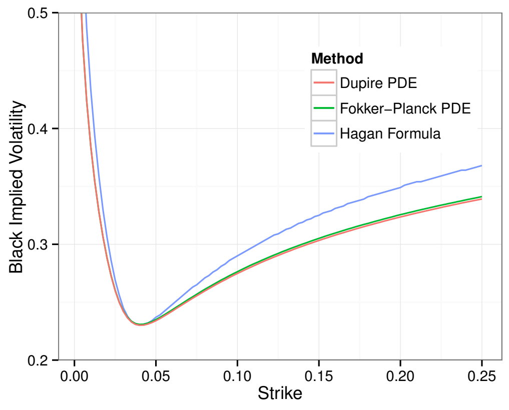


(b) with 2 time-steps

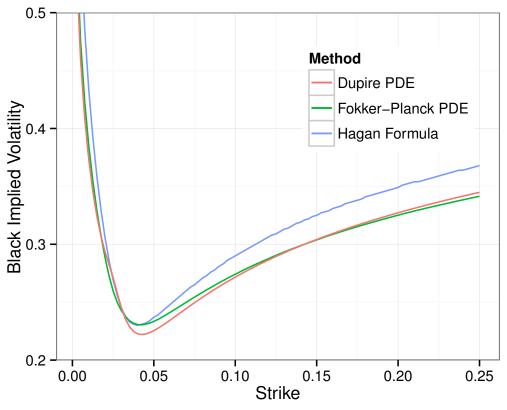
Figure 7.: Numerical Probability Density on Andreasen-Huge example

density. As Lawson-Swayne is strongly L-stable, it disappears very quickly: with three time-steps or more.

Electronic copy available at: https://ssrn.com/abstract=2402001



(a) with 5 time-steps



(b) with 2 time-steps

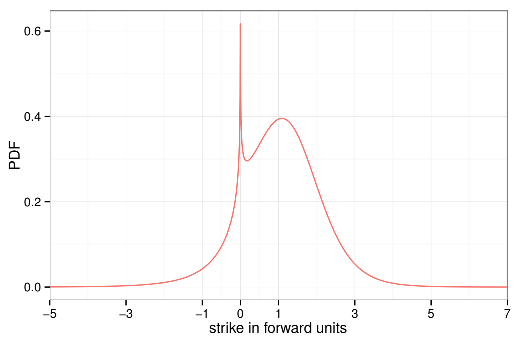
Figure 8.: Black implied volatility on Andreasen-Huge example

10. Free-boundary SABR model behavior

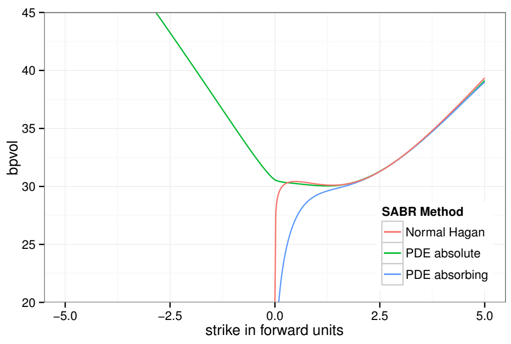
10.1 Smile

We first consider the same parameters as in (Antonov *et al.*, 2015) to validate the PDE approach. The arbitrage-free PDE reproduces the same probability density (Figure 9(a)). We observe a bigger and narrower spike around zero than Antonov *et al.* (2015). We would obtain a smaller spike had we employed a larger steps to compute the density. Note that this spike is not the result of a numerical error propagated in the finite difference scheme, but is inherent to the free boundary SABR model. The finite difference scheme actually behaves very well presenting no oscillation around the spike.

Normal volatilities are quite close to the reference values (Table 2) and exhibits a similar behavior. The maximum error from their analytic approximation on the same data is 0.20 b.p. while the PDE leads to a maximum error of 0.30 b.p. when compared to their Monte-Carlo reference values. When compared to the classic Hagan normal formula from (Hagan *et al.*, 2002), the error is much lower for positive strikes: the PDE stays very close to it rather than to the theoretical Monte-Carlo model.



(a) probability density function



(b) b.p. vols

Figure 9.: Lawson-Swayne on the free-boundary SABR PDE with $f = 50$ b.p., $\beta = 0.1, \alpha = 0.5f^{1-\beta}, \rho = -30\%, \nu = 30\%, \tau_{ex} = 3$

Electronic copy available at: https://ssrn.com/abstract=2402001

K	Lawson-Swayne	Reference	Difference
-0.95	30.72	30.93	-0.21
-0.80	29.70	29.95	-0.25
-0.65	28.69	28.97	-0.28
-0.5	27.70	27.99	-0.29
-0.35	26.74	27.04	-0.30
-0.2	25.87	26.15	-0.28
-0.05	25.21	25.46	-0.25
0.1	25.71	25.85	-0.14
0.25	26.65	26.69	-0.04
0.4	27.43	27.39	0.04
0.55	28.06	27.97	0.09
0.7	28.58	28.45	0.13
0.85	29.03	28.87	0.16
1.0	29.42	29.25	0.17
1.15	29.79	29.60	0.19
1.3	30.14	29.94	0.20
1.45	30.48	30.29	0.19
1.6	30.84	30.63	0.21
1.75	31.20	30.99	0.21
1.9	31.58	31.37	0.21

Table 2.: PDE accuracy with $f = 50$ b.p., $\beta = 0.25, \alpha = 0.5f^{1-\beta}, \rho = -30\%, \nu = 30\%, \tau_{ex} = 3$.The reference values are obtained by Monte-Carlo simulation in (Antonov *et al.*, 2015).

Figure 9(b) shows that the smile presents an inflection point at $F = 0$. This is a typical behavior of the free boundary SABR model under low rates. In contrast, the shifted SABR model assumes that this point is not special as absorption occurs at a different point, the shift $F = -b$ that no forward rate is supposed to be able to reach.

10.2 The absence of knee

Different recent studies show that the at-the-money normal volatilies observed in the market have a knee like shape (Hull and White, 2014; Deguillaume *et al.*, 2013): they are linear under very low-rate and are reasonably constant from low to high forward rates.

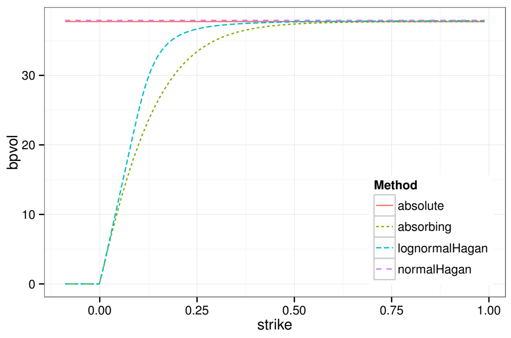
Hagan explains that this has led people to believe that the market switches from normal to lognormal when the rates become very low, but that in reality, absorption at zero creates this behavior even in the normal model (Hagan *et al.*, 2014). We reproduce his example in the normal SABR case $\beta = 0$, with absorption at $F = 0$ and see that the free boundary SABR model behaves just like the standard normal SABR model without absorption in this case (Figure 10(a)).

An increase of β results in only a very mild knee (Figure 10(b)).

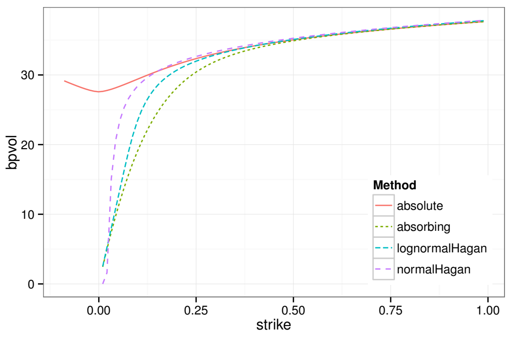
11. Conclusion

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 way to improve accuracy significantly compared to Crank-Nicolson on this problem.

Electronic copy available at: https://ssrn.com/abstract=2402001



(a) $\beta = 0$



(b) $\beta = 0.1$

Figure 10.: At-the-money b.p. volatilities with $f = 1, \alpha = 0.35, \rho = 0\%, \nu = 100\%, \tau_{ex} = 1$

However, as the number of time-steps decreases, the lower convergence of the Euler smoothing steps becomes more apparent. The simpler BDF2 is more efficient on this problem. Other less well known schemes such as TR-BDF2 or Lawson-Swayne further increase efficiency. 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, the arbitrage-free PDE of (Hagan *et al.*, 2014) is particularly competitive to the one step finite difference approach of (Andreasen and Huge, 2011). It can also lead to arbitrage-free pricing of alternative SABR models such as the free-boundary SABR model.

12. Declaration of Interest

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

References

Andersen, L. B. and Piterbarg, V. V. (2010) *Interest Rate Modeling, Volume I: Foundations and Vanilla Models*, (Atlantic Financial Press London).

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

Antonov, A., Konikov, M. and Spector, M. (2015) The Free Boundary SABR: Natural Extension to Negative Rates, *Available at SSRN 2557046*.

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.

Breeden, D. T. and Litzenberger, R. H. (1978) Prices of state-contingent claims implicit in option prices, *Journal of business*, , pp. 621–651.

Deguillaume, N., Rebonato, R. and Pogudin, A. (2013) The nature of the dependence of the magnitude of rate moves on the rates levels: a universal relationship, *Quantitative Finance*, 13(3), pp. 351–367.

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

Doust, P. (2012) No-arbitrage SABR, *The Journal of Computational Finance*, 15(3), p. 3.

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*.

Electronic copy available at: https://ssrn.com/abstract=2402001

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. (2013) Change of Variables and Conservative Numerical Schemes, *Not published*.

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. (2014) Arbitrage free SABR, *Wilmott magazine*.

Hull, J. and White, A. (2014) A generalized procedure for building trees for the short rate and its application to determining market implied volatility functions, *Quantitative Finance*, , pp. 1–12.

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) Exact Forward and Put-Call Parity with TR-BDF2, *SSRN eLibrary*.

Le Floc'h, F. (2014) TR-BDF2 for Stable American Option Pricing, *Journal of Computational Finance*.

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*.

Pooley, D., Vetzal, K. and Forsyth, P. (2003) Convergence remedies for non-smooth payoffs in option pricing, *Journal of Computational Finance*, 6(4), pp. 25–40.

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.

Tavella, D. and Randall, C. (2000) *Pricing financial instruments: The finite difference method*, (Wiley).

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.

Electronic copy available at: https://ssrn.com/abstract=2402001

Appendix A. Example Code

For illustration purpose, we detail here Octave code (also working with Matlab) for pricing vanilla options under the Arbitrage Free SABR model using Lawson-Swayne method.

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

```
function [P, PL, PR, zm, zmin, zmax, h] = makeTransformedSABRDensityLawsonSwayne(
    alpha, beta, nu, rho, forward, T, N, timesteps, nd)
  [zmin, zmax] = computeBoundaries(alpha, beta, nu, rho, forward, T, nd);
  J = N-2;  h0 = (zmax-zmin)/(J); j0 = int32((0-zmin)/h0);
  h = (0-zmin)/(double(j0)-0.5);
  z = (0:(J+1))*h + zmin; zmax = z(J+2); zm = z - 0.5*h;
  ym = Y(alpha, nu, rho, zm);
  ymax = Y(alpha, nu, rho, zmax); ymin = Y(alpha, nu, rho, zmin);
  Fm = F(forward, beta, ym);
  Fmax = F(forward, beta, ymax); Fmin = F(forward, beta, ymin);
  Fm(1) = 2*Fmin-Fm(2); Fm(J+2)= 2*Fmax - Fm(J+1);
  Cm = C(alpha, beta, rho, nu, ym, Fm); Cm(1) = Cm(2); Cm(J+2) = Cm(J+1);
  Gammam =  G(forward, beta, Fm, j0);
  dt = T/timesteps;
  b = 1 - sqrt(2)/2; %Lawson Swayne param
  dt1 = dt*b; dt2 = dt*(1-2*b); Em = ones(1,J+2);
  Emdt1 = exp(rho*nu*alpha*Gammam*dt1); Emdt1(1)= Emdt1(2); Emdt1(J+2)= Emdt1(J+1);
  Emdt2 = exp(rho*nu*alpha*Gammam*dt2); Emdt2(1)= Emdt2(2); Emdt2(J+2)= Emdt2(J+1);
  PL = 0.0; PR = 0.0; P = zeros(J+2,1); P(j0+1,1)=1.0/h;
  for t = 1:timesteps
    Em = Em .* Emdt1; [P1, PL1, PR1] = solveStep(Fm, Cm, Em, dt1, h, P, PL, PR);
    Em = Em .* Emdt1; [P2, PL2, PR2] = solveStep(Fm, Cm, Em, dt1, h, P1, PL1, PR1);
    P=(sqrt(2)+1)*P2-sqrt(2)*P1;
    PL=(sqrt(2)+1)*PL2-sqrt(2)*PL1;
    PR=(sqrt(2)+1)*PR2-sqrt(2)*PR1;
    Em = Em .* Emdt2;
  end
end
function [P, PL, PR] = solveStep(Fm, Cm, Em, dt, h, P, PL, PR)
  frac = dt/(2*h); M = length(P);
  B(2:M-1) = 1.0 + frac*(Cm(2:M-1).*Em(2:M-1).*(1./(Fm(3:M)-Fm(2:M-1))+1./(Fm(2:M
      -1)-Fm(1:M-2))));
  C(2:M-1) = -frac* Cm(3:M).*Em(3:M)./(Fm(3:M)-Fm(2:M-1));
  A(1:M-2) = -frac* Cm(1:M-2).*Em(1:M-2)./(Fm(2:M-1)-Fm(1:M-2));
  B(1) = Cm(1)/(Fm(2)-Fm(1))*Em(1); C(1) = Cm(2)/(Fm(2)-Fm(1))*Em(2);
  B(M) = Cm(M)/(Fm(M)-Fm(M-1))*Em(M); A(M-1) = Cm(M-1)/(Fm(M)-Fm(M-1))*Em(M-1);
  tri = diag(sparse(B))+diag(sparse(A),-1)+diag(sparse(C),1);
  %tri(2,1:end);
  P(1) = 0; P(M) = 0;
  P = tri\P;
  PL = PL + dt*Cm(2)/(Fm(2)-Fm(1))*Em(2)*P(2);
  PR = PR + dt*Cm(M-1)/(Fm(M)-Fm(M-1))*Em(M-1)*P(M-1);
end
function [zmin, zmax] = computeBoundaries(alpha, beta, nu, rho, forward, T, nd)
  zmin = -nd*sqrt(T); zmax = -zmin;
  if (beta < 1)
    ybar = -forward^(1-beta)/(1-beta);
    zbar = -1/nu*log((sqrt(1-rho^2+(rho+nu*ybar/alpha)^2)-rho-nu*ybar/alpha)/(1-rho
        ));
    if (zbar > zmin)
      zmin = zbar;
    end
  end
end
function Y = Y(alpha, nu, rho, zm)
  Y = alpha/nu*(sinh(nu*zm)+rho*(cosh(nu*zm)-1));
end
function F = F(forward, beta, ym)
  F = (forward^(1-beta)+(1-beta)*ym).^(1/(1-beta));
end
function Cm = C(alpha, beta, rho, nu, ym, Fm)
  Cm = sqrt(alpha^2+2*rho*alpha*nu*ym+nu^2*ym.^2).*Fm.^(beta);
end
```

Electronic copy available at: https://ssrn.com/abstract=2402001

```
function G = G(forward, beta, Fm, j0)
  G = (Fm.^beta-forward^beta)./(Fm-forward);
  G(j0+1) = beta/forward.^(1-beta);
end
```

The performance numbers in this paper come from an optimized Scala implementation, not from this Octave code.

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

```
function p = priceCallTransformedSABRDensity(strike, alpha, beta, nu, rho, forward,
     T, P, PL,PR, zmin, zmax, h)
  ystrike = yOfStrike(strike, forward, beta);
  zstrike = -1/nu*log((sqrt(1-rho^2+(rho+nu*ystrike/alpha)^2)-rho-nu*ystrike/alpha)
      /(1-rho));
  if (zstrike <= zmin)
    p = forward - strike;
  else
    if (zstrike >= zmax)
      p = 0;
    else
      Fmax = makeForward(alpha, beta, nu, rho, forward, zmax);
      p = (Fmax - strike) * PR;
      k0 = ceil((zstrike - zmin)/h);
      ztilde = zmin + k0*h;
      ftilde = makeForward(alpha, beta, nu, rho, forward, ztilde);
      term = ftilde - strike;
      if (term > 1e-5)
        zm = zmin + (k0 - 0.5) * h;
        Fm = makeForward(alpha, beta, nu, rho, forward, zm);
        dFdz = (ftilde-Fm) / (ztilde - zm);
        p = p + 0.5 * term * term * P(k0+1) / dFdz;
      end
      k = k0+1:length(P)-2;
      zm = zmin + (k - 0.5) * h;
      Fm = makeForward(alpha, beta, nu, rho, forward, zm);
      p = p + (Fm - strike) * h * P(k+1);
    end
  end
end
function F = makeForward(alpha, beta, nu, rho, forward, z)
  F = F(forward, beta, Y(alpha,nu,rho,z));
end
function y = yOfStrike(strike, forward, beta)
  y = (strike^(1-beta)-forward^(1-beta))/(1-beta);
end
```

Listing 3: functions override for the free-boundary SABR

```
function [zmin, zmax] = computeBoundaries(alpha, beta, nu, rho, forward, T, nd)
  zmin = -nd*sqrt(T); zmax = -zmin;
end
function F = F(forward, beta, ym)
  u = sign(forward)*abs(forward)^(1-beta)+(1-beta)*ym;
  F = sign(u).*abs(u).^(1/(1-beta));
end
function Cm = C(alpha, beta, rho, nu, ym, Fm)
  Cm = sqrt(alpha^2+2*rho*alpha*nu*ym+nu^2*ym.^2).*abs(Fm).^(beta);
end
function G = G(forward, beta, Fm, j0)
  G = (abs(Fm).^beta-abs(forward)^beta)./(Fm-forward);
  G(j0+1) = sign(forward)*beta/abs(forward).^(1-beta);
end
function y = yOfStrike(strike, forward, beta)
  y = (sign(strike)*abs(strike)^(1-beta)-sign(forward)*abs(forward)^(1-beta))/(1-
      beta);
end
```

Electronic copy available at: https://ssrn.com/abstract=2402001

Appendix B. Reference Implementation Values

Scheme	ATM price	$\theta(0)$	P_L	P_R
CN	0.156227316001	-75.391631075105	0.036151920718	0.000014227771
RAN	0.149166031026	0.486588975069	0.037035726447	0.000023444260
BDF2	0.149369112184	0.478480554553	0.036571170374	0.000036350018
RE	0.149622595293	0.482424678160	0.036971313633	-0.000001440333
LMG2	0.149449019862	0.486727660422	0.037356585469	-0.000003103630
LS	0.149701955629	0.482422521404	0.036472664324	0.000011244607
TRBDF2	0.149703527234	0.482401023656	0.036469263805	0.000011230925
Bathe	0.149631007454	0.486420051293	0.036725562889	0.000008932123

Table B1.: Sample values using 500 points and 5 time-steps,
$\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%, T = 1, f = 1, n_{sd} = 4, \delta = 0.2, h = 0.012018637349$

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. Extrapolation methods like RE and LMG2 produce here a very low negative accumulated probability at the higher boundary: there are very small oscillations in those schemes around the higher boundary. In practice it is unlikely to matter, furthermore, they quickly converge to the true value when the number of time-steps is increased.

Appendix C. Convergence Tables

Electronic copy available at: https://ssrn.com/abstract=2402001

J	N	ATM Value	Change	Ratio	Time(s)	ATM Value	Change	Ratio	Time(s)
CN						LMG2			
80	5	38.92945097	NaN	NaN	1.1e-04	37.66503968	NaN	NaN	1.4e-04
160	10	37.13441397	-1.8e+00	NaN	3.0e-04	37.70865944	4.4e-02	NaN	3.9e-04
320	20	37.42600113	2.9e-01	-6.2	7.8e-04	37.72199774	1.3e-02	3.3	1.3e-03
640	40	37.57545976	1.5e-01	2.0	2.1e-03	37.72578272	3.8e-03	3.5	4.9e-03
1280	80	37.65103999	7.6e-02	2.0	7.3e-03	37.72680412	1.0e-03	3.7	1.8e-02
2560	160	37.68906248	3.8e-02	2.0	2.7e-02	37.72707055	2.7e-04	3.8	7.1e-02
RAN						LMG3			
80	5	37.59281281	NaN	NaN	1.4e-04	37.70255505	NaN	NaN	1.8e-04
160	10	37.69385914	1.0e-01	NaN	2.9e-04	37.72145184	1.9e-02	NaN	5.7e-04
320	20	37.71890369	2.5e-02	4.0	6.8e-04	37.72587154	4.4e-03	4.3	2.0e-03
640	40	37.72510265	6.2e-03	4.0	2.2e-03	37.72686398	9.9e-04	4.5	7.6e-03
1280	80	37.72664749	1.5e-03	4.0	7.6e-03	37.72709114	2.3e-04	4.4	2.9e-02
2560	160	37.72703321	3.9e-04	4.0	2.7e-02	37.72714460	5.3e-05	4.2	1.2e-01
BDF2						LS			
80	5	37.64496421	NaN	NaN	9.7e-05	37.72979145	NaN	NaN	1.2e-04
160	10	37.70648716	6.2e-02	NaN	2.1e-04	37.72762970	-2.2e-03	NaN	2.9e-04
320	20	37.72178602	1.5e-02	4.0	7.3e-04	37.72729564	-3.3e-04	6.5	9.5e-04
640	40	37.72578924	4.0e-03	3.8	2.1e-03	37.72719717	-9.8e-05	3.4	3.4e-03
1280	80	37.72681485	1.0e-03	3.9	7.3e-03	37.72717084	-2.6e-05	3.7	1.2e-02
2560	160	37.72707451	2.6e-04	3.9	2.8e-02	37.72716402	-6.8e-06	3.9	4.9e-02
RE						TRBDF2			
80	5	37.70951734	NaN	NaN	1.8e-04	37.73019364	NaN	NaN	1.6e-04
160	10	37.72310519	1.4e-02	NaN	4.2e-04	37.72772657	-2.5e-03	NaN	3.0e-04
320	20	37.72621751	3.1e-03	4.4	1.6e-03	37.72731963	-4.1e-04	6.1	9.5e-04
640	40	37.72693472	7.2e-04	4.3	5.5e-03	37.72720314	-1.2e-04	3.5	3.5e-03
1280	80	37.72710615	1.7e-04	4.2	2.0e-02	37.72717233	-3.1e-05	3.8	1.3e-02
2560	160	37.72714796	4.2e-05	4.1	7.2e-02	37.72716439	-7.9e-06	3.9	5.0e-02

Table C1.: Convergence of at-the-money implied volatility using $\alpha = 35\%, \beta = 0.25, \rho = -10\%, \nu = 100\%, T = 1, f = 1, n_{sd} = 4$

Electronic copy available at: https://ssrn.com/abstract=2402001