A Modified Crank-Nicolson Method for Valuing Option Embedded Bonds using the Hull-White interest rate model.

Kiplin Perkins, Robert L Navin
Real Time Risk Systems
82 Wall St, Ste 500, New York, NY 10005
www.realtimerisksystems.com

1 Abstract

We demonstrate a modified Crank-Nicolson finite-difference diffusion algorithm for valuing option-embedded bonds using the Hull-White model of the short rate process. In particular, this method allows the practitioner to maintain model stability with relatively large values of bond tenor even when inputting both relatively large values of the *annualised* short-rate standard deviation and vanishingly small values of the mean reversion constant.

Indeed, typical finite-difference diffusion methods such as the fully explicit 3–1 and Crank-Nicolson 3–3 algorithms are demonstrated here to be unstable for values of short-rate standard deviation, σ, and bond tenor, T, such that $\sigma^2 T^3 > 3.6$ and $\sigma^2 T^3 > 17.1$ respectively, given sensible bond market tolerance requirements for the model's accuracy. In what follows, we show how the practitioner must modify these algorithms so that instability only occurs above an arbitrary user-determined value of the product $\sigma^2 T^3$.

As a concrete example, using a short-rate standard deviation of two points, $\sigma = 0.02$, the typical Crank-Nicolson algorithm does not accurately value bonds of tenor greater than 35 years if the number of modelling time steps is less than $N \approx 200$. We then demonstrate the general modifications one can make to the Crank-Nicolson algorithm in order to model bonds of any short-rate deviation and any tenor.

Our principal result is that for practitioners valuing long-dated option-embedded bonds in the Hull-White paradigm, modifying the diffusion algorithm, either fully-explicit or Crank-Nicolson, to achieve the desired accuracy is unavoidable.

2 Introduction

Many bonds contain options that may take effect during the life of the bond. For example, the issuer of a *callable bond* can prematurely terminate the bond by opting to prepay the holder some prescribed amount of cash, the *call amount*, on some prescribed *call date*. In so doing, the issuer settles (*calls*) their outstanding debt represented by that bond. A similar type of *option-embedded bond* is a *putable bond*, in which the holder has the option to sell back (*put*) the bond to the issuer at some predetermined *put amount* and *put date*.

The determination of whether to call or put a bond is highly sensitive to the present interest rate environment, which changes over time due to market influences. Consequently, these bonds are often priced using stochastic interest rate models. We will follow this trend here.

In particular, in Section 3, we will set up the formalism of the Hull-White model[1] of the interest rate curve in order to price interest-rate derivatives. A pricing equation, the *Hull-White equation*, will be presented.

Section 4 then outlines a standard fully-explicit 3–1 finite-difference algorithm for a numerical solution of the Hull-White equation. At the end of the section, we demonstrate that this method is unstable, or *stiff*, for derivatives of large tenor.

In this paper, we will consider two types of instability for two-dimensional finite-difference methods. The first, which we call *instability due to stiffness*, involves numerical solutions that need unreasonably many time steps, Δt, in order to maintain stability. (Note that the word "stiff" is ambiguously defined in the community. Our usage will always refer instability

due to the number of time steps being too large.) The second type, *Von Neumann mode instability*, concerns the lattice spacing of finite-difference grids, ie the relationship between Δt and the step size of the state variable, Δx. Clearly the two types are intimately related and, indeed, will be both analysed under one prescription, the standard *Von Neumann mode analysis*. It is important to note, however, that stabilising Von Neumann modes does *not* eliminate the stiffness problem – an additional analysis is necessary.

Accordingly, Section 5 presents the principal result of this work: a generalised Crank-Nicolson method with a prescription for evaluating the weights of the finite-difference diffusion operator. This generalisation is simply changing the standard 3–3 *molecule* of the Crank-Nicolson method into an n–m molecule. (Note that fully-explicit methods, n–1, are a subset.) As the molecule size increases, the number of weights increases, allowing the propagation error on the grid to be increasingly fine-tuned. Because this fine-tuning turns out to allow increasing accuracy for a given set of bond tenors and Hull-White input parameters, the practitioner can then decide which generalised Crank-Nicolson method to use to meet a required level of accuracy.

3 The Hull-White Model

This model is a stochastic, one-factor model of the interest rate curve, specified as a mean-reverting short-rate process plus a deterministic term structure of forward curve changes for each value of the short rate. The model takes the current curve of forward interest rates as an input, together with two constants, the annualised short-rate standard deviation, σ, and the mean reversion constant, k.

According to the Hull-White model, which is a simplified HJM model[3], derivatives are priced as expectation values under the risk-neutral mean-reverting stochastic short-rate process

$$dr = \left(\frac{df\left(t\right)}{dt} + \nu^2\left(t\right) - k\left[r - f\left(t\right)\right] \right) dt + \sigma dz\left(t\right) \quad , \tag{1}$$

where the function $f\left(t\right)$ is the *observed* current forward curve of interest rates, the constant k is the rate of mean reversion of the short rate, the constant σ is the annualised standard deviation of the short-rate, and $dz\left(t\right)$ is a Wiener process. The short rate mean-reverts to

$$f\left(t\right) + R\left(t\right) \quad , \tag{2}$$

where $R\left(t\right)$ is defined by

$$\frac{\partial R\left(t\right)}{\partial t} + kR\left(t\right) = \nu^2\left(t\right) \quad , \tag{3}$$

and $\nu^2\left(t\right)$ is given by[4]

$$\nu^2\left(t\right) = \frac{\sigma^2}{2k}\left(1 - e^{-2kt}\right) \quad . \tag{4}$$

The meaning of Eqs (2) and (3) is that the short rate only approximately reverts to the forward curve, $f\left(t\right)$, up to a correction, $R\left(t\right)$. This correction arises from the HJM[3] methodology and is due to the arbitrage freedom constraint imposed on stochastic *bond*

prices. Since bond prices are functions of the short rate, the arbitrage freedom constraint on all zero-coupon bond prices of maturity T observed at time $t \leq T$,

$$dB_T(t) = B_T(t)\, r dt + \Sigma_T(k,\, t,\, \sigma)\, B_T(t)\, dz(t) \quad , \tag{5}$$

leads to an equivalent constraint on the short rate process. This constraint is not trivial for two reasons. The first reason is the algebraic complexity of the transformation from a stochastic price-change process for zero-coupon bonds of all maturities to a single stochastic process for the short rate. The second is the application of Ito's lemma to this transformation. Heuristically, we may say the former complexity gives rise to a mean-reverting short-rate process, and the latter gives rise to the term $\nu^2(t)$ in Eq (1), which ensures that the process *does not* mean-revert precisely to the initially observed forward rates curve.

For simplicity, rewrite Eq (1) by letting

$$\begin{aligned} & x \equiv r - f(t) \quad , \\ \Rightarrow \quad & dx = \left(\nu^2(t) - kx\right) dt + \sigma dz(t) \quad . \end{aligned} \tag{6}$$

This new stochastic state variable, x, will be used for the remainder of this paper. It follows that when valuing functions of x today ($t = 0$) it is necessary that $x_0 = 0$ so that $r_0 = f(0)$.

From Eq (6), it follows that the pricing equation obeyed by all derivatives is[1, 3, 4, 5]

$$0 = \frac{\partial B_T(t)}{\partial t} + \frac{\sigma^2}{2}\frac{\partial^2 B_T(t)}{\partial x^2} + \left(\nu^2(t) - kx\right)\frac{\partial B_T(t)}{\partial x} - (x + f(t))\, B_T(t) \quad . \tag{7}$$

Given the boundary conditions of a zero-coupon bond, the closed-form solution to Eq (7) represents a stochastic *Hull-White zero-coupon bond*:

$$B_T^{\mathrm{HW}}(x,t) = \exp\left\{-a_T(t)\, x - b_T(t) - \int_t^T ds\, f(s)\right\} \quad , \tag{8}$$

where the functions $a_T(t)$ and $b_T(t)$ are given by

$$a_T(t) = \frac{1}{k}\left(1 - e^{-k(T-t)}\right) \tag{9}$$

and

$$b_T(t) = \frac{1}{2}\nu^2(t)\, a_T^2(t) \quad . \tag{10}$$

This solution describes a zero-coupon bond, or *zero*, that stochastically evolves from a known price today, $B_T^{\mathrm{HW}}(0,0)$, to a known terminal payout, the bond's par.

The analytic solution in Eq (8) is useful for boundary conditions of finite-difference grid valuations of fixed-income derivatives. The payout curve of a callable bond, for example, will asymptotically approach a value of a sum of Hull-White zeros as the value of x is further and further away from the call strike in either direction. This is analogous to the deep-in-the-money regime of a European equity call option, for which the payout value asymptotically becomes the stock forward minus the strike forward.

To obtain the solution for a fixed-coupons bond with known coupons c_i and par p, generalise Eq (8) in the obvious way:

$$B_T^{\mathrm{fixed}}(x,t) = \sum_{t_i \geq t}^{T} c_i B_{t_i}^{\mathrm{HW}}(x,t) + p B_T^{\mathrm{HW}}(x,t) \quad . \tag{11}$$

4 Solving the Pricing Equation Using a Fully Explicit Diffusion Operator

After some preliminary remarks, Section (4.2) presents a transformation of the Hull-White pricing equation, Eq (7), to the heat equation.

In Section (4.3), we outline an algorithm for a numerical solution of the heat equation in order to introduce the machinery and the main concepts of this work. For simplicity, we work through the example of the standard fully explicit 3–1 finite-difference method.

In Section (4.4), the standard convergence and stability analysis of Von Neumann[6] will be applied for optimisation. It turns out that ensuring fast convergence using Von Neumann stability analysis is equivalent to matching as many moments as possible of the distribution propagated on our finite-difference lattice to those of the Hull-White distribution. This will be investigated by propagating Hull-White zeros on the lattice. Furthermore, we demonstrate that even after ensuring Von Neumann stability for the 3–1 diffusion, the algorithm still contains a remarkable instability due to stiffness. This stiffness instability can be measured by the product $\sigma^2 T^3$, where σ is the annualised volatility of the short rate and T is the tenor of the Hull-White zero bond that is numerically propagated.

4.1 Preamble

For the sake of simple comparison, consider equity options numerically modelled over a normal distribution of stock-returns using a fully explicit 3–1 finite-difference diffusion. Convergence of such numerical solutions, as the number of time steps increases, may be achieved by enforcing that the first three moments of the *stock-price distribution* – cash forwards, stock forwards, and stock-squared forwards (to fix volatility) – are all propagated on the lattice identically to the exact continuous solutions. These three constraints fix the three weights of the diffusion operator. This choice of values for the weights is sometimes called *superweights*. It guarantees that the in-the-money- (ITM) and out-of-the-money (OTM) numerical results are exact irrespective of the number of time steps – as long as the ITM and OTM forms are linear combinations of these moments, as is usually the case. The central limit theorem then ensures that all higher moments are numerically approximated, with accuracy increasing as the number of time steps. This, in turn, ensures convergence to the correct answer with increasing number of time steps.

Specifically, the equity option pricing model is the expectation value under the risk-neutral process described by

$$dx_S = \left(r_\$ - r_S - \frac{1}{2}\sigma_S^2 \right) dt + \sigma_S dz_S(t) \quad , \tag{12}$$

where

$$S_t = S_0 e^{x_S} \quad , \tag{13}$$

$$r_\$ = \text{ instantaneous, risk-less rate} \quad , \tag{14}$$

$$r_S = \text{ stock-borrow/loan-fee rate} \quad , \tag{15}$$

$$\sigma_S = \text{ stock volatility} \quad . \tag{16}$$

The superweights are determined by ensuring that the following three analytic solutions of the pricing equation, the Black-Scholes-Merton equation, are numerically propagated exactly over every time step on our lattice:

$$\text{zeroth moment: cash} \quad Ke^{-r_{\$}(T-t)} \quad , \tag{17}$$

$$\text{first moment: stock} \quad S_t e^{-r_{\$}(T-t)} e^{\left(r_{\$}-r_S\right)(T-t)} \quad , \tag{18}$$

$$\text{second moment: stock-squared} \quad S_t^2 e^{-r_{\$}(T-t)} e^{\left(2r_{\$}-2r_S+\sigma_S^2\right)(T-t)} \quad . \tag{19}$$

By contrast, the collection of zero coupon bonds of all maturities in the Hull-White model,

$$B_T^{\mathrm{HW}}(x,t) = \exp\left\{-a_T(t)\,x - b_T(t) - \int_t^T ds\, f(s)\right\} \quad , \tag{8}$$

$$a_T(t) = \frac{1}{k}\left(1-e^{-k(T-t)}\right) \quad , \tag{9}$$

$$b_T(t) = \frac{1}{2}\nu^2(t)\,a_T^2(t) \quad . \tag{10}$$

may be viewed as forming a continuous set of moments,

$$B_T^{\mathrm{HW}}(x,t) \approx e^{-a_T(t)x} \sim S^{a_T(t)} \quad , \tag{20}$$

$$S = e^{-x} \quad . \tag{21}$$

Thus, without singling out any particular maturities (*moments*) we can only hope, at best, to make many – if not all – moments propagate on the lattice approximately correctly.

The stability and convergence of numerical solutions are properly investigated using von Neumann stability mode analysis. Applying this analysis to the numerical pricing of equity derivatives using superweights results in a constraint on the sizes of temporal- and state-variable (ie stock-return) grid steps.

In the Hull-White model we have an analogous situation. A relationship between temporal- and state variable intervals in the lattice is still an important result of the stability analysis. In this case, luckily, the analysis of the propagation of Fourier modes, parametrised by p

$$e^{ipx} \tag{22}$$

on the lattice, per the Von Neumann stability analysis recipe, is equivalent to an analysis of the propagation of Hull-White zeros of arbitrary maturity, ie

$$e^{ipx} \sim e^{-a_T(t)x} \quad . \tag{23}$$

Explicitly, one examines the properties of numerical propagation of a Hull-White zero of arbitrary tenor across one time step, ie one application of the diffusion operator, and matches the result to the desired analytic result. The way one performs this matching is by Taylor expanding the *ratio* of the numerical result and the analytic formula of the Hull-White zero, Eq (8). This analysis is precisely equivalent to the prescription of Von Neumann, which

requires an analysis of the amplitude of numerically propagated Fourier modes. The natural parameter of this expansion is

$$a_T(t)\,\Delta x \quad , \tag{24}$$

where Δx is the grid spacing of the state variable, x, defined in Eq (6). This expansion parameter corresponds to

$$p\Delta x \quad , \tag{25}$$

which is the natural parameter of the Von Neumann analysis. To see this equivalence, one needs to consider the propagation of Fourier modes on the lattice, and then compare these to the exact Fourier mode propagation. Luckily however, if one allows the function $a_T(t)$ to take complex values, the Hull-White solution itself describes propagation of a Fourier mode. We then compare this to the result of one application of the numerical diffusion operator,

$$\begin{aligned} &e^{\Phi(t)_{\text{numerical}}} \times e^{ipx} \quad , \\ &e^{\Phi(t)_{\text{numerical}}} = \text{numerical amplification factor} \quad . \end{aligned} \tag{26}$$

The expansion parameter, $a_T(t)\,\Delta x$, which, heuristically, we may think of as $T\Delta x$, turns out to demonstrate that the stability expansion parameter is

$$\sigma\, a_T(t) \sqrt{\frac{T}{N}} \quad , \tag{27}$$

where

$$T = \text{tenor} \quad , \quad N = \text{number of (uniform) time steps} \quad . \tag{28}$$

This analysis is detailed explicitly below. The principal result is that as $k \to 0$ (see Eq (9) for $a_T(t)$ above), very large tenor zeros correspond to high-frequency modes which do not necessarily accurately propagate on a finite-difference lattice. The only way that we can ensure fast convergence to a pre-specified accuracy for all values of k – particularly very small values – is to match higher and higher order Taylor terms, ie moments of the numerical distribution, and this can only be done by increasing the number of weights in the diffusion operator. This takes us from the 3–1 fully explicit method to 3–3 Crank-Nicolson and on to 5–3, 5–5, 7–1, 7–3 molecules, etc.

More importantly, when modelling option-embedded bonds – our ultimate goal – the kinks in the bond's payout curve caused by optionality also introduce high-frequency modes, regardless of the input value of k. This again forces us to consider diffusion methods of larger molecules.

Finally, we note, as an aside, that one *can* accurately propagate a large tenor zero bond using, say, the 3–1 fully explicit method by selecting a large enough value of the mean-reversion constant k. This has the effect of dampening the error terms that cause instability (see Section (4.4.2), Eq (77)). The large k technique is useless, however, since it corresponds to a reversion speed that approaches infinity as the tenor increases. Clearly, in order to sensibly evaluate large tenor bonds, the numerical algorithm must change, not the input values.

4.2 Transformation from the Hull-White Equation to the Heat Equation

In order to obtain the heat equation from the Hull-White pricing equation, begin with the transformation

$$B_T(x,t) = X_T(x,t)\, B^{\mathrm{HW}}_{T^*}(x,t) \quad . \tag{29}$$

Here, the function $X_T(x,t)$ is parametrised by the bond's maturity T, and the multiplicative Hull-White zero $B^{\mathrm{HW}}_{T^*}(x,t)$ has a different maturity T^*, which is taken to be a free parameter. We will use this transformation to implement a fully explicit finite-difference method, and we will choose this free parameter to be the grid time-step end, $T^* = t_{\mathrm{end}}$. Note that this choice effectively means we are using a different transformation for each time step. While this might seem unnecessarily complicated, it actually greatly simplifies the analysis of the properties of the numerical solutions.

Proceeding with the transformation to the heat equation, plug Eq (29) into Eq (7) and simplify to get a differential equation in X_T:

$$0 = \frac{\partial X_T(x,t)}{\partial t} + \left(\nu^2(t) - kx - \sigma^2 a_{T^*}(t)\right)\frac{\partial X_T(x,t)}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 X_T(x,t)}{\partial x^2} \quad . \tag{30}$$

Making the variable transformation

$$t = t_y \tag{31}$$

$$x = g_{T^*}(t_y)\, y \; + h_{T^*}(t_y) \tag{32}$$

$$\Rightarrow \quad X_T(x,t) \to \hat{X}_T(y,t_y) \quad , \tag{33}$$

leads us to the heat equation,

$$0 = \frac{\hat{X}_T(y,t_y)}{\partial t_y} + \frac{\sigma^2}{2g^2_{T^*}(t_y)}\frac{\partial^2 \hat{X}_T(y,t_y)}{\partial y^2} \quad , \tag{34}$$

where the functions $g_{T^*}(t_y)$ and $h_{T^*}(t_y)$ must satisfy

$$g_{T^*}(t_y) = \frac{1}{1 - k a_{T^*}(t)} \quad , \tag{35}$$

$$h_{T^*}(t_y) = -a_{T^*}(t_y)\,\nu^2(t_y) \quad . \tag{36}$$

with $\nu^2(t_y)$ and $a_{T^*}(t_y)$ given in Eqs (4) and (9), respectively.

The instantaneous variance,

$$\Sigma^2(t_y)\, dt_y \equiv \frac{\sigma^2}{g^2_{T^*}(t_y)} dt_y \quad , \tag{37}$$

is for the variance of a normal distribution. As such – looking forward to the finite-difference methods – we will choose our grid mesh in the (y, t_y)-coordinates to be uniform in y (but not necessarily in t_y) in order to guarantee convergence to a normal distribution over y by way of the central limit theorem.

We then demonstrate a numerical solution to this heat equation, Eq (34), in its local coordinates, $(y,\, t_y)$, by using a fully explicit finite-difference method. Note however, after every time step we revert back to the global (x,t)-coordinates in order to get the bond's pricing curve at that time.

This time-step dependent variable transformation is convenient for three reasons: First, we have the bond's price curve in its "natural coordinates" so that we may test embedded call- or put options easily. Second, reverting to the natural coordinates at each step allows us to inspect the evolving bond curve (by graphing it, for example) which greatly simplifies implementation and diagnosis of the model. Third, this algorithm is bond specification independent and allows us to easily make observations about the properties of the diffusion operator, ie using Von Neumann stability analysis, that are derivative-contract independent.

4.3 A Numerical Solution of the Heat Equation

We are now in a position to solve Eq (34) with the fully explicit 3–1 finite-difference method. The general method will be to start our solution algorithm at the known payout of an interest-rate derivative and diffuse – iteratively move backwards one time step at a time – back to today's value.

This method hinges on the analysis of the properties of a numerical diffusion operator for each time step. Hence, suppose that $t_{i+1} = t_{\text{end}}$ is the end of a time step that is midway through the diffusion process. At this time, we have a *uniform* set of x values, x^j_{i+1}, also called a *vector* or a *slide*. Over this slide there is a discretised derivative price curve,

$$B_T\left(x^j, t_{i+1}\right) = B^j_{T,\,i+1} \quad , \tag{38}$$

that satisfies Eq (7). We now proceed to find the numerical solution at $t_i = t_{\text{start}}$.

At the beginning of the time step algorithm (and away from the boundaries, which we will address later), we have that

$$t_{i+1} = t_{\text{end}} \quad , \tag{39}$$

$$X^j_{i+1} = B^j_{T,\,i+1} \quad , \tag{40}$$

$$y^j_{i+1} = x^j_{i+1} \quad , \tag{41}$$

$$\hat{X}^j_{i+1} = X^j_{i+1} \quad . \tag{42}$$

In other words, at t_{end} we map the natural coordinates (x,t) identically onto the local coordinates (y, t_y), hence Eqs (39) and (41). Also at t_{end}, the transformation in Eq (29) yields

$$B_T\left(x, t_{\text{end}}\right) = X_T\left(x, t_{\text{end}}\right) B^{\text{HW}}_{t_{\text{end}}}(x, t_{\text{end}}) = X_T\left(x, t_{\text{end}}\right) \quad , \tag{43}$$

which leads to Eq (40). Switching Eq (40) to the local coordinates leads to Eq (42).

At the end of the algorithm time step (ie at the chronological beginning of the time step)

we have that

$$t_i = t_{\rm start} \quad , \tag{44}$$

$$\hat{X}_i^j = p^{j+1}\hat{X}_{i+1}^{j+1} + p^j \hat{X}_{i+1}^j + p^{j-1}\hat{X}_{i+1}^{j-1} \qquad \text{(the } p\text{'s are weights)} \quad , \tag{45}$$

$$y_i^j = y_{i+1}^j \quad , \tag{46}$$

$$x_i^j = g_{i+1,\,i}\, y_i^j + h_{i+1,i} \qquad \left(g_{t_{\rm end}}(t_{\rm start}) = g_{i+1,\,i}\,, \text{ etc; see Eqs (9) \& (10)}\right) \quad , \tag{47}$$

$$X_i^j \underset{x\leftarrow y}{\longleftarrow} \hat{X}_i^j \quad , \tag{48}$$

$$B_i^j = X_i^j\, B_{i+1,\,i}^{{\rm HW},\,j} \qquad \left(B_{t_{\rm end}}^{\rm HW}(x^j, t_{\rm start}) = B_{i+1,\,i}^{{\rm HW},\,j}\ ;\ \text{see Eq (8)}\right) \quad . \tag{49}$$

The three p's are the weights of the 3–1 finite-difference diffusion process. They are chosen by imposing convergence and stability on the solution – which leads to the familiar choice of[5, 6, 7],

$$\left(p^{j+1},\, p^j,\, p^{j-1}\right) = \left(\frac{1}{6},\, \frac{2}{3},\, \frac{1}{6}\right) \quad . \tag{50}$$

Section 4.4.2 will explore the derivation of these weights and their associated stability properties in detail.

Note that at the algorithmic beginning of each time step, we start with a uniformly spaced set of x's. Next, the procedure defines a local set of uniform y's (see Eq (41)) and then diffuses along these to $t_{\rm start}$, where the result is then transformed back to the global coordinate x. At the algorithmic end of the time step, plots of x versus t and y versus t would look like Figure 1. (For the plots, $t_{\rm end} - t_{\rm start} = 10$ years, $\sigma = 10\%$, and $k = 0.1$. These values were chosen to emphasise the differences between the x and y coordinates.)

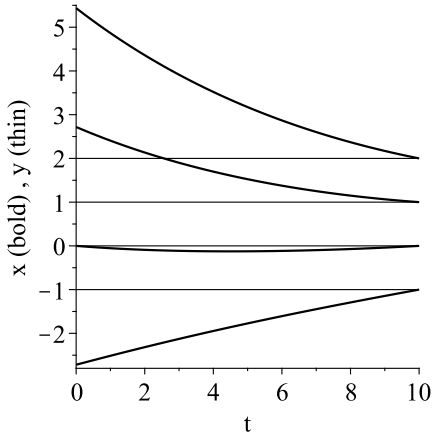

Figure 1: x-coordinate drift for one time step.

Clearly, at the algorithmic end of the time step, the values of the $x_{\rm start}$'s are still uniformly spaced. But in order to iterate the diffusion process, the grid spacing needs to be optimised according to a stability analysis (see Section 4.4 below). With the stability condition on the grid spacing known, we can simply repartition the x slide to have the proper spacing. This repartitioning will clearly require interpolation into the slide to construct a new slide. As an

aside, it is also the case that the practitioner may need to re-centre the slide anyway, and so this interpolation is not extraneous. (Grid re-centring is often done near to gamma-source kinks that are caused by optionality[5]. See next paragraph.)

The last thing to mention about the algorithm is the boundary conditions. Firstly, we use analytic Hull-White bond prices at the boundaries. This is indeed possible if the slides are big enough. The size of the slides is determined by the required accuracy tolerance of the practitioner and will be a certain multiple of standard deviations on either side of all future gamma sources. Secondly, we implement American-style choices by maximising or minimising the various optionality embedded in the derivative. It is known that numerical accuracy can be improved by carefully offsetting grid points from the kinks, or gamma sources, introduced by these choices[5].

4.4 Von Neumann Stability Analysis and Stiffness

In the above sections, we mentioned several times that uniform spacing of the coordinates at algorithm time t_{end} depends on a stability condition. In this section, we will briefly analyse the stability for the 3–1 diffusion algorithm given in the previous section. In Section 4.4.1 we focus on the simple heat equation, and in Section 4.4.2, we extend the analysis to the Hull-White equation. Moreover, we introduce and describe the stiffness instability of the 3–1 diffusion algorithm.

4.4.1 Stability of the 3–1 Solution of the Heat Equation

Recall our heat equation of Section (4):

$$0 = \frac{\hat{X}_T\left(y, t_y\right)}{\partial t_y} + \frac{\Sigma^2\left(t_y\right)}{2}\frac{\partial^2 \hat{X}_T\left(y, t_y\right)}{\partial y^2} \quad . \tag{34}$$

In an approximate discrete form, with (y, t_y)-space labelled by (j, i), this becomes

$$0 \approx \frac{\hat{X}^j_{T,\,i+1} - \hat{X}^j_{T,\,i}}{\Delta t} + \frac{\left(\Sigma^2\right)^j_{i+1}}{2}\left(\frac{\hat{X}^{j+1}_{T,\,i+1} - 2\hat{X}^j_{T,\,i+1} + \hat{X}^{j-1}_{T,\,i+1}}{\left(\Delta y\right)^2}\right) \quad . \tag{51}$$

The Δt and Δy define the grid spacing for the lattice. The above can be written as

$$\hat{X}^j_i \approx \hat{p}^{j+1}\hat{X}^{j+1}_{T,\,i+1} + \hat{p}^j\hat{X}^j_{T,\,i+1} + \hat{p}^{j-1}\hat{X}^{j-1}_{T,\,i+1} \quad , \tag{52}$$

where the $\hat{p}$'s are the finite-difference weights in the (y, t_y)-space and are given by

$$\begin{aligned} \hat{p}^{j+1} = \hat{p}^{j-1} &= \frac{\left(\Sigma^2\right)^j_{i+1}}{2}\frac{\Delta t}{\left(\Delta y\right)^2} \quad , \\ \hat{p}^j &= 1 - \hat{p}^{j+1} - \hat{p}^{j-1} \quad . \end{aligned} \tag{53}$$

Given a set of boundary conditions, Eq (52) generates an explicit-method, iterative algorithm for diffusing backwards in time from $t_{i+1} = t_{\text{end}}$ to obtain a numerical solution to the heat equation at $t_i = t_{\text{start}}$.

It is well known[6] that the lattice intervals Δt and Δy cannot be independent of one another – they must satisfy a stability condition. In fact, it can be shown by Von Neumann stability analysis[6], along with a convergence optimisation[5], that the ratio of the local variance to the squared interval $(\Delta y)^2$ optimally satisfies

$$\frac{(\Sigma^2)^j_{i+1}\,\Delta t}{(\Delta y)^2} = \frac{1}{3} \quad . \tag{54}$$

The above can be obtained by propagating a Fourier mode, $\exp(ipy_j)$, of mode number p, along the lattice. Stability is guaranteed if the mode propagates with an amplitude of modulus less than one, which leads to

$$\frac{(\Sigma^2)^j_{i+1}\,\Delta t}{(\Delta y)^2} < \frac{1}{2} \quad . \tag{55}$$

A further requirement that as $\Delta y \to 0$, the propagation amplitude converges to the Fourier transform of a Gaussian distribution, yields[5] Eq (54). With the above stability condition in Eq (54), the weights take their familiar form,

$$\left(\hat{p}^{j+1},\, \hat{p}^{j},\, \hat{p}^{j-1}\right) = \left(\frac{1}{6},\, \frac{2}{3},\, \frac{1}{6}\right) \quad . \tag{56}$$

4.4.2 Stability of the 3–1 Solution of the Hull-White Equation

It turns out, rather remarkably, that a similar "mode analysis" can be done to test solutions of our pricing equation,

$$0 = \frac{\partial B_T(t)}{\partial t} + \left(\nu^2(t) - kx\right)\frac{\partial B_T(t)}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 B_T(t)}{\partial x^2} - (x + f(t))\,B_T(t) \quad . \tag{7}$$

In this case, however, the "mode" to be tested is the Hull-White zero,

$$B^{\mathrm{HW}}_T(x,t) = \exp\left\{-a_T(t)\,x - b_T(t) - \int_t^T ds\, f(s)\right\} \quad , \tag{8}$$

of "mode number"

$$p = i a_T(t) \qquad \left(i = \sqrt{-1}\right) \quad , \tag{57}$$

and of maturity T.

Using the transformations of Eqs (29), (31) and (32), the finite-difference algorithm of Eq (52) can be converted into a finite-difference algorithm for the full differential equation. With $(j,\, i)$ now labelling the "natural coordinates" $(x,\, t)$, the 3–1 finite-difference version of Eq (7) becomes

$$B^j_{T,\,i} \approx \left(p^{j+1}B^{j+1}_{T,\,i+1} + p^j B^j_{T,\,i+1} + p^{j-1}B^{j-1}_{T,\,i+1}\right) B^{\mathrm{HW},\,j}_{i+1,\,i} \quad , \tag{58}$$

where, as before, the discrete Hull-White zero over the time step is given by

$$B^{\mathrm{HW},\,j}_{i+1,\,i} = B^{\mathrm{HW}}_{t_{\mathrm{end}}}\left(x^j, t_{\mathrm{start}}\right) \quad . \tag{59}$$

The goal will be to determine both the weights and the stability condition for the lattice spacing. As we will see, and perhaps not surprisingly, the weights and stability condition will be the same as for the heat equation. Nevertheless, the method is still instructive, as we will then use it to find weights and stability for "larger molecule" algorithms, eg 7–3 diffusion.

Furthermore, this technique will automatically introduce a further stability condition that is in addition to Eq (54). As we will show, the maturity T above is not entirely arbitrary. Of course, it will be equal to the bond maturity or option expiration of the particular derivative that one wants to model. More importantly, however, this maturity will be subject to a stability condition that will relate it to the input short-rate volatility σ of Eq (1).

Proceeding with the Hull-White mode analysis, we insert a Hull-White zero, Eq (8), into Eq (58) to obtain

$$\begin{aligned} e^{-a_T(t_i)x_i^j - b_T(t_i)} \approx\ & e^{-a_T(t_{i+1})x_{i+1}^j - b_T(t_{i+1})} \times e^{-a_{i+1}(t_i)x_{i+1}^j - b_{i+1}(t_i)} \\ & \times \left(p^{j+1} e^{-a_T(t_{i+1})\Delta x_{i+1}} + p^j + p^{j-1} e^{+a_T(t_{i+1})\Delta x_{i+1}} \right) \quad . \end{aligned} \tag{60}$$

A few comments are needed about the above equation. First, since the integrated forward rates curve,

$$\int_t^T ds\, f(s) \quad , \tag{61}$$

does not depend on x or σ, we have dropped it from the analysis without a loss of generality. This is equivalent to having a constant forward rate over the time step. Thus, we are effectively propagating the "mode"

$$e^{-a_T(t)x - b_T(t)} \quad . \tag{62}$$

Second, recall that, because of Eq (32) and Figure (1), the grid spacing in x is *not* constant over a time step. For this reason, the x's in the above expression are given time labels, the superscript i's.

Third, notice that there are two Hull-White zeros on the right-hand side of the equation. The first, of maturity T, is the Hull-White mode that is being propagated to probe stability. The second, of maturity $t_{i+1} = t_{\text{end}}$, is the time-step discount factor of transformation Eq (29), with $T^* = t_{\text{end}}$.

The next step in the analysis is to write everything on the right-hand side of Eq (60) in terms of $t_i = t_{\text{start}}$ and x_i. To do this, recall that at t_{end}, the local y variable was defined so that

$$x_{\text{end}} = y_{\text{end}} \quad . \tag{63}$$

Using that along with Eq (32) and

$$t_{i+1} = t_i + \Delta t \quad , \tag{64}$$

we can find a relationship between $x_{\text{end}} = x_{i+1}$ and $x_{\text{start}} = x_i$:

$$x_{i+1}^j = x_i^j e^{-k\Delta t} + \frac{\sigma^2 e^{-k\Delta t}}{2k^2} \left(1 - e^{-k\Delta t}\right) \left(1 - e^{-2kt_i}\right) \quad . \tag{65}$$

A further substitution to the right-hand side of Eq (60) comes from the stability condition for the standard deviation and the squared interval $\left(x_{i+1}\right)^2$. First, the average standard deviation over the time step is found to be, using Eq (37),

$$\Sigma^2_{\text{avg}}\Delta t = \int_{t_y=t_{\text{start}}}^{t_y=t_{\text{end}}} dt_y \frac{\sigma^2}{g^2_{t_{\text{end}}}(t_y)} = \frac{\sigma^2}{2k}\left(1-e^{-2k\Delta t}\right) \quad . \tag{66}$$

By requiring that this standard deviation has a to-be-determined stability condition,

$$\frac{\Sigma^2_{\text{avg}}\Delta t}{\left(\Delta x_{\text{end}}\right)^2} = \Pi \quad , \tag{67}$$

we obtain

$$\left(\Delta x^j_{i+1}\right)^2 = \frac{\sigma^2}{2k\Pi}\left(1-e^{-2k\Delta t}\right) \quad . \tag{68}$$

Finally, inserting Eqs (64), (65), and (68) into the right-hand side (RHS) of Eq (60), and expanding about $\Delta t = 0$, we find that (with the assistance of Maple™ 11)

$$\text{RHS} \approx e^{-a_T\left(t_i\right)x^j_i - b_T\left(t_i\right)}\left(1+\epsilon_{\frac{1}{2}}\left(\Delta t\right)^{\frac{1}{2}}+\epsilon_1\left(\Delta t\right)^1+\epsilon_{\frac{3}{2}}\left(\Delta t\right)^{\frac{3}{2}}+\epsilon_2\left(\Delta t\right)^2+...\right) \quad , \tag{69}$$

where the coefficients ϵ_n are error terms and are functions of everything,

$$\epsilon_n = \epsilon_n\left(p\text{'s}, \Pi, \sigma^2, k, x_{\text{start}}, t_{\text{start}}, T\right) \quad . \tag{70}$$

Using Eq (69), rewrite Eq (60) as

$$0 \approx \epsilon_{\frac{1}{2}}\left(\Delta t\right)^{\frac{1}{2}}+\epsilon_1\left(\Delta t\right)^1+\epsilon_{\frac{3}{2}}\left(\Delta t\right)^{\frac{3}{2}}+\epsilon_2\left(\Delta t\right)^2+... \quad . \tag{71}$$

Imposing normalised, centred weights,

$$p^j = 1 - p^{j+1} - p^{j-1} \quad , \tag{72}$$

$$p^{j+1} = p^{j-1} \quad , \tag{73}$$

in Eq (71) eliminates the error terms of fractional power and greatly simplifies the remaining ones. Normalised, centred weights are fundamental to the analysis presented in this paper and *always* eliminate error terms in fractional powers of the expansion parameter, labelled above using a fraction in the subscript. These terms are then effectively dropped.

After the substitution of normalised centred weights, one sees that the choice

$$p^{j+1} = p^{j-1} = \frac{1}{2}\Pi \quad , \tag{74}$$

eliminates the first error term, ϵ_1. Subsequently, one also sees that the choice

$$\Pi = \frac{1}{3} \quad , \tag{75}$$

eliminates the second term, ϵ_2. Thus, we recover the same weights and stability condition as we had for the heat equation:

$$\left(\Pi;\, p^{j+1},\, p^{j},\, p^{j-1}\right) = \left(\frac{1}{3};\, \frac{1}{6},\, \frac{2}{3},\, \frac{1}{6}\right) \quad . \tag{76}$$

Furthermore, with the Von Neumann stability and the weights given as in Eq (76), the third error term becomes

$$\epsilon_3 \left(\Delta t\right)^3 = -\frac{1}{120} \left(\sigma\, a_T(t_i)\right)^6 \left(\Delta t\right)^3 \quad , \tag{77}$$

where, from before,

$$a_T\left(t\right) = \frac{1}{k}\left(1 - e^{-k(T-t)}\right) \quad . \tag{9}$$

This is the leading proportional error that is added to the Hull-White model after each algorithmic time step. Note that, because of the dependence on $a_T\left(t\right)$, each successive error term *grows* as the algorithm steps back to today. For N steps from maturity to today, the cumulative error of the model, E, is

$$\begin{aligned} E &\approx -\frac{1}{120} \sum_{i=1}^{N} \left(\sigma\, a_T(t_{N-i})\right)^6 \left(\Delta t_{N-i}\right)^3 + O\left(\left(\Delta t\right)^4\right) \quad , \\ t_N &= T \quad , \quad t_{N-i} = t_{N-i+1} - \Delta t_{N-1} \quad , \quad t_0 = 0 \quad \text{(today)} \quad , \end{aligned} \tag{78}$$

where the Δt_i are the (possibly varying) algorithmic time steps.

By inspection, the maximum value of $a_T\left(t\right)$ is the maturity T. It follows that as we diffuse back to today, the magnitude of the time-step error, Eq (77), has the upper bound

$$\left|\epsilon_3 \left(\Delta t\right)^3\right| = \frac{1}{120} \left(\sigma\, a_T(t_i)\right)^6 \left(\Delta t\right)^3 \leq \frac{1}{120} \left(\sigma\, T\right)^6 \left(\Delta t\right)^3 \quad , \tag{79}$$

After N steps of uniform time step

$$\Delta t = \frac{T}{N} \quad , \tag{80}$$

the upper bound of the cumulative error is

$$\left|E\right|_{3-1} = \frac{N}{120} \left(\sigma\, T\right)^6 \left(\Delta t\right)^3 = \frac{N}{120} \left(\frac{\sigma^2\, T^3}{N}\right)^3 \quad . \tag{81}$$

Considering the value of N as fixed (ie not directly chosen by the user of the model) it is clear that the product $\sigma^2\, T^3$ is a measure of the the model's stiffness. In other words, even if the model has Von Neumann stability, as the above does, it is still possible to have model instability if $\sigma^2\, T^3$ is too large. In fact, it can be shown (see Section 5.1, for example) that for a diffusion algorithm using a molecule of size n–m, the maximum cumulative error boundary will have the form

$$\begin{aligned} \left|E\right|_{n-m} &= \alpha_{n-m} N \left(\frac{\sigma^2\, T^3}{N}\right)^{\beta_{n-m}} \quad , \\ \alpha_{n-m} &= \text{numerical coefficient of Taylor expansion} \quad , \\ \beta_{n-m} &= \text{power of Taylor expansion} \quad . \end{aligned} \tag{82}$$

Therefore, the product $\sigma^2\,T^3$ is the *relative* stiffness measure for *all* molecules being used to diffuse Hull-White zeros. Note that it is not an absolute measure of stiffness since the error boundary for each n–m molecule is also sensitive to the coefficient α_{n-m}. Molecules with a smaller coefficient can withstand a larger value of $\sigma^2\,T^3$.

As a specific example of stiffness, we model a bond of maturity $T = 50$ years using even time steps of $\Delta t = 0.25$ years – not an unusual case. Clearly, we cannot arbitrarily choose the short rate volatility σ. In fact, choosing the short rate volatility to be two hundred basis points (large but not unreasonable),

$$\sigma = 0.02 \quad , \tag{83}$$

causes the upper bound of the cumulative error to be

$$|E| \approx 2.6 \times 10^{-2} \quad , \tag{84}$$

which is unacceptably large. Clearly, this value of the volatility allows the model to only handle bonds of much shorter tenor. If we choose to have a zero bond model with a maximum cumulative error bound of

$$|E| \leq 10^{-5} \quad , \tag{85}$$

for example, then the 3–1 diffusion with the above rates volatility and time steps ($\sigma = 0.02$, $N = 200$) can at most model zeros with tenor

$$T \approx 20.8 \text{ years} \quad , \tag{86}$$

as one can see in Figure 2.

If we allow both the tenor and the volatility to vary, but keep the number of time steps and the desired maximum error bound the same, then we find that the 3–1 molecule works in the regime

$$\sigma^2\,T^3 \leq 3.6 \quad . \tag{87}$$

Alternatively one can vary the number of time steps. In order to model a zero of tenor $T = 50$ years with a short-rate volatility $\sigma = 0.02$, one would need to use a rather large number of time steps,

$$N \approx 10200 \quad . \tag{88}$$

Clearly this model would be too computationally slow to get the desired accuracy – it is too stiff to be practical. Accordingly, the 3–1 diffusion fails to efficiently model long-dated bonds when the volatility is large.

5 Diffusion Operators with More Weights

From the above mode analysis, it is clear that basic 3–1 diffusion is simply not robust enough to model long-dated bonds with a relatively large standard deviation of the short rate. To overcome this limitation, we will generalise these results to any type of n–m diffusion algorithm in this section.

In particular, we will see that 7–1 fully explicit diffusion and 7–3 modified Crank-Nicolson diffusion have excellent convergence and stability properties and allow rather large input

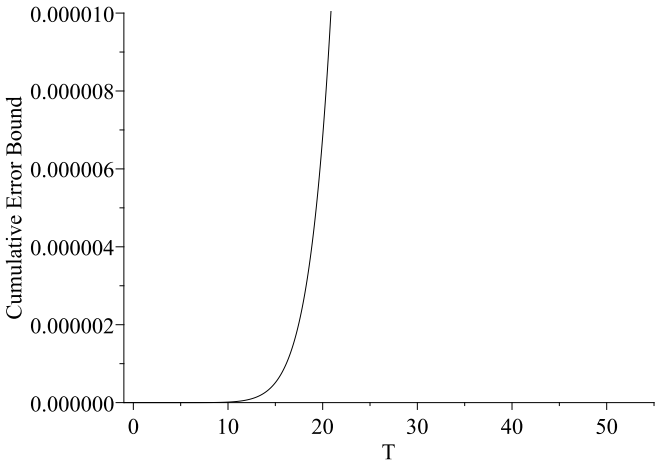

Figure 2: Upper bound of the 3–1 diffusion cumulative error magnitude versus tenor, with $\sigma = 0.02$ and $N = 200$ time steps.

values for the standard deviation of the Hull-White model. Put another way, most "lesser" algorithms (eg: 3–1, 3–3, 6–1) have errors that blow up for large but reasonable standard deviation inputs.

As a specific example, our goal might be to find the diffusion molecule that allows us to model a bond with parameters

$$\sigma = 0.02 \quad , \quad T = 50 \text{ years} \quad , \tag{89}$$

such that for 200 uniform times steps, the upper bound of the cumulative error's magnitude is

$$|E| \leq 10^{-5} \quad . \tag{90}$$

If we can constrain the upper bound of the error for this rather restrictive situation then we have confidence in the model for larger N or smaller σ.

5.1 Von Neumann Analysis and Stiffness for n–m Diffusion

Before proceeding to the larger molecules, first note that fully implicit finite difference diffusion methods, ie 1–3, 2–4, etc, produce the exact same error structure as the fully explicit methods. Therefore, we will restrict our development to fully explicit methods, ie 3–1, 5–1, etc, and (modified) Crank-Nicolson methods, ie 3–3, 4–2, etc.

The general method for n–m diffusion methods is just a simple generalisation of the 3–1 method outlined in Section 4.4.2. Recall that in that case, we analysed Eq (60),

$$\begin{aligned} e^{-a_T(t_i)x_i^j - b_T(t_i)} \approx\ & e^{-a_T(t_{i+1})x_{i+1}^j - b_T(t_{i+1})} \times e^{-a_{i+1}(t_i)x_{i+1}^j - b_{i+1}(t_i)} \\ & \times \left(p^{j+1} e^{-a_T(t_{i+1})\Delta x_{i+1}} + p^j + p^{j-1} e^{+a_T(t_{i+1})\Delta x_{i+1}} \right) \quad , \end{aligned} \tag{60}$$

which is the 3–1 diffusion of the Hull-White "mode" (zeros of all maturities),

$$e^{-a_T(t)x - b_T(t)} \quad . \tag{91}$$

To generalise, simply insert the appropriate weight terms on both sides of Eq (60). For example, the standard 3–3 Crank-Nicolson diffusion would be

$$\begin{aligned} & e^{-a_T(t_i)x_i^j - b_T(t_i)} \times \left(q^{j+1} e^{-a_T(t_i)\Delta x_i} + q^j + q^{j-1} e^{+a_T(t_i)\Delta x_i} \right) \\ & \approx \\ & e^{-a_T(t_{i+1})x_{i+1}^j - b_T(t_{i+1})} \times e^{-a_{i+1}(t_i)x_{i+1}^j - b_{i+1}(t_i)} \\ & \quad \times \left(p^{j+1} e^{-a_T(t_{i+1})\Delta x_{i+1}} + p^j + p^{j-1} e^{+a_T(t_{i+1})\Delta x_{i+1}} \right) \quad , \end{aligned} \tag{92}$$

where the q's are the weights at $t_i = t_{\text{start}}$ and, as before, the p's are the weights at $t_{i+1} = t_{\text{end}}$. To simplify the discussion, rewrite Eq (92) as

$$\text{LHS}\,(t_i) \approx \text{RHS}\,(t_{i+1}) \quad . \tag{93}$$

Then, as in Section 4.4.2, use Eqs (64), (65), and (68),

$$t_{i+1} = t_i + \Delta t \quad , \tag{64}$$

$$x_{i+1}^j = x_i^j e^{-k\Delta t} + \frac{\sigma^2 e^{-k\Delta t}}{2k^2} \left(1 - e^{-k\Delta t}\right) \left(1 - e^{-2kt_i}\right) \quad , \tag{65}$$

$$\left(\Delta x_{i+1}^j\right)^2 = \frac{\sigma^2}{2k\Pi} \left(1 - e^{-2k\Delta t}\right) \quad , \tag{68}$$

to rewrite all of the functions in $\text{RHS}\,(t_{i+1})$ in terms of t_i and x_i. After a bit of algebra (using Maple™ 11) we find that

$$\frac{\text{RHS}\,(t_{i+1})}{\text{LHS}\,(t_i)} - 1 \approx \epsilon_{\frac{1}{2}} (\Delta t)^{\frac{1}{2}} + \epsilon_1 (\Delta t)^1 + \epsilon_{\frac{3}{2}} (\Delta t)^{\frac{3}{2}} + \epsilon_2 (\Delta t)^2 + ... \tag{94}$$

As before, choosing normalised and centred weights,

$$p^j = 1 - p^{j+1} - p^{j-1} \quad , \tag{95}$$

$$q^j = 1 - q^{j+1} - q^{j-1} \quad , \tag{96}$$

$$p^{j+1} = p^{j-1} \quad , \quad q^{j+1} = q^{j-1} \quad , \tag{97}$$

eliminates the error terms of fractional power and simplifies the others. Then, successively setting

$$\epsilon_1, \epsilon_2, \epsilon_3 = 0 \quad , \tag{98}$$

allows us to determine all of the weights *and* the stability condition on the lattice. For illustration, they are

$$\Pi \approx 0.45 \quad , \tag{99}$$

$$p^j \approx 0.61 \quad , \tag{100}$$

$$p^{j+1} = p^{j-1} \approx 0.20 \quad , \tag{101}$$

$$q^j \approx 1.06 \quad , \tag{102}$$

$$q^{j+1} = q^{j-1} \approx -0.03 \quad . \tag{103}$$

Subsequently, the magnitude of the leading error term of one time step becomes

$$|\epsilon_4| \approx \left(9.2 \times 10^{-4}\right) \left(\sigma a_T\left(t\right)\right)^8 \left(\Delta t\right)^4 \leq \left(9.2 \times 10^{-4}\right) \left(\sigma T\right)^8 \left(\Delta t\right)^4 \quad . \tag{104}$$

For our test case with

$$\sigma = 0.02 \quad , \quad T = 50 \text{ years} \quad , \quad |E| \leq 10^{-5} \quad , \quad N = 200 \quad , \quad \Delta t = \frac{T}{N} \quad , \tag{105}$$

we find that the 3–3 diffusion fails since

$$|E|_{3-3} \approx \left(9.2 \times 10^{-4}\right) N \left(\frac{\sigma^2 T^3}{N}\right)^4 \approx 7.2 \times 10^{-4} \quad . \tag{106}$$

In fact, all else being the same, the 3–3 molecule can only model bonds of tenor

$$T \approx 35.0 \text{ years} \quad . \tag{107}$$

(In other words, the regime of (σ, T) is $\sigma^2 T^3 \leq 17.2$.)

Carrying on in this manner, we find that the 7–1 and 7–3 molecules satisfy our requirements since

$$|E|_{7-1} \approx 1.6 \times 10^{-6} \quad , \tag{108}$$

$$|E|_{7-3} \approx 1.4 \times 10^{-8} \quad . \tag{109}$$

This can be easily seen in Figure 3.

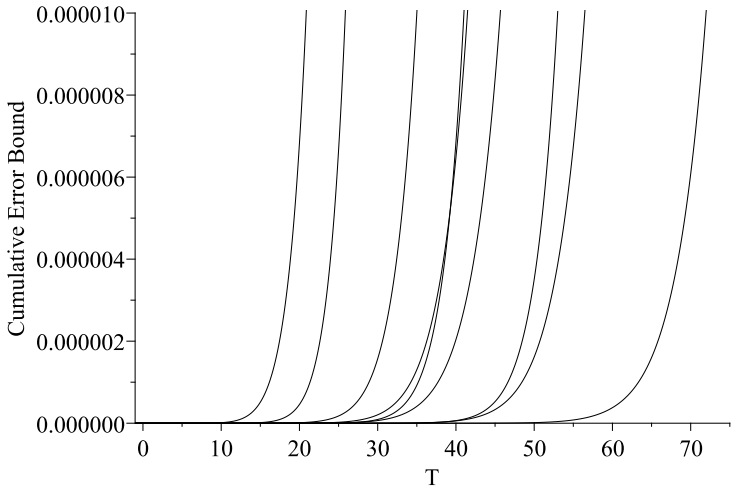

Figure 3: Error bounds for various molecules with $\sigma = 0.02$ and N=200 time steps. From left to right at the top: 3–1, 4–3, 3–3, 5–3, 6–1, 5–2, 5–5, 7–1, 7–3.

A few comments are needed about Figure 3. First, there are clearly many possible molecules that one could experiment with. We have just plotted a selection to capture the essence of how the cumulative error bounds grow with tenor, T.

Second, note that the error bounds for the 5–3 and 6–1 molecules cross one another as T increases. This is a common feature of these error bounds. Their positions relative to one another are highly dependent on the input short-rate volatility σ, the number of time steps N, and the desired maximum error (1×10^{-5} for this case). Changing any one of these values, ie adjusting the model's performance threshold, causes the arrangement order of the bounds to change, and some may even cross through one another.

Third, there are a few molecules, namely 4–1, 5–1, 6–2, etc, that we have purposely omitted from this plot because of their anomalous stability behaviour. Recall that one uses Von Neumann stability analysis to find both the set of diffusion weights and the value of Π, which represents optimised diffusion stability. (See the examples of 3–1 and 3–3 diffusion above.) The values of Π are usually found by finding the positive real roots that solve a polynomial equation. The optimal root is the one that minimises the error term of the diffusion. But, when performing the same stability analysis on the 5–1 diffusion, for example, one finds that there are no positive real roots for Π. In this case, one has to find a minimum positive value that minimises the error term. Note that this value need not be the global minimum. Also note that we have not observed any molecules with this anomalous error feature above the the n–2 set of molecules, though larger molecules with this type of error may exist. Moreover, the 2–2 molecule in particular has the curious feature of having no dependence on Π and is very unstable. See Table 1 below for a collection of all molecules used and omitted.

Fourth, from the graph, one sees that the 5–5 molecule also meets our modelling requirement, and in fact,

$$|E|_{5-5} \approx 3.5 \times 10^{-6} \quad . \tag{110}$$

Certainly, there are many other molecules that also satisfy our requirement. The point is that each practitioner must decide which of these allowable molecules (5–5, 7–1, 7–3, etc) to use. Comparing 5–5 to 7–1, for example, it would make more sense to use 7–1. The 5–5 molecule requires the inversion of a 5×5 matrix in order to solve the system at each time step. By contrast, the 7–1 molecule is just a simple explicit diffusion and thus is easier to implement and a faster model to compute. Furthermore, it has slightly better error-bound properties than the former. If the practitioner desires a model that has much better error properties than 7–1, however, then 7–3 is a possible choice. The trade-off for more stability in that case is that one has to invert a 7×7 matrix at every time step, which is computationally more expensive than an explicit diffusion model. Again, it is up to the individual to decided which quality is more important: a faster model or a more stable one.

5.1.1 Analysis of the Cumulative Error

As we have mentioned, when numerically propagating a Hull-White zero bond using a diffusion algorithm of molecule size n–m, the maximum cumulative error boundary will be of the form

$$\begin{aligned} |E|_{n-m} &= \alpha_{n-m} N \left(\frac{\sigma^2 \, T^3}{N} \right)^{\beta_{n-m}} \quad , \\ \alpha_{n-m} &= \text{numerical coefficient of Taylor expansion} \quad , \\ \beta_{n-m} &= \text{power of Taylor expansion} \quad . \end{aligned} \tag{82}$$

In our study of different diffusion molecules, we found the error boundaries for all of the molecules in Table 1.

2–1	3–1	~~4–1~~	~~5–1~~	6–1	7–1	~~8–1~~	~~9–1~~
~~2–2~~	3–2	4–2	5–2	~~6–2~~	7–2	8–2	9–2
	3–3	4–3	5–3	6–3	7–3	8–3	
		4–4	5–4	6–4	7–4	8–4	9–4
			5–5	6–5	7–5		
				6–6	7–6	8–6	9–6

Table 1: Molecules we investigated. Slashed entries represent those with anomalous error boundaries and were not used to estimate α and β.

Excluding the molecules with anomalous errors, it can be shown that the order of the Taylor expansion for each n–m molecule can be written as

$$\beta_{n-m} = \frac{1}{2}\left[\text{even}\,(n, n+1) + \text{even}\,(m, m+1)\right] \quad , \tag{111}$$

where the function $\text{even}(\cdot\,,\cdot)$ selects the even number of its pair of arguments. The above equation can be derived by assuming that all sets of molecules have normalised and symmetric weights, and then by counting all of the error terms in the Taylor expansion that will be eliminated by finding these weights and the associated Von Neumann stability condition of Eq (67).

Unfortunately, there does not seem to be an analytic way to describe the Taylor coefficient α_{n-m} as a function of an n–m molecule. The best that we can do is a least-squares plane fit. Using the errors of the molecules in Table 1 (except for those with anomalous errors), we find that

$$\ln \alpha_{n-m} \approx 2.98 - 1.82n - 0.74m \quad , \tag{112}$$

with a standard deviation

$$\text{fit deviation} \approx 2.73 \quad . \tag{113}$$

This fit tells us that the cumulative error bound for an n–m molecule is more sensitive to the size of n than to the size of m. In other words, if a practitioner wanted to decrease the error bound of an algorithm, the general technique would be to change the diffusion molecule from n–m to $(n+2)$–m. Note however, that the deviation of the plane fit is rather large. Accordingly, moving from n–m to n–$(m+2)$ might produce better accuracy than the former change. Here a bit of testing must be undertaken. It is useful to test several molecules, given the individuals tolerance requirements.

5.2 Optionality and higher-frequency modes

Thus far, we have only analysed the propagation of Hull-White zeros in modified Crank-Nicolson diffusion models. Of course, the main application of these models is to value option-embedded bonds (and their risk numbers). This raises the issue of what happens when a practitioner attempts to propagate a piece-wise continuous curve containing a strike kink on a lattice.

As we have observed, a particular n–m model has a regime of allowable tenor and standard deviations. (The practitioner will choose the model to suit the required tenor and deviations to be input.) Also we have observed, heuristically at least, that given the maximum allowable standard deviation for a chosen model, the tenor of a bond, T, is basically the largest frequency mode that can be propagated without exploding error.

For a bond with optionality, however, the strike kink will in fact introduce higher-order modes. In other words, in the at-the-money (ATM) range of the payout, there are bonds of tenor greater than T being propagated. To handle this properly, the practitioner must be careful. If the appropriate minimal number of modelling time steps is chosen, the higher modes are damped out and only marginally contribute to the overall error of the model. This can be empirical tested by testing convergence of the model. One way to do this is to model a particular American-style option-embedded bond on a particular n–m model, and then double the number of time steps and remodel. Another way is to run the model twice, increasing the length of the x-slide on the second run.

From our observations, an acceptable minimum number of time steps is about 200 (see Section 5.3, below), ie going from 200 time steps to 400 satisfies our convergence criteria. Also, our accepted x-slide length is five standard deviations – measured in local coordinates over a time step, see Eq (66) – on either side of a strike. Other practitioners may wish to use larger values to suit their purposes. We present these, qualitatively, as the minimum values that we trust. Smaller values will, in general, lead to higher modes of Hull-White zeros that do not get damped away fast enough.

5.3 Observations about n–m diffusion methods

As we have seen, the 7–1 fully explicit diffusion and the 7–3 modified Crank-Nicolson diffusion are two algorithms that pass our example requirement of valuing long dated bonds with a large short-rate volatility and all values of the mean reversion constant. They are relatively straight forward to implement, though one does need to invert a septa-diagonal matrix for the latter. This matrix inversion is straightforward and is accomplished by the generalisation of the tri-diagonal matrix inversion used in the standard 3–3 Crank-Nicolson method[6].

For bonds with embedded optionality, of various tenor or even American style, an efficient trick is to make the time step spacing chronologically grow exponentially in size. We have shown experimentally that this requires fewer time steps than an algorithm with uniform spacing requires. Heuristically, one can understand the reason for this as being that all the various embedded option tenors, from very short dated to the whole life of the bond, have a similar number of time steps, ensuring optimal advantage of the central limit theorem. Our experimentally optimised value for the number of time steps is

$$N_{\text{exp opt}} \approx 200 \quad . \tag{114}$$

Specifically for American options, we note that because of their "always-optional" nature we *cannot* do better than using at least $N \approx 200$, even by going to larger molecules, eg 7–5. Molecules larger than 7–3 *are* more accurate but will generally be slower to implement.

6 Conclusion

In this work, we have analysed the numerical stability properties of diffusion algorithms using generalised Crank-Nicolson n–m molecules. In particular, we probed the stability of the algorithms by modelling long-dated Hull-White zero bonds with large tenor and large volatility of the short-rate. The main idea was to find molecules that exhibited a desired level of stability when diffusing the Hull-White zeros. We could then be confident that the algorithms of these molecules could be used to model option-embedded bonds, such as American style callable bonds.

During our analysis, we demonstrated that there are two types of numerical stability to consider: 1) Von Neumann stability, which concerns the relationship between the size of the time steps, Δt, and the size of the intervals of the state variable Δx, and 2) Stiffness, which, for a given number of time steps, concerns the size of the product $\sigma^2 T^3$, where T is the tenor the bond being diffused and σ is the annualised volatility of the short rate.

Von Neumann stability for a n–m molecule was obtained by Von Neumann mode analysis. For the standard one-dimensional heat equation, one probes Von Neumann stability of an n–m molecule by diffusing (propagating) Fourier modes

$$e^{ipx} \quad , \tag{115}$$

where p is the frequency of the mode. After one numerical time step of the algorithm, a Taylor expansion is performed over the parameter

$$p\Delta x \quad , \tag{116}$$

and the leading error terms are eliminated in order to find the diffusion weights of the molecule *and* to optimise the Von Neumann stability. In the case of the standard 3–3 Crank-Nicolson molecule, the algorithm turns out to be unconditionally stable for the heat equation, ie stable for all values of p.

For the Hull-White equation, however, none of the n–m molecules are unconditionally stable. Instead, there is an additional stability to consider – stiffness. The stiffness of the diffusion algorithms arises from the fact that Hull-White zero bonds are the modes to analyse for Von Neumann stability. In other words, instead of Fourier modes we have

$$e^{-a_T(t)x} \quad , \tag{117}$$

where

$$\begin{aligned} a_T(t) &= \frac{1}{k}\left(1 - e^{-k(T-t)}\right) \quad , \\ k &= \text{rate of mean reversion of the short rate} \quad . \end{aligned} \tag{118}$$

In this case, the parameter for the Taylor expansion is

$$a_T(t)\,\Delta x \quad . \tag{119}$$

Noting that Von Neumann stability relates Δx to $\sigma\sqrt{\Delta t}$, and also noting that the function $a_T(t)$ has the upper bound T, we have that error boundary for the Hull-White modes is parametrised by

$$\sigma T\sqrt{\frac{T}{N}} \quad , \tag{120}$$

where N is the uniform number of time steps in the algorithm. Clearly the upper bound of the error is sensitive to the values of σ and T, which gives rise to model stiffness. Recalling that the fractional powers of $\Delta t = \frac{T}{N}$ are eliminated by choosing normalised symmetric weights in the Von Neumann analysis, we see that the stiffness measure (for a fixed number of time steps) is given by the product

$$\sigma^2 \, T^3 \quad . \tag{121}$$

Note, however, that stiffness is also controlled by the order of the remaining Taylor expansion term and by the coefficient of that term. Because of this, the allowable regime of $\sigma^2 \, T^3$ is relative to the molecule used and the practitioners desired model tolerance.

For modelling long dated (option-embedded) bonds with large short-rate volatility, stiffness is extremely important. In particular, given the practitioner's target bond specifications, it is likely that the standard 3–3 Crank-Nicolson algorithm will have an exploding error or be insufficiently fast. Thus, larger molecules and their associated stiffness must be investigated in order to have a fast, accurate model.

This work raises a few questions immediately. We are not entirely sure of the exact role that the particular local coordinates we chose, ie $(y, \, t_y)$, are playing in this work. We will leave this to others to investigate. Our hunch is this general result will hold for any coordinates chosen to value the derivative, but it would be very interesting to investigate this assertion.

Furthermore, it seems that the choice of n–m finite-difference algorithm plays a crucial role. In our example of the 3–1 explicit diffusion algorithm (see Section 4.4.2), we saw that, given a user-defined error threshold, the model failed with exploding error when passing a critical value,

$$\sigma^2 \, T^3 > 3.6 \quad . \tag{122}$$

In fact, the error grows so rapidly that, in effect, the critical value of $\sigma^2 \, T^3$ acts as a switch: exceeding the critical value of a model "turns off" successful valuation. This critical behaviour raises the question of the performance of other numerical methods, eg finite element methods, and their interaction with derivative maturities and parameters. It would be quite exciting to explore other methods.

References

[1] J. Hull, A. White "Pricing Interest Rate Derivative Securities", *The Review of Financial Studies*, 3(4), 1990, pp. 573-592.

[2] J. Topper 2005, *Financial Engineering with Finite Elements* Chichester, England: John Wiley & Sons Ltd.

[3] D. Heath, R. Jarrow, A. Morton "Bond Pricing and the Term Structure of Interest Rates: A New Methodology", *Econometrica*, 60(1), 1992, pp. 77-105.

[4] O. Cheyette, "Term Structure Dynamics and Mortgage Valuation", *The Journal of Fixed Income*, 2, 1992, pp. 28-41.

[5] R. Navin, 2007, *The Mathematics of Derivatives*, Hoboken, NJ: John Wiley & Sons, Inc.

[6] R. Burden, J. Faires 1985, *Numerical Analysis, 3rd Ed.*, Boston, MA: PWS Publishers.

[7] J. Hull 2006, *Options, Futures, and Other Derivatives*, Upper Saddle River, NJ: Prentice Hall.