2-2007

On Stiffness in Affine Asset Pricing Models

Shirley Junying HUANG
Singapore Management University, shirleyhuang@smu.edu.sg

Jun Yu
Singapore Management University, yujun@smu.edu.sg

Journal of Computational Finance (99–123) Volume 10/Number 3, Spring 2007

On stiffness in affine asset pricing models

Shirley J. Huang
Lee Kong Chian School of Business, Singapore Management University,
50 Stamford Road, Singapore 178899; email: shirleyhuang@smu.edu.sg

Jun Yu
School of Economics and Social Sciences, Singapore Management University,
90 Stamford Road, Singapore 178903; email: yujun@smu.edu.sg

Economic and econometric analysis of continuous-time affine asset pricing models often necessitates solving systems of ordinary differential equations (ODEs) numerically. Explicit Runge–Kutta (ERK) methods have been suggested to solve these ODEs both in the theoretical finance literature and in the financial econometrics literature. In this paper we show that under many empirically relevant circumstances the ODEs involve stiffness, a phenomenon which leads to some practical difficulties for numerical methods with a finite region of absolute stability, including the whole class of ERK methods. The difficulties are highlighted in the present paper in the context of pricing zero-coupon bonds as well as econometric estimation of dynamic term structure models via the empirical characteristic function. To overcome the numerical difficulties, we propose to use implicit numerical methods for the ODEs. The performance of these implicit methods relative to certain widely used ERK methods are examined in the context of bond pricing and parameter estimation. The results show that the implicit methods greatly improve the numerical efficiency.

1 INTRODUCTION

> "· · · around 1960, things became completely different and everyone became aware that world was full of stiff problems."
>
> Dahlquist (1985)

When valuing financial assets in a continuous-time, arbitrage-free framework, one often needs to find the numerical solution to a partial differential equation (PDE) (examples include the Feynman–Kac PDE for bond prices; see Duffie

Both authors gratefully acknowledge financial support from the Research Office at Singapore Management University. We would like to thank the anonymous referee for constructive comments that have substantially improved the article. We also wish to thank John Butcher, David Chen, Peter Phillips, Dai Min and participants in the Econometric Society World Congress in London, 2005 International Symposium on Econometrics Development in Beijing, the Inaugural Saw Centre for Financial Studies Conference on Quantitative Finance and the Singapore Econometric Study Group Meeting for comments on an earlier version of the paper. All computations were performed using MATLABr12 and Compact Visual FORTRAN 6.6a.

(2001) for details and references). Given the fact that in many practically relevant cases solving the PDE is computationally demanding and even becomes impractical when the number of states is modestly large, considerable attention has been paid to the class of affine asset pricing models where the risk-neutral drift and volatility functions of the process for the state variables are affine. Under the affine specification, many asset prices have either completely or nearly closed-form expressions. Important examples from the first category include Black and Scholes (1973) for pricing equity options, Vasicek (1977) and Cox, Ingersoll, and Ross ((1985); hereafter CIR) for pricing bonds and bond options, and Heston (1993) for pricing equity and currency options. Important examples from the second category include Duffie and Kan (1996) for pricing bonds, Chacko and Das (2002) for pricing interest derivatives, Bates (1996) for pricing currency options, and Duffie *et al* (2000) for a treatment of very general pricing relations. The solutions have nearly closed-form expressions in the sense that the PDE is decomposed into a system of ordinary differential equations (ODEs) and hence only a system of ODEs, as opposed to a PDE, has to be solved numerically. Such decomposition greatly facilitates the numerical implementation of pricing (Piazzesi (2003)).

Computational burdens are even heavier for econometric analysis of continuous-time dynamic asset pricing models based on discretely sampled observations. The reasons for this are: (i) the implied transition density of discretely sampled observations are solutions to PDEs which have to be solved numerically at every data point and at each iteration in the numerical optimizations (examples include the apparent need to solve the Kolmogorov forward and backward equations for the transition density in the maximum likelihood (ML) approach; see Lo (1988) for details and references[1]); (ii) as argued above, the asset prices themselves are numerical solutions to PDEs. However, the additional affine specification of the drift and volatility functions of the state process under the physical measure will greatly facilitate econometric estimation of continuous-time models via the empirical characteristic function (ECF). In particular, in an influential study Duffie *et al* (2000) showed that the conditional characteristic function (CCF) of the state variable(s) and hence the CCF of the observable(s) have nearly closed-form expressions for the class of affine models in the sense that only a system of ODEs has to be solved, although the likelihood functions are more difficult to obtain. Exploiting these nearly closed-form expressions of CCF, Singleton (2001) proposed several estimation procedures based on the ECF and CCF, circumventing the numerical problem in solving the PDEs. Yu (2004) provides a detailed review of the literature on the estimation problem via the ECF.

Due to the specific functional form of the ODEs found in the literature (ie, the so-called Ricatti equations), it is generally believed by many researchers that these ODEs can be solved fast and numerically efficiently using traditional

[1]In a recent contribution, however, Aït-Sahalia and Kimmel (2003) showed that it is no longer necessary to solve the Kolmogorov forward and backward equations to implement the approximate ML approach of Aït-Sahalia (2003).

numerical solvers for initial problems, such as explicit Runge–Kutta method. Specifically, Piazzesi (2003) recommended the MATLAB command ode45[2] to solve the ODEs.[3] In spite of its high order of accuracy, like any other explicit Runge–Kutta (ERK) method, ode45 has a finite region of absolute stability (see Huang (2005) for a detailed discussion of stability of Runge–Kutta methods and Butcher (2003) for a textbook treatment). The stability properties of numerical methods are important for obtaining a good approximation to the true solution. As is known, at each mesh point there are differences between the exact solution and the numerical solution, known as the error. Sometimes the accumulation of the error will cause instability and the numerical solution will no longer follow the path of the true solution. Therefore, a method must satisfy the stability condition so that the numerical solution will converge to the exact solution. In this paper, we show that under many situations that are empirically relevant in finance, the ODEs involve stiffness; that is, there are processes in the system with significantly different time scales (Dahlquist and Björck (1974)). In the term structure of interest rates literature, a typical finding in multi-factor models is that one factor has a very slow mean reversion and another has a much faster reverting rate; see, for example, Chen and Scott (1993), Geyer and Pichler (1999), and Jagannathan *et al* (2003). Moreover, in the financial volatility literature, it is empirically found that, when two volatility factors are used, one factor corresponds to the short memory component which has a very fast decay rate and the other corresponds to a long memory component which has a very slow decay rate; see, for example, Chernov *et al* (2003) and Engle and Lee (1999). In both cases, the processes in the system have very different time scales and hence exhibit stiffness. When an explicit method is used to solve a stiff problem, a small stepsize has to be chosen to ensure stability and hence the algorithm becomes numerically inefficient.

To address the problem of computational inefficiency of ERK methods for stiff systems, this paper introduces implicit numerical methods which ensure the so-called A-stability (that is, the stability region includes all of the left half-plane; see Section 2 for the formal definition). In particular, we examine the performance of two implicit methods in two different contexts. The two implicit methods considered are an implicit Runge–Kutta method and an implicit linear multistep method. In the first context we price zero-coupon bonds, while in the second context we estimate an affine term structure model. While implicit methods have received much attention in the recent numerical ODE literature (see Huang (2005) for a survey on implicit methods) and have been extensively used in engineering, physics, and chemistry (Aiken (1985)), to the best of our knowledge, we know of no earlier application in the finance literature.

As the proposed implicit solvers have an unbounded stability region, the stepsize can be large or small, depending on the degree of stiffness. This flexibility

[2]It is an order four embedded in order five Runge–Kutta method; see Shampine and Reichelt (1997) for details.

[3]From personal communications, we have found that many researchers who are working in this area have used ode45 to solve the Ricatti equations numerically.

in the choice of the stepsize makes the procedures numerically more efficient than the ERK methods when the underlying system involves stiffness. For example, when a 20-year zero coupon bond is priced under a stiff three-factor model, we find that an implicit method is about 70 times more efficient than ode45. More details of this implementation and comparison are provided in Section 4.

Compared with the situation in asset pricing, the numerical efficiency would be more of an issue in the econometric analysis of continuous-time affine models via the ECF. This is because: (i) estimation requires numerical optimizations and at each iteration in the numerical optimizations a system of ODEs has to be solved numerically; (ii) pricing only requires solving one system of ODEs, which is associated with the risk-neutral measure, to be solved whereas estimation requires an additional system of ODEs, which is associated with the physical measure, to be solved. Our finding indicates that, without sacrificing the statistical efficiency of parameter estimates, the implicit methods substantially reduce the computer time when a stiff two-factor CIR model and a three-factor affine model are estimated.

This paper is organized as follows. Section 2 defines the stiffness for initial value problems, reviews general approaches for non-stiff and stiff problems, and discusses the stability properties of numerical solvers. Section 3 explains the practical relevance of stiffness in the context of asset pricing and econometric estimation. In Section 4 we compare the performance of implicit methods with ERK methods in the two contexts. Section 5 concludes and outlines some further applications of implicit methods.

2 STIFFNESS AND IMPLICIT METHODS

2.1 What is stiffness?

Any initial problem of ODEs can be classified into one of two types, non-stiff and stiff. Although the difference between the two types of problems is not clear cut, stiffness is widely found (Aiken (1985)) in many applied sciences such as robotics, fluid dynamics, electric circuits, and chemical kinetics. However, to the best of our knowledge, stiffness has not been reported in finance. Several attempts at a rigorous definition of stiffness have been made in the numerical analysis literature. In this paper, we follow the definition of Gear (1971, p. 209).

DEFINITION 1 Stiff differential equations are differential equations with greatly differing time constants (ie, rates of decay).

Since, in a stiff system, some components of the solution decay much more rapidly than the others, for a numerical method with a finite region of stability, we are forced to use excessively small stepsizes in relation to the smoothness of the solution, which inevitably decreases computational efficiency and accumulates more machine roundoff error. As a result, stiffness leads to some practical difficulties for explicit methods.

To illustrate the problem, we consider the following simple, hypothetical example,

$$\frac{\mathrm{d}y_1(t)}{\mathrm{d}t} = y_1(t) - 2y_2(t) \tag{1}$$

$$\frac{\mathrm{d}y_2(t)}{\mathrm{d}t} = 1{,}001y_1(t) - 1{,}002y_2(t) \tag{2}$$

with initial conditions $y_1(0) = 1$ and $y_2(0) = -1$. This linear system has the following exact solution,

$$\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \frac{1{,}003}{999}\begin{pmatrix} 1 \\ 1 \end{pmatrix}\mathrm{e}^{-t} - \frac{2{,}002}{999}\begin{pmatrix} \dfrac{2}{1{,}001} \\ 1 \end{pmatrix}\mathrm{e}^{-1{,}000t} \tag{3}$$

The component of the solution involving the term $\mathrm{e}^{-1{,}000t}$ decays very fast whereas the other component involving e^{-t} decays very slowly. This feature can obviously be captured by the Jacobian matrix of the ODEs

$$\begin{pmatrix} 1 & -2 \\ 1{,}001 & -1{,}002 \end{pmatrix}$$

which has two distinct eigenvalues $\lambda_1 = -1$, $\lambda_2 = -1{,}000$, and hence the system can be rotated into a system of two independent differential equations,

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = -z(t) \tag{4}$$

$$\frac{\mathrm{d}w(t)}{\mathrm{d}t} = -1{,}000w(t) \tag{5}$$

To explain the numerical difficulty of explicit methods, we consider the explicit Euler method (other explicit methods share the same spirit) and obtain

$$w_{n+1} = (1 - 1{,}000h)w_n = (1 - 1{,}000h)^{n+1}w(0) \tag{6}$$

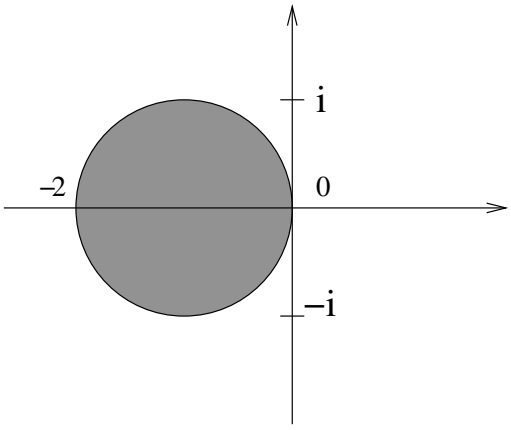
where h is the stepsize and w_{n+1} is the numerical approximation to $w((n+1)h)$. Note that the exact solution decays as t increases, thus the numerical solution should do the same. This clearly requires that $|1 - 1{,}000h| < 1$ and hence $0 < h < 0.002$ for a real value of h to fulfill the stability requirement. For this reason, we say the explicit Euler method has a *finite region of absolute stability*. It is clear that after a few steps, the magnitude of $w(t)$ will be very small compared with $z(t)$ which still contains the significant information that we are interested in.

Formally, for a linear problem $y'(t) = \lambda y(t)$ with $\lambda < 0$, the stability region of the explicit Euler method is given by $|1 + z| \leq 1$ where $z = \lambda L$ with L being a lag operator. The stability region is plotted in Figure 1.

Stiffness can be understood in the following linear system

$$y'(t) = My(t) \tag{7}$$

FIGURE 1 Stability region of explicit Euler method (shaded area).



where M is an $N \times N$ matrix which has distinct eigenvalues λ_j $(j = 1, \ldots, N)$. The system can be rotated into a system of N independent differential equations

$$z'(t) = \text{diag}(\lambda_1, \ldots, \lambda_N) z(t) \tag{8}$$

The solution of the above equation is of the form

$$z(t) = (C_1 \, \mathrm{e}^{\lambda_1 t}, \ldots, C_N \, \mathrm{e}^{\lambda_N t})^\top \tag{9}$$

where the C_i are constants which depend on the initial conditions. Let $\text{Re}(\cdot)$ be the real part of a complex number. If $\text{Re}(\lambda_j) < 0$ for $j = 1, \ldots, N$, then $C_i \, \mathrm{e}^{\lambda_i t} \to 0$ when $t \to \infty$. Consider the case where these eigenvalues are very different, for example, one of the eigenvalues has a very negative real part relative to the others. This means that

$$R = \frac{\max_j |\text{Re}(\lambda_j)|}{\min_j |\text{Re}(\lambda_j)|} \gg 1 \tag{10}$$

Denote $\max_j |\text{Re}(\lambda_j)|$ by λ_0. Hence, the exponential function $\exp(\lambda_0 t)$ decays to zero much more rapidly than the function based on the eigenvalue, $\min_j |\text{Re}(\lambda_j)|$. As a result, the system is difficult to solve for explicit methods. The ratio R is called the "stiffness ratio" which provides a measure of stiffness (only eigenvalues with negative real parts are considered).

As many ODEs encountered in continuous-time finance are actually non-linear, it is useful to generalize the stiffness ratio to non-linear systems:

$$y'(t) = f(y(t)), \quad f : R^N \to R^N$$
$$y_0 = y(t_0)$$

For non-linear systems the Jacobian is used to determine the stiffness. If the eigenvalues of the Jacobian $J = \partial f / \partial y$ (an $N \times N$ matrix of partial derivatives,

$\partial f_i/\partial y_j$) satisfy the same condition as in (10), then the initial value problem is stiff. In general, there are two types of error introduced by a numerical method, truncation error and roundoff error. On the one hand, stability requires the use of a small stepsize for the component with the fastest rate, which of course decreases the truncation error. On the other hand, the choice of a small stepsize leads to a large number of steps and hence long computer time. Moreover, because roundoff error will accumulate, after a large number of steps there will be a large accumulation of roundoff error for the slowest moving component where the derivatives are small and relatively constant. Clearly, for a stiff system, the trade-off is between large stepsizes to reduce computer time and roundoff error and small stepsizes to reduce truncation error.

2.2 A-stability and implicit methods

In this section we review certain numerical solvers and their properties that are relevant in finance, paying particular attention to the so-called A-stability and its linkage to implicit methods.

We begin the review with the definition of order. For any point t_n, let Y denote the approximation of $y(t_n)$. For some positive integer p, if we have $\|Y(t_n) - y(t_n)\|/h^p$ bounded, then we would define "the numerical solution to be of order p". If p is the highest value such that $\|Y(t_n) - y(t_n)\|/h^p$ is bounded, we would define "the order of the numerical solution to be p". A comparison of the Euler solution $Y(t_n)$ with the Taylor series of $y(t_n)$ gives the order of the Euler method as at least 1.

The Euler method discussed above is explicit. The next formula is referred to as the implicit Euler method,

$$y_n = y_{n-1} + h_n f(y_n), \quad n = 1, 2, \ldots, N, \quad y_0 = y(t_0) \tag{11}$$

It is implicit as Equation (11) is an implicit function in y_n. The stability region for the implicit method is $|1 - z| \geq 1$, which is shown in Figure 2. Obviously the stability region of the implicit Euler method is unbounded.

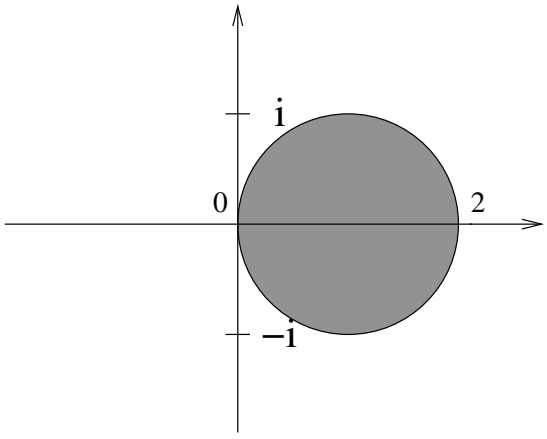
Runge–Kutta methods can be understood as one way to generalize the Euler method. They allow for a multiplicity of evaluations of function f within one step, and then use the information obtained to match a Taylor series expansion up to some higher order. For an ODE system $y' = f(y)$, a general Runge–Kutta method is of the form

$$Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f(Y_j), \quad i = 1, \ldots, s \tag{12}$$

$$y_n = y_{n-1} + h \sum_{i=1}^{s} b_i f(Y_i) \tag{13}$$

where the quantities $Y_1, Y_2, \ldots, Y_s$ are called stage values, which are approximations to solution values $y(t_{n-1} + c_i h)$ at points $t_{n-1} + c_i h$. The integer s is

FIGURE 2 Stability region of implicit Euler method (unshaded area).



the number of stages of the method. The c_i represent the position of the internal stages within one step. The Runge–Kutta formula can be conveniently represented by Butcher tableau,

$$\begin{array}{c|c} c & A \\ \hline & b^\top \end{array}$$

where $A = \{a_{ij}\}$, $b^\top = \{b_i\}$, and $c = \{c_i\}$. The set of numbers a_{ij} are the coefficients used to find the internal stages using linear combinations of the stage derivatives. The components of the vector b are coefficients which represent how the numerical solution at this step depends on the derivatives of the internal stages. The vector $c = [c_1, c_2, \ldots, c_s]^\top$ is called the abscissae. If matrix A is strictly lower triangular, ie, the internal stages can be calculated without depending on later stages, the method is explicit.

One widely used ERK method was proposed by Dormand and Prince (1980) (ode45 is the corresponding MATLAB command). Another example of ERK methods was proposed by Bogacki and Shampine (1989) (ode23 is the corresponding MATLAB command). It is of smaller order than ode45, but may be more efficient at crude tolerances and in the presence of moderate stiffness (Shampine and Reichelt (1997)).

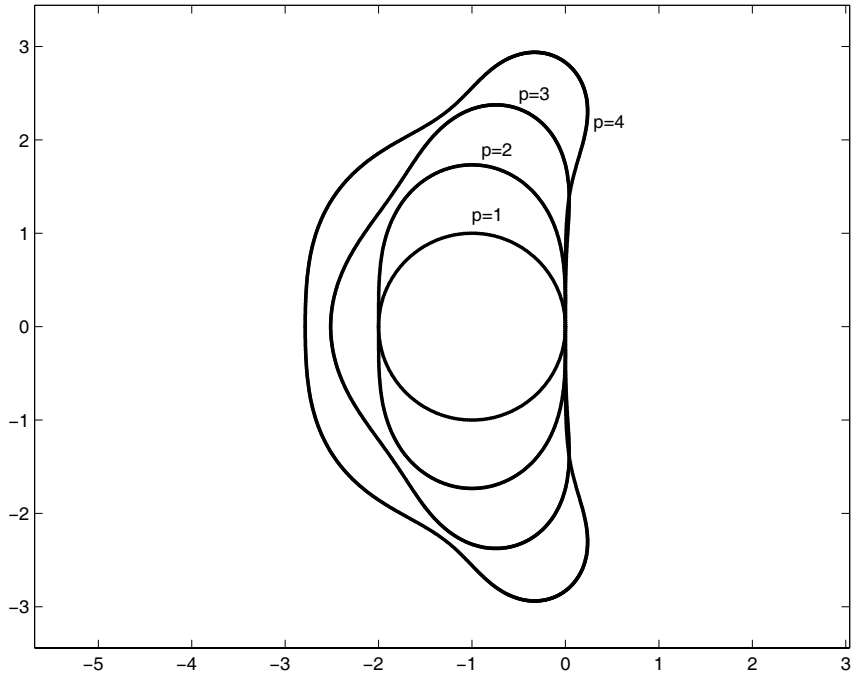
A well-known result in the numerical ODE literature (see, for example, Butcher (2003)) is that an ERK method with order p has the stability function of

$$R(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^p}{p!} + O(z^{p+1}) \tag{14}$$

which determines the stability region of the underlying method via

$$S = \{z \in C : |R(z)| \leq 1\}$$

FIGURE 3 The boundaries of the stability regions of ERK of order one, two, three, four, correspond to $f(z) = 1 + z + z^2/2! + \cdots + z^p/p!$ with $p = 1, 2, 3, 4$. In each case the stability region is the bounded set enclosed by these curves.



As a result, all ERK methods have bounded stability regions. The stability regions of order one, two, three, and four ERK methods with $p = 1, 2, 3, 4$ are given in Figure 3. It can be seen that although the stability region of a higher-order ERK method is slightly wider than that of the explicit Euler method, it is still highly restrictive. This will create, as shown above, numerical inefficiency for stiff systems. To deal with stiffness, a method with a much wider stability region is called for. A typical approach in the numerical ODE literature is to use a method with an unbounded stability region. This is the motivation behind the concept of A-stability (Dahlquist (1963)).

DEFINITION 2 A method whose stability region contains the whole of the left half-plane is called A-stable.

Since all of the ERK methods have bounded stability regions, obviously they are not A-stable (Iserles (1996)). In a ground-breaking paper, Dahlquist (1963) showed that explicit linear multistep methods cannot be A-stable either. These results lead to the following recommendation: for a stiff problem, it is always better to use implicit procedures; see, for example, Dahlquist and Björck (1974) and Huang (2005).

A simple implicit method is the implicit Euler method defined by Equation (11). As already demonstrated in Figure 2, the stability region of the implicit Euler method includes the whole half-plane and hence is A-stable. For the Runge–Kutta methods, if matrix A in the Butcher tableau is *not* strictly lower triangular, the method is implicit. A particular type of implicit Runge–Kutta method is the modified Rosenbrock method of Zedan (1990). The MATLAB code ode23s implements the method of order two.

Another way to generalize the Euler method is the use of linear multistep methods. The idea is to use several previous solutions and derivative values in computing the updated solution. The general form of a linear k-step method for an ODE $y' = f(y)$ is

$$y_n = \alpha_1 y_{n-1} + \cdots + \alpha_k y_{n-k} + h(\beta_0 f(y_n) + \cdots + \beta_k f(y_{n-k}))$$

where y_n is the numerical approximation to the exact solution at the point t_n and $\alpha_1, \alpha_2, \ldots, \alpha_k, \beta_0, \beta_1, \ldots, \beta_k$ are fixed numbers. The values of $\alpha_1, \alpha_2, \ldots, \alpha_k, \beta_0, \beta_1, \ldots, \beta_k$ are chosen to obtain the highest possible order and characterize a method. If $\beta_0 \neq 0$ and $\beta_i = 0,\ i = 1, \ldots, k$, the methods are obviously implicit and known as backward difference formulae (BDF; see, for example, Gear (1971)). A closely related implicit linear multistep method is based on the numerical differential formulae (NDF) of Klopfenstein (1971). The MATLAB code ode15s is a variable order solver which implements these two sets of formulae, with the default being NDF.

3 STIFFNESS IN ASSET PRICING MODELS

Affine asset pricing models have recently received much attention both in the theoretical finance literature and in the financial econometrics literature. Important contributions in this area include Duffie and Kan (1996), Dai and Singleton (2000), Duffie *et al* (2000), and Singleton (2001).

3.1 Stiffness in zero-coupon bonds

The multi-factor affine term structure model, introduced in Duffie and Kan (1996) and empirically investigated in Dai and Singleton (2000), adopts the following specifications.

1. Under the risk-neutral measure, the state variables, denoted by $Y(t) = (Y_1(t), \ldots, Y_N(t))^\top$, follow an affine diffusion,

$$\mathrm{d}Y(t) = \tilde{K}(\tilde{\theta} - Y(t))\,\mathrm{d}t + \Sigma\,\mathrm{diag}(\sqrt{\alpha_j + \beta_j^\top Y(t)})\,\mathrm{d}\tilde{W}(t)$$

   where $\tilde{K}$ and Σ are $N \times N$ matrices, α_j is an scalar, β_j is an N-dimensional vector, for $j = 1, \ldots, N$, and $\tilde{W}(t)$ is an N-dimensional independent standard Brownian motion under the risk-neutral measure.

2. The short rate is an affine function of $Y(t)$,

$$r(t) = \delta_0 + \delta_y^\top Y(t)$$

3. The market price of risk associated with factor j takes the form of

$$\lambda_j(t) = \lambda_j \sqrt{\alpha_j + \beta_j^\top Y(t)} \tag{15}$$

The specification implies that the physical measure of state variables has the affine form of

$$\mathrm{d}Y(t) = K(\theta - Y(t))\,\mathrm{d}t + \Sigma\,\mathrm{diag}(\sqrt{\alpha_j + \beta_j^\top Y(t)})\,\mathrm{d}W(t)$$

where $W(t)$ is an N-dimensional independent standard Brownian motion under the physical measure, $K = \tilde{K} - \Sigma\Phi$, $\theta = K^{-1}(\tilde{K}\tilde{\theta} + \Sigma\psi)$, with the jth row in matrix Φ being given by $\lambda_j\beta_j$ and the ith element in vector ψ being given by $\lambda_j\alpha_j$.

Under these three assumptions, the yield-to-maturity at time t of a zero-coupon bond that matures at $t + \tau$ takes the following affine expression (Duffie and Kan (1996)),

$$y(t;\tau) = -\frac{A(\tau)}{\tau} + \frac{B(\tau)^\top}{\tau} Y(t) = a(\tau) + b(\tau)^\top Y(t) \tag{16}$$

where $A(\tau)$, $B(\tau)$ satisfy a system of $N + 1$ ODEs (known as Ricatti equations)

$$\frac{\mathrm{d}A(\tau)}{\mathrm{d}\tau} = -\tilde{\theta}^\top \tilde{K}^\top B(\tau) + \frac{1}{2}\sum_{j=1}^{N} [\Sigma^\top B(\tau)]_j^2 \alpha_j - \delta_0 \tag{17}$$

$$\frac{\mathrm{d}B(\tau)}{\mathrm{d}\tau} = -\tilde{K}^\top B(\tau) - \frac{1}{2}\sum_{j=1}^{N} [\Sigma^\top B(\tau)]_j^2 \beta_j + \delta_y \tag{18}$$

with initial conditions $A(0) = 0$ and $B(0) = 0$. Note that the assumption about market price of risk in Equation (15) is more than enough for deriving the affine pricing relation in Equation (16). This assumption was relaxed in Duffee (2002) and further relaxed in Duarte (2004) without changing the pricing result.

It is known that the above affine model may not be econometrically identified. Dai and Singleton (2000) defined a model to be "maximal" if it has the maximum number of identifiable parameters and if it generates admissible (ie, well-defined) dynamics of state vectors. To find the maximal model, Dai and Singleton classified the affine models into $N + 1$ subfamilies according to the number "m" of the Y that determines the conditional variance of Y. Dai and Singleton called this subfamily $A_m(N)$ and illustrated the idea in the case of $N = 3$. For example, the

maximum $A_1(3)$ model (called $A_1(3)_{\text{Max}}$) takes the form

$$\begin{bmatrix} \mathrm{d}\nu(t) \\ \mathrm{d}\theta(t) \\ \mathrm{d}r(t) \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \nu & 0 \\ \kappa_{r\nu} & -\kappa & \kappa \end{bmatrix} \left(\begin{bmatrix} \bar{\nu} \\ \bar{\theta} \\ \bar{\theta} \end{bmatrix} - \begin{bmatrix} \nu(t) \\ \theta(t) \\ r(t) \end{bmatrix} \right) \mathrm{d}t + \begin{bmatrix} \eta & 0 & 0 \\ \eta\sigma_{\theta\nu} & 1 & \sigma_{\theta r} \\ \eta\sigma_{r\nu} & \sigma_{r\theta} & 1 \end{bmatrix} \times \begin{bmatrix} \sqrt{\nu(t)} & 0 & 0 \\ 0 & \sqrt{\zeta^2 + \beta_\theta \nu(t)} & 0 \\ 0 & 0 & \sqrt{\alpha_r + \nu(t)} \end{bmatrix} \begin{bmatrix} \mathrm{d}W_\nu(t) \\ \mathrm{d}W_\theta(t) \\ \mathrm{d}W_r(t) \end{bmatrix} \tag{19}$$

So,

$$\alpha_1 = \alpha_2 = \zeta^2, \quad \alpha_3 = \alpha_r, \quad \beta_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \beta_1 = \begin{pmatrix} \beta_\theta \\ 0 \\ 0 \end{pmatrix}, \quad \beta_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

As a result,

$$\tilde{K} = \begin{pmatrix} \mu + \eta\lambda_1 & 0 & 0 \\ \eta\sigma_{\theta\nu}\lambda_1 + \lambda_2\beta_\theta + \lambda_3\sigma_{\theta r} & \nu & 0 \\ \kappa_{r\nu} + \eta\lambda_1\sigma_{r\nu} + \lambda_2\beta_\theta\sigma_{r\theta} & -\kappa & \kappa \end{pmatrix}$$

With suitable restrictions on certain parameters, one obtains the $A_1(3)_{\text{BDFS}}$ and $A_1(3)_{\text{DS}}$ models. For example, if $\kappa_{r\nu} = \sigma_{\theta\nu} = \sigma_{r\theta} = \sigma_{r\theta} = \alpha_r = \beta = 0$, the $A_1(3)_{\text{Max}}$ model becomes the $A_1(3)_{\text{BDFS}}$ model.

The maximum $A_2(3)$ model (called $A_2(3)_{\text{Max}}$) takes the form

$$\begin{bmatrix} \mathrm{d}\nu(t) \\ \mathrm{d}\theta(t) \\ \mathrm{d}r(t) \end{bmatrix} = \begin{bmatrix} \mu & \kappa_{\nu\theta} & 0 \\ \kappa_{\theta\nu} & \nu & 0 \\ \kappa_{r\nu} & \kappa_{r\theta} - \kappa & \kappa \end{bmatrix} \left(\begin{bmatrix} \bar{\nu} \\ \bar{\theta} \\ \bar{r} \end{bmatrix} - \begin{bmatrix} \nu(t) \\ \theta(t) \\ r(t) \end{bmatrix} \right) \mathrm{d}t + \begin{bmatrix} \eta & 0 & 0 \\ 0 & \zeta & 0 \\ \eta\sigma_{r\nu} & \zeta\sigma_{r\theta} & 1 \end{bmatrix} \times \begin{bmatrix} \sqrt{\nu(t)} & 0 & 0 \\ 0 & \sqrt{\theta(t)} & 0 \\ \sqrt{\alpha_r + \beta_\theta \nu(t) + \theta(t)} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{d}W_\nu(t) \\ \mathrm{d}W_\theta(t) \\ \mathrm{d}W_r(t) \end{bmatrix} \tag{20}$$

As a result,

$$\tilde{K} = \begin{pmatrix} \mu + \eta\lambda_1 & \kappa_{r\theta} & 0 \\ \kappa_{r\theta} & \nu + \zeta\lambda_2 & 0 \\ \kappa_{r\nu} + \lambda_1\eta\sigma_{r\nu} + \lambda_3 & \kappa_{r\theta} - \kappa + \lambda_2\zeta\sigma_{r\theta} + \lambda_3\beta_\theta & \kappa \end{pmatrix}$$

With suitable restrictions on certain parameters, one obtains the $A_2(3)_{\text{Chen}}$ and $A_2(3)_{\text{DS}}$ models. For example, if $\kappa_{\nu\theta} = \sigma_{r\theta} = \alpha_r = \beta_\theta = 0$, the $A_2(3)_{\text{Max}}$ model becomes the $A_2(3)_{\text{DS}}$ model.

Dai and Singleton (2000) further empirically estimated and tested two maximal three-factor models and some of the restricted specifications in each of the two families. Using the empirical results reported by Dai and Singleton, we find the stiffness in the ODEs (18). For example, the estimates of $(\mu, \eta, \lambda_1, \nu, \kappa)$ in the $A_1(3)_{\text{BDFS}}$ model are (0.602, 0.007197, 67,900, 0.0523, 2.05). Using these

estimates we obtain

$$\tilde{K} = \begin{bmatrix} 489.3 & 0 & 0 \\ 0 & 0.0523 & 0 \\ 230{,}617.2 & -2.05 & 2.05 \end{bmatrix}$$

Three eigenvalues of $\tilde{K}$ are 489.3, 0.0523, 2.05 which lead to a stiffness ratio of 9,355.6 in (18). For another example, the estimates of $(\mu, \eta, \lambda_1, \nu, \zeta, \lambda_2, \kappa)$ in the $A_2(3)_{\mathrm{DS}}$ model are (0.636, 0.0109, 13,000, 0.103, 0.055857, −152, 2.7). Using these estimates we obtain

$$\tilde{K} = \begin{bmatrix} 142.45 & 0 & 0 \\ -33.9 & -8.387 & 0 \\ -26{,}537.3 & -2.7 & 2.7 \end{bmatrix}$$

The implied stiffness ratio is 52.76 and hence there is mild stiffness involved.

3.2 Stiffness in parameter estimation

Based on the assumption that the state variables $Y(t)$ follow an affine diffusion under the following physical measure

$$\mathrm{d}Y(t) = K(\theta - Y(t))\,\mathrm{d}t + \Sigma\,\mathrm{diag}(\sqrt{\alpha_j + \beta_j^\top Y(t)})\,\mathrm{d}W(t)$$

Duffie *et al* (2000) derived the CCF of Y_{t+1} conditional on Y_t:

$$c_{Y(t)}(r) = E[\exp(\mathrm{i}r^\top Y(t+1))|Y(t)] = \exp(C(1) + D(1)^\top Y(t)) \tag{21}$$

where $\mathrm{i} = \sqrt{-1}$, r is the transform variable which takes real values, and $C(\cdot)$ and $D(\cdot)$ satisfy the following complex-valued ODEs (which are Ricatti equations once again):

$$\frac{\mathrm{d}C(\tau)}{\mathrm{d}\tau} = \theta^\top K^\top D(\tau) + \frac{1}{2}\sum_{i=1}^{N}[\Sigma^\top D(\tau)]_j^2\alpha_j \tag{22}$$

$$\frac{\mathrm{d}D(\tau)}{\mathrm{d}\tau} = -K^\top D(\tau) + \frac{1}{2}\sum_{i=1}^{N}[\Sigma^\top D(\tau)]_j^2\beta_j \tag{23}$$

with initial conditions $C(0) = 0$ and $D(0) = \mathrm{i}r$.[4]

Under an affine pricing environment where $y(t;\tau) = a(\tau) + b(\tau)^\top Y(t)$, the CCF of $y(t+1;\tau)$ conditional on $y(t;\tau)$ is given by

$$c_{y(t)}(r) = \exp(\mathrm{i}r^\top a(\tau))c_{Y(t)}(b(\tau)r) \tag{24}$$

Exploiting this nearly closed-form solution for the CCF of $y(t)$, Singleton (2001) and Chacko and Viceira (2003) proposed several estimation methods based

[4]The ODEs were given as a boundary problem by Duffie *et al* (2000). Piazzesi (2003) and Yu (2004) provided equivalent initial value representations for $C(\tau)$ and $D(\tau)$.

on the ECF. Yu (2004) provided a detailed review of these methods and more generally the literature on the parameter estimation via the ECF.

Stiffness may be present in the ODEs (23). For example, taking the estimates reported in Geyer and Pichler (1999), the stiffness ratio is 2,847.2 in the estimated five-factor CIR model. The estimates reported in Chen and Scott (1991, 1992) for a two-factor CIR model imply a stiffness ratio of 351.9. These numbers indicate the presence of modest to severe stiffness. The implied stiffness ratios in $A_1(3)_{\mathrm{BDFS}}$, $A_1(3)_{\mathrm{DS}}$, and $A_1(3)_{\mathrm{Max}}$ of Dai and Singleton (2001) are 39.2, 77.0, and 78.9, respectively. The implied stiffness ratios in $A_2(3)_{\mathrm{Chen}}$, $A_2(3)_{\mathrm{DS}}$, and $A_2(3)_{\mathrm{Max}}$ of Dai and Singleton are 28.9, 26.2, and 40.7, respectively. These numbers indicate the presence of mild stiffness.

4 PERFORMANCE OF NON-STIFF SOLVERS AND STIFF SOLVERS

To investigate the numerical inefficiency of ERK methods for stiff systems and to compare the performance of explicit solvers with implicit solvers, we focus on a two-factor CIR model (ie, an $A_2(2)$ model) and a three-factor affine model of Balduzzi *et al* (1996) (ie, the $A_1(3)_{\mathrm{BDFS}}$ model).

4.1 A two-factor CIR model

There are several reasons why we use the two-factor CIR model to study the numerical efficiency. First, the CIR model is a special case of affine models. As a result, the bond price can be calculated by solving a system of real-valued ODEs and the CCF can be obtained by solving a system of complex-valued ODEs. Second, the CIR model has a closed-form expression for bond prices, enabling an examination of approximation errors of alternative numerical methods for bond pricing. Third, the model has only two factors so that a large-scale Monte Carlo study of parameter estimation via the ECF is computationally feasible.

In particular, we design two experiments, one for pricing bonds and the other for estimating model parameters. In both experiments, we use four numerical solvers for ODEs, two ERK methods, one implicit Runge–Kutta method and one implicit linear multistep method. The two explicit methods are ode45 and ode23 while the two implicit methods are ode23s and ode15s. All four methods are built-in functions in MATLAB.

The two-factor CIR model has two independent factors, each evolving over time according to a square-root process:

$$\mathrm{d}Y_i(t) = \kappa_i(\mu_i - Y_i(t))\,\mathrm{d}t + \sigma_i Y_i^{1/2}(t)\,\mathrm{d}W_i(t), \quad i = 1, 2 \tag{25}$$

The market price of risk for each factor is assumed to be $\lambda_i Y_i^{1/2}(t)$. As a result the risk-neutral measure is

$$\mathrm{d}Y_i(t) = (\kappa_i + \lambda_i)\left(\frac{\kappa_i\mu_i}{\kappa_i + \lambda_i} - Y_i(t)\right)\mathrm{d}t + \sigma_i Y_i^{1/2}(t)\,\mathrm{d}\tilde{W}_i(t), \quad i = 1, 2 \tag{26}$$

TABLE 1 Computational efficiency of explicit and implicit solvers for bong pricing under the two-factor CIR model. CPU is the CPU time in seconds when the system is solved repeatedly 5,000 times, Yield is the yield-to-maturity, and Stepsize is the average value of stepsizes.

	10-year bond			20-year bond		
	CPU (s)	**Yield**	**Stepsize**	**CPU (s)**	**Yield**	**Stepsize**
ode45	310	0.097471474	0.025	562	0.0978857433	0.026
ode23	237	0.097471469	0.077	367	0.0978857286	0.080
ode23s	122	0.097471778	0.345	115	0.0978866689	0.667
ode15s	82	0.097470910	0.218	82	0.0978864351	0.417
Exact		0.097471410			0.0978857088	

The instantaneous default-adjusted rate is assumed to be the sum of the two state variables:

$$r(t) = Y_1(t) + Y_2(t)$$

The parameters in the two experiments are set to be

$$\kappa_1 = 0.06, \quad \mu_1 = 0.03, \quad \sigma_1 = 0.03, \quad \lambda_1 = -0.01$$

and

$$\kappa_2 = 100, \quad \mu_2 = 0.02, \quad \sigma_2 = 0.1, \quad \lambda_2 = -70$$

so that

$$K = \begin{pmatrix} 0.06 & 0 \\ 0 & 100 \end{pmatrix} \quad \text{and} \quad \tilde{K} = \begin{pmatrix} 0.05 & 0 \\ 0 & 30 \end{pmatrix}$$

Obviously there exists severe stiffness in the ODEs (18) and ODEs (23) with stiffness ratios of 600 and 1,667, respectively.

4.1.1 Comparison for pricing bonds

Using the two-factor CIR model, we price two long-term bonds with maturities of 10 and 20 years. Both the analytical and numerical approaches are used to compute the yields of zero-coupon bonds. The analytical expression for prices of zero-coupon bonds is obtained from CIR (1985) and used to calculate the exact values for the yields. The numerical calculation is performed in MATLAB. The initial state variables are assumed to take the values of 0.03 and 0.02. To obtain the CPU time, we follow the suggestion made by a referee of replicating the experiment 5,000 times. Table 1 shows the CPU time (in seconds) on a Pentium-M laptop (1,600 MHz), the yield, and the average value of stepsizes used by each solver.

First, all four methods provide very accurate yields for both bonds. Second, although ode45 and ode23 are slightly more accurate, the marginal improvement in precision comes with higher computational cost. Compared with ode15s, for instance, ode45 is nearly a factor of three and six less efficient for the two bonds while ode23 is nearly a factor of two and three and a half times

TABLE 2 Computational efficiency and statistical efficiency of explicit and implicit solvers for parameter estimation under the two-factor CIR model. CPU is the averaged CPU time in seconds. The first number in each cell is the mean of the estimates while the second number is the standard deviation of the estimates.

	κ_1	μ_1	σ_1	λ_1	κ_2	μ_2	σ_2	λ_2	CPU (s)
True	0.06	0.03	0.03	−0.01	100	0.02	0.1	−70	
ode45	0.0903	0.0292	0.0308	−0.0088	105.9	0.0205	0.096	−66.5	28,578.0
	0.0100	0.0055	0.0101	0.0068	11.30	0.0041	0.0255	9.11	
ode23	0.0902	0.0293	0.0307	−0.0088	105.9	0.0206	0.096	−66.8	21,619.1
	0.0099	0.0055	0.0102	0.0068	11.31	0.0040	0.0252	9.08	
ode23s	0.0903	0.0293	0.0305	−0.0086	105.8	0.0207	0.097	−66.6	10,152.6
	0.0099	0.0054	0.0103	0.0069	11.32	0.0041	0.0256	9.10	
ode15s	0.0904	0.0294	0.0304	−0.0087	105.9	0.0205	0.096	−66.8	6,460.2
	0.0099	0.0055	0.0102	0.0069	11.29	0.0040	0.0252	9.09	

less efficient. The longer the maturity, the less the relative efficiency of the explicit methods. Third, the improvement in numerical efficiency by the implicit methods is manifest in the average value of stepsizes used by each solver. For both bonds, the two explicit methods use much smaller stepsizes than the two implicit methods. This is because the ODE is stiff and the ERK methods must choose small enough stepsizes to ensure stability. Moreover, when the maturity increases, the stepsize increases nearly in the same proportion for the two implicit methods while it remains constant for the two explicit methods. Therefore, it is expected that the computational efficiency of the implicit methods relative to the explicit methods increases with the maturity and with the stiffness.

4.1.2 Comparison for parameter estimation

In this section we estimate the two-factor CIR model based on 600 discretely sampled observations on six-month and 10-year zero-coupon bonds that are simulated from the same model. We repeat the experiment 1,000 times. As before, the numerical calculation is performed in MATLAB. Although the likelihood function is available for the two-factor CIR model and hence ML is feasible (see Chen and Scott (1992) and Phillips and Yu (2005) for the implementation of ML in this context), we estimate the model using the ECF method of Chacko and Viceira (2003) in order to highlight the importance of numerical efficiency when solving the stiff Ricatti equations.

The basic idea for the ECF method is to match the CCF implied from the model with the ECF obtained from the data over a grid of discrete points (ie, transform variables). One should keep in mind that the ECF method of Chacko and Viceira (2003) does not address the issue of the optimal choice of transform variables, and hence in general their estimator is statistically inefficient. However, their method is advantageous in that it is computationally less intensive than the method of Singleton (2001). This computational advantage makes a comparison of alternative ODE solvers feasible.

To use the method of Chacko and Viceira (2003), a grid of discrete points has to be chosen for $r = (r_1, r_2)^\top$. In this paper 30 points are used and listed in Appendix A. The only guidance for the choice is that the characteristic function contains more information around the origin (Yu (1998)). At each iteration in the numerical optimizations, one stiff system of real-valued ODEs in the form of (18) and 30 stiff systems of complex-valued ODEs in the form of (23) are solved numerically.

Table 2 shows the average values of the estimates (the first number in each cell), the standard errors of the estimates (the second number in each cell), and the CPU time (in seconds), all across 1,000 simulated series.

Several interesting results emerge from Table 2. First, the ECF method works well in all cases for all of the parameters, suggesting that the ECF method is a viable estimation technique for continuous-time affine models. Second, similar to a finding reached in the previous experiment where the four methods provide almost identical prices, the four methods produce almost the same results for the parameter estimation. Third, given the fact that the estimation involves the numerical bond pricing and the numerical calculation of the characteristic function, it is computationally expensive to obtain parameters estimates, consistent with our expectation. For example, it took approximately 8 and 6 hours to complete 1,000 replications for ode45 and ode23. Most importantly, similar to the situation in bond pricing where the two implicit methods are more efficient than the explicit methods, the two implicit methods continue to perform better than the explicit methods for the parameter estimation. For example, it took ode23s and ode15s approximately 2.8 and 1.8 hours of CPU time to complete 1000 replications. They are a factor of 1.8 and 3.4 more efficient than ode45, and a factor of 1.3 and 2.3 more efficient than ode23. Fourth, numerical efficiency is accomplished by the two implicit methods without compromising the gains by less statistical efficiency. Finally, a comparison of the two implicit solvers suggests that we should use ode15s as it not only provides accurate bond prices, as shown in Table 1, but also obtains parameter estimates in a shorter time.

4.2 A three-factor model

Although we have clearly identified the numerical problems of explicit methods in the context of the CIR model, CIR itself is a textbook example of a term structure model and has been overwhelmingly rejected by actual data; see Dai and Singleton (2000). To demonstrate the numerical problems of explicit methods in more complex and realistic frameworks, we consider a three-factor model of Balduzzi *et al* (1996) which is defined below. For this $A_1(3)_{\text{BDFS}}$ model, there is no closed-form expression for bond prices, neither is there a closed-form expression for the likelihood function.

In this comparison, we also designed two experiments, one for pricing bonds and the other for estimating model parameters. In both experiments, we used four numerical solvers for ODEs, namely ode45, ode23, ode23s, and ode15s, and MATLAB to perform numerical computations.

TABLE 3 Computational efficiency of explicit and implicit solvers for bong pricing under a three-factor model. CPU is the CPU time in seconds when the system is solved repeatedly 100 times, Yield is the yield-to-maturity, and Stepsize is the average value of stepsizes.

	10-year bond			20-year bond		
	CPU (s)	Yield	Stepsize	CPU (s)	Yield	Stepsize
ode45	99	0.0681304278	0.0018	202	0.0740219723	0.0018
ode23	63	0.0681304857	0.0055	125	0.0740219785	0.0055
ode23s	4.9	0.0681278444	0.204	5.2	0.0740219671	0.385
ode15s	2.6	0.0681304661	0.143	2.7	0.0740231149	0.263

The model proposed by Balduzzi *et al* (1996) is a three-factor model with stochastic mean and stochastic variance and, under the physical measure, takes the form of

$$\begin{bmatrix} \mathrm{d}\nu(t) \\ \mathrm{d}\theta(t) \\ \mathrm{d}r(t) \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \nu & 0 \\ 0 & -\kappa & \kappa \end{bmatrix} \left(\begin{bmatrix} \bar{\nu} \\ \bar{\theta} \\ \bar{\theta} \end{bmatrix} - \begin{bmatrix} \nu(t) \\ \theta(t) \\ r(t) \end{bmatrix} \right) \mathrm{d}t + \begin{bmatrix} \eta & 0 & 0 \\ 0 & 1 & 0 \\ \eta\sigma_{r\nu} & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \sqrt{\nu(t)} & 0 & 0 \\ 0 & \zeta & 0 \\ 0 & 0 & \sqrt{\nu(t)} \end{bmatrix} \begin{bmatrix} \mathrm{d}W_\nu(t) \\ \mathrm{d}W_\theta(t) \\ \mathrm{d}W_r(t) \end{bmatrix} \tag{27}$$

By assuming that λ_i is the market price of risk associated with factor i, the model is affine under the risk-neutral measure.

The parameters are set at

$$\mu = 1, \quad \nu = 0.5, \quad \kappa = 30, \quad \bar{\nu} = 0.002, \quad \bar{\theta} = 0.1, \quad \sigma_{r\nu} = 3$$

and

$$\zeta = 0.03, \quad \eta = 0.03, \quad \lambda_1 = 15{,}000, \quad \lambda_2 = 10, \quad \lambda_3 = 50$$

so that

$$K = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & -30 & 30 \end{pmatrix} \quad \text{and} \quad \tilde{K} = \begin{pmatrix} 451 & 0 & 0 \\ 0 & 0.5 & 0 \\ 1400 & -30 & 30 \end{pmatrix}$$

Obviously there exists severe stiffness in the ODE (18) and moderate stiffness in the ODE (23) with stiffness ratios of 902 and 60, respectively. However, the implied stiffness ratio in (18) is much smaller than that in the estimated $A_1(3)_{\mathrm{BDFS}}$ model of Dai and Singleton (2000).

Using this three-factor model, we price two long-term bonds with maturities of 10 and 20 years. The initial state variables are assumed to take the values of 0.008, 0.02, and 0.08. To obtain the CPU time, we replicate the experiment 100 times. Table 3 shows the CPU time (in seconds) on a Pentium-M laptop (1,600 MHz), the yield, and the average value of stepsizes used by each solver.

TABLE 4 Computational efficiency of explicit and implicit solvers for parameter estimation under a three-factor model. CPU is the CPU time in seconds.

Method	μ	ν	κ	$\bar{\nu}$	$\bar{\theta}$	$\sigma_{r\nu}$	ξ	η	λ_1	λ_2	λ_3	CPU (s)
ode45	1.15	0.71	33.4	0.00188	0.108	2.96	0.031	0.029	13,286	9.1	45.3	47,549.1
ode23	1.15	0.71	33.5	0.00191	0.108	2.95	0.031	0.029	13,294	9.1	45.0	42,784.3
ode23s	1.16	0.70	33.4	0.00190	0.111	2.97	0.032	0.029	13,277	8.9	44.7	6,317.4
ode15s	1.16	0.72	33.6	0.00191	0.110	2.96	0.032	0.028	13,271	8.7	45.5	4,961.2

Several interesting results emerge from Table 3. First, all four methods provide almost identical yields. Second, the two explicit methods are numerically much less efficient than the two implicit methods. For example, compared with ode15s, ode45 is a factor of 37 and 76 less efficient for the two bonds while ode23 is a factor of 23 and 47 less efficient. The longer the maturity, the less the relative efficiency in the explicit methods. Third, the improvement in numerical efficiency by the implicit methods is manifest in the difference of the average values of stepsizes used by each solver. For both bonds, the two explicit methods use much smaller stepsizes than the two implicit methods. Comparing the results in Table 2 with those in Table 1, it seems reasonable to claim that the computational efficiency of the implicit methods relative to the explicit methods increases with the number of factors.

To estimate the three-factor model, we simulate 600 discretely sampled observations on six-month, two-year and 10-year zero-coupon bonds from the same model. Different from the CIR model where we repeated the experiment 1,000 times, we only estimate the three-factor model once due to the high computational cost involved. Although the approximate ML method of Aït-Sahalia (2003) can be used to estimate this model, we use the ECF method for estimation, which requires solving two systems of ODEs numerically. A grid of 52 discrete points is chosen for $r = (r_1, r_2, r_3)^\top$ and listed in Appendix A.

Table 4 shows all of the estimates and CPU time (in seconds) obtained from all four solvers. Several interesting results emerge. First, the ECF method seems to yield reasonable estimates. Second, the four methods produce almost identical parameter estimates. Third, compared with bond pricing, parameter estimation is computationally much more demanding. For example, it took approximately 13 and 12 hours to get a set of parameter estimates when ode45 and ode23 are used. Most importantly, the two implicit methods continue to achieve a great deal of numerical efficiency relative to the explicit methods. For example, it only took 1.75 and 1.38 hours of CPU time to get a set of parameter estimates when ode23s and ode15s are used.

4.3 Implementation in FORTRAN

As MATLAB is an interpreted computing language, it may not be accurate in obtaining CPU timing. In this section we use FORTRAN to compare relative performances of explicit and implicit solvers in terms of computational efficiency.

TABLE 5 Computational efficiency of explicit and implicit solvers for bong pricing under a three-factor model. CPU is the CPU time in seconds when the system is solved repeatedly 5,000 times; Yield is the yield-to-maturity, Total function calls is the number of function evaluations.

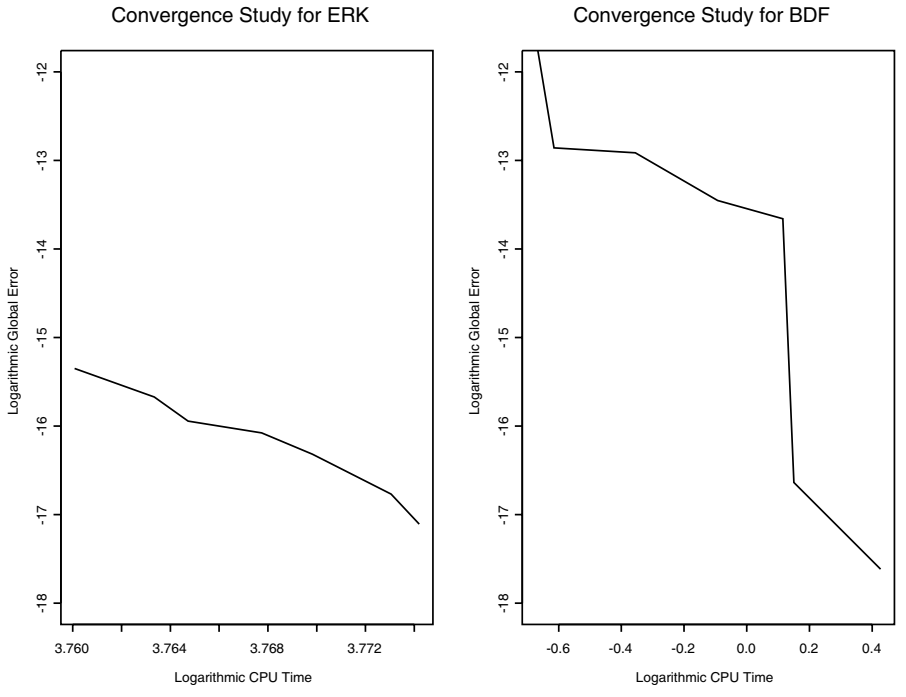
	20-year bond		
	CPU (s)	**Yield**	**Total function calls**
ERK	43.513	0.0740220547	19,968
Implicit BDF	1.532	0.0740220249	109

In particular, we use the explicit fifth-order and fourth-order Runge–Kutta method and the implicit BDF method of Gear (1971) to price two long-term bonds with maturities of 10 and 20 years based on $A_1(3)_{\text{BDFS}}$. The same parameter settings are used as in Section 4.2. The CPU time is obtained by replicating the experiment 5,000 times. In addition to its ability to more accurately obtain CPU time, FORTRAN also allows us to obtain the number of function calls.

Table 5 shows the CPU time (in seconds) on a Pentium-M laptop (1600 MHz), the yield, and the number of function evaluations by each solver. Several findings emerge from the table. First, the yields obtained from the FORTRAN routines are very similar to those calculated from MATLAB, suggesting that MATLAB is reliable in terms of computing yields. Second, the yields obtained from the two FORTRAN routines are very close to each other. Third, the implicit BDF method is more than 27 times as efficient as the ERK method. The order of magnitude in the relative efficiency seems to be consistent with those obtained from MATLAB, suggesting that the MATLAB ODE solvers can reasonably accurately obtain the CPU time. Fourth, the relative efficiency is manifest in the difference in the numbers of function evaluations by the two methods. While the ERK method has to evaluate 19,968 functions, the implicit BDF method only needs to calculate 109 functions.

As suggested by the referee, we conduct a convergence study by examining the relationship between the global error and the CPU time for both the ERK method and the implicit BDF method. To obtain the global error, we assume than the true solution is the same as the numerical solution with a very small tolerance level for error control (we use 10^{-9}). This is a standard approach in the numerical analysis literature. Figure 4 plots the logarithmic global error against the logarithmic CPU time for both methods. There are some interesting results. First, it can be seen that ERK is always much more time consuming than the implicit BDF method. Second, in both cases, the global error is a decreasing function of the CPU time. Third, initially ERK has a smaller global error than the implicit BDF method. The error slowly decreases with the CPU time for ERK. This result is consistent with what was reported by Hull *et al* (1972). On the other hand, the error decreases with the CPU time in a faster fashion for the implicit BDF method, suggesting that the implicit BDF is indeed more efficient than ERK and converges faster.

FIGURE 4 The logarithmic global error as a function of the logarithmic CPU time for both the ERK method and the implicit BDF method.



5 CONCLUSIONS AND IMPLICATIONS

Stiffness in ODEs exists widely in the continuous-time finance literature. The presence of stiffness causes all of the explicit solvers that have been suggested and widely used in the literature to be numerically inefficient. The procedures proposed in the present paper are based on implicit numerical methods which have unbounded stability regions, greater flexibility on the choice of the stepsize, and hence lead to improved numerical efficiency.

We apply the implicit methods to price zero-coupon bonds and to estimate affine dynamic term structure models via ECF under two affine models. We show that the procedures are highly effective to improve numerical efficiency without sacrificing statistical efficiency. As a result, affine models can be much more efficiently analyzed and hence computationally even more attractive than non-affine models. However, the techniques themselves are quite general and can be applied in many other contexts. For example, the methods can be used in connection with more efficient ECF procedures for estimation, including the procedures of Singleton (2001) and Carrasco *et al* (2006). Since these more efficient ECF procedures require more intensive numerical calculation of ODEs, one naturally expects greater gains in using the implicit methods. Also, the methods extend

naturally to more complicated affine models, including the affine jump-diffusion models (Duffie *et al* (2000); Chacko and Das (2002)) and semi-affine models (Duffee (2002); Duarte (2004)), where greater gains should be expected in the implicit methods as the dimension of the system grows. Furthermore, the methods can be applied to specifications outside of the affine family. An example is the quadratic affine model of Ahn *et al* (2002), where bond prices are shown to be dependent on a system of ODEs, although not Ricatti equations.

We have provided evidence of stiffness in bond markets. However, other financial markets can also have stiff problems. For example, using the estimates in the two-factor volatility models (termed LL2V, LL2V1, and LL2VF) obtained by Chernov *et al* (2003) from stock market data, we obtain the following stiffness ratios: 8,886.6, 1,028.6 and 426.5. These suggest severe stiffness.

Our methods can be applied to price many other financial assets as long as stiffness is a feature in a system. Examples include coupon-bearing bonds, swaptions, captions, mortgage-back securities, bond options, stock options, and currency options. As the calculation of these financial instruments is usually computationally more demanding than zero-coupon bonds, there may be further advantages of using the implicit methods over the explicit methods.

The implicit methods examined in the present paper include an implicit Runge–Kutta method and an implicit linear multi-step method. Both methods have their own merits. To take advantage of the merits of both methods, a new type of hybrid method, called the implicit general linear method, has appeared in the numerical ODE literature in recent years (Butcher (2003); Huang (2005)). How well the implicit general linear methods perform in financial stiff problems will be reported later.

APPENDIX A

When estimating the two-factor CIR model via ECF, the following 30 points were used to evaluate the CCF and ECF: (0.01, 0), (0, 0.01), (0.01, 0.01), (0.02, 0.01), (0.01, 0.02), (0.02, 0.02), (0.03, 0.02), (0.02, 0.03), (0.03, 0.03), (0.04, 0.03), (0.03, 0.04), (0.04, 0.04), (0.04, 0.05), (0.05, 0.04), (0.05, 0.05), (−0.01, 0), (0, −0.01), (−0.01, −0.01), (−0.02, −0.01), (−0.01, −0.02), (−0.02, −0.02), (−0.03, −0.02), (−0.02, −0.03), (−0.03, −0.03), (−0.04, −0.03), (−0.03, −0.04), (−0.04, −0.04), (−0.04, −0.05), (−0.05, −0.04), (−0.05, −0.05).

When estimating the three-factor model of Balduzzi *et al* (1996) via ECF, the following rule is used to select 52 points to evaluate the CCF and ECF. First, r_i takes one of the three values 0, 0.01, or 0.02. All possible combinations in (r_1, r_2, r_3), excluding the origin, yield 26 different points. We then take negative values of these 26 points to get another 26 points.

REFERENCES

Ahn, D., Dittmar, R. F., and Gallant, A. R. (2002). Quadratic term structure models: theory and evidence. *Review of Financial Studies* **15**, 243–288.

Aiken, R. C. (1985). *Stiff Computation*. Oxford University Press, Oxford.

Aït-Sahalia, Y. (2003). Closed-form likelihood expansion for multivariate diffusion. Working Paper, Princeton University.

Aït-Sahalia, Y., and Kimmel, R. (2003). Estimating affine multifactor term structure models using closed-form likelihood expansion. Working Paper, Princeton University.

Balduzzi, P., Das, S., Foresi, S., and Sundaram, R. (1996). A simple approach to three factor affine term structure models. *Journal of Fixed Income* **6**, 43–53.

Bates, D. (1996). Jumps and stochastic volatility: exchange rate processes implicit in PHLX deutschmark options. *Review of Financial Studies* **9**, 69–107.

Black, F., and Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy* **81**, 637–659.

Bogacki, P., and Shampine, L. F. (1989). A 3(2) pair of Runge–Kutta formulas. *Applied Mathematics Letters* **2**, 1–9.

Butcher, J. C. (2003). *The Numerical Analysis of Ordinary Differential Equations*. John Wiley & Sons, Chichester.

Carrasco, M., Chernov, M., Florens, J., and Ghysels, E. (2006). Efficient estimation of jump diffusions and general dynamic models with a continuum of moment conditions. *Journal of Econometrics*, to appear.

Chacko, G., and Das, S. (2002). Pricing interest rate derivatives: a general approach. *Review of Financial Studies* **15**, 195–241.

Chacko, G., and Viceira, L. M. (2003). Spectral GMM estimation of continuous-time processes. *Journal of Econometrics* **116**, 259–292.

Chen, R. R., and Scott, L. (1992). Pricing interest rate options in a two-factor Cox–Ingersoll–Ross model of the term structure. *Review of Financial Studies* **5**, 613–636.

Chen, R. R., and Scott, L. (1993). Maximum likelihood estimation of a multi-factor equilibrium model of the term structure of interest rates. *Journal of Fixed Income* **3**, 14–31.

Chernov, M., Gallant, A. R., Ghysels, E., and Tauchen, G. (2003). Alternative models of stock prices dynamics. *Journal of Econometrics* **116**, 225–257.

Cox, J., Ingersoll, J. and Ross, S. (1985). A Theory of the term structure of interest rates. *Econometrica* **53**, 385–407.

Dahlquist, G. (1963). A special stability problem for linear multistep methods. *BIT* **3**, 27–43.

Dahlquist, G. (1985). Some comments on stability and error analysis for stiff non-linear differential systems. *Stiff Computation*, Aiken., R. C. (ed.). Oxford University Press, Oxford.

Dahlquist, G., and Björck, A. (1974). *Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ.

Dai, Q., and Singleton, K. J. (2000). Specification analysis of affine term structure models. *Journal of Finance* **55**, 1943–1978.

Dormand, J. R., and Prince, P. J. (1980). A family of embedded Runge–Kutta formulae. *Journal of Computational Applied Mathematics* **6**, 19–26.

Duarte, J. (2004). Evaluating an alternative risk preference in affine term structure models. *Review of Financial Studies* **17**, 379–404.

Duffee, G. R. (2002). Term premia and interest rate forecasts in affine models. *Journal of Finance* **57**, 405–443.

Duffie, D. (2001). *Dynamic Asset Pricing Theory*. Princeton University Press, New Jersey.

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

Duffie, D., Pan, J., and Singleton, K. J. (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica* **68**, 1343–1376.

Engle, R., and Lee, G. (1999). A long-run and short-run component model of stock return volatility. *Cointegration, Causality and Forecasting*, Engle, R. and White, H. (eds). Oxford University Press, Oxford.

Gear, G. W. H. (ed.) (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ.

Geyer, A., and Pichler, S. (1999). A state-space approach to estimate and test multifactor Cox–Ingersoll–Ross models of the term structure. *Journal of Financial Research* **22**, 107–130.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility, with application to bond and currency options. *Review of Financial Studies* **6**, 327–343.

Huang, S. J. (2005). Implementation of general linear methods for stiff ordinary differential equations. PhD Thesis, Department of Mathematics, University of Auckland.

Hull, T. E., Enright, W. E., Fellen, B. M., and Sedgwick, A. E. (1972). Comparing numerical methods for ordinary differential equations. *SIAM Journal of Numerical Analysis* **9**, 603–637.

Iserles, A. (1996). *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, Cambridge.

Jagannathan, R., Kaplin, A., and Sun, S. (2003). An evaluation of multi-factor CIR models using LIBOR, swap rates, and cap and swaption prices. *Journal of Econometrics* **116**, 113–146.

Klopfenstein, R. W. (1971). Numerical differential formulas for stiff systems of ordinary differential equations. *RCA Review* **32**, 447–462.

Lo, A. W. (1988). Maximum likelihood estimation of generalized Itô processes with discretely sampled data. *Econometric Theory* **4**, 231–247.

Phillips, P. C. B., and Yu, J. (2005). Jackknifing bond option prices. *Review of Financial Studies* **18**, 707–742.

Piazzesi, M. (2003). Affine term structure models. *Handbook of Financial Econometrics*, to appear.

Shampine, L. F., and Reichelt, M. W. (1997). The MATLAB ODE suite. *SIAM Journal of Scientific Computing* **18**, 1–22.

Singleton, K. J. (2001). Estimation of affine asset pricing models using the empirical characteristic function. *Journal of Econometrics* **102**, 111–141.

Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics* **5**, 177–186.

Yu, J. (1998). The empirical characteristic function in time-series estimation. PhD Thesis, Department of Economics, University of Western Ontario.

Yu, J. (2004). Empirical characteristic function estimation and its applications. *Econometric Reviews* **23**, 93–123.

Zedan, H. (1990). Avoiding the exactness of the Jacobian matrix in Rosenbrock formulae. *Computers and Mathematics with Applications* **19**, 83–89.