SOUTH AFRICAN ACTUARIAL JOURNAL
SAAJ **20** (2020) 109–47
http://dx.doi.org/10.4314/saaj.v20i1.5

Comparison of numerical methods to price zero coupon bonds in a two-factor CIR model

by S Emslie & S Mataramvura

Submission date 20 November 2019
Acceptance date 9 October 2020

ABSTRACT
In this paper we price a zero coupon bond under a Cox–Ingersoll–Ross (CIR) two-factor model using various numerical schemes. To the best of our knowledge, a closed-form or explicit price functional is not trivial and has been less studied. The use and comparison of several numerical methods to determine the bond price is one contribution of this paper. Ordinary differential equations (ODEs) , finite difference schemes and simulation are the three classes of numerical methods considered. These are compared on the basis of computational efficiency and accuracy, with the second aim of this paper being to identify the most efficient numerical method. The numerical ODE methods used to solve the system of ODEs arising as a result of the affine structure of the CIR model are more accurate and efficient than the other classes of methods considered, with the Runge–Kutta ODE method being the most efficient. The Alternating Direction Implicit (ADI) method is the most efficient of the finite difference scheme methods considered, while the simulation methods are shown to be inefficient. Our choice of considering these methods instead of the other known and apparently new numerical methods (eg Fast Fourier Transform (FFT) method, Cosine (COS) method, etc.) is motivated by their popularity in handling interest rate instruments.

KEYWORDS
Cox–Ingersoll–Ross model; numerical methods; Runge–Kutta method; zero-coupon bonds; Alternating Direction Implicit method

CONTACT DETAILS
Stuart Emslie, Discovery Limited; Email: emsliestuart@gmail.com
Sure Mataramvura, Department of Actuarial Science, University of Cape Town
Email: sure.mataramvura@uct.ac.za

1. INTRODUCTION

1.1 Equilibrium term-structure models like the ones we consider in this paper (two-factor Cox–Ingersoll–Ross (CIR) models) are common in non-defaultable bond pricing. Since a coupon-paying bond can be stripped into sums of zero-coupon bonds, the focus has always been on pricing zero-coupon bonds. With affine term structures, the expectation is to get an affine yield curve whose components can be explicitly obtained. There are other cases where closed-form pricing solutions of such bonds may not be either easy to obtain (like in our case) or almost impossible to get using classical methods. Our paper will not concentrate on justifying the right term-structure model, but on comparison of numerical methods used to price the zero coupon bond given a CIR two-factor model structure. To the best of our knowledge this is the first time such rigorous numerical comparison and subsequent simulation using real-life data has been done.

1.2 The model considered in this paper is a time-homogeneous short-rate equilibrium two-factor model. The two most prominent short-rate models are arguably the single-factor models proposed by Vasicek (1977) and Cox et al. (1985). Arguments for more than one factor models can be found, for example, in Stambaugh (1988) and Chen & Scott (2003) who argue that two- or three-factor models are required to adequately represent market dynamics. Litterman & Scheinkman (1991) propose the use of three-factor models while Jamshidian & Zhu (1996) argue that a single factor explains approximately 70 per cent of the variation in yield curves, with a second factor explaining approximately 15 per cent. The focus of this paper is on pricing a zero-coupon bond in a two-factor extension of the CIR model by means of numerical methods.

1.3 As mentioned, to the best of our knowledge, a closed-form solution to the price of the non-defaultable zero-coupon bond in this term-structure setup has not been dealt with adequately. We fill the gap by showing that several numerical methods adequately give best approximations and this is shown in the numerical results provided. The use of numerical methods by Duffie & Kan (1996) to estimate the solution of a partial differential equation (PDE) of a similar form to that presented in this paper and the argument by Brennan & Schwartz (1979) that no closed-form solution exists for a similar PDE is reason to believe that a closed-form solution does not exist. This allows for the identification of efficient methods that can be used to construct yield curves.

1.4 Literature comparing some of the numerical methods used in this paper is considered in section 3, although few comparisons of applications similar to those presented in this paper were found. One of the contributions of this paper is the comparison of several numerical methods across three different classes of methods: ordinary differential equation (ODE) methods used to solve the system of ODEs arising as a result of the affine structure of the CIR model; finite difference scheme methods applied to the PDE of the CIR model; and simulation methods applied to the equation of the bond price. A second contribution of this paper is the identification of the most efficient numerical method of all those considered. The results in this paper may also be relevant to other short-rate models, although not all classes of numerical methods are applicable to other short-rate models.

1.5 This paper begins with a description of the two-factor CIR model in section 2. This section also contains the derivation of three equations, any of which can be solved to obtain the price of zero-coupon bonds in the two-factor CIR model. Section 3 describes the numerical methods considered in this paper, after which a description of the methodology used for the comparison is described in section 4. Section 5 contains the results of the comparison, with section 6 containing the conclusions.

2. THE BLACK–SCHOLES EQUATION FOR THE PRICE

2.1 Without loss of generality, the two-factor CIR model considered in this paper is the specification of the model according to Shreve (2004), and is given by the short rate under the risk neutral measure Q, where

$R(t)=\delta_0+\delta_1 Y_1(t)+\delta_2 Y_2(t)$; $R(0)\geq 0$; $\delta_0\geq 0; \delta_1>0; \delta_2>0$ and

$Y_1(t)$ and $Y_2(t)$ are two-factor CIR models satisfying the stochastic differential equations:

$$dY_1(t)=\left(\mu_1-\lambda_{11}Y_1(t)-\lambda_{12}Y_2(t)\right)dt+\sqrt{Y_1(t)}d\tilde{B}_1(t) \text{ and}$$

$$dY_2(t)=\left(\mu_2-\lambda_{21}Y_1(t)-\lambda_{22}Y_2(t)\right)dt+\sqrt{Y_2(t)}d\tilde{B}_2(t),$$

where $\tilde{B}_1(t)$ and $\tilde{B}_2(t)$ are independent standard Brownian motions under Q and $\mu_1\geq 0$, $\mu_2\geq 0$, $\lambda_{11}>0$, $\lambda_{22}>0$, $\lambda_{12}\leq 0$, $\lambda_{21}\leq 0$, $Y_1(0)\geq 0$. The processes $Y_1(t)$, $Y_2(t)$ and $R(t)$ are adapted to the natural filtration $\mathcal{F}_t=\sigma_{0\leq s\leq t}\left(Y_1(s),Y_2(s)\right)$. By construction, we expect that in probability, $R(t)\geq 0$ for all $t\geq 0$, ensuring consistency with the market dynamic of positive interest rates. The above CIR model is in canonical form (specified such that it contains the least number of parameters), from which more complex two-factor affine yield models can be derived (Shreve, 2004).

2.2 Throughout this paper it is assumed the non-defaultable zero-coupon bond (simply referred to as bond in this paper) will mature at time T with redemption of 1, with the bond price being calculated at the current time $t\leq T$. The remaining time to maturity is $\tau=T-t$. The bond price is denoted as $P(t,T)$ and since the price process is Markovian, we have $P(t,T)=f(t,y_1,y_2)$ for some measurable function f.

2.3 In the rest of the paper, we shall use the notation f_x to imply $\frac{\partial f}{\partial x}$. Then the Black–Scholes partial differential equation (PDE) for the price

$$f(t,y_1,y_2)=\mathbb{E}_Q\left[e^{-\int_t^T R(s)ds} \mid \mathcal{F}_t\right];\ \left(f(T,y_1,y_2)=1\right) \tag{1}$$

is given by:

$$\begin{aligned}&-[\delta_0+\delta_1 y_1+\delta_2 y_2]f(t,y_1,y_2)+f_t(t,y_1,y_2)+f_y\ (t,y_1,y_2)[\mu_1-\lambda_{11}y_1-\lambda_{12}y_2]+\\&f_{y_2}(t,y_1,y_2)[\mu_2-\lambda_{21}y_1-\lambda_{22}y_2]+\frac{1}{2}f_{y_1y_1}(t,y_1,y_2)y_1+\frac{1}{2}f_{y_2y_2}(t,y_1,y_2)y_2=0\end{aligned} \tag{2}$$

with boundary condition $f(T,y_1,y_2)=1$ for all $y_1\geq 0, y_2\geq 0$ and for all $0\leq t<T$. The above PDE is linear and of order two in three variables. The bond price $f(t,y_1,y_2)$ (assuming $Y_1(t)=y_1$ and $Y_2(t)=y_2$) can be calculated directly (where possible) by finding closed-form solutions to this PDE given the unique boundary condition. Where that is not easy or possible, like in this case, we use numerical solutions to PDEs applied to eq. (2). We shall be using finite difference methods as part of solution methods in the next section.

2.4 Pricing this bond could be done in many different ways. One way is through simulation (eg Monte Carlo simulation) of eq. (1). As stated before, if a closed-form solution to the price exists, we could, as another method, explicitly solve the PDE eq. (2). That could easily be achieved, seeing that the bond price can be specified as:

$$f(t,y_1,y_2)=e^{-y_1C_1(\tau)-y_2C_2(\tau)-A(\tau)} \tag{3}$$

for some functions $C_1(\tau)$, $C_2(\tau)$ and $A(\tau)$ to be determined, so that eq. (2) subsequently becomes:

$$\begin{aligned}&\left(-\delta_1+C_1'(\tau)+C_1(\tau)\lambda_{11}+C_2(\tau)\lambda_{21}+\frac{1}{2}C_1(\tau)^2\right)y_1f+(A'(\tau)-C_1(\tau)\mu_1-\\&C_2(\tau)\mu_2-\delta_0)f+\left(-\delta_2+C_2'(\tau)+C_1(\tau)\lambda_{12}+C_2(\tau)\lambda_{22}+\frac{1}{2}C_2(\tau)^2\right)y_2f=0\end{aligned}$$

which must hold for all $y_1\geq 0, y_2\geq 0$.

2.5 Solving for $C_1(\tau), C_2(\tau)$ and $A(\tau)$ leads to the following system of three ODEs:

$$\begin{aligned}&-\delta_0+A'(\tau)-C_1(\tau)\mu_1-C_2(\tau)\mu_2 &&=0\\&-\delta_1+C_1'(\tau)+C_1(\tau)\lambda_{11}+C_2(\tau)\lambda_{21}+\frac{1}{2}C_1(\tau)^2 &&=0\\&-\delta_2+C_2'(\tau)+C_1(\tau)\lambda_{12}+C_2(\tau)\lambda_{22}+\frac{1}{2}C_2(\tau)^2 &&=0\end{aligned} \tag{4}$$

The initial conditions are $C_1(0)=0;\ C_2(0)=0;\ A(0)=0.$

2.6 We posit that the non-linear system of ODEs given by eq. (4) are not trivial in finding explicit solutions. We are less interested in the resulting phase planes and thus we in part use the ODE class of numerical methods to estimate the bond price apart from the finite difference methods proposed before. We show that our choice of numerical methods works perfectly because all give good approximations to at least three decimal places.

3. NUMERICAL METHODS

3.1 Simulation methods, finite difference scheme (FD) methods and ODE methods are the three categories of numerical methods considered in this paper. Multiple methods within each category are considered. The two simulation methods considered are Monte Carlo simulation and antithetic variates. The five FD methods considered are the alternating direction implicit (ADI) method, the explicit method, the Hopscotch method, the implicit method and the Crank–Nicolson (CN) method. The five numerical methods considered for ODE are the implicit method, the Euler/explicit method, the Runge–Kutta method, the Taylor method (second order) and the Crank–Nicolson (CN) method. Note that the last group target the nonlinear system of ODEs (4) while the FD methods are used for the PDE boundary value eq. (2). The simulation methods are used in eq. (1). We show by way of results for chosen parameters how the three sets of methods give approximate values for $f\left(t, y_1, y_2\right)$.

3.2 Little prior research has been found on the comparison of numerical methods in pricing non-defaultable zero-coupon bonds across the three classes of methods considered, with this comparison being one of the contributions of this paper. An immediate comparison between FD methods and simulation methods can be found in Boyle (1977) who proposes that simulation methods are computationally inefficient but have the advantage of flexibility while Wilmott (2006) argues that for models containing less than four random factors, finite difference schemes are more efficient than simulation methods. We shall look more closely into these comparisons in section 5.

3.3 Few examples have been found in prior research of comparisons of some of the FD methods considered in this paper in an application that is similar to the one in this paper. Geske & Shastri (1985) compare the computational efficiency of the implicit and explicit methods when used to solve the Black–Scholes PDE, showing the explicit method to take roughly 60 per cent of the computation time taken by the implicit method. Hull & White (1990) argue that the explicit scheme uses between 40 and 70 per cent of the computation time of the implicit method. Although not in an application similar to that in this paper, previous results when using these methods (see eg Cairns, 2004) show that the CN method converges more quickly than the explicit and implicit methods, while the rate of convergence of the implicit method may be quicker than the explicit method in certain applications, but not all. Duffie & Kan (1996) argues that the ADI method is less computationally intensive than the standard implicit method and the Crank–Nicolson method, but is not guaranteed to result in stable solutions. They further showed that the errors resulting from the ADI method are small when used to solve a three dimensional PDE similar to that in this paper. The results from some of theses papers are compared to the results in this paper in section 5.

3.4 Simulation methods

3.4.1 We briefly summarise the Monte Carlo simulation. For more detailed treatment, the reader can refer, for example, to Cairns (2004:185) and references therein.

3.4.2 The time interval $(0,T-t)$ is divided into $n=\dfrac{T-t}{\Delta_t}$ equally spaced intervals $t_{k+1}-t_k$, where $t_k=\dfrac{k}{n}$ and $0\le k\le n$. A simulated path of the short rate is obtained as

$$R(t_k)=\delta_0+\delta_1Y_1(t_k)+\delta_2Y_2(t_k)\text{, using:}$$

$$Y_1(t_k)=Y_1(t_{k-1})+\left(\mu_1-\lambda_{11}Y_1(t_{k-1})-\lambda_{12}Y_2(t_{k-1})\right)\Delta_t+\sqrt{Y_1(t_{k-1})\Delta_t}\,z_{1k}\text{ and}$$

$$Y_2(t_k)=Y_2(t_{k-1})+\left(\mu_2-\lambda_{21}Y_1(t_{k-1})-\lambda_{22}Y_2(t_{k-1})\right)\Delta_t+\sqrt{Y_2(t_{k-1})\Delta_t}\,z_{2k},$$

where z_{1k} and z_{2k} are independent random simulations of the standard normal distribution. The integral $\int_t^T R(s)$ is estimated as $I_i=\Delta_t\sum_{k=0}^{n-1}R(t_k)$, from which $e^{-\int_t^T R(s)ds}$ is estimated as $A_i=e^{-I_i}$ and A_i is estimated for $i=1,\ldots,N$. The sample expectation $\hat{P}(t,T)=f(t,y_1,y_2)=\dfrac{1}{N}\sum_{k=1}^{N}A_i$ is the estimated bond price.

3.4.3 The antithetic variates method is a variance reduction method using a similar procedure to that specified above. Each iteration involves two simulated paths $R_1(t_k)$ and $R_2(t_k)$. Two estimates of $e^{-\int_t^T R(s)ds}$ are obtained, labelled A_{i1} and A_{i2}. The first simulated path uses the values z_{1i} and z_{2i} as specified above, the second uses $-z_{1i}$ and $-z_{2i}$. The estimate A_i in each iteration is calculated as the average of A_{i1} and A_{i2}.

3.4.4 There are two causes of error in the estimated price $\hat{P}(t,T)$. The first is the variation in the sample estimate of the expectation, reduced as the number of simulations increases. The second is the discretisation error of the short rate and the error in estimating an integral with a discrete sum, reduced as $\Delta_t\to0$. The antithetic variates method is expected to result in a lower variance of the estimated price than Monte Carlo simulation (Cairns, 2004). Boyle et al. (1997), however, argues that the antithetic variates method is inefficient in reducing the variance of the estimated price, and suggests the alternatives of control variates, Latin hypercube sampling and moment matching. Joy et al. (1996) suggests quasi-Monte Carlo methods as an additional alternative. These alternative methods are not considered in this paper.

3.5 Finite difference scheme (FD) methods

3.5.1 OVERVIEW

3.5.1.1 In order to specify eq. (2) in terms of bounded variables for easier application of the FD methods, y_1 and y_2 were transformed to the variables θ_1 and θ_2. The bond price is subsequently represented as $f(t,\theta_1,\theta_2)$. The common transformation used (see for example, Brennan & Schwartz (1979: 152)) is:

$$\theta_1=\frac{1}{1+k_1y_1};\quad\theta_2=\frac{1}{1+k_2y_2}\quad\Rightarrow\quad0\le\theta_1,\,\theta_2\le1. \tag{5}$$

3.5.1.2 The following notation is used in describing the FD methods:

$$a_0 = \delta_0 + \delta_1 y_1 + \delta_2 y_2$$
$$a_1 = y_1 k_1^2 \theta_1^3 - k_1 \theta_1^2 \left(\mu_1 - \lambda_{11} y_1 - \lambda_{12} y_2\right)$$
$$a_2 = y_2 k_2^2 \theta_2^3 - k_2 \theta_2^2 \left(\mu_2 - \lambda_{21} y_1 - \lambda_{22} y_2\right)$$
$$b_1 = y_1 k_1^2 \theta_1^4$$
$$b_2 = y_2 k_2^2 \theta_2^4$$

3.5.1.3 Using the transformation in eq. (5), eq. (2) becomes (details are given in Appendix B1):

$$-a_0 f + f_t - a_1 f_{\theta_1} - a_2 f_{\theta_2} + \frac{1}{2} b_1 f_{\theta_1 \theta_1} + \frac{1}{2} b_2 f_{\theta_2 \theta_2} = 0 \qquad (6)$$

3.5.1.4 The FD methods used in this paper numerically estimate $f\left(t, \theta_1, \theta_2\right)$ by discretising eq. (6). An $\left(n+1\right) \times \left(n+1\right)$ grid of (θ_{1i}, θ_{2j}) points, where $\Delta_{\theta_1} = \Delta_{\theta_2} = \frac{1}{n}$ and $\theta_{1j} = \theta_{2j} = \frac{j}{n}$ for $0 \le i, j \le n$, is created. It is assumed that the time interval (t,T) is divided into $x = \frac{T-t}{\Delta_t}$ equal steps for some Δ_t. The FD method estimate of $f\left(t, \theta_{1i}, \theta_{2j}\right)$ is denoted U_{ij}^t. The FD methods iteratively calculate U_{ij}^s using the values of $U_{ij}^{s+\Delta_t}$ (input grid) for all $0 \le i,\ j \le n$ and $s = t,\ t + \Delta_t,\ \ldots,\ T - \Delta_t$, starting with $U_{ij}^T = 1$ for all i, j. A higher value for k_1 and k_2 results in the grid being more concentrated around smaller values of y_1 and y_2. The values used for k_1 and k_2 differed for each set of parameters considered, but were the same for each of the five FD methods for a given set of parameters.

3.5.1.5 The initial conditions used were $U_{00}^s = U_{i0}^s = U_{0j}^s = 0$ for $0 < i,\ j \le n$ and $t \le s < T$. The initial conditions for $\theta_1 = 1$ and/or $\theta_2 = 1$ are found by substituting $y_1 = 0$ and/or $y_2 = 0$ into the discretised equations (described below). The values of $U_{(n+1)j}^t$ and $U_{i(n+1)}^t$ were therefore not required. A detailed discussion of the initial conditions is available in Appendix B2.

3.5.2 IMPLEMENTATION OF THE FD METHODS

3.5.2.1 The following five paragraphs will summarise each of the five FD methods used in this paper. We refer the reader to eg to Wilmott (2006: 1260) and references therein for a more elaborate treatment.

3.5.2.2 The explicit method involves discretising eq. (6) using the approximations in Appendix B4 as:

$$-a_0U_{ij}^{t+\Delta_t}+\frac{U_{ij}^{t+\Delta_t}-U_{ij}^{t}}{\Delta_t}+a_1\frac{U_{ij}^{t+\Delta_t}-U_{(i-1)j}^{t+\Delta_t}}{\Delta_{\theta_1}}+a_2\frac{U_{ij}^{t+\Delta_t}-U_{i(j-1)}^{t+\Delta_t}}{\Delta_{\theta_2}}+ \\ \frac{1}{2}b_1\frac{U_{(i+1)j}^{t+\Delta_t}-2U_{ij}^{t+\Delta_t}+U_{(i-1)j}^{t+\Delta_t}}{\Delta_{\theta_1}{}^2}+\frac{1}{2}b_2\frac{U_{i(j+1)}^{t+\Delta_t}-2U_{ij}^{t+\Delta_t}+U_{i(j-1)}^{t+\Delta_t}}{\Delta_{\theta_2}{}^2}=0. \quad (7)$$

The value of U_{ij}^t is the only unknown in eq. (7) and subsequently easily found, detailed in Appendix B5. Solving U_{ij}^t for all $0 < i, j \leq n$ defines the new grid for each iteration. For the scheme to work, we must check that the 'probabilities' given by the parameters in eq. (6) connecting U_{ij}^t with the three known values $U_{i(j-1)}^{t+\Delta t}$, $U_{ij}^{t+\Delta t}$ and $U_{i(j+1)}^{t+\Delta t}$ are always non-negative.

3.5.2.3 The implicit method involves discretising eq. (6) using the approximations in Appendix B4 as:

$$-a_0U_{ij}^{t}+\frac{U_{ij}^{t+\Delta_t}-U_{ij}^{t}}{\Delta_t}+a_1\frac{U_{ij}^{t}-U_{(i-1)j}^{t}}{\Delta_{\theta_1}}+a_2\frac{U_{ij}^{t}-U_{i(j-1)}^{t}}{\Delta_{\theta_2}}+ \\ \frac{1}{2}b_1\frac{U_{(i+1)j}^{t}-2U_{ij}^{t}+U_{(i-1)j}^{t}}{\Delta_{\theta_1}{}^2}+\frac{1}{2}b_2\frac{U_{i(j+1)}^{t}-2U_{ij}^{t}+U_{i(j-1)}^{t}}{\Delta_{\theta_2}{}^2}=0 \quad (8)$$

The values of $U_{ij}^t, U_{(i+1)j}^t$, $U_{(i-1)j}^t$, $U_{i(j+1)}^t$ and $U_{i(j-1)}^t$ are unknown in eq. (8) above. A set of n^2 equations is obtained using eq. (8) for $0 < i, j \leq n$. The equations are solved simultaneously to find U_{ij}^t, $0 < i, j \leq n$ in each iteration, with more detail in Appendix B6.

3.5.2.4 The discretised version of eq. (6) used in implementing the CN method is an average of eq. (7) and eq. (8). As with the implicit method, n^2 equations containing U_{ij}^t, $U_{(i+1)j}^t$, $U_{(i-1)j}^t$, $U_{i(j+1)}^t$ and $U_{i(j-1)}^t$ for $0 < i, j \leq n$ are solved simultaneously in each iteration, with more detail in Appendix B7.

3.5.2.5 When using the Hopscotch method, each iteration consists of two stages. In the first stage, U_{ij}^t is calculated as was done in the explicit method using eq. (7) for all the odd or even points (a point is odd or even depending on whether its grid reference $i+j$ is odd or even). The remaining grid points are then calculated in stage two as was done in the explicit method using eq. (7), but where the input grid is updated with the values from stage one (i.e. $U_{(i+1)j}^{t+\Delta_t}$, $U_{(i-1)j}^{t+\Delta_t}$, $U_{i(j+1)}^{t+\Delta_t}$ and $U_{i(j-1)}^{t+\Delta_t}$ are replaced by the values calculated in stage one). The odd points are calculated in stage one and even points in stage two in the first iteration. The stages in which the odd and even points are calculated swap for each subsequent iteration. More detail is found in Appendix B8.

3.5.2.6 The ADI method also consists of two stages in each iteration. The first stage calculates a new grid at $t+\frac{1}{2}\Delta_t$ by solving θ_1 implicitly and θ_2 explicitly. The values of

$U_{ij}^{t+\frac{1}{2}\Delta_t}$ for $0 < i, j < n$ are found using the grid containing $U_{ij}^{t+\Delta_t}$ for $0 < i, j \leq n$. The second stage calculates a new grid at time t by solving θ_2 implicitly and θ_1 explicitly, using the grid from stage one as an input. Each stage results in n independent sets of n simultaneous equations, the solution of which is the new grid. More detail is found in Appendix B9.

3.5.3 Comparisons of the FD methods

3.5.3.1 Finite difference methods should preferably be stable, consistent and convergent. An FD method is stable if an error arising in any iteration remains bounded. An FD method is consistent if the truncation error $\left|f\left(t,\theta_{1i},\theta_{2i}\right)-U_{ij}^{t}\right|$ approaches zero as $\Delta_{\theta_1},\Delta_{\theta_2}$ and Δ_t approach zero for all iterations. An FD method is convergent if the estimated bond price approaches the true solution of the PDE as $\Delta_{\theta_1},\Delta_{\theta_2}$ and Δ_t approach zero.

3.5.3.2 The explicit method is stable if Δ_t is sufficiently small. The implicit and CN methods are expected to be stable for larger values of Δ_t. The ADI and Hopscotch methods attempt to retain the stability of the implicit method while increasing the rate of convergence of the implicit method. Both, however, are not guaranteed to result in stable solutions.

3.6 ODE methods

3.6.1 Five ODE methods used to estimate the solution to eq. (3) using eq. (4) are considered. Comparison of additional methods such as those proposed by Bulirsch & Stoer (1966) and Bashforth & Adams (1883) is an avenue for further research. The ODE methods involve iteratively calculating the values of $C_1\left(\tau+\Delta_\tau\right)$, $C_2\left(\tau+\Delta_\tau\right)$ and $A\left(\tau+\Delta_\tau\right)$ for some Δ_τ time step using $C_1\left(\tau\right)$, $C_2\left(\tau\right)$ and $A\left(\tau\right)$ as inputs, starting with $C_1\left(0\right)=C_2\left(0\right)=A\left(0\right)=0$ and ending when $\tau=T-t$ (Shampine, 1994). The bond price is estimated as $\hat{P}\left(t,T\right)=\hat{f}\left(t,y_1,y_2\right)=e^{-y_1C_1(T-t)-y_2C_2(T-t)-A(T-t)}$. The true bond price is $P\left(t,T\right)=f\left(t,y_1,y_2\right)$.

3.6.2 An explicit method uses $C_1\left(\tau\right)$, $C_2\left(\tau\right)$ and $A\left(\tau\right)$ in the formulation of each iteration, while an implicit method uses $C_1\left(\tau+\Delta_\tau\right)$, $C_2\left(\tau+\Delta_\tau\right)$ and $A\left(\tau+\Delta_\tau\right)$. A Crank–Nicolson method uses the average of the functions at τ and $\tau+\Delta_\tau$. The first method considered is the explicit Euler (or Euler/explicit) method. The implicit method considered is an implicit implementation of the Euler method and the CN method is an average of the explicit and implicit Euler methods. The Taylor method is explicit, although implicit and CN versions can be developed. The Runge–Kutta method considered is explicit, with implicit examples found in Alexander (1977).

3.6.3 The following functions of C_1 and C_2 are used for the ODE methods:

$$r_1=\delta_1-C_1\lambda_{11}-C_2\lambda_{21}-\frac{1}{2}C_1^{\,2};\ s_1=\delta_2-C_1\lambda_{12}-C_2\lambda_{22}-\frac{1}{2}C_2^{\,2};\ u_1=\delta_0+C_1\mu_1+C_2\mu_2;$$

$$f_1=-r_1\lambda_{11}-s_1\lambda_{21}-r_1C_1;\ f_2=-r_1\lambda_{12}-s_1\lambda_{22}-s_1C_2;\ f_3=r_1\mu_1+s_1\mu_2;$$

$$r_i=r_1\left(C_1+\frac{1}{2}\Delta_\tau r_{i-1},C_2+\frac{1}{2}\Delta_\tau s_{i-1}\right);\ s_i=s_1\left(C_1+\frac{1}{2}\Delta_\tau r_{i-1},C_2+\frac{1}{2}\Delta_\tau s_{i-1}\right);$$

$$u_i=u_1\left(C_1+\frac{1}{2}\Delta_\tau r_{i-1},C_2+\frac{1}{2}\Delta_\tau s_{i-1}\right)\text{ for } i=2,3;$$

$r_4 = r_1\left(C_1 + \Delta_\tau r_3, C_2 + \Delta_\tau s_3\right)$; $s_4 = s_1\left(C_1 + \Delta_\tau r_3, C_2 + \Delta_\tau s_3\right)$ and $u_4 = u_1\left(C_1 + \Delta_\tau r_3, C_2 + \Delta_\tau s_3\right)$.

3.6.4 For the Euler method we use a first order Taylor expansion in each iteration as follows:

$$C_1\left(\tau + \Delta_\tau\right) = C_1 + \Delta_\tau r_1;\ C_2\left(\tau + \Delta_\tau\right) = C_2 + \Delta_\tau s_1;\ \text{and}\ A\left(\tau + \Delta_\tau\right) = A + \Delta_\tau u_1.$$

3.6.5 For the second order Taylor method we use a second order expansion in each iteration as follows:

$$C_1\left(\tau + \Delta_\tau\right) = C_1 + \Delta_\tau r_1 + \frac{1}{2}\Delta_\tau{}^2 f_1;\ C_2\left(\tau + \Delta_\tau\right) = C_2 + \Delta_\tau s_1 + \frac{1}{2}\Delta_\tau{}^2 f_2;\ \text{and}$$

$$A\left(\tau + \Delta_\tau\right) = A + \Delta_\tau u_1 + \frac{1}{2}\Delta_\tau{}^2 f_3.$$

3.6.6 For the fourth order explicit Runge–Kutta method each iteration is calculated as:

$$C_1\left(\tau + \Delta_\tau\right) = C_1 + \frac{\Delta_\tau}{6}\left(r_1 + 2r_2 + 2r_3 + r_4\right);\ C_2\left(\tau + \Delta_\tau\right) = C_2 + \frac{\Delta_\tau}{6}\left(s_1 + 2s_2 + 2s_3 + s_4\right);\ \text{and}$$

$$A\left(\tau + \Delta_\tau\right) = A + \frac{\Delta_\tau}{6}\left(u_1 + 2u_2 + 2u_3 + u_4\right).$$

3.6.7 The implicit method requires $C_1\left(\tau + \Delta_\tau\right)$ and $C_2\left(\tau + \Delta_\tau\right)$ to be solved for simultaneously using the two equations:

$$\left(\frac{1}{\Delta_\tau} + \lambda_{11}\right)C_1\left(\tau + \Delta_\tau\right) + C_2\left(\tau + \Delta_\tau\right)\lambda_{21} + \frac{1}{2}C_1(\tau + \Delta_\tau)^2 = \delta_1 + \frac{1}{\Delta_\tau}C_1\left(\tau\right)\ \text{and}$$

$$\left(\frac{1}{\Delta_\tau} + \lambda_{22}\right)C_2\left(\tau + \Delta_\tau\right) + C_1\left(\tau + \Delta_\tau\right)\lambda_{12} + \frac{1}{2}C_2(\tau + \Delta_\tau)^2 = \delta_2 + \frac{1}{\Delta_\tau}C_2\left(\tau\right).$$

The value for $A\left(\tau + \Delta_\tau\right)$ in each iteration is then calculated as:

$$A\left(\tau + \Delta_\tau\right) = A\left(\tau\right) + \Delta_\tau\left(\delta_0 + C_1\left(\tau + \Delta_\tau\right)\mu_1 + C_2\left(\tau + \Delta_\tau\right)\mu_2\right).$$

3.6.8 The CN method requires $C_1\left(\tau + \Delta_\tau\right)$ and $C_2\left(\tau + \Delta_\tau\right)$ to be solved for simultaneously using the two equations:

$$\left(\frac{1}{\Delta_\tau} + \frac{\lambda_{11}}{2}\right)C_1\left(\tau + \Delta_\tau\right) + C_2\left(\tau + \Delta_\tau\right)\frac{\lambda_{21}}{2} + \frac{1}{4}C_1(\tau + \Delta_\tau)^2 = \delta_1 + \frac{1}{\Delta_\tau}C_1\left(\tau\right) + \frac{1}{2}f_1\ \text{and}$$

$$\left(\frac{1}{\Delta_\tau} + \lambda_{22}\right)C_2\left(\tau + \Delta_\tau\right) + C_1\left(\tau + \Delta_\tau\right)\lambda_{12} + \frac{1}{2}C_2(\tau + \Delta_\tau)^2 = \delta_2 + \frac{1}{\Delta_\tau}C_2\left(\tau\right) + \frac{1}{2}f_2.$$

The value for $A\left(\tau + \Delta_\tau\right)$ in each iteration is then calculated as:

$$A(\tau+\Delta_\tau)=A(\tau)+\Delta_\tau\left(\delta_0+\frac{1}{2}\left(C_1(\tau+\Delta_\tau)+C_1(\tau)\right)\mu_1+\frac{1}{2}\left(C_2(\tau+\Delta_\tau)+C_2(\tau)\right)\mu_2\right).$$

3.6.9 The implicit and CN methods are expected to be more stable for larger Δ_τ than the explicit method (Granville, 1988). The local truncation error, defined as

$$LTE=\left|\hat{P}(t,y_1,y_2)-P(t,y_1,y_2)\right| \text{ given } P(t-\Delta_t,y_1,y_2),$$

measures accuracy for each iteration. As $\Delta_t\to 0$, *LTE* converges to 0 slower for the Taylor series than the Runge–Kutta method and even slower for the Euler method (Shampine, 1994).

4. METHODOLOGY

Without loss of generality, the numerical methods are compared for the sets of parameters in Table 1 below and the six times to maturity (in years) τ = 2, 5, 10, 15, 20, 30. To stress test the results, further comparisons are made for selected numerical methods across a broader random selection of fifty sets of parameters at all times to maturity τ = 1, 2, …, 30.

4.1 Selection of parameters

4.1.1 Five sets of parameters were selected to ensure generality across different yield curve shapes and different average bond yields. Increasing, decreasing and humped yield curves are represented within these sets of parameters, as well as average bond yields ranging from $i=0{,}02$ p.a. to $i=0{,}13$ p.a. The yield curves obtained from the five parameter sets are found in Appendix A.

4.1.2 Three of the five sets of parameters were chosen to roughly represent yield curves based on data obtained from Bloomberg for government bonds in the countries of South Africa, Brazil and the United States of America (USA) on 1 June 2016. Since the focus of this paper is not parameter estimation, the parameters are chosen only to roughly represent the market yield curves. There exists a substantial amount of literature on estimation procedures that could be used for more accurate estimates, including the papers by Chen & Scott (2003) and Pearson & Sun (1994).

4.1.3 We posit that the choice of the three countries was motivated by the fact that South Africa and Brazil are developing BRICS countries while the USA is a developed country. What is evident in all cases, as expected, is that zero-coupon bond prices decrease with term.

4.1.4 The other two sets of parameters were chosen to obtain a humped yield curve (the arbitrary set of parameters) and a decreasing yield curve (the check set of parameters).

4.1.5 The majority of the analysis is based on the five sets of parameters specified in Table 1. As previously mentioned, further analysis is done for selected numerical methods at all integer times to maturity τ = 1, 2, …, 30 based on an additional fifty sets of parameters. The additional sets of parameters are chosen randomly and used only for the methods appearing computationally efficient for the parameters in Table 1. The use of random parameters is based on the method used by Broadie & Detemple (1996), with these additional random parameters being selected as follows: The values for δ_0, δ_1 and δ_2 are selected from a uniform distribution with range $(0;0{,}04)$ randomly and independently . The values for μ_1, μ_2, λ_{11} and λ_{22} are selected randomly and independently from a uniform distribution with range $(0;0{,}5)$. The values for λ_{12} and λ_{21} are selected randomly and independently from a uniform distribution with range $(-0{,}5;0)$. Y_1 and Y_2 are set equal to 1. The random parameter selection is used to stress test the results and ensure generality of the results.

TABLE 1. Sets of parameter values used for comparison of numerical methods

	South Africa	Brazil	USA	Arbitrary	Check
Y_1	1	1	1	1	1
Y_2	1	1	1	1	1
δ_0	0,04334	0,06251	0,00469	0,04663	0,02500
δ_1	0,01705	0,06910	0,00220	0,02200	0,02500
δ_2	0,01114	0,00799	0,00230	0,00636	0,02500
μ_1	0,20342	0,08955	0,62433	0,01304	0,10000
μ_2	0,26024	0,89128	1,55256	0,00854	0,10000
λ_{11}	0,36545	0,54832	0,04106	0,86828	0,10000
λ_{22}	0,25620	0,28266	0,46624	0,01275	0,10000
λ_{12}	–0,17552	–0,12509	–0,00576	0,18118	0,00000
λ_{21}	–0,37807	–0,03003	–0,01812	–0,779155	0,00000
k_1	$\frac{1}{7}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{6}{34}$	$\frac{1}{19}$
k_2	$\frac{1}{7}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{6}{34}$	$\frac{1}{19}$

4.2 Criteria used for comparison

4.2.1 The numerical methods are compared in this paper based on the two criteria of accuracy and computational efficiency.

4.2.2 The price of bonds considered in this paper assume a face value of USD1. Accuracy, however, is measured in dollars and cents for a bond with a face value of $100 (i.e. accuracy is measured to four decimal places for a bond with face value of USD1). Accuracy error is defined as $e_i = 100\left|\hat{P}(t,T) - P(t,T)\right|$ where $P(t,T)$ is the true bond price and $\hat{P}(t,T)$ is the estimated price. For example, the accuracy for a bond with price of $0,8575 is measured assuming the face value is $100 and the price is $85,75. Two additional measures of accuracy are further used in this paper, with these being the root mean square relative error (RMS) and the mean absolute error (MAE). The RMS is defined as $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{P}_i(t,T) - P_i(t,T)}{P_i(t,T)}\right)^2}$. The MAE is defined as $\frac{1}{n}\Sigma_{i=1}^{n},\left|\hat{P}_i(t,T) - P_i(t,T)\right|$. The RMS penalises large errors more than the MAE, and defines the error relative to the true bond price (Broadie & Detemple, 1996).

4.2.3 Computational efficiency is measured as the number of seconds taken to obtain the estimated bond price. Computational efficiency depends largely on the efficiency of the computer code used in implementing the numerical method, and is also influenced by the computer used to run the code. An effort was made to ensure the efficiency of the code, but the results may be impacted by inefficiencies in the code.

4.3 Determining a baseline for comparison of accuracy

4.3.1 The absence of a closed-form solution results in complexity in determining a true value to be used as a baseline for comparison of accuracy. The restriction of $\lambda_{12} = \lambda_{21} = 0$ in eq. (4), however, results in a system of ODEs for which a closed-form solution can be found. The check set of parameters was chosen with this restriction (last column of parameters), for which the closed-form solution (as calculated using WolframAlpha (2016)) is $P(t, y_1, y_2) = e^{-y_1C_1(T-t)-y_2C_2(T-t)-A(T-t)}$, where

$$C_1(\tau) = C_2(\tau) = 0{,}144949\frac{e^{0{,}244949\tau}-1}{0{,}420204+e^{0{,}244949\tau}}, \text{ and}$$

$$A(\tau) = -\delta_0\tau + \int_0^\tau C_1(s)\mu_1 + C_2(s)\mu_2 ds.$$

4.3.2 The five ODE methods all converge to the true solution (based on accuracy to seven decimal places) for the check set of parameters. For the remaining four sets of parameters and all additional random sets of parameters for which $\lambda_{12}<0$ and $\lambda_{21}<0$, the value to which the ODE methods converge (using sufficiently small Δ_τ) is used as the true value. It is further shown in the results that the other numerical methods converge to this value.

5. RESULTS

The results begin with comparisons across the three categories of methods, followed by detailed comparisons of the methods within each category.

5.1 Overview

5.1.1 The accuracy and computational efficiency of the numerical methods used to estimate the bond prices in Table 2 is summarised in Table 3. RMS and MAE are used as accuracy measures, calculated across the estimated bond prices for the six times to maturity for each parameter set and method. The total computational time required to estimate the six prices for each parameter set and method is used as the measure of efficiency.

5.1.2 The ODE methods have higher levels of accuracy while maintaining quicker computational times than the FD and simulation methods. The least efficient ODE method is a minimum of 14,7 times more computationally efficient than the most efficient of the FD and simulation methods over the five sets of parameters, while maintaining an RMS of 0 (rounded to four decimal places – referred to as 0 in the remainder of this paper). The simulation and FD methods are unable to achieve this level of accuracy in a reasonable computation time (explored further below). The ODE methods are clearly preferred on the basis of accuracy and efficiency but are only applicable to affine short-rate models.

TABLE 2. Estimated bond prices using various numerical methods for the five sets of parameters in Table 1 at six specific times to maturity[1]

	SA						Brazil					
	τ						τ					
	2	5	10	15	20	30	2	5	10	15	20	30
Runge	0,8595	0,6674	0,4199	0,2579	0,1569	0,0576	0,7735	0,5333	0,2765	0,1404	0,0710	0,0181
Euler	0,8595	0,6674	0,4199	0,2579	0,1569	0,0576	0,7735	0,5333	0,2765	0,1404	0,0710	0,0181
Taylor	0,8595	0,6674	0,4199	0,2579	0,1569	0,0576	0,7735	0,5333	0,2765	0,1404	0,0710	0,0181
Implicit (ODE)	0,8595	0,6674	0,4199	0,2579	0,1569	0,0576	0,7735	0,5333	0,2765	0,1404	0,0710	0,0181
CN (ODE)	0,8595	0,6674	0,4199	0,2579	0,1569	0,0576	0,7735	0,5333	0,2765	0,1404	0,0710	0,0181
Explicit	0,8594	0,6673	0,4199	0,2578	0,1568	0,0574	0,7736	0,5331	0,2762	0,1403	0,0709	0,0181
Implicit	0,8594	0,6673	0,4198	0,2577	0,1567	0,0574	0,7738	0,5334	0,2765	0,1405	0,0711	0,0182
CN	0,8594	0,6674	0,4200	0,2579	0,1568	0,0574	0,7737	0,5332	0,2763	0,1403	0,0710	0,0181
Hopscotch	0,8594	0,6673	0,4199	0,2578	0,1568	0,0574	0,7737	0,5332	0,2762	0,1403	0,0710	0,0181
ADI	0,8594	0,6673	0,4200	0,2579	0,1568	0,0574	0,7737	0,5332	0,2763	0,1403	0,0710	0,0181
Monte Carlo	0,8594	0,6670	0,4202	0,2569	0,1556	0,0570	0,7739	0,5333	0,2752	0,1399	0,0704	0,0179
Antithetic	0,8596	0,6670	0,4199	0,2573	0,1564	0,0570	0,7733	0,5322	0,2753	0,1397	0,0706	0,0179

	USA						Arbitrary					
	τ						τ					
	2	5	10	15	20	30	2	5	10	15	20	30
Runge	0,9756	0,9249	0,8250	0,7203	0,6196	0,4456	0,8687	0,7005	0,4724	0,3184	0,2239	0,1237
Euler	0,9756	0,9249	0,8250	0,7203	0,6196	0,4456	0,8687	0,7005	0,4724	0,3184	0,2239	0,1237
Taylor	0,9756	0,9249	0,8250	0,7203	0,6196	0,4456	0,8687	0,7005	0,4724	0,3184	0,2239	0,1237
Implicit (ODE)	0,9756	0,9249	0,8250	0,7203	0,6196	0,4456	0,8687	0,7005	0,4724	0,3184	0,2239	0,1237
CN (ODE)	0,9756	0,9249	0,8250	0,7203	0,6196	0,4456	0,8687	0,7005	0,4724	0,3184	0,2239	0,1237
Explicit	0,9754	0,9245	0,8243	0,7197	0,6192	0,4459	0,8688	0,7006	0,4719	0,3180	0,2239	0,1238
Implicit	0,9754	0,9245	0,8243	0,7197	0,6193	0,4460	0,8689	0,7006	0,4722	0,3183	0,2240	0,1239
CN	0,9754	0,9245	0,8243	0,7197	0,6193	0,4460	0,8688	0,7006	0,4720	0,3181	0,2240	0,1238
Hopscotch	0,9754	0,9245	0,8243	0,7197	0,6192	0,4459	0,8688	0,7006	0,4720	0,3181	0,2240	0,1238
ADI	0,9754	0,9245	0,8243	0,7197	0,6193	0,4460	0,8689	0,7006	0,4720	0,3181	0,2240	0,1238
Monte Carlo	0,9756	0,9248	0,8248	0,7199	0,6211	0,4462	0,8685	0,6991	0,4682	0,3066	0,2031	0,0989
Antithetic	0,9756	0,9249	0,8251	0,7202	0,6190	0,4457	0,8685	0,6994	0,4671	0,3062	0,2063	0,0992

[1] i. Grid size = 1600; ii. 25 000 simulations for Monte Carlo; iii. 5000 simulations for antithetic variates due to its lower variance; iv. $\Delta_t = 0{,}005$ for simulation methods to achieve some degree of accuracy; v. Δ_t for the FD methods is chosen such that using ½Δ_t decreased average e_i by less than 0,01. Specifically, Δ_t equals: 0,01 for Hopscotch and implicit, 0,005 for explicit, 0,1 for ADI and 0,25 for CN; and vi. Δ_τ for the ODE methods is chosen as the maximum for which RMS = 0.

	Check					
	τ					
	2	5	10	15	20	30
Runge	0,8619	0,6988	0,5120	0,3855	0,2930	0,1705
Euler	0,8619	0,6988	0,5120	0,3855	0,2930	0,1705
Taylor	0,8619	0,6988	0,5120	0,3855	0,2930	0,1705
Implicit (ODE)	0,8619	0,6988	0,5120	0,3855	0,2930	0,1705
CN (ODE)	0,8619	0,6988	0,5120	0,3855	0,2930	0,1705
Explicit	0,8620	0,6988	0,5117	0,3850	0,2925	0,1702
Implicit	0,8621	0,6990	0,5119	0,3851	0,2927	0,1703
CN	0,8620	0,6988	0,5117	0,3850	0,2926	0,1702
Hopscotch	0,8620	0,6989	0,5117	0,3850	0,2926	0,1702
ADI	0,8620	0,6989	0,5117	0,3850	0,2926	0,1702
Monte Carlo	0,8617	0,6960	0,5022	0,3714	0,2763	0,1535
Antithetic	0,8618	0,6957	0,5020	0,3724	0,2766	0,1539

5.1.3 The FD methods have higher levels of accuracy and quicker computational times than the simulation methods. The most efficient FD method is a minimum of 49,7 times more computationally efficient than the most efficient simulation method, while having a 94,8 per cent lower MAE across the five parameter sets and 50,8 per cent lower MAE on average. The FD methods have greater consistency in accuracy across the five parameter sets. These results are consistent with the comparison between FD methods and simulation methods found in Boyle (1977) who proposes that simulation methods are computationally inefficient. These results are also consistent with those in Wilmott (2006) in which it is argued that for models containing less than four random factors, finite difference schemes are more efficient than simulation methods.

5.1.4 The Euler, CN, Taylor and Runge–Kutta ODE methods as well as the ADI and CN finite difference schemes (selected due to their apparent computational efficiency) are analysed further using fifty additional randomly selected sets of parameters as previously described. The accuracy and computational efficiency of these methods when applied to the randomly selected sets of parameters is summarised and compared in Table 4, with the full results available in Appendix C.

5.1.4 The relative ranking of theses numerical methods based on efficiency and accuracy does not differ between the parameter sets in Table 1 and the random parameter sets. The conclusions drawn above based on the results for the sets of parameters in Table 1 hold when considering the random selected sets of parameters. In particular, the ODE methods remain more efficient and accurate across the random parameter sets. In Table 4 it can be seen that the least efficient of the four ODE methods shown is more efficient than the most efficient FD method on average, while the most efficient ODE method is more than 400 times more efficient on average. Furthermore, the least accurate ODE method has a 98 per cent lower MAE than the most accurate FD method on average. An improvement in the accuracy of the FD methods causes a significant worsening of their already low relative efficiency.

TABLE 3. RMS, MAE and total computation time (in seconds) taken over the six integer times to maturity for each numerical method used

	SA			Brazil			USA			Arbitrary			Check		
	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time
Runge	0,0000	0,0000	0,00	0,0000	0,0000	0,00	0,0000	0,0000	0,00	0,0000	0,0000	0,00	0,0000	0,0000	0,00
Euler	0,0000	0,0000	0,14	0,0000	0,0000	0,06	0,0000	0,0000	0,06	0,0000	0,0000	0,08	0,0000	0,0000	0,05
Taylor	0,0000	0,0000	0,22	0,0000	0,0000	2,56	0,0000	0,0000	0,05	0,0000	0,0000	0,47	0,0000	0,0000	0,01
Implicit (ODE)	0,0000	0,0000	4,30	0,0000	0,0000	2,17	0,0000	0,0000	1,96	0,0000	0,0000	2,31	0,0000	0,0000	0,89
CN (ODE)	0,0000	0,0000	0,01	0,0000	0,0000	0,03	0,0000	0,0000	0,02	0,0000	0,0000	0,06	0,0000	0,0000	0,02
Explicit	0,0013	0,0001	632,75	0,0011	0,0002	634,12	0,0006	0,0004	631,82	0,0007	0,0002	632,12	0,0011	0,0003	631,14
Implicit	0,0016	0,0001	6611,59	0,0012	0,0001	6609,81	0,0007	0,0004	6610,11	0,0009	0,0002	6608,85	0,0008	0,0002	6608,03
CN	0,0010	0,0001	273,06	0,0006	0,0001	271,98	0,0006	0,0004	272,64	0,0007	0,0002	271,79	0,0010	0,0003	271,37
Hopscotch	0,0012	0,0001	344,58	0,0008	0,0001	343,55	0,0006	0,0004	346,07	0,0007	0,0002	344,22	0,0010	0,0003	343,33
ADI	0,0010	0,0001	63,19	0,0004	0,0001	63,02	0,0006	0,0004	63,25	0,0007	0,0002	62,86	0,0010	0,0003	62,75
Monte Carlo	0,0054	0,0006	7618,61	0,0065	0,0005	7614,81	0,0011	0,0005	7616,34	0,0914	0,0105	7614,88	0,0498	0,0101	7616,23
Antithetic	0,0044	0,0004	3139,13	0,0059	0,0006	3132,61	0,0004	0,0002	3128,77	0,0884	0,0101	3114,52	0,0485	0,0099	3125,64

TABLE 4. Summary of accuracy and efficiency for 50 random parameter sets for selected numerical methods. Accuracy is measured using RMS and MAE and efficiency (labelled Time) is measured using total calculation time in seconds, calculated over 30 integer times to maturity $\tau = 1, \ldots, 30$.[2]

Method	Measure	Mean	Standard deviation	Min	25th percentile	Median	75th percentile	Max
Runge–Kutta	RMS	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0.0005
	MAE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Time	0.0297	0.0066	0.0200	0.0300	0.0300	0.0300	0.0600
Euler	RMS	0.0005	0.0014	0.0000	0.0000	0.0001	0.0002	0.0074
	MAE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Time	1.3543	0.9049	0.0700	0.5280	1.3045	2.0235	3.3120
Taylor	RMS	0.0004	0.0011	0.0000	0.0000	0.0001	0.0002	0.0054
	MAE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Time	3.3387	3.2052	0.1300	1.1595	2.2400	4.7438	14.4500
CN ODE	RMS	0.0004	0.0011	0.0000	0.0000	0.0001	0.0002	0.0048
	MAE	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Time	0.2703	0.1447	0.0800	0.1613	0.2160	0.3310	0.8000
ADI	RMS	0.0205	0.0528	0.0003	0.0012	0.0032	0.0142	0.3284
	MAE	0.0007	0.0005	0.0001	0.0003	0.0006	0.0008	0.0027
	Time	12.9710	1.7857	12.6260	12.6973	12.7220	12.7473	25.4690
CN (FD method)	RMS	0.0174	0.0460	0.0002	0.0013	0.0034	0.0124	0.2832
	MAE	0.0006	0.0004	0.0001	0.0003	0.0005	0.0008	0.0022
	Time	77.9153	23.7836	67.9311	68.1659	68.2941	68.5509	137.2282

5.2 ODE methods

5.2.1 As shown in Tables 3 and 4, the Runge–Kutta method is the most computationally efficient method for all sets of parameters considered. Table 5 further considers the relative efficiency of the ODE methods when requiring a greater degree of accuracy.

TABLE 5. Computation time taken in order to obtain estimated bond prices with e_i<0,00005 for five sets of parameters at selected times to maturity

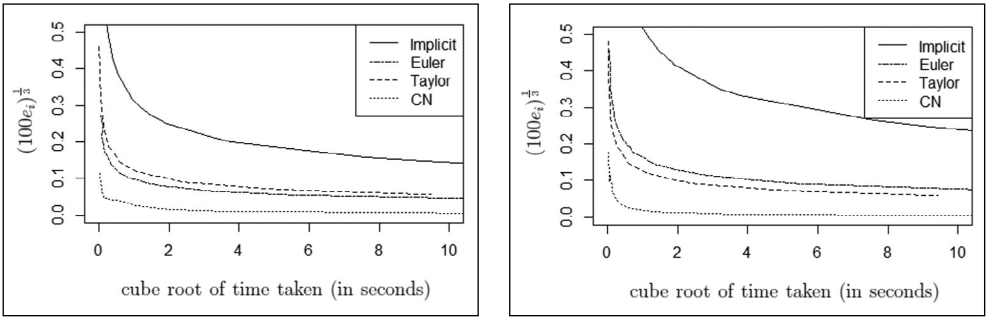
	SA (τ=30)	Brazil (τ=5)	USA (τ=30)	Arbitrary (τ=30)	Check (τ=10)	Cumulative sum
Runge–Kutta	0,00	0,00	0,00	0,00	0,00	0,01
Euler/Explicit	0,22	0,39	0,93	0,12	0,15	1,81
Taylor	0,43	7,69	0,45	0,95	0,01	9,53
Implicit	7,35	12,17	30	3,74	5,00	58,26
CN	0,01	0,01	0,03	0,08	0,05	0,18

[2] i. Grid size = 1600 for the two FD methods; ii. Δ_t for the two FD methods is the largest value for which a reduction in Δ_t by a factor of two results in a reduction in e_i by less than 0,01; and iii. Δ_τ for the ODE methods is the maximum value for which e_i < 0,001.

5.2.2 The Runge–Kutta method estimates bond prices accurately (measured to six decimal places) in a computation time of less than 0,01 seconds for all $1 \leq \tau \leq 30$ for all parameter sets in Table 1, on average 10 times quicker than the second most efficient method (determined separately for each parameter set). The Runge–Kutta method further estimates bond prices accurately ($e_i < 0{,}001$) for all times $1 \leq \tau \leq 30$ for 50 random parameter sets within 1.5 seconds. The other four ODE methods are compared in Figure 1 in addition to the results above by considering the rates of convergence to the true bond price (Runga–Kutta is excluded due to its efficiency making graphical comparison less relevant).

5.2.3 For each of the five parameter sets in Table 1, the relative efficiencies of the methods are consistent whether computation time is based on the time required to achieve an RMS of 0 or MAE of less than 0,0000005 (Table 3 and Table 5 respectively). The CN method converges quicker than the Euler method which in turn converges quicker than the implicit method (for all five parameter sets). The Euler method requires between two and six per cent of the computation time required by the implicit method. The efficiency of the CN method relative to the Euler method is less consistent, with the CN method requiring between two and 75 per cent of the computation time required by the Euler method. The efficiency of the Taylor method depends on the particular set of parameters. The Taylor method appears to be more efficient (relative to the efficiency of the other methods) if λ_{12} and λ_{21} are close to 0, This is the case for the USA and check parameter sets, where the Taylor method is more efficient than the Euler method. On average, the CN method converges the second quickest of the ODE methods, although the Taylor method converges marginally quicker for the check parameter set. The efficiency of the CN method is due to the target level of accuracy being obtained using a larger Δ_τ than the explicit and Taylor methods, which appears to offset the complexity of needing to solve two simultaneous equations.

5.2.4 Similar results are obtained for the random parameter sets (the implicit method is excluded). The Runge–Kutta method is on average 8,9 times more efficient than the second most efficient method (determined independently for each parameter set). The



South Africa (τ=30) USA (τ=30)

FIGURE 1. Rates of convergence of the ODE methods (excluding Runge–Kutta), represented as the cube root of time taken (in seconds) plotted against $(100e_i)^{\frac{1}{3}}$, for two sets of parameters at selected times to maturity

CN method is the second most efficient method for 84 per cent of the parameter sets, and is more efficient than the Euler method for 88 per cent of parameter sets. The CN method, on average, is 5,1 times more efficient than the Euler method and 17,8 times more efficient than the Taylor method. The relative ranking of the Taylor and Euler methods is dependent on the parameter set. On average, however, the Euler method is more efficient than the Taylor method, being 6,9 times more efficient on average and more efficient for 72 per cent of the parameter sets.

5.3 Finite difference scheme methods

5.3.1 Finite difference scheme methods can be adapted to non-affine models of the short rate, unlike the ODE methods which rely on the affine structure of the model considered in this paper (Shreve, 2004). We do not suggest the results in this paper will hold for non-affine models without having tested this assertion. We merely propose that the results may be of interest in spite of having previously shown the superior efficiency of the ODE methods. We further consider the results in comparison to the previous studies mentioned in ¶3.3.

5.3.2 The accuracy and efficiency of the five FD methods depends on both the grid size and Δ_t. An understanding of the accuracy of the FD methods is first considered in Table 6, after which the efficiency to achieve similar levels of accuracy is considered. Table 6 represents how the accuracy of these methods changes for selected sets of parameters with an increase in the size of the grid and reduction in the size of the time step used Δ_t.

5.3.3 As can be seen in Table 6, assuming the same grid size is used that is not too small, the five FD methods converge to the same estimated bond price as Δ_t is reduced (referred to as the limited estimate in this paper), but the rate of convergence differs between the methods. The implicit and explicit methods converge to the limited estimate from opposite directions. The size of the grid determines the level of accuracy attainable with a reduction in Δ_t (i.e. the accuracy of the limited estimate). An increase in the size of the grid increases the accuracy of the limited estimate.

5.3.4 Given the five FD methods converge to the same estimated bond price for the same size of grid, the efficiency of the FD methods can be compared for a defined grid size. This is done in the following paragraphs. Although the relationship between grid size and accuracy is an important consideration in overall efficiency of the FD methods, it is not a core result of this paper apart from what is discussed in ¶5.3.10 and as such is explored in more detail in Appendix D.

5.3.5 Table 7 compares the efficiency of the FD methods for a defined grid size by comparing the computational time required to achieve a given level of accuracy through a reduction in Δ_t. The Δ_t required to achieve the defined level of accuracy is also shown.

5.3.6 The results in Table 7 are analysed in the paragraphs below, but the distinction between implicit$_1$ and implicit$_2$ is first considered as this impacts the results. For all parameter sets and times to maturity shown in Table 7, the implicit method has an estimated bond price that decreases towards the true bond price with a reduction in Δ_t but then away from the true bond price towards the limited estimate. The results labelled implicit$_1$ are for Δ_t being selected as the maximum value for which e_i is less than 0,0025 greater than that for the limited estimate. The results labelled implicit$_2$ are for Δ_t being selected as the maximum value for which the difference between the estimated bond price and the limited estimate is less than 0,0001. Implicit$_2$ is preferred for comparison as the limited estimate is the estimated price to which the FD methods converge, and as such considered the base value around which comparisons of efficiency of the FD methods should be made. Comparisons of the implicit method in the following paragraphs refer to implicit$_2$.

TABLE 6. Error, represented as $100\left(\hat{P}(t,T)-P(t,T)\right)$, of the FD schemes for the selected parameter sets and τ for varying grid sizes and values of Δ_t

		South Africa (τ=30)			Arbitrary (τ=2)			Check (τ=10)		
Method	Δ_t	Grid size			Grid size			Grid size		
		100	400	1 600	100	400	1 600	100	400	1 600
Explicit	0,100	–0.20	–0.15	–5.76	–18.17	–1.90	–47.24	0.08	–0.17	–0.33
	0,050	–0.16	–0.10	–5.69	–12.82	–0.16	–47.24	0.13	–0.11	–0.08
	0,010	–0.12	–0.07	–0.02	4.21	–0.10	–0.05	0.18	–0.06	–0.04
	0,005	–0.12	–0.06	–0.02	7.00	–0.09	–0.05	0.18	–0.06	–0.03
	0,001	–0.11	–0.06	–0.01	9.22	–0.08	–0.04	0.19	–0.05	–0.03
Implicit	0,500	0.33	0.38	0.42	–41.98	0.69	0.72	0.75	0.52	0.54
	0,250	0.11	0.16	0.21	–21.19	0.31	0.35	0.47	0.23	0.26
	0,100	–0.02	0.03	0.07	–6.49	0.07	0.12	0.30	0.06	0.09
	0,010	–0.10	–0.05	0.00	8.07	–0.07	–0.02	0.20	–0.04	–0.02
	0,005	–0.11	–0.05	–0.01	8.93	–0.08	–0.03	0.19	–0.05	–0.02
CN	1,000	–0.13	–0.07	–0.03	–47.24	–0.08	–0.04	0.17	–0.08	–0.05
	0,500	–0.11	–0.06	–0.02	–32.61	–0.08	–0.04	0.18	–0.06	–0.03
	0,250	–0.11	–0.06	–0.01	–25.57	–0.08	–0.04	0.19	–0.05	–0.03
	0,100	–0.11	–0.06	–0.01	–22.84	–0.08	–0.04	0.19	–0.05	–0.03
	0,050	–0.11	–0.06	–0.01	–22.09	–0.08	–0.04	0.19	–0.05	–0.03
Hopscotch	0,500	–0.48	–1.45	–5.76	0.14	–3.07	–5.84	–0.22	–0.73	–3.15
	0,250	–0.20	–0.40	–1.80	–16.59	–0.95	–2.99	0.09	–0.22	–0.69
	0,100	–0.13	–0.11	–0.30	–19.49	–0.22	–0.56	0.17	–0.08	–0.13
	0,010	–0.11	–0.06	–0.02	3.83	–0.08	–0.04	0.19	–0.05	–0.03
	0,005	–0.11	–0.06	–0.01	6.83	–0.08	–0.04	0.19	–0.05	–0.03
ADI	1,000	–0.12	–0.07	–5.76	–47.24	–0.23	–1.15	0.18	–0.06	–0.03
	0,500	–0.11	–0.06	–5.31	–23.88	–0.09	–0.04	0.19	–0.05	–0.03
	0,250	–0.11	–0.06	–0.01	5.39	–0.08	–0.04	0.19	–0.05	–0.03
	0,100	–0.11	–0.06	–0.01	12.59	–0.08	–0.04	0.19	–0.05	–0.03
	0,050	–0.11	–0.06	–0.01	11.81	–0.08	–0.04	0.19	–0.05	–0.03

TABLE 7. Computational times and accuracy for finite difference methods for a selection of times to maturity for the five sets of parameters[3]

Method	Measure	SA ($\tau=30$)	Brazil ($\tau=5$)	USA ($\tau=15$)	Arbitrary ($\tau=10$)	Check ($\tau=10$)	Sum of time	RMS	MAE
Implicit$_1$	Δ_t	0,025	0,01	0,5	0,025	0,025			
	Price	0,0576	0,5333	0,7203	0,4724	0,5120			
	Time	2666,50	36,32	66,67	885,00	890,57	4545,06	0,0007	0,0000
Implicit$_2$	Δ_t	0,025	0,005	0,05	0,005	0,005			
	Price	0,0576	0,5332	0,7198	0,4721	0,5118			
	Time	2667,60	68,23	663,27	4425,53	4427,00	12251,63	0,0008	0,0002
Explicit	Δ_t	0,0025	0,0025	0,01	0,001	0,001			
	Price	0,0574	0,5331	0,7197	0,4720	0,5117			
	Time	1247,37	51,14	155,85	1038,52	1044,55	3537,43	0,0013	0,0003
Hopscotch	Δ_t	0,005	0,01	0,025	0,005	0,01			
	Price	0,0574	0,5331	0,7197	0,4720	0,5117			
	Time	725,28	14,14	68,24	227,88	116,46	1151,98	0,0012	0,0003
CN	Δ_t	0,25	0,25	1	1	0,25			
	Price	0,0574	0,5331	0,7198	0,4720	0,5117			
	Time	281,84	1,70	34,13	22,96	91,55	432,17	0,0012	0,0003
ADI	Δ_t	0,25	1	1	0,25	0,5			
	Price	0,0574	0,5334	0,7198	0,4720	0,5117			
	Time	26,45	0,27	3,18	8,42	4,20	42,51	0,0012	0,0003
True Bond Price		0,0576	0,5333	0,7203	0,4724	0,5120			

5.3.7 Table 7 shows that the ADI method is the most computationally efficient of the FD methods across all five parameter sets, followed by the CN method. The Hopscotch method is more efficient than the explicit method. The implicit method is the least efficient method on average, with its theoretical stability not seeming to translate into improved computational efficiency. The time step required to obtain a given level of accuracy (relative to the limited estimate) is the largest for the CN and ADI methods, significantly larger than that required by the other three methods (based on the Δ_t values used for Table 7). This offsets the complexity of these two methods (although the ADI method is significantly less complex), resulting in their efficiency.

5.3.8 Based on the computation times in Table 7, the explicit method requires on average 38 per cent of the computation time required by the implicit method, while the

[3] i. Brazil parameter: grid size = 400; ii. Other parameter sets: grid size = 1600; iii. The Δ_t value is chosen from: 1; 0,5; 0,25; 0,1; 0,05; 0,025; 0,01; 0,005; 0,0025; 0,001; iv. For the four methods other than the implicit method, the Δ_t value used is the maximum for which e_i is less than 0,0025 greater than that for the limited estimate.

Hopscotch method requires on average 33 per cent of the computation time required by the explicit method. The ADI method requires on average 3.5 per cent of the computation time required by the Hopscotch method and 15 per cent of that required by the CN method. For the random parameter sets (Table 4), the ADI method is six times more efficient on average than the CN method with similar levels of accuracy.

5.3.9 As detailed in ¶3.3, Geske & Shastri (1985) showed the explicit method to take roughly 60 per cent of the computation time taken by the implicit method, while Hull & White (1990) argue that the explicit scheme uses between 40 and 70 per cent of the computation time of the implicit method. The results in this paper similarly show the explicit method to be computationally more efficient. The results in Table 7 show the explicit method requiring between 25 and 75 per cent of the computation time of the implicit method. Previous results when using these methods (see eg Cairns (2004)) further showed that the CN method converges more quickly than the explicit and implicit method, which is the result of this paper.

5.3.10 The analysis in ¶¶5.3.7 and 5.3.8 has shown the ADI method to be the most efficient of the FD methods. The efficiency of this method is considered in more detail in Table 8 to draw a comparison relative to the efficiency of the Runga–Kutta method (the most efficient ODE method). The efficiency is considered for the Brazil parameter set as the smallest grid size was required to achieve a given degree of accuracy for this set of parameters.

TABLE 8. True bond prices and estimated bond prices using the ADI method for the Brazil parameter set at selected times to maturity[4]

	$\tau=2$	$\tau=5$	$\tau=10$	$\tau=15$	$\tau=20$	$\tau=30$	MAE	RMS	Time
True price	0,7735	0,5333	0,2765	0,1404	0,0710	0,0181			
Estimated price$_1$	0,7736	0,5333	0,2764	0,1404	0,0710	0,0181	0,0000	0,0002	432,93
Estimated price$_2$	0,7735	0,5333	0,2764	0,140	0,0710	0,0181	0,0000	0,0001	3857,21

5.3.11 The ADI method is unable to obtain an RMS of 0 within one hour of computation time. This is in contrast to the Runge–Kutta method which obtains an RMS of 0 within 0.01 seconds of computation time. Given the ADI is the most efficient of the FD methods, it can be concluded that the FD methods are unable to obtain the accuracy of the ODE methods (in this case measured as an RMS of 0) within a reasonable computation time.

5.4 Simulation methods

5.4.1 This section considers the simulation methods in more detail due to their flexibility. In particular consideration is given to the impact of the number of simulations and the size of the Δ_t on the accuracy and efficiency of the simulation methods. As mentioned

[4] i. Estimated price$_1$: a grid size of 10 000 is used; ii. Estimated price$_2$: a grid size of 40 000 is used; iii. Δ_t is selected as the maximum value (from 0.25; 0.1; 0.05; 0.025; 0.01) such that the RMS is the same as the limited estimate; iv. Estimated price$_1$: $\Delta_t = 0.1$; and v. Estimated price$_2$: $\Delta_t = 0.05$

previously, these impact the two causes of error in the estimated price $\hat{P}(t,T)$ for the simulation methods. The number of simulations will determine the variation in the sample estimate of the expectation. Δ_t will determine the accuracy of the estimate as the discretisation error of the short rate and the error in estimating an integral with a discrete sum reduces as $\Delta_t \rightarrow 0$. The variation in the sample estimate as measured with standard deviation is analysed first. This is done by considering the change in the standard deviation of the estimates with a change in the number of simulations for a fixed Δ_t, enabling a comparison of the relative efficiency of the two simulation methods. The impact of Δ_t on the accuracy of the simulation methods is then considered in ¶¶ 5.4.5 and 5.4.6.

5.4.2 Table 9 below compares the reduction in the observed and theoretical standard deviations of the estimated bond price with an increase in the number of simulations the South African parameter set.

TABLE 9. Observed and theoretical standard deviations of the sample estimate of the bond price assuming $\tau = 10$ for the SA parameter set using a varying number of iterations[5]

Method	Measure	500	1000	2500	5000	10000	25000
Monte Carlo	Sample σ	0,00576	0,00397	0,00262	0,00185	0,00113	0,00089
	Theoretical σ	0,00563	0,00398	0,00252	0,00180	0,00126	0,00080
	Difference	0,00013	–0,00001	0,00010	0,00005	–0,00013	0,00009
Antithetic variates	Sample σ	0,00189	0,00134	0,00084	0,00057	0,00042	0,00028
	Theoretical σ	0,00185	0,00131	0,00083	0,00059	0,00041	0,00026
	Difference	0,00004	0,00003	0,00001	–0,00002	0,00001	0,00002

5.4.3 Similar results to those in Table 9 were found when considering the Brazil (setting $\Delta_t = 10$) and Check (setting $\Delta_t = 5$) parameter sets, with the standard deviation of the estimated bond price being between 40 and 70 per cent lower using the antithetic variates method as opposed to Monte Carlo simulation, assuming the same Δ_t and number of simulations. Furthermore, the observed sample standard deviation does not differ significantly from the theoretical standard deviation of the estimated bond price in any of the parameter sets considered. The theoretical standard deviation is a quick measure of the number of simulations required to reduce the sample variance to within an acceptable limit, and highlights the inefficiency if very low standard deviation is required.

5.4.4 To achieve a variance reduction of 40 per cent in Monte Carlo simulation, the number of simulations and therefore the computation time must be increased by a factor of $\frac{1}{1-0.4}^{\,2} = 2.78$. The antithetic variates method increases the computation time of

[5] The theoretical standard deviations are calculated by reducing the sample standard deviation obtained for 250 simulations in proportion to the change in $\frac{1}{\sqrt{N}}$.Δ_t of 0,01 is used.

that required by Monte Carlo simulation by a factor of 2. It, therefore, has an estimated 28 per cent lower computation time (based on assuming that it results in a 40 per cent lower variance than Monte Carlo simulation). Similarly, if the antithetic variates method results in a 70 per cent lower variance than Monte Carlo simulation, it will have an estimated 82 per cent lower computation time. It, therefore, has a computation time of between 28 and 80 per cent lower than that of Monte Carlo simulation in order to obtain the same level of variance in the estimated bond price. This is in spite of the antithetic variates method requiring approximately double the computation time of Monte Carlo simulation for the same Δt and number of simulations. This is as a result of the lower variance offsetting the doubling of the computation time.

5.4.5 The above has analysed the effect of increasing the number of simulations on the standard deviation of the estimate. For a given number of simulations, the accuracy of the estimates will depend on Δ_t. This is considered in Table 10.

TABLE 10. Error e_i of the simulation methods at different time steps Δ_t[6]

	South Africa (τ=5)			Brazil (τ=10)				Check (τ=5)			
Time step	0,50	0,10	0,01	0,50	0,10	0,01	0,005	0,50	0,10	0,01	0,005
Monte Carlo	0,35	0,17	0,01	1,53	0,42	0,08	0,07	1,64	0,78	0,23	0,18
Antithetic	0,38	0,10	0,01	1,57	0,46	0,13	0,09	1,71	0,55	0,40	0,33

5.4.6 The error decreases as the time step decreases. The level of accuracy, however, depends on the set of parameters considered. This is similar to the results in Table 2, where the average accuracy (using MAE) across the South African, Brazil and USA parameter sets is more than 10 times higher than that across the check and arbitrary parameter sets using the same Δ_t and number of simulations.

6. CONCLUSIONS

6.1 Numerical methods used to estimate the price of zero-coupon bonds in a two-factor CIR model were compared on the basis of accuracy and computational efficiency. Five ODE methods, five FD methods and two simulation methods were considered. The ODE methods were shown to be significantly more efficient, consistent and accurate than the FD methods, which in turn were more efficient, consistent and accurate than the simulation methods. The ODE methods can only be used in affine short-rate models, while the FD methods can be applied to many non-affine short-rate models (Shreve, 2004).

6.2 The Runge–Kutta method is the most preferred of the numerical methods considered, having the quickest computation time and highest accuracy for all parameters and times to maturity considered. The Runge–Kutta method is, on average, between nine and ten times

[6] Based on 25 000 simulations for Monte Carlo and 5 000 for antithetic variates

more computationally efficient than the second most efficient method. This paper would suggest that the lower theoretical truncation error of the Runge–Kutta method translates into improved efficiency. The Crank–Nicolson ODE method is the second most computationally efficient method, while the relative efficiency of the Taylor and Euler methods is dependent on the parameter set considered. The implicit method is the least efficient of the ODE methods, in spite of its theoretical stability.

6.3 The FD methods are unable to obtain the same level of accuracy as the ODE methods in a reasonable computation time. This is the result of the accuracy of the FD methods being largely constrained by the size of grid used. The ADI method is the most efficient FD method and was on average six times more efficient than the Crank–Nicolson method, the second most efficient FD method. The use of the antithetic variates methods improved the efficiency of Monte Carlo simulation, but these two methods were the most inefficient.

6.4 The numerical methods used in this paper are some of the most commonly used numerical methods in mathematics of finance. In mathematics of finance derivative pricing, a bigger proportion of derivatives are over the counter traded and their prices sometimes may not be obtained in closed-form due to their exotic nature. Our paper contributes to the literature in terms of providing evidence that the chosen numerical methods, though applied to a particular bond pricing problem, could be used in cases where closed-form solutions either do not exist or are difficult to obtain using classical methods. We demonstrated the advantages of each method based on efficiency and accuracy. We acknowledge that recent numerical methods, eg Fast Fourier transform (FFT), Cosine method (COS) etc, require knowledge of the distribution or characteristic function of the underlying, which is one of their weaknesses.

REFERENCES

Alexander, R (1977). Diagonally implicit Runge–Kutta methods for stiff ODEs. *SIAM Journal on Numerical Analysis*, **14**(6), 1006–21

Bashforth, F & Adams, JC (1883). An attempt to test the theories of capillary action: by comparing the theoretical and measured forms of drops of fluid. With an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops. University Press.

Boyle, PP (1977). Options: A Monte Carlo approach. *Journal of Financial Economics*, **4**(3), 323–38

Boyle, P, Broadie, M, & Glasserman, P (1997). Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, **21**(8), 1267–321

Brennan, MJ & Schwartz, ES (1979). A continuous time approach to the pricing of bonds. *Journal of Banking & Finance*, **3**(2), 133–55

Broadie, M & Detemple, J (1996). American option valuation: new bounds, approximations, and a comparison of existing methods. *Review of Financial Studies*, **9**(4), 1211–50

Bulirsch, R & Stoer, J (1966). Numerical treatment of ordinary differential equations by extrapolation methods. *Numerische Mathematik*, **8**(1), 1–13

Cairns, AJ (2004). *Interest rate models: an introduction*, volume 10. Princeton University Press

Chen, R-R & Scott, L (2003). Multi-factor Cox–Ingersoll–Ross models of the term structure: Estimates and tests from a Kalman filter model. *The Journal of Real Estate Finance and Economics*, **27**(2), 143–72

Cox, JC, Ingersoll, JE, & Ross, SA (1985). A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, 385–407

Duffie, D & Kan, R (1996). A yield-factor model of interest rates. *Mathematical Finance*, **6**(4), 379–406

Geske, R & Shastri, K (1985). Valuation by approximation: a comparison of alternative option valuation techniques. *Journal of Financial and Quantitative Analysis*, **20**(01), 45–71

Granville, S (1988). *The numerical solution of ordinary and partial differential equations*. Academic Press, New York and London.

Hull, JC & White, AD (1990). Valuing derivative securities using the explicit finite difference method. *Journal of Financial and Quantitative Analysis*, **25**(01), 87–100

Jamshidian, F & Zhu, Y (1996). Scenario simulation: Theory and methodology. *Finance and Stochastics*, **1**(1), 43–67

Joy, C, Boyle, PP, & Tan, KS (1996). Quasi-Monte Carlo methods in numerical finance. *Management Science*, **42**(6), 926–38

Litterman, RB & Scheinkman, J (1991). Common factors affecting bond returns. *The Journal of Fixed Income*, **1**(1), 54–61

Pearson, ND & Sun, T (1994). Exploiting the conditional density in estimating the term structure: An application to the Cox, Ingersoll, and Ross model. *The Journal of Finance*, **49**(4), 1279–304

Shampine, LF (1994). *Numerical solution of ordinary differential equations*, volume 4. CRC Press

Shreve, SE (2004). *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer Science & Business Media

Stambaugh, RF (1988). The information in forward rates: Implications for models of the term structure. *Journal of Financial Economics*, 21(1), 41–70

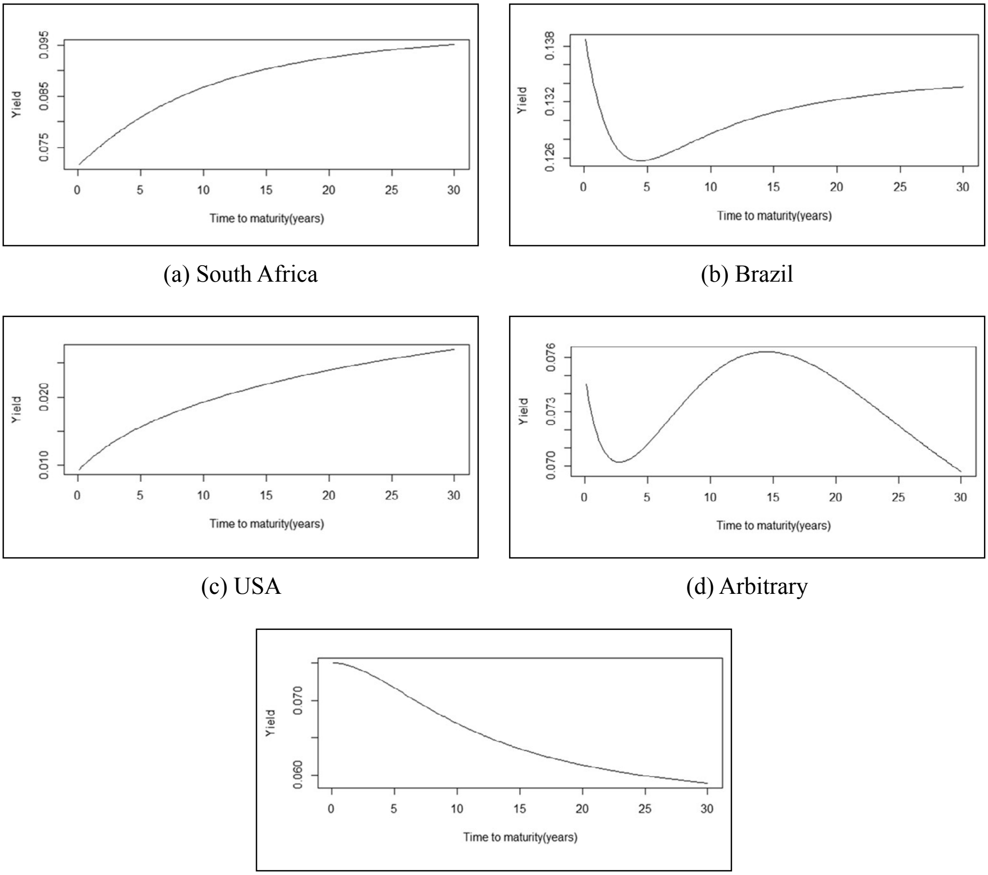
Vasicek, O (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics*, **5**(2), 177–88

Wilmott, P (2006). *Paul Wilmott on quantitative finance*, volume 3. John Wiley & Sons, second edition

WolframAlpha (2016). https://www.wolframalpha.com/input/?i=y% 27+-+0.025+% 2B0.1y+% 2B0.5*y% 5E2+% 3D+0;+y(0)% 3D0. Last visited on 18/8/2016

APPENDIX A
Yield curves

The following diagrams illustrate the shape of the yield curves arising from the five sets of parameters found in Table 1.



(a) South Africa

(b) Brazil

(c) USA

(d) Arbitrary

(e) Check

FIGURE 2. Yield curves for the five parameter sets

APPENDIX B
Further description of finite difference scheme methods

B1. Description of how eq. (6) is obtained

B1.1 The transformation in eq. (5), as presented by Brennan & Schwartz (1979: 152), results in the following:

$$\text{Let } \theta_1 = \frac{1}{1+k_1 y_1}\,;\ \Rightarrow 0 \le \theta_1 \le 1$$

$$\text{Let } \theta_2 = \frac{1}{1+k_2 y_2}\,;\ \Rightarrow 0 \le \theta_2 \le 1$$

$$\Rightarrow y_1 = \frac{1}{k_1}\left(\frac{1}{\theta_1} - 1\right)$$

$$\Rightarrow y_2 = \frac{1}{k_2}\left(\frac{1}{\theta_2} - 1\right)$$

B1.2 The partial derivatives of f with respect to Y_1 and Y_2 as functions of θ_1 and θ_2 are obtained as follows:

$$\frac{\partial f}{\partial y_1} = \frac{\partial f}{\partial \theta_1}\frac{\partial \theta_1}{\partial y_1} = -\frac{\partial f}{\partial \theta_1}\left(k_1\theta_1{}^2\right)$$

$$\frac{\partial f}{\partial y_2} = \frac{\partial f}{\partial \theta_2}\frac{\partial \theta_2}{\partial y_2} = -\frac{\partial f}{\partial \theta_2}\left(k_2\theta_2{}^2\right)$$

$$\frac{\partial^2 f}{\partial y_1^2} = \frac{\partial}{\partial y_1}\left(-\frac{\partial f}{\partial \theta_1}\left(k_1\theta_1^2\right)\right) = \frac{\partial}{\partial \theta_1}\left(\frac{\partial f}{\partial \theta_1}\left(k_1\theta_1{}^2\right)\right)\left(k_1\theta_1{}^2\right) = \frac{\partial^2 f}{\partial \theta_1^2}\left(k_1^2\theta_1{}^4\right) + 2\frac{\partial f}{\partial \theta_1}k_1^2\theta_1^3$$

$$\frac{\partial^2 f}{\partial y_2^2} = \frac{\partial}{\partial y_2}\left(-\frac{\partial f}{\partial \theta_2}\left(k_2\theta_2{}^2\right)\right) = \frac{\partial}{\partial \theta_2}\left(\frac{\partial f}{\partial \theta_2}\left(k_2\theta_2{}^2\right)\right)\left(k_2\theta_2{}^2\right) = \frac{\partial^2 f}{\partial \theta_2^2}\left(k_2^2\theta_2{}^4\right) + 2\frac{\partial f}{\partial \theta_2}k_2^2\theta_2^3$$

B1.3 It is now possible to specify eq. (2) in terms of θ_1 and θ_2 as follows:

$$\begin{aligned}&-[\delta_0 + \delta_1 y_1 + \delta_2 y_2]f + f_t - f_{\theta_1}k_1\theta_1^2[\mu_1 - \lambda_{11}y_1 - \lambda_{12}y_2] - \\ &f_{\theta_2}k_2\theta_2^2[\mu_2 - \lambda_{21}y_1 - \lambda_{22}y_2] + \frac{1}{2}f_{\theta_1\theta_1}y_1k_1^2\theta_1^4 + y_1k_1^2\theta_1^3 f_{\theta_1} + \frac{1}{2}f_{\theta_2\theta_2}y_2k_2^2\theta_2^4 + \\ &y_2k_2^2\theta_2^3 f_{\theta_2} = 0\end{aligned} \tag{9}$$

Eq. (6) is subsequently obtained using a_0, a_1, a_2, b_1 and b_2 for notational compactness.

B2. Initial conditions

B2.1 The initial conditions are found by substituting $y_1 = 0$ and/or $y_2 = 0$ ($\theta_1 = 1$ and/or $\theta_2 = 1$) into eq. (9), resulting in the second order partial derivative $f_{\theta_1\theta_1}$ and/or $f_{\theta_2\theta_2}$ terms falling away as their co-efficient becomes 0.

B2.2 If $\theta_1 = 1$, eq. (9) becomes:

$$-[\delta_0 + \delta_2 y_2]f + f_t - f_{\theta_1} k_1 [\mu_1 - \lambda_{12} y_2] - f_{\theta_2} k_2 \theta_2^2 [\mu_2 - \lambda_{22} y_2] + \frac{1}{2} f_{\theta_2\theta_2} y_2 k_2^2 \theta_2^4 + y_2 k_2^2 \theta_2^3 f_{\theta_2} = 0$$

B2.3 If $\theta_2 = 1$, eq. (9) becomes:

$$-[\delta_0 + \delta_1 y_1]f + f_t - f_{\theta_1} k_1 \theta_1^2 [\mu_1 - \lambda_{11} y_1] - f_{\theta_2} k_2 [\mu_2 - \lambda_{21} y_1] + \frac{1}{2} f_{\theta_1\theta_1} y_1 k_1^2 \theta_1^4 + y_1 k_1^2 \theta_1^3 f_{\theta_1} = 0$$

B2.4 If $\theta_1 = 1$ and $\theta_2 = 1$, eq. (9) becomes:

$$-[\delta_0]f + f_t - f_{\theta_1} k_1 [\mu_1] - f_{\theta_2} k_2 [\mu_2] = 0$$

B2.5 The boundary conditions as $y_1 \to \infty$ and/or $y_2 \to \infty$ ($\theta_1 \to 0$ and/or $\theta_2 \to 0$) are obtained based on the method used by Brennan & Schwartz (1979) as below.

B2.6 Firstly, specify eq. (9) as:

$$\begin{aligned}
&-\left[\delta_0 + \delta_1 \frac{1}{k_1}\left(\frac{1}{\theta_1} - 1\right) + \delta_2 \frac{1}{k_2}\left(\frac{1}{\theta_2} - 1\right)\right] f + f_t \\
&-f_{\theta_1} k_1 \theta_1^2 \left[\mu_1 - \lambda_{11} \frac{1}{k_1}\left(\frac{1}{\theta_1} - 1\right) - \lambda_{12} \frac{1}{k_2}\left(\frac{1}{\theta_2} - 1\right)\right] - \\
&f_{\theta_2} k_2 \theta_2^2 \left[\mu_2 - \lambda_{21} \frac{1}{k_1}\left(\frac{1}{\theta_1} - 1\right) - \lambda_{22} \frac{1}{k_2}\left(\frac{1}{\theta_2} - 1\right)\right] + \\
&\frac{1}{2} f_{\theta_1\theta_1} \frac{1}{k_1}\left(\frac{1}{\theta_1} - 1\right) k_1^2 \theta_1^4 + \frac{1}{k_1}\left(\frac{1}{\theta_1} - 1\right) k_1^2 \theta_1^3 f_{\theta_1} + \\
&\frac{1}{2} f_{\theta_2\theta_2} \frac{1}{k_2}\left(\frac{1}{\theta_2} - 1\right) k_2^2 \theta_2^4 + \frac{1}{k_2}\left(\frac{1}{\theta_2} - 1\right) k_2^2 \theta_2^3 f_{\theta_2} = 0
\end{aligned} \tag{10}$$

B2.7 To obtain the boundary condition when $\theta_1 = 0$ and $\theta_2 = 0$, multiply eq. (10) by $min\left(\frac{k_1\theta_1}{\delta_1}; \frac{k_2\theta_2}{\delta_2}\right)$ and take the limit as $\theta_1 \to 0; \theta_2 \to 0$. All terms fall away except for the coefficient of *f*. As such $U_{00}^t = 0$ for all $t < T$.

B2.8 To obtain the boundary condition when $\theta_1 = 0; \theta_2 > 0$, multiply eq. (10) by $k_1\theta_1$ and take the limit as $\theta_1 \to 0$. Eq. (10) reduces to $-\delta_1 f + \lambda_{21} f_{\theta_2} k_2 \theta_2^2 = 0$. Using the condition that $U_{00}^t = 0$, iteratively it is obtained that $U_{0j}^t = 0$ for all $t < T$. Similarly it can be shown that $U_{i0}^t = 0$ for all $t < T$.

B3. Notation

The following notation is used in the following sections.

$$c \; = a_0 - \frac{a_1}{\Delta_{\theta_1}} - \frac{a_2}{\Delta_{\theta_2}} + \frac{b_1}{\Delta_{\theta_1}{}^2} + \frac{b_2}{\Delta_{\theta_2}{}^2};$$

$$d_1 = -\frac{a_1}{\Delta_{\theta_1}} + \frac{b_1}{2\Delta_{\theta_1}{}^2};$$

$$d_2 = -\frac{a_2}{\Delta_{\theta_2}} + \frac{b_2}{2\Delta_{\theta_2}{}^2};$$

$$e_1 = \frac{b_1}{2\Delta_{\theta_1}{}^2};$$

$$e_2 = \frac{b_2}{2\Delta_{\theta_2}{}^2}.$$

B4. Approximations used in discretisation of eq. (6) for the FD methods

The following approximations are used in discretising eq. (6) to obtain eq. (7) and eq. (8). Backward differences are used for f_{θ_1}, f_{θ_2} and f_t while central second differences are used for $f_{\theta_2\theta_2}$ and $f_{\theta_1\theta_1}$.

$$f_{\theta_1}(t,\theta_1,\theta_2) \approx \frac{f(t,\theta_1,\theta_2) - f(t,\theta_1 - \Delta_{\theta_1},\theta_2)}{\Delta_{\theta_1}} = \frac{U_{\theta_1\theta_2}^t - U_{\theta_1-\Delta_{\theta_1}\theta_2}^t}{\Delta_{\theta_1}}$$

$$f_{\theta_2}(t,\theta_1,\theta_2) \approx \frac{f(t,\theta_1,\theta_2) - f(t,\theta_1,\theta_2 - \Delta_{\theta_2})}{\Delta_{\theta_2}} = \frac{U_{\theta_1\theta_2}^t - U_{\theta_1\theta_2-\Delta_{\theta_2}}^t}{\Delta_{\theta_2}}$$

$$f_t(t,\theta_1,\theta_2) \approx \frac{f(t+\Delta_t,\theta_1,\theta_2) - f(t,\theta_1,\theta_2)}{\Delta_t} = \frac{U_{\theta_1\theta_2}^{t+\Delta_t} - U_{\theta_1\theta_2}^t}{\Delta_t}$$

$$f_{\theta_1\theta_1}(t,\theta_1,\theta_2) \approx \frac{U_{\theta_1+\Delta_{\theta_1}\theta_2}^t - 2U_{\theta_1\theta_2}^t + U_{\theta_1-\Delta_{\theta_1}\theta_2}^t}{\Delta_{\theta_1}{}^2}$$

$$f_{\theta_2\theta_2}(t,\theta_1,\theta_2) \approx \frac{U_{\theta_1\theta_2+\Delta_{\theta_2}}^t - 2U_{\theta_1\theta_2}^t + U_{\theta_1\theta_2-\Delta_{\theta_2}}^t}{\Delta_{\theta_2}{}^2}$$

B5. Solving for U_{ij}^{t} in the explicit method

B5.1 The explicit method is developed from Cairns (2004:169). The value of U_{ij}^{t} for $0<i$, $j<n$ is found using eq. (7) as follows:

$$U_{ij}^{t}=U_{ij}^{t+\Delta_t}\left(1-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)j}^{t+\Delta_t}+e_2U_{i(j+1)}^{t+\Delta_t}+d_1U_{(i-1)j}^{t+\Delta_t}+d_2U_{i(j-1)}^{t+\Delta_t}\right].$$

B5.2 The value of U_{in}^{t} for $0<i<n$ is found using eq. (7) as follows:

$$U_{in}^{t}=U_{in}^{t+\Delta_t}\left(1-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)n}^{t+\Delta_t}+d_1U_{(i-1)n}^{t+\Delta_t}+d_2U_{i(n-1)}^{t+\Delta_t}\right].$$

B5.3 The value of U_{nj}^{t} for $0<j<n$ is found using eq. (7) as follows:

$$U_{nj}^{t}=U_{nj}^{t+\Delta_t}\left(1-\Delta_t c\right)+\Delta_t\left[e_2U_{n(j+1)}^{t+\Delta_t}+d_1U_{(n-1)j}^{t+\Delta_t}+d_2U_{n(j-1)}^{t+\Delta_t}\right].$$

B6. Solving for U_{ij}^{t} in the implicit method

B6.1 The implicit method is developed from Cairns (2004:170). Eq. (8) for $0<i, j<n$ can be represented as:

$$U_{ij}^{t}\left(-1-\Delta_t c\right)+U_{ij}^{t+\Delta_t}+\Delta_t\left[e_1U_{(i+1)j}^{t}+e_2U_{i(j+1)}^{t}+d_1U_{(i-1)j}^{t}+d_2U_{i(j-1)}^{t}\right]=0.$$

B6.2 Eq. (8) for $0<i<n$ and $j=n$ can be represented as:

$$U_{in}^{t}\left(-1-\Delta_t c\right)+U_{in}^{t+\Delta_t}+\Delta_t\left[e_1U_{(i+1)n}^{t}+d_1U_{(i-1)n}^{t}+d_2U_{i(n-1)}^{t}\right]=0.$$

B6.3 Eq. (8) for $0<j<n$ and $i=n$ can be represented as:

$$U_{nj}^{t}\left(-1-\Delta_t c\right)+U_{nj}^{t+\Delta_t}+\Delta_t\left[e_2U_{n(j+1)}^{t}+d_1U_{(n-1)j}^{t}+d_2U_{n(j-1)}^{t}\right]=0.$$

B6.4 The above system of equations is solved simultaneously to obtain U_{ij}^{t} for $0<i$, $j\le n$ as $U=Q^{-1}X$ where Q is an $n^2\times n^2$ co-efficient matrix and X is an $n^2\times 1$ target matrix, obtained from the equations above. Each row and column index x in Q and each row index x in X specifies the value relating to the (i,j) grid point as follows: $x=1$ is point grid point (1,1), $x=2$ is point grid point (1,2), …, $x=n$ is point grid point (1,n), $x=n+1$ is point grid point (2,1), …, $x=n^4-1$ is point grid point $(n,n-1)$ and $x=n^4$ is point grid point (n, n).

B7. Solving for U_{ij}^t in the CN method

B7.1 The CN method is developed from Cairns (2004:172). The discretised version of eq. (6) used for the CN method is:

$$
\begin{aligned}
&-\frac{1}{2}a_0\left(U_{ij}^{t+\Delta_t}+U_{ij}^t\right)+\frac{U_{ij}^{t+\Delta_t}-U_{ij}^t}{\Delta_t}+\frac{1}{2}a_1\left(\frac{U_{ij}^{t+\Delta_t}-U_{(i-1)j}^{t+\Delta_t}}{\Delta_{\theta_1}}+\frac{U_{ij}^t-U_{(i-1)j}^t}{\Delta_{\theta_1}}\right)+\\
&\frac{1}{2}a_2\left(\frac{U_{ij}^{t+\Delta_t}-U_{i(j-1)}^{t+\Delta_t}}{\Delta_{\theta_2}}+\frac{U_{ij}^t-U_{i(j-1)}^t}{\Delta_{\theta_2}}\right)+\\
&\frac{1}{4}b_1\left(\frac{U_{(i+1)j}^{t+\Delta_t}-2U_{ij}^{t+\Delta_t}+U_{(i-1)j}^{t+\Delta_t}}{\Delta_{\theta_1}{}^2}+\frac{U_{(i+1)j}^t-2U_{ij}^t+U_{(i-1)j}^t}{\Delta_{\theta_1}{}^2}\right)+\\
&\frac{1}{4}b_2\left(\frac{U_{i(j+1)}^{t+\Delta_t}-2U_{ij}^{t+\Delta_t}+U_{i(j-1)}^{t+\Delta_t}}{\Delta_{\theta_2}{}^2}+\frac{U_{i(j+1)}^t-2U_{ij}^t+U_{i(j-1)}^t}{\Delta_{\theta_2}{}^2}\right)=0.
\end{aligned}
\tag{11}
$$

B7.2 Eq. (11) for $0<i,j<n$ can be represented as:

$$U_{ij}^t\left(-2-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)j}^t+e_2U_{i(j+1)}^t+d_1U_{(i-1)j}^t+d_2U_{i(j-1)}^t\right]=-x_1\,.$$

B7.3 Eq. (11) for $0<i<n$ and $j=n$ can be represented as:

$$U_{in}^t\left(-2-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)n}^t+d_1U_{(i-1)n}^t+d_2U_{i(n-1)}^t\right]=-x_2\,.$$

B7.4 Eq. (11) for $0<j<n$ and $i=n$ can be represented as:

$$U_{nj}^t\left(-2-\Delta_t c\right)+\Delta_t\left[e_2U_{n(j+1)}^t+d_1U_{(n-1)j}^t+d_2U_{n(j-1)}^t\right]=-x_3\,.$$

where x_1, x_2 and x_3 are:

$$
\begin{aligned}
x_1&=U_{ij}^{t+\Delta_t}\left(2-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)j}^{t+\Delta_t}+e_2U_{i(j+1)}^{t+\Delta_t}+d_1U_{(i-1)j}^{t+\Delta_t}+d_2U_{i(j-1)}^{t+\Delta_t}\right]\\
x_2&=U_{in}^{t+\Delta_t}\left(2-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)n}^{t+\Delta_t}+d_1U_{(i-1)n}^{t+\Delta_t}+d_2U_{i(n-1)}^{t+\Delta_t}\right]\\
x_3&=U_{nj}^{t+\Delta_t}\left(2-\Delta_t c\right)+\Delta_t\left[e_2U_{n(j+1)}^{t+\Delta_t}+d_1U_{(n-1)j}^{t+\Delta_t}+d_2U_{n(j-1)}^{t+\Delta_t}\right].
\end{aligned}
$$

The above system is solved as in the implicit method.

B8. Solving for U_{ij}^{t} in the Hopscotch method

B8.1 The Hopscotch method is developed from Wilmott (2006:1260). U_{ij}^{t} is calculated as in section B5 in the first stage of each iteration. U_{ij}^{t} is calculated as follows in the second stage of each iteration, using the values calculated in the first stage:

B8.2 The value of U_{ij}^{t} for $0<i,j<n$ is found as follows:

$$U_{ij}^{t}=U_{ij}^{t+\Delta_t}\left(1-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)j}^{t}+e_2U_{i(j+1)}^{t}+d_1U_{(i-1)j}^{t}+d_2U_{i(j-1)}^{t}\right].$$

B8.3 The value of U_{in}^{t} for $0<i<n$ is found as follows:

$$U_{in}^{t}=U_{in}^{t+\Delta_t}\left(1-\Delta_t c\right)+\Delta_t\left[e_1U_{(i+1)n}^{t}+d_1U_{(i-1)n}^{t}+d_2U_{i(n-1)}^{t}\right].$$

B8.4 The value of U_{nj}^{t} for $0<j<n$ is found as follows:

$$U_{nj}^{t}=U_{nj}^{t+\Delta_t}\left(1-\Delta_t c\right)+\Delta_t\left[e_2U_{n(j+1)}^{t}+d_1U_{(n-1)j}^{t}+d_2U_{n(j-1)}^{t}\right].$$

B9. Solving for U_{ij}^{t} in the ADI method

B9.1 The ADI method is developed from Wilmott (2006:1259). The discretised version of eq. (6) used for the ADI method if θ_1 is solved implicitly is:

$$\begin{aligned}&-\frac{1}{2}a_0\left(U_{ij}^{t+\Delta_t}+U_{ij}^{t}\right)+\frac{U_{ij}^{t+\Delta_t}-U_{ij}^{t}}{\Delta_t}+a_1\frac{U_{ij}^{t}-U_{(i-1)j}^{t}}{\Delta_{\theta_1}}+a_2\frac{U_{ij}^{t+\Delta_t}-U_{i(j-1)}^{t+\Delta_t}}{\Delta_{\theta_2}}+\\&\frac{1}{2}b_1\frac{U_{(i+1)j}^{t}-2U_{ij}^{t}+U_{(i-1)j}^{t}}{\Delta_{\theta_1}{}^2}+\frac{1}{2}b_2\frac{U_{i(j+1)}^{t+\Delta_t}-2U_{ij}^{t+\Delta_t}+U_{i(j-1)}^{t+\Delta_t}}{\Delta_{\theta_2}{}^2}=0\end{aligned}\qquad(12)$$

B9.2 The discretised version of eq. (6) used for the ADI method if θ_2 is solved implicitly is:

$$\begin{aligned}&-\frac{1}{2}a_0\left(U_{ij}^{t+\Delta_t}+U_{ij}^{t}\right)+\frac{U_{ij}^{t+\Delta_t}-U_{ij}^{t}}{\Delta_t}+a_1\frac{U_{ij}^{t+\Delta_t}-U_{(i-1)j}^{t+\Delta_t}}{\Delta_{\theta_1}}+a_2\frac{U_{ij}^{t}-U_{i(j-1)}^{t}}{\Delta_{\theta_2}}+\\&\frac{1}{2}b_1\frac{U_{(i+1)j}^{t+\Delta_t}-2U_{ij}^{t+\Delta_t}+U_{(i-1)j}^{t+\Delta_t}}{\Delta_{\theta_1}{}^2}+\frac{1}{2}b_2\frac{U_{i(j+1)}^{t}-2U_{ij}^{t}+U_{i(j-1)}^{t}}{\Delta_{\theta_2}{}^2}=0\end{aligned}\qquad(13)$$

B9.3 The values for U_{ij}^{t} for $0<i,j\leq n$ are solved simultaneously using the above set of equations. Each iteration contains n sets of n simultaneous equations. Each set of simultaneous equations relates to solving U_{ij}^{t} for $0<i\leq n$ and fixed j if θ_1 is solved implicitly, solved as $U=Q_1^{-1}X_1$. Each set of simultaneous equations relates to solving U_{ij}^{t} for $0<j\leq n$ and fixed

i if θ_2 is solved implicitly, solved as $U = Q_2^{-1} X_2$. Q_1 and Q_2 are $n \times n$ co-efficient matrices obtained from eq. (12) and eq. (13) above. X_1 and X_1 are $n \times 1$ target matrices obtained from eq. (12) and eq. (13) above. Each row and column index x in Q_1 and each row index x in X_1 specifies the value relating to the (i, j) grid point for fixed j and $0 < i \leq n$. Each row and column index x in Q_2 and each row index x in X_2 specifies the value relating to the (i, j) grid point for fixed i and $0 < j \leq n$.

APPENDIX C

Results of the fifty random parameter sets

TABLE 11. Accuracy and efficiency for the fifty random parameter sets for selected numerical methods. Accuracy is measured using RMS and MAE and efficiency is measured using total calculation time in seconds, calculated over thirty integer times to maturity $\tau = 1,\ldots,30$[7]

Runge–Kutta			Euler			Taylor			CN ODE			ADI			CN (FD method)		
RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time
0,0000	0,0000	0,03	0,0000	0,0000	0,63	0,0000	0,0000	0,74	0,0000	0,0000	0,29	0,0017	0,0004	12,70	0,0023	0,0006	68,05
0,0000	0,0000	0,03	0,0000	0,0000	0,58	0,0000	0,0000	5,66	0,0000	0,0000	0,15	0,0009	0,0002	12,70	0,0013	0,0003	67,93
0,0000	0,0000	0,02	0,0029	0,0000	1,45	0,0035	0,0000	1,12	0,0030	0,0000	0,33	0,0404	0,0005	25,47	0,0348	0,0005	136,86
0,0000	0,0000	0,03	0,0000	0,0000	0,07	0,0000	0,0000	7,21	0,0000	0,0000	0,08	0,0010	0,0006	12,73	0,0010	0,0005	68,65
0,0000	0,0000	0,03	0,0000	0,0000	0,09	0,0000	0,0000	3,60	0,0000	0,0000	0,11	0,0012	0,0007	12,77	0,0012	0,0006	68,60
0,0000	0,0000	0,03	0,0000	0,0000	1,08	0,0000	0,0000	1,29	0,0000	0,0000	0,21	0,0036	0,0007	12,73	0,0041	0,0008	68,19
0,0000	0,0000	0,03	0,0004	0,0000	1,99	0,0004	0,0000	2,88	0,0004	0,0000	0,47	0,0429	0,0010	12,66	0,0356	0,0007	68,20
0,0000	0,0000	0,03	0,0001	0,0000	2,71	0,0001	0,0000	4,56	0,0001	0,0000	0,50	0,0106	0,0009	12,71	0,0097	0,0007	68,38
0,0000	0,0000	0,03	0,0000	0,0000	0,97	0,0000	0,0000	10,44	0,0000	0,0000	0,18	0,0010	0,0003	12,70	0,0010	0,0003	68,11
0,0000	0,0000	0,03	0,0012	0,0000	2,32	0,0011	0,0000	0,13	0,0012	0,0000	0,60	0,1817	0,0026	12,75	0,1696	0,0022	68,25
0,0000	0,0000	0,03	0,0000	0,0000	0,18	0,0000	0,0000	1,03	0,0000	0,0000	0,14	0,0006	0,0002	12,75	0,0003	0,0001	68,43
0,0000	0,0000	0,02	0,0002	0,0000	1,14	0,0002	0,0000	0,52	0,0002	0,0000	0,18	0,0014	0,0002	12,74	0,0013	0,0002	136,91
0,0000	0,0000	0,03	0,0000	0,0000	0,17	0,0000	0,0000	1,28	0,0000	0,0000	0,17	0,0003	0,0001	12,74	0,0005	0,0001	68,47
0,0000	0,0000	0,03	0,0012	0,0000	2,32	0,0013	0,0000	3,76	0,0011	0,0000	0,39	0,0350	0,0006	12,68	0,0187	0,0005	68,18
0,0000	0,0000	0,03	0,0000	0,0000	0,21	0,0000	0,0000	4,83	0,0000	0,0000	0,10	0,0006	0,0003	12,70	0,0005	0,0002	68,31
0,0000	0,0000	0,03	0,0003	0,0000	1,54	0,0002	0,0000	0,76	0,0003	0,0000	0,33	0,0287	0,0010	12,75	0,0241	0,0007	68,13
0,0000	0,0000	0,03	0,0001	0,0000	3,26	0,0001	0,0000	5,75	0,0001	0,0000	0,33	0,0037	0,0008	12,75	0,0054	0,0009	68,18
0,0000	0,0000	0,03	0,0001	0,0000	0,99	0,0001	0,0000	2,34	0,0001	0,0000	0,19	0,0036	0,0004	12,70	0,0039	0,0005	136,35
0,0000	0,0000	0,02	0,0004	0,0000	1,18	0,0003	0,0000	0,25	0,0003	0,0000	0,19	0,0015	0,0002	12,68	0,0017	0,0003	136,37
0,0005	0,0000	0,04	0,0046	0,0000	3,31	0,0040	0,0000	1,85	0,0035	0,0000	0,80	0,3284	0,0027	12,70	0,2832	0,0020	68,44
0,0000	0,0000	0,03	0,0000	0,0000	0,29	0,0000	0,0000	2,87	0,0000	0,0000	0,10	0,0005	0,0002	12,72	0,0004	0,0002	68,40
0,0000	0,0000	0,02	0,0000	0,0000	1,42	0,0000	0,0000	5,27	0,0000	0,0000	0,41	0,0026	0,0006	12,70	0,0024	0,0006	68,16
0,0000	0,0000	0,03	0,0000	0,0000	2,03	0,0000	0,0000	9,27	0,0000	0,0000	0,33	0,0030	0,0007	12,73	0,0035	0,0007	68,56
0,0000	0,0000	0,03	0,0001	0,0000	2,79	0,0001	0,0000	2,54	0,0001	0,0000	0,33	0,0031	0,0007	12,75	0,0045	0,0008	68,24
0,0001	0,0000	0,03	0,0002	0,0000	1,79	0,0002	0,0000	5,61	0,0002	0,0000	0,18	0,0039	0,0006	12,70	0,0076	0,0007	68,89

Runge–Kutta			Euler			Taylor			CN ODE			ADI			CN (FD method)		
RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time	RMS	MAE	Time
0,0001	0,0000	0,06	0,0001	0,0000	1,20	0,0001	0,0000	3,39	0,0001	0,0000	0,26	0,0015	0,0002	12,73	0,0015	0,0003	68,16
0,0000	0,0000	0,02	0,0000	0,0000	0,17	0,0000	0,0000	1,88	0,0000	0,0000	0,19	0,0004	0,0002	12,75	0,0002	0,0001	68,65
0,0000	0,0000	0,03	0,0001	0,0000	0,93	0,0001	0,0000	6,13	0,0001	0,0000	0,18	0,0037	0,0005	12,68	0,0040	0,0005	137,20
0,0000	0,0000	0,03	0,0000	0,0000	0,68	0,0000	0,0000	1,55	0,0000	0,0000	0,16	0,0012	0,0004	12,74	0,0014	0,0004	68,12
0,0000	0,0000	0,03	0,0000	0,0000	0,46	0,0000	0,0000	14,45	0,0000	0,0000	0,13	0,0006	0,0002	12,73	0,0007	0,0002	68,17
0,0000	0,0000	0,03	0,0000	0,0000	0,19	0,0000	0,0000	2,05	0,0000	0,0000	0,22	0,0005	0,0002	12,77	0,0007	0,0001	68,13
0,0000	0,0000	0,03	0,0001	0,0000	2,34	0,0001	0,0000	1,15	0,0001	0,0000	0,14	0,0063	0,0011	12,73	0,0080	0,0011	68,44
0,0000	0,0000	0,03	0,0000	0,0000	0,46	0,0000	0,0000	1,10	0,0000	0,0000	0,13	0,0022	0,0003	12,66	0,0033	0,0005	68,15
0,0000	0,0000	0,05	0,0074	0,0000	1,49	0,0021	0,0000	0,30	0,0040	0,0000	0,32	0,0483	0,0005	12,70	0,0147	0,0005	68,15
0,0000	0,0000	0,03	0,0001	0,0000	1,25	0,0001	0,0000	1,18	0,0001	0,0000	0,28	0,0048	0,0003	12,78	0,0032	0,0002	68,09
0,0001	0,0000	0,03	0,0002	0,0000	2,30	0,0002	0,0000	4,78	0,0002	0,0000	0,43	0,0200	0,0007	12,63	0,0158	0,0005	68,07
0,0000	0,0000	0,03	0,0001	0,0000	0,51	0,0001	0,0000	2,79	0,0001	0,0000	0,16	0,0007	0,0001	12,65	0,0010	0,0002	137,10
0,0000	0,0000	0,03	0,0000	0,0000	2,24	0,0000	0,0000	14,37	0,0000	0,0000	0,21	0,0022	0,0007	12,67	0,0022	0,0007	68,21
0,0000	0,0000	0,03	0,0001	0,0000	1,36	0,0001	0,0000	1,37	0,0001	0,0000	0,25	0,0033	0,0004	12,70	0,0033	0,0004	137,23
0,0000	0,0000	0,03	0,0001	0,0000	1,04	0,0001	0,0000	2,26	0,0001	0,0000	0,28	0,0110	0,0007	12,69	0,0088	0,0005	68,27
0,0000	0,0000	0,02	0,0000	0,0000	0,29	0,0000	0,0000	1,53	0,0000	0,0000	0,28	0,0008	0,0006	12,75	0,0007	0,0005	68,48
0,0000	0,0000	0,03	0,0000	0,0000	1,44	0,0000	0,0000	4,62	0,0000	0,0000	0,21	0,0038	0,0008	12,72	0,0046	0,0008	68,52
0,0000	0,0000	0,02	0,0000	0,0000	3,16	0,0000	0,0000	5,23	0,0000	0,0000	0,16	0,0019	0,0008	12,67	0,0019	0,0008	68,44
0,0000	0,0000	0,03	0,0001	0,0000	2,10	0,0001	0,0000	1,84	0,0001	0,0000	0,31	0,0152	0,0011	12,69	0,0132	0,0008	68,20
0,0001	0,0000	0,03	0,0003	0,0000	1,55	0,0003	0,0000	1,38	0,0003	0,0000	0,48	0,0375	0,0011	12,70	0,0336	0,0009	68,16
0,0000	0,0000	0,02	0,0001	0,0000	1,82	0,0001	0,0000	3,17	0,0001	0,0000	0,15	0,0057	0,0008	12,75	0,0071	0,0009	68,52
0,0000	0,0000	0,03	0,0000	0,0000	0,15	0,0000	0,0000	1,05	0,0000	0,0000	0,16	0,0015	0,0008	12,73	0,0015	0,0007	68,52
0,0000	0,0000	0,03	0,0001	0,0000	2,27	0,0001	0,0000	0,89	0,0001	0,0000	0,50	0,0352	0,0022	12,76	0,0329	0,0019	68,28
0,0001	0,0000	0,03	0,0044	0,0000	2,00	0,0054	0,0000	2,22	0,0048	0,0000	0,42	0,0321	0,0009	12,73	0,0155	0,0009	69,12
0,0001	0,0000	0,03	0,0015	0,0000	1,76	0,0012	0,0000	0,71	0,0012	0,0000	0,40	0,0837	0,0012	12,74	0,0697	0,0008	68,19

[7] i. Grid size = 1600 for the two FD methods; ii. Δ_t for the two FD methods is the largest value for which a reduction in Δ_t by a factor of two results in a reduction in e_i by less than 0,01; and iii. Δ_τ for the ODE methods is the maximum value for which $e_i < 0{,}001$.

APPENDIX D
Impact of grid size on computation time for FD methods

D1. The overall efficiency of the FD methods is dependant on the grid size. An increase in the size of the grid increases the accuracy of the limited estimate but also increases the computational time of the method. The tables below consider the change in the computation time with a change in the size of the grid for selected sets of parameters at selected times to maturity. The multiplier represents the factor at which the computational time for a given size of grid increased relative to the previous grid size, i.e. $\frac{T_{i+1}}{T_i}$ where T_i represents the computational time for grid size *i*. The average multiplier is an average of these values.

TABLE 12. Computation times for USA with $\tau=2$[8]

	Grid size				Multiplier			Average multiplier
	100	400	1 600	6 400				
Explicit	2,58	10,54	42,73	170,70	4,08	4,05	3,99	4,04
Implicit	1,64	15,40	443,01	28758,49	9,39	28,76	64,92	34,35
CN	0,02	0,19	4,61	288,00	9,50	24,24	62,53	32,09
Hopscotch	1,40	5,79	23,06	98,63	4,14	3,99	4,28	4,13
ADI	0,06	0,21	0,85	3,61	3,50	4,05	4,24	3,93

TABLE 13. Computation times for South Africa with $\tau=10$[8]

	Grid size					Multiplier				Average multiplier
	100	400	900	1 600	2 500					
Explicit	6,39	26,18	59,39	108,40	167,29	4,10	2,27	1,83	1,54	2,43
Implicit	3,92	39,65	252,10	1109,51	3994,09	10,10	6,36	4,40	3,60	6,12
CN	0,11	0,96	5,45	22,91	82,39	8,65	5,68	4,20	3,60	5,53
Hopscotch	3,53	14,36	32,62	58,25	90,65	4,07	2,27	1,79	1,56	2,42
ADI	0,14	0,52	1,21	2,18	3,31	3,71	2,33	1,80	1,52	2,34

D.2 The computation time for the explicit, Hopscotch and ADI methods increase largely in direct proportion to an increase in the grid size as shown above in Tables 12 and 13. The computation time for the implicit and CN methods increases exponentially with an increase in grid size as a result of needing to create an $n \times n$ matrix to solve the n simultaneous equations, where n is the size of the grid. If an increase in the level of accuracy of the limited estimate is required (and subsequently a larger grid size needs to be used), the CN and implicit methods

[8] The values for Δ_t used are as follows: i. Δ_t = 0,005 for the explicit method; ii. Δ_t = 0,01 for the Hopscotch and implicit methods; iii. Δ_t = 0,5 for the ADI method; and iv. Δ_t = 1 for the CN method.

will become relatively less efficient. The CN method may, therefore, become more inefficient than the explicit and Hopscotch methods if an increase in the accuracy of the limited estimate is required.