PDE Models of The Term Structure: Matching Yield and Volatility Curves

Kenneth R. Vetzal
Centre for Advanced Studies in Finance
University of Waterloo

May 2, 1996

Outline

1. Background on the term structure
2. Background on finite difference methods
3. Matching the yield curve
4. Matching yield and volatility curves
5. Concluding remarks

1. Background on the Term Structure

Assume for simplicity a one-factor model and a risk-neutral world:

$$dP(t,T) = \mu(t,T)P(t,T)dt + \sigma(t,T)P(t,T)dB$$

Absence of arbitrage $\Rightarrow \mu(t,T) = r(t)$

Three Points of Departure

1. Claim values as expectations:

$$\text{Let } \; Q(t,T) = P(t,T)\exp\{-\int_0^t r(u)du\}$$

It is easy to show Q is a martingale, implying:

$$\mathrm{E}_t \exp\{-\int_0^T r(u)du\} = P(t,T)\exp\{-\int_0^t r(u)du\}$$

$$\Rightarrow P(t,T) = \mathrm{E}_t \exp\{-\int_t^T r(u)du\}$$

2. Traditional PDE approach:

Assume a model for r such as:

$$dr = \kappa(\theta - r)dt + \sigma dB$$

Then:

$$dP = \left[\frac{1}{2}\sigma^2 P_{rr} + \kappa(\theta - r)P_r + P_t\right] dt + \sigma P_r dB$$

But $\mathrm{E}(dP) = rPdt$, implying:

$$\frac{1}{2}\sigma^2 P_{rr} + \kappa(\theta - r)P_r + P_t - rP = 0$$

This is relatively easy to implement, but typically inconsistent with the current yield curve.

3. Heath, Jarrow, and Morton approach:

$$\text{Define}\;\; f(t,T) \equiv \frac{-\partial \ln P(t,T)}{\partial T}$$

Then

$$df(t,T) = \sigma_T(t,T)\sigma(t,T)dt + \sigma_T(t,T)dB$$

Since $r(t) = f(t,t)$,

$$\begin{aligned} dr(t) &= \frac{\partial f(0,t)}{\partial t} + \int_0^t \left[\sigma(u,t)\sigma_{tt}(u,t) + \sigma(u,t)^2\right] du \\ &+ \int_0^t \sigma_{tt}(u,t)dB + \sigma_t(t,t)dB \end{aligned}$$

This is automatically consistent with the current term structure, but path-dependence makes it relatively hard to implement.

Recent work

- special cases of HJM
  - Carverhill
  - Cheyette
  - Li, Ritchken and Sankarasubramian
- PDE models with time-dependent parameters
  - Black, Derman, and Toy
  - Black and Karasinski
  - Hull and White
  - Chen

2. Background on Finite Difference Methods

Consider a PDE of the form:

$$\frac{1}{2}\sigma^2 u_{rr} + \kappa(\theta - r)u_r + u_t - ru = 0$$

We solve this using a discrete grid. Let $u_{i,j}$ denote the value of u at grid point i (in r space) and j (in time). We approximate the derivatives using something like:

$$\begin{aligned} u_{rr} &= \left[u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right]/(\Delta r)^2 \\ u_r &= \left[u_{i+1,j} - u_{i-1,j}\right]/2\Delta r \\ u_t &= \left[u_{i,j+1} - u_{i,j}\right]/\Delta t \end{aligned}$$

Using similar expressions for r derivatives at time step $j+1$, and substituting into the PDE, we get a set of equations:

$$\begin{aligned}\lambda c_1 u_{i-1,j+1} \quad &+ \quad (1+\lambda c_2)u_{i,j+1} + \lambda c_3 u_{i+1,j+1} = \\ (1-\lambda)d_1 u_{i-1,j} \quad &+ \quad (1-(1-\lambda)d_2)u_{i,j} + (1-\lambda)d_3 u_{i+1,j}\end{aligned}$$

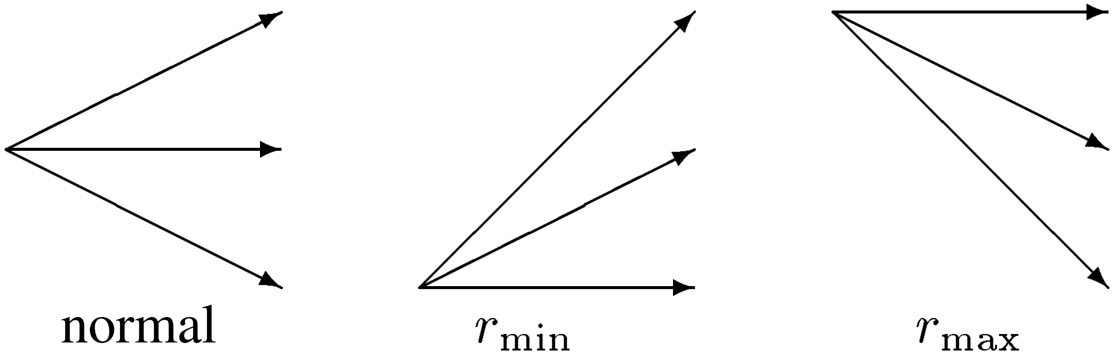
which can be written in matrix form as $Ax = b$, where x is the u vector at time step $j+1$ and b is it at j.

<u>Comments</u>

- Explicit methods $(\lambda = 0)$ are fast, but not necessarily stable if Δt is not small enough. Problems get worse with multiple factors.
- Implicit $(\lambda = 1)$ and Crank-Nicholson (e.g. $\lambda = 1/2$) methods are slower, but stable. No significant problems (aside from computation time) with more factors.
- HW "trinomial trees" are just explicit methods. Inappropriate for some models (not mean-reverting, stochastic volatility).

Boundary conditions

The terminal condition determines the type of claim, what about the maximum and minimum values of r? HW use modified branching such that "probabilities" are positive:



It is easy to show that HW's method is equivalent to calculating the r derivatives slightly differently at the boundary points, so we can use exactly the same "boundary conditions" in a CN scheme.

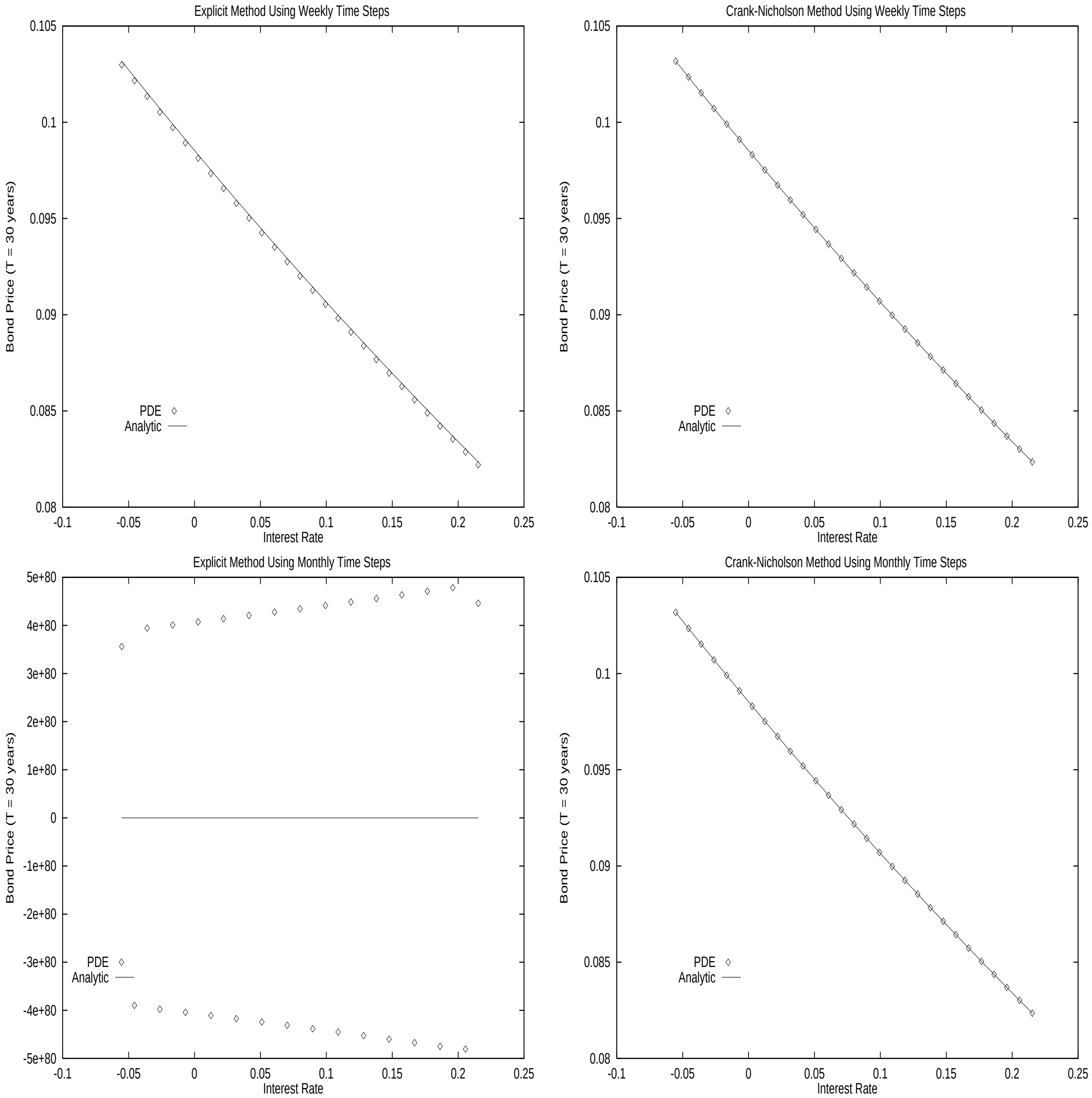
For a CN scheme, how significant is the speed disadvantage and can we take big enough time steps to overcome it?

Execution Time for a 30 Year Bond

Vasicek Model: $dr = \kappa(\theta - r)dt + \sigma dB$

DEC Alpha 2100

Weekly time steps	Explicit	0.3172 seconds
	CN	0.7603 seconds
Monthly time steps	CN	0.1972 seconds


Explicit Method Using Weekly Time Steps
Crank-Nicholson Method Using Weekly Time Steps
Explicit Method Using Monthly Time Steps
Crank-Nicholson Method Using Monthly Time Steps
Bond Price (T = 30 years)
Interest Rate
PDE
Analytic
0.105
0.1
0.095
0.09
0.085
0.08
5e+80
4e+80
3e+80
2e+80
1e+80
0
-1e+80
-2e+80
-3e+80
-4e+80
-5e+80
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25

3. Matching the Yield Curve

Note that if we are only interested in pricing European-style options consistently with the current yield curve, it is easy to just use Dybvig's simple trick (divide strike by observed forward bond price/model forward bond price, multiply option value by observed price of underlying bond/model price of underlying bond).

Aside from that, the basic method is to make some parameter in the PDE a function of time, then solve for values of it to match the yield curve. For example:

$$\frac{1}{2}\sigma^2 u_{rr} + \kappa(\theta(t) - r)u_r + u_t - ru = 0$$

or:

$$\frac{1}{2}\sigma^2 u_{rr} + \kappa(\theta - r)u_r + u_t - (r + \gamma(t))u = 0$$

which will typically be somewhat faster since all terms involving the unknown parameter lie on the diagonal of A.

The matching procedure

Suppose we write the PDE as:

$$\frac{1}{2}\sigma^2 u_{rr} + (p - \kappa r)u_r + u_t - ru = 0 \tag{1}$$

and we impose the extra condition $u(\overline{r}, \overline{s}) = \overline{u}$. Let $f(u,p) = u(\overline{r}, \overline{s}) - \overline{u}$. Then our problem is to find p such that $f(u,p) = 0$.

We use Newton's method:

$$p^{n+1} = p^n - \frac{f(u,p^n)}{f'(u,p^n)}$$

where $f'(u,p^n) = \partial u / \partial p^n$. Let $w = \partial u / \partial p$ and differentiate (1) to obtain:

$$\frac{1}{2}\sigma^2 w_{rr} + (p - \kappa r)w_r + w_t - rw = -u_r \tag{2}$$

We solve this for a given p and use it to compute w, then repeat until convergence.

How long does it take?

Execution Time for a 30 Year Bond

Vasicek Model: $dr = \kappa(\theta - r)dt + \sigma dB$

(Matching a yield curve generated by the CIR model)

DEC Alpha 2100

	Diagonal	Off-diagonal
Weekly time steps	3.29 seconds	5.64 seconds
Monthly time steps	0.93 seconds	1.17 seconds

Note that we can also use this method to match at only some time steps, rather than each one.

4. Matching Yield and Volatility Curves

If desired, it is straightforward to extend this procedure to matching volatilities/option prices. Suppose we want to match the price of a one year option on a two year bond. The first step is to match the price of a one year bond. Consider the PDE:

$$\frac{1}{2}\sigma^2 u_{rr} + (p_1 - \kappa r)u_r + u_t - ru = 0$$

Find p_1 as before. Next, we match the option price and the price of a two year bond. We have:

$$\frac{1}{2}\sigma^2 u_{rr} + (p_2 + p_3 r)u_r + u_t - ru = 0$$

with "derivative" PDES:

$$\begin{aligned}\frac{1}{2}\sigma^2 w_{rr} + (p_2 + p_3 r) w_r + w_t - rw &= -u_r \\ \frac{1}{2}\sigma^2 v_{rr} + (p_2 + p_3 r) v_r + v_t - rv &= -ru_r\end{aligned}$$

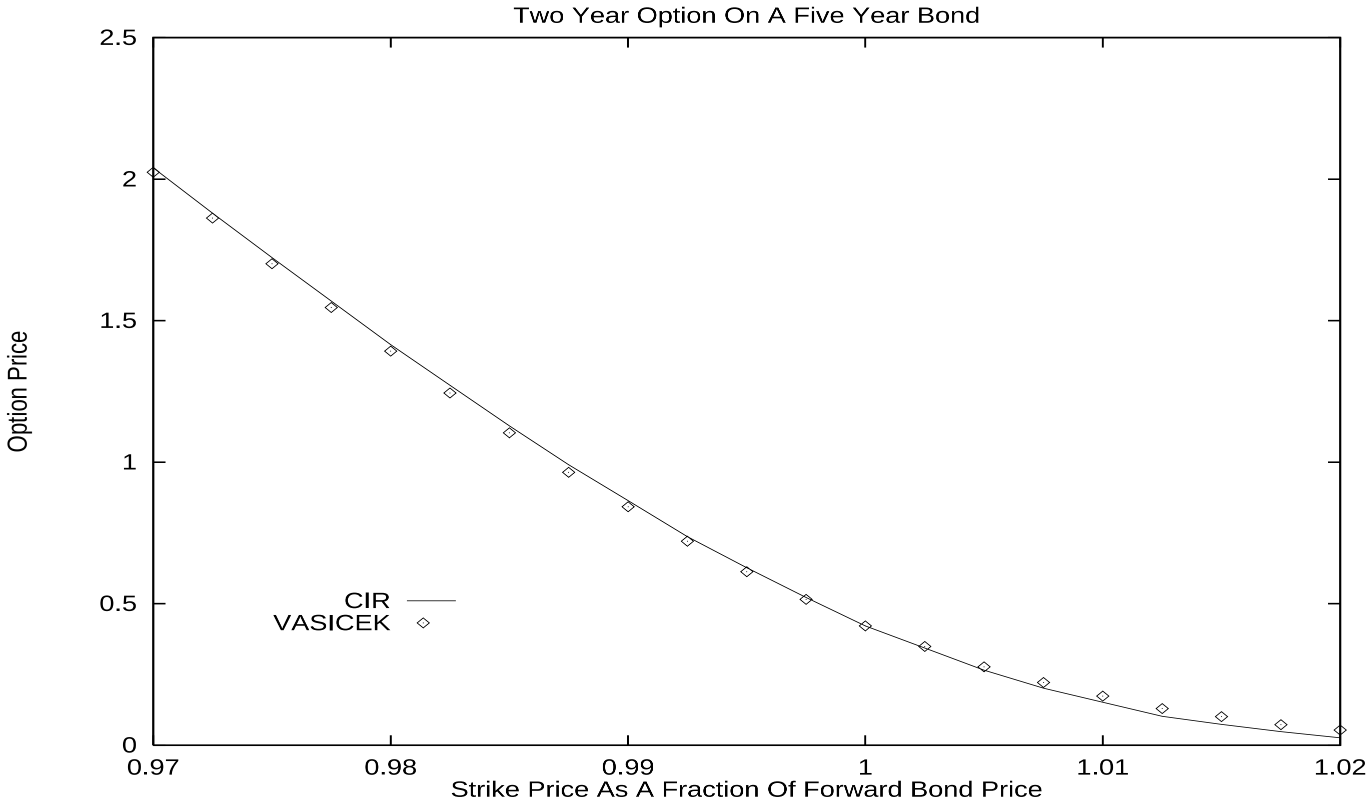
where $w = \partial u / \partial p_2$, $v = \partial u / \partial p_3$. We now have two functions, say f_1 and f_2. Using Taylor's series:

$$f(p + \delta p) = f(p) + J\delta p + O(\delta p^2)$$

where $J_{ij} = \partial f_i / \partial p_j$. Ignoring higher than first order terms, we get $J\delta p = -f$, which can be used to get updates for p. Again, we repeat until convergence.

Example

Using the Vasicek model to match an option price generated by the CIR model (two year call on a five year bond) takes 3.49 seconds with weekly time steps, 0.81 seconds with monthly).



5. Concluding Remarks

- CN methods over some potential advantages including:
  - grid flexibility (more control over time steps, larger time steps)
  - ability to handle wider class of models
- the method presented here is fairly flexible (matching term structure, options, yield volatilities at selected time points)
- preliminary results suggest that it is relatively easy to match the yield curve using two factor models, but option prices are more difficult (computing time is very sensitive to starting values, need to try other optimization algorithms)
- existence of solutions?