Interpolation Methods for Curve Construction

Article *in* Applied Mathematical Finance · February 2006

DOI: 10.1080/13504860500396032 · Source: RePEc

Patrick S Hagan
Gorilla Science

Applied Mathematical Finance,
Vol. 13, No. 2, 89–129, June 2006

Routledge
Taylor & Francis Group

Interpolation Methods for Curve Construction

PATRICK S. HAGAN* & GRAEME WEST**

**Bloomberg, LP, 499 Park Avenue, New York, NY 10022, USA, **Programme in Advanced Mathematics of Finance, School of Computational & Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, South Africa*

(Received 30 November 2004; in revised form 6 July 2005)

ABSTRACT *This paper surveys a wide selection of the interpolation algorithms that are in use in financial markets for construction of curves such as forward curves, basis curves, and most importantly, yield curves. In the case of yield curves the issue of bootstrapping is reviewed and how the interpolation algorithm should be intimately connected to the bootstrap itself is discussed. The criterion for inclusion in this survey is that the method has been implemented by a software vendor (or indeed an inhouse developer) as a viable option for yield curve interpolation. As will be seen, many of these methods suffer from problems: they posit unreasonable expections, or are not even necessarily arbitrage free. Moreover, many methods lead one to derive hedging strategies that are not intuitively reasonable. In the last sections, two new interpolation methods (the monotone convex method and the minimal method) are introduced, which it is believed overcome many of the problems highlighted with the other methods discussed in the earlier sections.*

KEY WORDS: Yield curve, interpolation, bootstrap

Curve Fitting

There is a need to value all instruments consistently within a single valuation framework. For this we need a risk-free yield curve which will be a continuous zero curve (because this is the standard format, for all option pricing formulae). Thus, a yield curve is a function $r=r(\tau)$, where a single payment investment for time τ will earn a continuous rate $r=r(\tau)$, that is, a payment of 1 at initiation will be redeemed by a payment of $\exp(r(\tau)\tau)$ at time τ.

As explained in Zangari (1977) and Lin (2002) term structure estimation methods can be classified into two groups: theoretical and empirical. Theoretical term structure methods typically posit an explicit structure for a variable known as the short rate of interest, whose value depends on a set of parameters that might be

Correspondence Address: Graeme West, Programme in Advanced Mathematics of Finance, School of Computational & Applied Mathematics, University of the Witwatersrand, Private Bag 3, Wits 2050, South Africa. Tel.: +27-11-4472901; Email: graeme@finmod.co.za

1350-486X Print/1466-4313 Online/06/020089–41 © 2006 Taylor & Francis
DOI: 10.1080/13504860500396032

determined using statistical analysis of market variables. Early examples of theoretical methods include Vasicek (1977) and Cox *et al.* (1985). From such a method the yield curve can be derived. Because the theoretical method is parsimonious, the yield curve will fall into one of a few basic categories in terms of shape. In some circumstances, negative rates are possible.

Empirical methods are available to compute spot interest rates. Unlike the theoretical methods, the empirical methods are independent of any model or theory of the term structure. Whereas the theoretical methods attempt to explain typical features of the term structure, which may include how the term structure evolves through time, the empirical methods merely try to find a close representation of the term structure at any point in time, given some observed interest rate data.

Later developments, in particular the approach of Hull and White (1990), allowed the use of a empirically determined yield curve in a theoretical model. Furthermore, the classification scheme of Heath *et al.* (1990) takes as input the same empirically determined yield curve. Thus, while the practitioner has several choices for the theoretical model that will govern their evolution of the yield curve, and hence govern their pricing of derivative products, they will almost certainly have as starting point an empirically determined yield curve. This document is concerned with that task of determining the yield curve, a process typically called bootstrapping. In fact, our treatment is slightly more general, as it covers the construction of spread curves, forward curves, etc. as well.

As explained in several sources, for example Ron (2000), there is no single correct way to complete the term structure of a yield curve from a set of rates. It is desired that the derived yield curve should be smooth, but there must not be over-smoothing, as this might cause the elimination of valuable market pricing information. It may or may not be a criterion that all inputs to the yield curve should price back exactly after the construction of that curve, although we certainly prefer an approach where there are fewer inputs and hence this perfect replication is feasible. We will typically be following this approach, although the issue of error minimization when there is a large set of inputs will be mentioned later. Certainly this approach is completely feasible when bootstrapping a swap curve, it may or may not be feasible when bootstrapping a bond curve, this will depend on the number of liquid bonds available in the market. Even when we require that the curve perfectly replicates the price of the input instruments, the yield curve is not constructed uniquely; we need to select an interpolation method with which to build the curve.

In this paper we survey a wide, but not exhaustive, selection of the interpolation methods that are in use in financial markets and their systems. In later sections we introduce two new interpolation methods, which we believe overcome many of the problems highlighted with other methods that have been discussed in the earlier sections.

Desirable Features of an Interpolation Scheme

The criteria to use in judging a curve construction and its interpolation method that we will consider are:

(1) In the case that we have a small set of instruments with which we are building an exact empirical curve, does this indeed occur? In the case that we are using a

large set of instruments, does the algorithm to find the best fit curve converge sufficiently rapidly, and is the degree of error in the created curve sufficiently small?

(2) In the case of yield curves, how good do the forward rates look? These are usually taken to be the 1 month or 3 month forward rates, but these are virtually the same as the instantaneous rates. We will want to have positivity and continuity of the forwards. It is required that forwards be positive to avoid arbitrage, while continuity is required as the pricing of interest-sensitive instruments is sensitive to the stability of forward rates. As pointed out in McCulloch & Kochin (2000), 'a discontinuous forward curve implies either implausible expectations about future short-term interest rates, or implausible expectations about holding period returns'. Thus, such an interpolation method should probably be avoided, especially when pricing derivatives whose value is dependent upon such forward values.

(3) How local is the interpolation method? If an input is changed, does the interpolation function only change nearby, with no or minor spill-over elsewhere, or can the changes elsewhere be material?

(4) Are the forwards not only continuous, but also stable? We can quantify the degree of stability by looking for the maximum basis point change in the forward curve given some basis point change (up or down) in one of the inputs. Many of the simpler methods can have this quantity determined exactly, for others we can only derive estimates.

(5) How local are hedges? Suppose we deal with an interest rate derivative of a particular tenor. We assign a set of admissible hedging instruments, for example, in the case of a swap curve, we might (even should) decree that the admissible hedging instruments are exactly those instruments that were used to bootstrap the yield curve. Does most of the delta risk get assigned to the hedging instruments that have maturities close to the given tenors, or does a material amount leak into other regions of the curve?

We will discuss criteria (1) and (2) as we proceed with each method that we analyse. Criteria (3), (4) and (5) will be discussed much later.

In most cases we have the rates $r_1, r_2, \ldots, r_n$ at the nodes $\tau_1, \tau_2, \ldots, \tau_n$ and need to determine the rate $r(\tau)$ where τ is not necessarily one of the τ_i. Occasionally we will have the forward rates rather than the rates themselves, and are required to perform the interpolation on these. In these cases, we may wish to recover the rates using the relationship $f(\tau) = \frac{\partial}{\partial \tau} r(\tau)\tau$.

For any $\tau \notin [\tau_1, \tau_n]$, the value of $r(\tau)$ or $f(\tau)$ will be that rate found at the nearer of τ_1 or τ_n.

Note that the forward is positive if and only if the capitalization function is increasing, equivalently, $r(\tau)\tau$ is increasing.

Interpolation and Bootstrap of Yield Curves – not two Separate Processes

As has been mentioned, many interpolation methods for curve construction are available. What needs to be stressed is that in the case of bootstrapping yield curves, the interpolation method is intimately connected to the bootstrap, as the bootstrap proceeds with incomplete information. This information is 'completed' (in a non-unique way) using the interpolation scheme.

Swap Curves

Let us first consider swap curves. Suppose a swap makes the fixed payments at time τ_1, τ_2, …, τ_n; time is measured in years. As explained in Hull (2002, Section 6.4), a swap just issued at par can be valued by

$$R_n \sum_{i=1}^{n} \alpha_i Z(\tau_i) + Z(\tau_n) = 1 \tag{1}$$

where R_n is the par swap rate, and α_i is the time in years from τ_{i-1} to τ_i, calculated with the relevant day count convention. In the theory, R_n is now solved for, as

$$R_n = \frac{1 - Z(\tau_n)}{\sum_{i=1}^{n} \alpha_i Z(\tau_i)} \tag{2}$$

Alternatively, we can inductively suppose that $Z(\tau_i)$ is known for $i=1, 2, \ldots, n-1$, and R_n is known, to get

$$Z(\tau_n) = \frac{1 - R_n \sum_{i=1}^{n-1} \alpha_i Z(\tau_i)}{1 + R_n \alpha_n} \tag{3}$$

At first blush, use of (3) assumes that inputs to the curve are available for all standard tenors[1] to maturity. This is typically not the case. For example, in constructing a swap curve, we might use deposit rates in the very short term, forward rate agreements or futures in the short to medium term, and swap rates in the longer term. Typically, the FRA or futures rates will be available for calculation of the relevant rates for all three-month tenors out to say two years.

The use of futures and FRAs will pose no difficulty. One applies a standard convexity adjustment to futures prices to get an equivalent FRA rate. This convexity adjustment will depend on some time and volatility parameters, but not on the yield curve itself. However, the swap rates may only be available in say 2, 3, 4, up to 10 year tenors. What to do about tenors which are not in whole number of years away? Even worse, the swap rates may only be available in say 2, 3, 4, 5 and 10 year tenors, with the 6 to 9 year tenors insufficiently liquid to use with confidence. Thus, lack of liquidity can reduce our information set dramatically.

One approach now advocated in some sources is to interpolate (linearly, say) the input swap rates to the expiries which are not quoted, and then proceed with a complete information set. However, this decouples the interpolation procedure from the bootstrap procedure, even if the chosen interpolation method here is the same as the interpolation method that will be used to find rates at points which are not nodes after the bootstrap is completed. Rather, we rewrite (3) as

$$r_n = \frac{-1}{\tau_n} \ln \left[\frac{1 - R_n \sum_{j=1}^{n-1} \alpha_j Z(t, t_j)}{1 + R_n \alpha_n} \right] \tag{4}$$

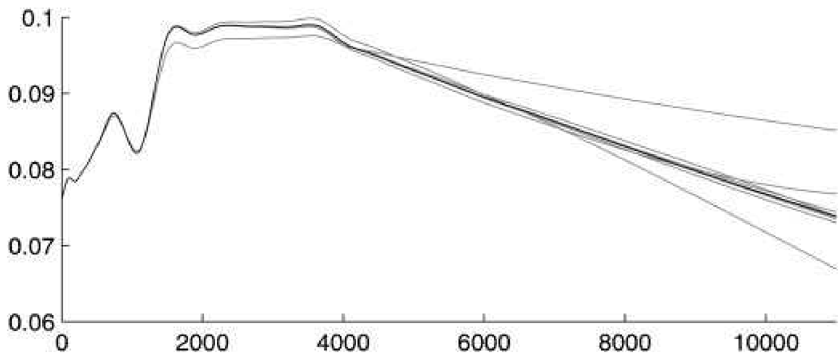

Figure 1. This method used in finding a swap curve, with the limiting curve in the contrasting colour.

and this gives us a very useful iterative formula: we guess initial rates r_n for each of the quoted expiries, perform interpolation using our chosen method of the yield curve itself to determine any missing r_j, and hence any $Z(t, t_j)$, and use this formula to extract new estimates of the r_n. The initial guess might, for example, be the continuous equivalent of the input swap rate, but in reality, any guess will suffice. We then iterate; convergence is fast over the entire yield curve.

Thus, the interpolation method applies not only to the spaces between standard tenors, but the (typically larger) spaces between the input tenors.

Bond Curves

Bootstrapping bond curves poses new problems. Let us first consider the case where there are only a few bonds for construction of the yield curve, and we require an output yield curve which prices those bonds exactly. In this case, we can consider two different ways of realizing the value of any of the bonds: the all-in (dirty) price of the bond, adjusted if necessary for any defined payment lags in the market, and the sum of the present value of all of the cash flows due to the owner as found off of the desired yield curve.

We easily set up equations very similar to (4): there will be one equation for each bond, and the rate on the left-hand side will be the rate for the maturity date of the bond. The first guess could, for example, be the continuous equivalent of the yield to maturity of the bond if such an input exists (in other words, if the market trades on or calculates yield to maturities of bonds). Again, in reality, any initial estimate will typically suffice.

In many markets, there will rather be a surfeit of bond information, with many bonds of different maturities trading. We must assume that, modulo liquidity issues, the bonds are reasonably homogenous, or can be homogenized using some procedure which will occur prior to input to a bootstrap algorithm.[2] Because of liquidity issues, one may prefer to exclude some of the bonds, and use only a subset of the bonds to bootstrap the yield curve; those left out are then deemed to be marked to market at the price one obtains by stripping them off the yield curve, rather than the illiquid (and hence by now 'erroneous') last price at which they traded.

A key issue is to decide on how many bonds to include as bona fide inputs to the bootstrap. To exclude too many runs the risk of excluding market information which is actually meaningful, on the other hand, including too many could result in a yield curve that is implausible, a yield curve that admits arbitrage, or a bootstrap algorithm that fails to converge. In this case, we need to consider constructing a yield curve that 'does as good a job as possible' in recovering the prices of the inputs.

In this case, what needs to be done can easily be understood; we will not deal with the specifics here, they will involve some multi-dimensional minimization problem. One needs to fix some set of node points, for example, they could be the maturity dates of those bonds that are deemed to be the most important, or could be the same nodes as exist in the swap curve, for example. One then postulates values of the yield curve at each of those node points, and completes the yield curve by using the chosen interpolation method. We can then calculate the value of each bond as stripped off this curve, versus the value that it is trading at in the market. The error (typically squared, and possibly weighted in order to attach more importance to some bonds than to others) is then summed across all the bonds. The values of the curve at the node points are perturbed, using some optimisation routine, to minimise this summed error.

We now go on to consider a variety of interpolation methods.

Simple Interpolation Methods

Suppose we are given some $\tau \in (\tau_1, \tau_n)$ which is not equal to any of the τ_i. First we determine i such that $\tau_i < \tau < \tau_{i+1}$.

The methods discussed in this section only use the rates r_i and r_{i+1} in order to estimate $r(\tau)$. Typically, the methods can be formulated as implicitly linear interpolation on the discount function, spot function, or some other transformation, such as the logarithm of the discount or spot function. Other methods will require some property of the forward function, for example, piecewise constant. Possible methods are:

Linear on Discount Factors

Let $d(\tau) = \exp(-r(\tau)\tau)$ be the discount function, with d_i and d_{i+1} having their obvious meanings. Then for this method we have

$$d(\tau) = \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} d_{i+1} + \frac{\tau_{i+1} - \tau}{\tau_{i+1} - \tau_i} d_i$$

and so

$$r(\tau) = \frac{-1}{\tau} \ln\left[\frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} d_{i+1} + \frac{\tau_{i+1} - \tau}{\tau_{i+1} - \tau_i} d_i\right] \tag{5}$$

For the forward function, we calculate

$$
\begin{aligned}
f(\tau) &= \frac{\partial}{\partial \tau} r(\tau)\tau \\
&= -\frac{\dfrac{1}{\tau_{i+1}-\tau_i} d_{i+1} - \dfrac{1}{\tau_{i+1}-\tau_i} d_i}{\dfrac{\tau-\tau_i}{\tau_{i+1}-\tau_i} d_{i+1} + \dfrac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i} d_i} \\
&= \frac{d_i - d_{i+1}}{(\tau-\tau_i)d_{i+1} + (\tau_{i+1}-\tau)d_i}
\end{aligned}
$$

which shows that the forward is not continuous (by the time τ reaches τ_{i+1}, the input from τ_i has not been 'forgotten').

Linear on Spot Rates

$$
r(\tau) = \frac{\tau-\tau_i}{\tau_{i+1}-\tau_i} r_{i+1} + \frac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i} r_i \tag{6}
$$

In this case clearly

$$
f(\tau) = \frac{2\tau-\tau_i}{\tau_{i+1}-\tau_i} r_{i+1} + \frac{\tau_{i+1}-2\tau}{\tau_{i+1}-\tau_i} r_i \tag{7}
$$

and, as before, the forward rates are not continuous.

Raw Interpolation

This method is linear on the logarithm of discount factors, and as we shall see, corresponds to piecewise constant forward curves. To a good approximation, any forward curve that has the same area between each node would work. This means that if a piecewise linear approximation starts too high, it has to go too low to average to the right value, but then it starts the next interval too low and has to go too high to average to the right value. This method is very stable, is trivial to implement, and is usually a base method one implements in a system before any others. One can often find mistakes in fancier methods by comparing the raw method with the more sophisticated method.

Since the instantaneous forward curve is $f(\tau) = \frac{\partial}{\partial \tau} r(\tau)\tau$, the interpolating function for the yield curve is $r(\tau) = K + \frac{C}{\tau}$. Given the two endpoints, this solves as

$$
\begin{aligned}
f(\tau) : = K &= \frac{r_{i+1}\tau_{i+1} - r_i\tau_i}{\tau_{i+1}-\tau_i} \\
C &= \frac{(r_i - r_{i+1})\tau_i\tau_{i+1}}{\tau_{i+1}-\tau_i}
\end{aligned}
$$

and after some manipulation, we get

$$
r(\tau) = \frac{\tau-\tau_i}{\tau_{i+1}-\tau_i}\frac{\tau_{i+1}}{\tau} r_{i+1} + \frac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i}\frac{\tau_i}{\tau} r_i \tag{8}
$$

Note that this method is occasionally called exponential interpolation, as it involves exponential interpolation of the discount factors i.e.

$$d(\tau)=d_{i+1}^{\frac{\tau-\tau_i}{\tau_{i+1}-\tau_i}}d_i^{\frac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i}}$$

This is equivalent to linear interpolation of the logarithm of the discount factors, and this should not be a surprise: one always has that

$$f(\tau)=-\frac{\partial}{\partial\tau}\ln d(\tau) \tag{9}$$

and so the constant forward model is easily seen to be equivalent to this type of linear interpolation.

Linear on the Logarithm of Rates

This method is called log-linear interpolation or even exponential interpolation. If

$$\ln r(\tau)=\frac{\tau-\tau_i}{\tau_{i+1}-\tau_i}\ln r_{i+1}+\frac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i}\ln r_i$$

then

$$r(\tau)=r_{i+1}^{\frac{\tau-\tau_i}{\tau_{i+1}-\tau_i}}r_i^{\frac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i}} \tag{10}$$

Since

$$\ln r(\tau)\tau=\frac{\tau-\tau_i}{\tau_{i+1}-\tau_i}\ln r_{i+1}+\frac{\tau_{i+1}-\tau}{\tau_{i+1}-\tau_i}\ln r_i+\ln\tau$$

we have

$$\frac{1}{r(\tau)\tau}f(\tau)=\frac{1}{\tau_{i+1}-\tau_i}\ln\frac{r_{i+1}}{r_i}+\frac{1}{\tau}$$

and so

$$f(\tau)=r(\tau)\left[\frac{\tau}{\tau_{i+1}-\tau_i}\ln\frac{r_{i+1}}{r_i}+1\right] \tag{11}$$

Remarkably, this method is quite popular, being provided as one of the default methods by many software vendors. However, it clear from (11) that this method does not guarantee positive forward rates. As a trivial (not necessarily practicable) example, if we have a two-point curve, with nodes (1,6%) and (30,2%) then the forward rates are negative from about the 26th year.

All these simple methods have continuity difficulties associated with them. Thus, they should not be used for anything but naive interpolation of yield curves, after which criteria such as rate smoothness, forward rate smoothness etc. are important.

Piecewise Linear Continuous Forwards

Let us quickly consider this method; just as quickly we will reject it.

When considering this method we find that the forward curve exhibits enormous zig-zag instability. As a simple example, suppose the continuous rates for a yield curve are specified at all whole number terms, with $r(t) = r$ for $t \leqslant T$, some specified T, and $r(t) = r+\varepsilon$ for $t \geqslant T+1$. Then $f(t) = r$ for $t \leqslant T$. The discrete forward from T to $T+1$ is $r+\varepsilon(1+T)$. Hence $f(T+1) = r+2\varepsilon(1+T)$. Then $f(T+2) = r-2\varepsilon T$, and the pattern repeats itself for odd and even increments from T.

Cubic Splines

As before, suppose $\tau_1, \tau_2, \ldots, \tau_n$ and $r_1, r_2, \ldots, r_n$, $r_i\!:=r(\tau_i)$ are known. To complete a cubic spline, we desire coefficients (a_i, b_i, c_i, d_i) for $1 \leqslant i \leqslant n-1$. Given these coefficients, the function value at any term τ will be

$$r(\tau) = a_i + b_i(\tau-\tau_i) + c_i(\tau-\tau_i)^2 + d_i(\tau-\tau_i)^3 \quad \tau_i \leq \tau \leq \tau_{i+1} \tag{12}$$

Note that

$$\begin{aligned} r'(\tau) &= b_i + 2c_i(\tau-\tau_i) + 3d_i(\tau-\tau_i)^2 \quad \tau_i < \tau < \tau_{i+1} \\ r''(\tau) &= 2c_i + 6d_i(\tau-\tau_i) \quad \tau_i < \tau < \tau_{i+1} \\ r'''(\tau) &= 6d_i \quad \tau_i < \tau < \tau_{i+1} \end{aligned}$$

Let $h_i = \tau_{i+1}-\tau_i$ throughout. The constraints common to all methods will be

- the interpolating function indeed meets the given data, so $a_i = r_i$ for $i = 1, 2, \ldots, n-1$ and $a_{n-1} + b_{n-1}h_{n-1} + c_{n-1}h_{n-1}^2 + d_{n-1}h_{n-1}^3 = r_n := a_n$;
- the entire interpolating function is continuous, so $a_i + b_ih_i + c_ih_i^2 + d_ih_i^3 = a_{i+1}$ for $i = 1, 2, \ldots, n-2$;
- the entire function is differentiable, so $b_i + 2c_ih_i + 3d_ih_i^2 = b_{i+1}$ for $i = 1, 2, \ldots, n-2$.

This is a system of $3n-4$ equations in $4n-4$ unknowns. Thus there are n linear constraints still to be specified.

A fundamental reason for requiring differentiability of the interpolating function is that then the forward function $f(\tau) = \frac{\partial}{\partial\tau} r(\tau)\tau$ is continuous. We clearly have

$$f(\tau) = a_i + b_i(2\tau-\tau_i) + c_i(\tau-\tau_i)(3\tau-\tau_i) + d_i(\tau-\tau_i)^2(4\tau-\tau_i) \quad \tau_i \leq \tau \leq \tau_{i+1} \tag{13}$$

Let us define

$$b_n := b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 \tag{14}$$

so that b_n is the derivative of the interpolating function at the right hand endpoint. In the most general case (de Boor 1978, 2001, Chapter IV), the specification of the remaining n linear constraints is equivalent to specifying $b_1, b_2, \ldots, b_n$ (as any n additional conditions will do, assuming there is no redundancy - the point in de

Boor, 1978, 2001, is that such a view can be an aid to classification). In particular, if we take this approach, defining $b_1, b_2, \ldots, b_n$, then $c_1, c_2, \ldots, c_{n-1}$ and $d_1, d_2, \ldots, d_{n-1}$ follow easily, as for each i, we have two equations in two unknowns, which easily solve as:

$$m_i = \frac{a_{i+1} - a_i}{h_i} \tag{15}$$

$$c_i = \frac{3m_i - b_{i+1} - 2b_i}{h_i} \tag{16}$$

$$d_i = \frac{b_{i+1} + b_i - 2m_i}{h_i^2} \tag{17}$$

for $1 \leqslant i \leqslant n-1$.

Natural Cubic Spline

The cubic spline method with the so-called natural boundary condition is described in Burden & Faires (1997, Section 3.4). This is the unique cubic spline interpolating function where the extra n conditions are

- the entire function is twice differentiable: $c_i + 3d_ih_i = c_{i+1}$ for $i = 1, 2, \ldots, n-2$;
- the second derivative at each endpoint is 0.

If this method is implemented, it would be completely satisfactory for curves with a fairly dense set of nodes (for example, swap curves) but is qualitatively unsatisfactory for curves with a more sparse set of nodes (for example, bootstrapped bond curves). The curve is too convex ('bulging') between points which are a fair distance away.

In particular, this method in no way guarantees that negative forward rates are avoided. Let us consider an example. Suppose we have the following curve:

Term	Continuous yield	Capitalization factor	Discrete forward
0.01	8.00%	1.0008	
5	7.00%	1.419068	7.00%
10	8.00%	2.225541	9.00%
15	7.00%	2.857651	5.00%
20	8.00%	4.953032	11.00%
30	7.00%	8.16617	5.00%

Then, using the natural cubic spline, forward rates after about 28 years are negative. On the other hand, with the inputs

Term	Continuous yield	Capitalization factor	Discrete forward
0.01	8.00%	1.0008	
5	8.00%	1.491825	8.00%
10	8.00%	2.225541	8.00%
15	8.00%	3.320117	8.00%
20	8.00%	4.953032	8.00%
30	7.00%	8.16617	5.00%

the forward rates are satisfactory. In both cases, the discrete forward rate in the 20–30 year period is 5%. This illustrates another property that is missing from these analytic splining methods: locality. The interpolation in a region should take into account the data in that region, and not the data some distance away.

One determines the coefficients using the well-known natural cubic spline algorithm (Burden & Faires 1997, Algorithm 3.4).

Bessel (Hermite) Cubic Spline

This method, discussed in de Boor (1978, 2001, Chapter IV) is a common choice with software vendors, by whom it is frequently called Hermite interpolation.[3] The values of b_i for $1<i<n$ are chosen to be the slope at τ_i of the quadratic that passes through (τ_j, r_j) for $j=i-1, i, i+1$. The value of b_1 is chosen to be the slope at τ_1 of the quadratic that passes through (τ_j, r_j) for $j=1, 2, 3$. The value of b_n is chosen likewise. As shown in de Boor (1978, 2001) for $1<i<n$, and by direct calculation for the extreme cases, this is given by

$$b_1 = \frac{1}{\tau_3-\tau_1}\left[\frac{(\tau_3+\tau_2-2\tau_1)(r_2-r_1)}{\tau_2-\tau_1} - \frac{(\tau_2-\tau_1)(r_3-r_2)}{\tau_3-\tau_2}\right] \tag{18}$$

$$b_i = \frac{1}{\tau_{i+1}-\tau_{i-1}}\left[\frac{(\tau_{i+1}-\tau_i)(r_i-r_{i-1})}{\tau_i-\tau_{i-1}} + \frac{(\tau_i-\tau_{i-1})(r_{i+1}-r_i)}{(\tau_{i+1}-\tau_i)}\right] \quad (1<i<n) \tag{19}$$

$$b_n = -\frac{1}{\tau_n-\tau_{n-2}}\left[\frac{(\tau_n-\tau_{n-1})(r_{n-1}-r_{n-2})}{\tau_{n-1}-\tau_{n-2}} - \frac{(2\tau_n-\tau_{n-1}-\tau_{n-2})(r_n-r_{n-1})}{\tau_n-\tau_{n-1}}\right] \tag{20}$$

Financial Cubic Spline

This is the cubic spline interpolating function which is the same as the natural method, except that the first (rather than second) derivative at the long endpoint be 0. This property is quite attractive because it ensures that there is a horizontal rate asymptote, which means that rates can be extrapolated.

The system is actually a bandwidth matrix with widths 2 and 1. As a banded matrix, we write it in the form suggested in Press *et al.* (1992, Section 2.4): so a $3(n-1)\times 4$ matrix A. The scheme below is $|A||x||b|$, where $Ax=b$, and A is written in 2-1-1 bandwidth form.[4]

$\times$	$\times$	0	1	b_1	0
$\times$	h_1	h_1^2	h_1^3	c_1	a_2-a_1
	$2h_1$	$3h_1^2$	-1	d_1	0
	$3h_1$	0	-1	b_2	0
	h_2	h_2^2	h_2^3	c_2	a_3-a_2
1	$2h_2$	$3h_2^2$	-1	d_2	0
1	$3h_2$	0	-1	b_3	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	$3h_{n-2}$	0	-1	b_{n-1}	0
0	h_{n-1}	h_{n-1}^2	h_{n-1}^3	c_{n-1}	a_n-a_{n-1}
1	$2h_{n-1}$	$3h_{n-1}^2$	$\times$	d_{n-1}	0

The first equation above, $c_1 = 0$, is the left-hand condition $f''(\tau_1) = 0$. The last equation above, $b_{n-1} + 2c_{n-1}h_{n-1} + 3d_{n-1}h_{n-1}^2 = 0$ is the right-hand condition $f'(\tau_n) = 0$.

Cubic Spline on $r(\tau)\tau$

This is a cubic spline where the cubic spline is applied not to the function $r(\tau)$ but to the function $r(\tau)\tau$; it is this interpolating function that is required to be twice differentiable.

Thus, the nodes τ_i and values $r_i\tau_i$ are passed into the cubic splining mechanism, and the value $r(\tau)\tau$ is returned. Thus, to find the value of $r(\tau)$ for any given τ, the returned value needs to be divided by τ:

$$r(\tau) = \frac{a_i + b_i(\tau-\tau_i) + c_i(\tau-\tau_i)^2 + d_i(\tau-\tau_i)^3}{\tau} \qquad \tau_i \le \tau \le \tau_{i+1} \tag{21}$$

Note that if f is the forward rate, then

$$\begin{aligned} f(\tau) &= \frac{\partial}{\partial\tau} r(\tau)\tau \\ &= b_i + 2c_i(\tau-\tau_i) + 3d_i(\tau-\tau_i)^2 \qquad \tau_i \le \tau \le \tau_{i+1} \end{aligned} \tag{22}$$

We now consider two ways of solving for the coefficients.

Quadratic natural spline. The quadratic-natural splining method is proposed in McCulloch & Kochin (2000). They argue that the usual natural cubic spline method can have a 'roller coaster' output curve, particularly in the longer part of the curve where there are fewer inputs. Essentially, the term is included as a weight in the interpolation scheme, so for longer terms, the curve will certainly be stabilised. They see as desirable properties that the curve be linear in the short end and asymptote in the long end (so that extrapolation can be applied). These criteria are achieved by making the endpoint constraints natural at the long end (i.e. the second derivative is 0), but quadratic at the short end (i.e. the third derivative is 0). Making the second derivative of $r(\tau)\tau$ zero at the short end would make the first derivative of $r(\tau)$ zero, and this is contrary to typically observed features of yield curves.

The entire system of equations can once again be rewritten into a tridiagonal system, in a way very much analogous to the ordinary cubic spline algorithm, and so this is a special case implementation of Crout's method of solving a tridiagonal system.

Bessel (Hermite) cubic spline. This method stands in relation to the Bessel method as the quadratic-normal method stands in relationship to the natural cubic spline. Thus, it is exactly the Bessel method, but applied to the function $r(\tau)\tau$ rather than $r(\tau)$.

Thus, the interpolation formulae are the same as (21) and (22).

Monotone Preserving Cubic Spline

This method is quite different from the others; it is a local method – the interpolatory values are only determined by local behaviour, not global behaviour. Thus we are not in the 'linear algebra' set-up of the previous methods, but have a different local approach. Many of the ideas here will have a natural development later.

The method specifies the values of b_i for $1 \leqslant i \leqslant n$. Note that $c_1, c_2, \ldots, c_{n-1}$ and $d_1, d_2, \ldots, d_{n-1}$ follow directly from this, as seen in (16) and (17).

It is clear that a desirable criterion for interpolation of the yield curve is that the geometry of the data points are preserved, i.e. there is no unexplained curvature, or roller coaster, introduced. Such a method, based on the method of Fritsch & Butland (1980), is developed in Hyman (1983). This method ensures that in regions of monotonicity of the inputs (so, three successive increasing or decreasing values) the interpolating function preserves this property. In the case of a local minimum or maximum in the input discrete data, the output continuous interpolatory function will preserve that property.

As before we have nodes $\tau_1, \tau_2, \ldots, \tau_n$ and values $r_1, r_2, \ldots, r_n$. Define mesh sizes and gradients

$$h_i = \tau_{i+1} - \tau_i \quad (1 \leq i \leq n-1) \tag{23}$$

$$m_i = \frac{r_{i+1} - r_i}{h_i} \quad (1 \leq i \leq n-1) \tag{24}$$

Now Hyman (1983) first defines

$$b_1 = \frac{(2h_1 + h_2)m_1 - h_1 m_2}{h_1 + h_2}$$

$$b_n = \frac{(2h_{n-1} + h_{n-2})m_{n-1} - h_{n-1} m_{n-2}}{h_{n-1} + h_{n-2}}$$

but simple examples show that this method may fail to be locally monotone immediately to the interior side of the endpoints. In particular, negative forward rates are possible, even likely. Thus, rather we define

$$b_1 = 0 = b_n \tag{25}$$

We now define the gradients b_i for $1<i<n$. The curve is locally monotone at τ_i $(1<i<n)$ if $m_{i-1}m_i \geqslant 0$.

If the curve is not locally monotone, so it has a turning point, then we define $b_i=0$, which ensures that the interpolant will also have a turning point there.

If the curve is locally monotone, we first define

$$b_i = \frac{3m_{i-1}m_i}{\max(m_{i-1}, m_i) + 2\min(m_{i-1}, m_i)} \quad (1<i<n) \tag{26}$$

Then we also include the adjustment (Hyman 1983, Equation 2.3), which ensures that no spurious extrema are introduced in the interpolated function. This adjustment is

$$b_i = \begin{cases} \min(\max(0, b_i), 3\min(m_{i-1}, m_i)) & \text{if the curve is locally increasing at } i \\ \max(\min(0, b_i), 3\max(m_{i-1}, m_i)) & \text{if the curve is locally decreasing at } i \end{cases} \tag{27}$$

Note that the requirement that the interpolatory curve preserves the geometry of the curve does not guarantee that the forward function is positive.

A Curve where all Cubic Methods Produce Negative Forward Rates

Consider the following rate curve:

Term	Continuous yield	Capitalization factor	Discrete forward
0.1	8.10%	1.00813289	
1	7.00%	1.07250818	6.88%
4	4.40%	1.22140276	4.33%
9	7.00%	1.87761058	8.60%
20	4.00%	2.22554093	1.55%
30	3.00%	2.45960311	1.00%

Although such a curve is perhaps odd looking, it is arbitrage free: the capitalization factor curve is increasing. However, when applying any of the six cubic spline variations discussed here, negative forward rates appear. See Figure 2.

Quartic Splines

Quartic Forward Spline

According to Adams (2001), the interpolation method that guarantees the smoothest interpolation of the continuous instantaneous forward rates is a quartic spline of that continuous forward curve. See also van Deventer & Inai (1997), Adams & van Deventer (1994), and Lim & Xiao (2002). A variation of this method is implemented in Quant Financial Research (2003).

As before, suppose $\tau_1, \tau_2, \ldots, \tau_n$ and $f_1, f_2, \ldots, f_n, f_i := f(\tau_i)$ are known. To complete the requisite spline for f, we desire coefficients $(a_i, b_i, c_i, d_i, e_i)$ for $1 \leqslant i \leqslant n-1$. Given

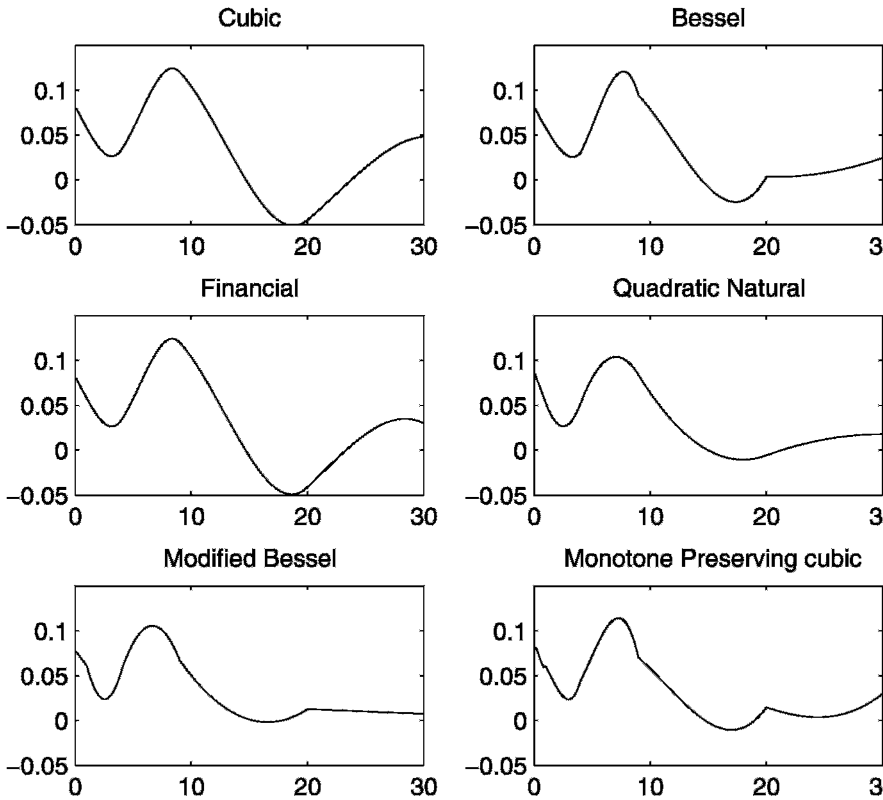

Figure 2. The forward curves under various cubic interpolation methods for the given rates

these coefficients, the function value at any term τ will be

$$f(\tau) = a_i + b_i\tau + c_i\tau^2 + d_i\tau^3 + e_i\tau^4 \quad \tau_i \le \tau \le \tau_{i+1}{}^5 \tag{28}$$

Hence

$$f'(\tau) = b_i + 2c_i\tau + 3d_i\tau^2 + 4e_i\tau^3$$
$$f''(\tau) = 2c_i + 6d_i\tau + 12e_i\tau^2$$
$$f'''(\tau) = 6d_i + 24e_i\tau$$

Requiring continuity of all of the above functions gives

$$f_i = a_i + b_i\tau_i + c_i\tau_i^2 + d_i\tau_i^3 + e_i\tau_i^4 \quad 1 \le i \le n-1$$
$$f_{i+1} = a_i + b_i\tau_{i+1} + c_i\tau_{i+1}^2 + d_i\tau_{i+1}^3 + e_i\tau_{i+1}^4 \quad 1 \le i \le n-1$$
$$b_i + 2c_i\tau_{i+1} + 3d_i\tau_{i+1}^2 + 4e_i\tau_{i+1}^3 = b_{i+1} + 2c_{i+1}\tau_{i+1} + 3d_{i+1}\tau_{i+1}^2 + 4e_{i+1}\tau_{i+1}^3 \quad 1 \le i \le n-2$$
$$2c_i + 6d_i\tau_{i+1} + 12e_i\tau_{i+1}^2 = 2c_{i+1} + 6d_{i+1}\tau_{i+1} + 12e_{i+1}\tau_{i+1}^2 \quad 1 \le i \le n-2$$
$$6d_i + 24e_i\tau_{i+1} = 6d_{i+1} + 24e_{i+1}\tau_{i+1} \quad 1 \le i \le n-2$$

Thus we have 5n-8 equations in 5n-5 unknowns. Thus, we need three more conditions. The following three conditions are specified in Adams (2001):

- $f''(\tau_1)=0$,
- $f'(\tau_n)=0$,
- $f''(\tau_n)=0$.

The system is actually a bandwidth matrix with widths 2 and 6. As a banded matrix, we write it in the form suggested in Press *et al.* (1992, Section 2.4): so a $5(n-1)\times 9$ matrix A. The scheme below is $[A\|x\|b]$, where $Ax=b$, and A is written in 2-1-6 bandwidth form.

$$
\left|\begin{array}{cc|c|cccccc||c||c}
\times & \times & 0 & 0 & 2 & 6\tau_1 & 12\tau_1^2 & 0 & 0 & a_1 & 0 \\
\times & 1 & \tau_i & \tau_i^2 & \tau_i^3 & \tau_i^4 & 0 & 0 & 0 & b_i & f_i \\
1 & \tau_{i+1} & \tau_{i+1}^2 & \tau_{i+1}^3 & \tau_{i+1}^4 & 0 & 0 & 0 & 0 & c_i & f_{i+1} \\
-1 & -2\tau_{i+1} & -3\tau_{i+1}^2 & -4\tau_{i+1}^3 & 0 & 1 & 2\tau_{i+1} & 3\tau_{i+1}^2 & 4\tau_{i+1}^3 & d_i & 0 \\
-2 & -6\tau_{i+1} & -12\tau_{i+1}^2 & 0 & 0 & 2 & 6\tau_{i+1} & 12\tau_{i+1}^2 & 0 & e_i & 0 \\
-6 & -24\tau_{i+1} & 0 & 0 & 0 & 6 & 24\tau_{i+1} & 0 & 0 & a_{i+1} & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 2\tau_n & 3\tau_n^2 & 4\tau_n^3 & \times & \times & \times & \times & \times & d_{n-1} & 0 \\
2 & 6\tau_n & 12\tau_n^2 & \times & \times & \times & \times & \times & \times & e_{n-1} & 0
\end{array}\right|
$$

The first equation above is the first extra condition, while the last two equations above are the other extra conditions.

This system is solved with the bandwidth matrix algorithms.

The bad news (and this should be expected) is that, like the cubic splining methods, there is no guarantee of the absence of negative forward rates. These methods are demanding such high smoothness criteria that any desired stiffness is completely lost from the system. Thus, we can have enormous and completely implausible fluctions in the output curve. For example, if we have the following forward curve:

Term	Instantaneous forward
0.1	2.00%
1	2.00%
2	2.00%
6	3.00%
7	2.00%
30	2.00%

then the quartic spline on forwards has negative values. See Figure 3. By simply adjusting the 6 year rate from 3% to 2%, we get a forward curve which is flat everywhere, at 2%.

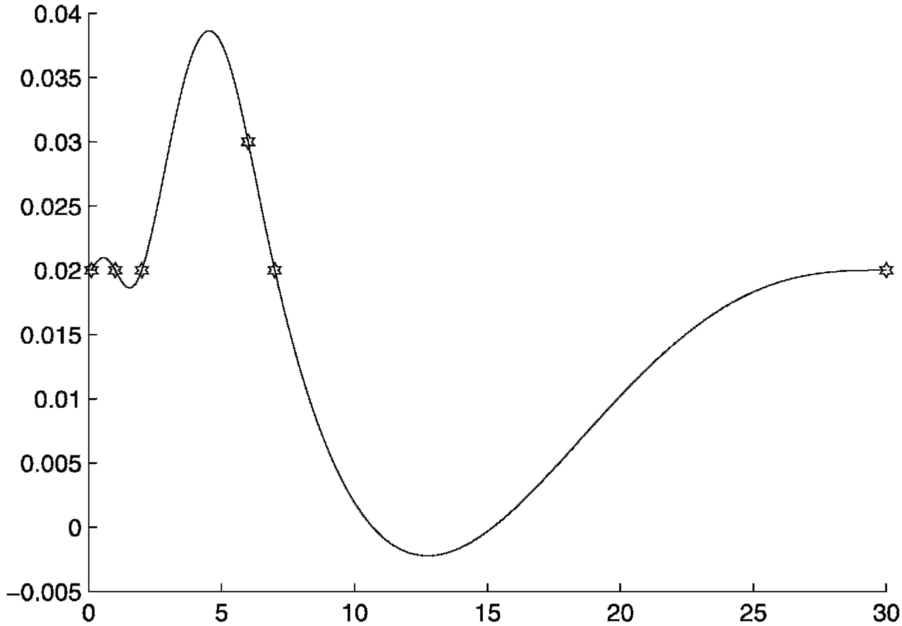

Figure 3. Quartic interpolation of the forward curve

Quartic Curve Spline

The problem arises in Adams (2001) that we may not know the forward rates for input to the spline, we know the actual risk free rates themselves. Thus it can be argued that the method of Adams (2001) is moot, and we need to rely on the mathematical relationship between the rates and the forward rates, viz. $f(\tau) = \frac{\partial}{\partial \tau} r(\tau)\tau$.

Since f is (locally) a quartic, its integral is a quintic i.e. $r(\tau)\tau$ is locally a quintic. Hence $r(\tau)$ is of the form of (29), to follow.

As before, suppose $\tau_1, \tau_2, \ldots, \tau_n$ and $r_1, r_2, \ldots, r_n$, $r_i := r(\tau_i)$ are known. To complete the requisite spline for r, we desire coefficients $(a_i, b_i, c_i, d_i, e_i, g_i)$ for $1 \leqslant i \leqslant n-1$. Given these coefficients, the function value at any term τ will be

$$r(\tau) = \frac{a_i}{\tau} + b_i + c_i\tau + d_i\tau^2 + e_i\tau^3 + g_i\tau^4 \quad \tau_i \leq \tau \leq \tau_{i+1} \tag{29}$$

Hence

$$\begin{aligned} f(\tau) &= b_i + 2c_i\tau + 3d_i\tau^2 + 4e_i\tau^3 + 5g_i\tau^4 \quad \tau_i \leq \tau \leq \tau_{i+1} \\ f'(\tau) &= c_i(2) + d_i(6\tau) + e_i\left(12\tau^2\right) + g_i\left(20\tau^3\right) \\ f''(\tau) &= d_i(6) + e_i(24\tau) + g_i\left(60\tau^2\right) \\ f'''(\tau) &= e_i(24) + g_i(120\tau) \end{aligned}$$

Requiring continuity of all of the above functions gives

$$r_i = \frac{a_i}{\tau_i} + b_i + c_i\tau_i + d_i\tau_i^2 + e_i\tau_i^3 + g_i\tau_i^4 \quad 1 \le i \le n-1$$

$$r_{i+1} = \frac{a_i}{\tau_{i+1}} + b_i + c_i\tau_{i+1} + d_i\tau_{i+1}^2 + e_i\tau_{i+1}^3 + g_i\tau_{i+1}^4 \quad 1 \le i \le n-1$$

$$\begin{aligned} & b_i + 2c_i\tau_{i+1} + 3d_i\tau_{i+1}^2 + 4e_i\tau_{i+1}^3 + 5g_i\tau_{i+1}^4 \\ &= b_{i+1} + 2c_{i+1}\tau_{i+1} + 3d_{i+1}\tau_{i+1}^2 + 4e_{i+1}\tau_{i+1}^3 + 5g_{i+1}\tau_{i+1}^4 \quad 1 \le i \le n-2 \end{aligned}$$

$$\begin{aligned} & 2c_i + 6d_i\tau_{i+1} + 12e_i\tau_{i+1}^2 + 20g_i\tau_{i+1}^3 \\ &= 2c_{i+1} + 6d_{i+1}\tau_{i+1} + 12e_{i+1}\tau_{i+1}^2 + 20g_{i+1}\tau_{i+1}^3 \quad 1 \le i \le n-2 \end{aligned}$$

$$\begin{aligned} & 6d_i + 24e_i\tau_{i+1} + 60g_i\tau_{i+1}^2 \\ &= 6d_{i+1} + 24e_{i+1}\tau_{i+1} + 60g_{i+1}\tau_{i+1}^2 \quad 1 \le i \le n-2 \end{aligned}$$

$$24e_i + 120g_i\tau_{i+1} = 24e_{i+1} + 120g_{i+1}\tau_{i+1} \quad 1 \le i \le n-2$$

Thus we have $6n-10$ equations in $6n-6$ unknowns. Thus, we need four more conditions. It could be any four of the following, the first three of which are specified in Adams (2001):[6]

- $f''(\tau_1)=0$,
- $f'(\tau_n)=0$,
- $f''(\tau_n)=0$,
- $f'''(\tau_n)=0$.
- $f'(\tau_1)=0$,
- $r(\tau_1)=f(\tau_1)$, (as this must be the case as $\tau_1 \downarrow 0$),

Consider the system with the first four extra constraints named above. The system is actually a bandwidth matrix with widths 2 and 8. As a banded matrix, we write it in the form suggested in Press *et al.* (1992, Section 2.4): so a $6(n-1) \times 11$ matrix A. The scheme below is $|A||x||b|$, where $Ax=b$, and A is written in 2-1-8 bandwidth form.

$$\left|\begin{array}{cc|c|cccccccc}
\times & \times & 0 & 0 & 0 & 6 & 24\tau_1 & 60\tau_1^2 & 0 & 0 & 0 \\
\times & \tau_i^{-1} & 1 & \tau_i & \tau_i^2 & \tau_i^3 & \tau_i^4 & 0 & 0 & 0 & 0 \\
\tau_{i+1}^{-1} & 1 & \tau_{i+1} & \tau_{i+1}^2 & \tau_{i+1}^3 & \tau_{i+1}^4 & 0 & 0 & 0 & 0 & 0 \\
-1 & -2\tau_{i+1} & -3\tau_{i+1}^2 & -4\tau_{i+1}^3 & -5\tau_{i+1}^4 & 0 & 1 & 2\tau_{i+1} & 3\tau_{i+1}^2 & 4\tau_{i+1}^3 & 5\tau_{i+1}^4 \\
-2 & -6\tau_{i+1} & -12\tau_{i+1}^2 & -20\tau_{i+1}^3 & 0 & 0 & 2 & 6\tau_{i+1} & 12\tau_{i+1}^2 & 20\tau_{i+1}^3 & 0 \\
-6 & -24\tau_{i+1} & -60\tau_{i+1}^2 & 0 & 0 & 0 & 6 & 24\tau_{i+1} & 60\tau_{i+1}^2 & 0 & 0 \\
-24 & -120\tau_{i+1} & 0 & 0 & 0 & 0 & 24 & 120\tau_{i+1} & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 2 & 6\tau_n & 12\tau_n^2 & 20\tau_n^3 & \times & \times & \times & \times & \times & \times \\
0 & 6 & 24\tau_n & 60\tau_n^2 & \times & \times & \times & \times & \times & \times & \times \\
0 & 24 & 120\tau_n & \times & \times & \times & \times & \times & \times & \times & \times
\end{array}\right\|
\begin{array}{c} a_1 \\ b_i \\ c_i \\ d_i \\ e_i \\ g_i \\ a_{i+1} \\ \vdots \\ d_{n-1} \\ e_{n-1} \\ g_{n-1} \end{array}
\left\|\begin{array}{c} 0 \\ r_i \\ r_{i+1} \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{array}\right|$$

The first equation above is the first extra condition, while the three final equations above, are, in order, the second, third and fourth extra conditions specified above.

This system is solved with the bandwidth matrix algorithms.

The problems are even worse than before. This method produces negative forward rates for the inputs seen earlier. As another tamer example, with the very innocuous rate inputs

Term	Continuous yield	Capitalisation factor	Discrete forward
0.1	6.00%	1.006018	6.00%
1	6.00%	1.061837	6.00%
2	5.00%	1.105171	4.00%
3	5.00%	1.161834	5.00%
4	5.00%	1.221403	5.00%
9	6.50%	1.794991	7.70%
20	6.00%	3.320117	5.59%
30	6.00%	6.049647	6.00%

one has a very unstable interpolant, with wild fluctuations in the output curve, in particular at the short end (where the tenors of the inputs are closer together). See Figure 4. Even though the discrete forwards are completely reasonable, lying in the range 4–8%, the interpolated curve is not.

The problem with many of the schemes we have seen so far is that we do not have instantaneous forward rates as inputs to a yield curve, we have (or can rearrange our inputs so that we have) discrete forwards for entire intervals. Many of the methods we have seen so far are implicitly treating discrete forwards not as a property of the

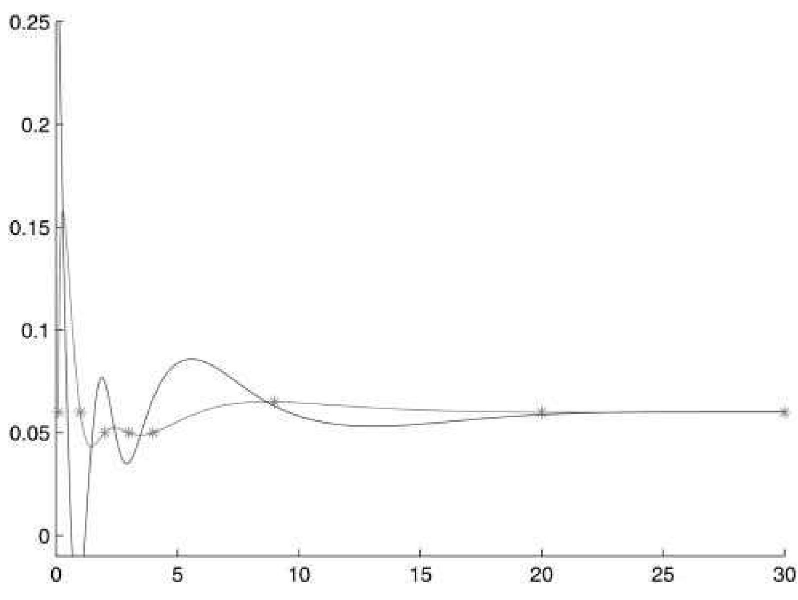

Figure 4. The quartic spline on forwards

entire interval, but as a property of the right endpoint of that interval, and ignore the interval itself. We will change focus appropriately in the following section.

Forward Monotone Convex Spline

We introduce a new method: this spline is constructed to preserve appropriate properties, principally among these the geometry of the inputs: the curve is locally monotone and convex if the inputs show the analogous discrete properties. Furthermore, if required, the curve is guaranteed to be positive if all the inputs are positive. Because the method will explicitly include the possibility of a zero day rate, we change notation slightly. Now we have terms $0=\tau_0, \tau_1, \ldots, \tau_n$ and the generic interval for consideration is $[\tau_{i-1}, \tau_i]$.

Construction of Suitable Discrete Forward Rates

Suppose the input at node i is f_i^d. Rather than interpolate as if f_i^d were the rate 'only' for time τ_i, we model f_i^d as belonging to the entire interval $[\tau_{i-1}, \tau_i]$, and the rate at the point τ_i as being

$$f_i = \frac{\tau_i - \tau_{i-1}}{\tau_{i+1} - \tau_{i-1}} f_{i+1}^d + \frac{\tau_{i+1} - \tau_i}{\tau_{i+1} - \tau_{i-1}} f_i^d, \quad \text{for } i = 1, 2, \ldots, n-1 \tag{30}$$

$$f_0 = f_1^d - \frac{1}{2}\left(f_1 - f_1^d\right), \tag{31}$$

$$f_n = f_n^d - \frac{1}{2}\left(f_{n-1} - f_n^d\right). \tag{32}$$

The interpolation algorithm will proceed on the rates f_i. For $i=1, 2, \ldots, n-1$ this choice amounts to interpolating f_i from the average values f_{i+1}^d and f_i^d at the midpoints of the adjacent intervals. The values f_0 and f_n were selected so that $f'(0)=0=f'(\tau_n)$.

For some emerging markets, we may know the overnight rate $f(0)$. If so, this should be used for the end-point in preference to (31).

The Basic Interpolator

We require an interpolation function that satisfies the following properties:

$$f(\tau_{i-1}) = f_{i-1}, \quad f(\tau_i) = f_i, \quad \frac{1}{\tau_i - \tau_{i-1}} \int_{\tau_{i-1}}^{\tau_i} f(t)\,dt = f_i^d \tag{33}$$

Let us postulate a quadratic of the form $K + Lx(\tau) + Mx(\tau)^2$; one quickly gets from (33) 3 linear equations in the unknowns K, L, M, which will solve as in (34):

$$f(\tau) = f_{i-1} - \left(4f_{i-1} + 2f_i - 6f_i^d\right)x(\tau) + \left(3f_{i-1} + 3f_i - 6f_i^d\right)x(\tau)^2 \tag{34}$$

$$= \left(1 - 4x(\tau) + 3x(\tau)^2\right)f_{i-1} + \left(-2x(\tau) + 3x(\tau)^2\right)f_i + \left(6x(\tau) - 6x(\tau)^2\right)f_i^d \tag{35}$$

where

$$x(\tau) = \frac{\tau - \tau_{i-1}}{\tau_i - \tau_{i-1}} \tag{36}$$

for i=1, 2, …, n.

Here and later we use the following simple fact: suppose $x := x(s) = \frac{s-\tau_{i-1}}{\tau_i - \tau_{i-1}}$ and $G' = g$. Then

$$\int_{\tau_{i-1}}^{\tau} g(x(s))\, ds = (\tau_i - \tau_{i-1})\left[G\left(\frac{\tau - \tau_{i-1}}{\tau_i - \tau_{i-1}}\right) - G(0)\right] \tag{37}$$

Beyond τ_n, we should use flat extrapolation: $f(\tau) = f(\tau_n)$ for all $\tau > \tau_n$.

In the next section we enforce monotonicity and convexity. Before doing this, however, let us note some properties of the basic interpolator.

First, the accuracy of the interpolator is $O(\Delta\tau)^2$ as $\Delta\tau \to 0$. This, because

(1) (30) can be viewed as first approximating $f(\tau)$ at the midpoints of the intervals by its average over the interval, and then linearly interpolating to find f_i at the end points of the interval;
(2) the value of any smooth function $f(\tau)$ at the midpoint of an interval is within $O(\Delta\tau)^2$ of the average value of the function over the interval;
(3) linear interpolation has an error of $O(\Delta\tau)^2$. Moreover, discount factors rely on the integrals $\int f(\tau')dt'$. Since $f(\tau)$ has an $O(\Delta\tau)^2$ error, the error in the discount factor is $O(\Delta\tau)^3$.

Second, (30) implies that f_i is between the average values of $f(\tau)$ on the adjacent intervals:

$$\min\left(f_i^d, f_{i+1}^d\right) \le f_i \le \max\left(f_i^d, f_{i+1}^d\right). \tag{38}$$

Third, if we re-write (34) and (30) as

$$f(\tau) = f_i^d - \frac{\tau_i - \tau_{i-1}}{\tau_i - \tau_{i-2}}\left(f_i^d - f_{i-1}^d\right)\left(1 - 4x + 3x^2\right) + \frac{\tau_i - \tau_{i-1}}{\tau_{i+1} - \tau_{i-1}}\left(f_{i+1}^d - f_i^d\right)\left(-2x + 3x^2\right) \tag{39}$$

where $x = x(\tau)$ as before, so $0 \leqslant x \leqslant 1$, we see that

$$-\frac{1}{3}\frac{\tau_i - \tau_{i-1}}{\tau_i - \tau_{i-2}} < \frac{\partial f}{\partial f_{i-1}^d} < \frac{\tau - \tau_{i-1}}{\tau_i - \tau_{i-2}} \tag{40}$$

$$0 < \frac{\partial f}{\partial f_i^d} < \frac{3}{2}, \tag{41}$$

$$-\frac{1}{3}\frac{\tau_i - \tau_{i-1}}{\tau_{i+1} - \tau_{i-1}} < \frac{\partial f}{\partial f_{i+1}^d} < \frac{\tau_i - \tau_{i-1}}{\tau_{i+1} - \tau_{i-1}} \tag{42}$$

So this interpolation method is stable.

Finally, (39) shows that if we change the value of f_i^d, then $f(\tau)$ would only change on the interval $\tau_{i-1} < \tau < \tau_i$ and the two immediately adjacent intervals: $\tau_{i-2} < \tau < \tau_{i-1}$ and $\tau_{i+1} < \tau < \tau_{i+2}$. So this method is local.

Enforcing Monotonicity and Convexity

We first analyse the monotonicity condition. Let us first focus on a single interval $\tau_{i-1}<\tau<\tau_i$. We require $f(\tau)$ to be monotone increasing in this interval if and only if

$$f_{i-1}^d < f_i^d < f_{i+1}^d$$

and to be monotone decreasing if and only if

$$f_{i-1}^d > f_i^d > f_{i+1}^d$$

Now, by considering (38), we see that a sufficient condition for our monotonicity requirements over the *ith* interval are:

$$f_{i-1} \le f_i^d \le f_i \Rightarrow f(\tau) \text{ is monotone increasing} \tag{43}$$

$$f_{i-1} \ge f_i^d \ge f_i \Rightarrow f(\tau) \text{ is monotone decreasing} \tag{44}$$

Clearly monotonicity is not possible if $f_{i-1} < f_i^d$ and $f_i^d > f_i$ or $f_{i-1} > f_i^d$ and $f_i^d < f_i$. Let us define

$$g(\tau) = f(\tau) - f_i^d \tag{45}$$

for the interval $[\tau_{i-1}, \tau_i]$. Denote $g_{i-1} =: g(\tau_{i-1})$ $g_i := g(\tau_i)$, and note that

$$\int_{\tau_{i-1}}^{\tau_i} g(\tau)d\tau = 0 \tag{46}$$

and via (35) we have

$$g(\tau) = g_{i-1}\left(1-4x+3x^2\right) + g_i\left(-2x+3x^2\right) \tag{47}$$

Now

$$\frac{\partial g}{\partial x} = g_{i-1}(-4+6x) + g_i(-2+6x) \tag{48}$$

In order to examine the extremum behaviour of g on [0, 1], we can simply analyse the behaviour at 0 and at 1. Note that

$$g'(0) = -4g_{i-1} - 2g_i$$
$$g'(1) = 2g_{i-1} + 4g_i$$

So, for example, if $g_{i-1}>0$, $g_i>0$ then $g'(0)<0$ and $g'(1)>0$, so g has a minimum on the interval. In full generality, the analysis breaks down into eight cases, where the values $g'(0)$ and $g'(1)$ are positive or negative, and where one (but not both) are zero. See Figure 5. The eight regions are the rays labelled (b), (d), (f), and (h) and the angled regions (a), (c), (e), (g) between them. The origin is excluded.[7]

Observe that

(a) If $g_{i-1}>-2g_i$ and $g_{i-1} > -\frac{1}{2}g_i$ then $g(x)$ has a minimum in $0<x<1$.
(b) If $g_{i-1}>0$ and $g_i = -\frac{1}{2}g_{i-1}$ then $g(x)$ has a minimum at $x=1$.
(c) If $g_{i-1}>0$ and $-2g_{i-1} < g_i < -\frac{1}{2}g_{i-1}$ then $g(x)$ is monotone decreasing.
(d) If $g_{i-1}>0$ and $g_i=-2g_{i-1}$ then $g(x)$ has a maximum at $x=0$.

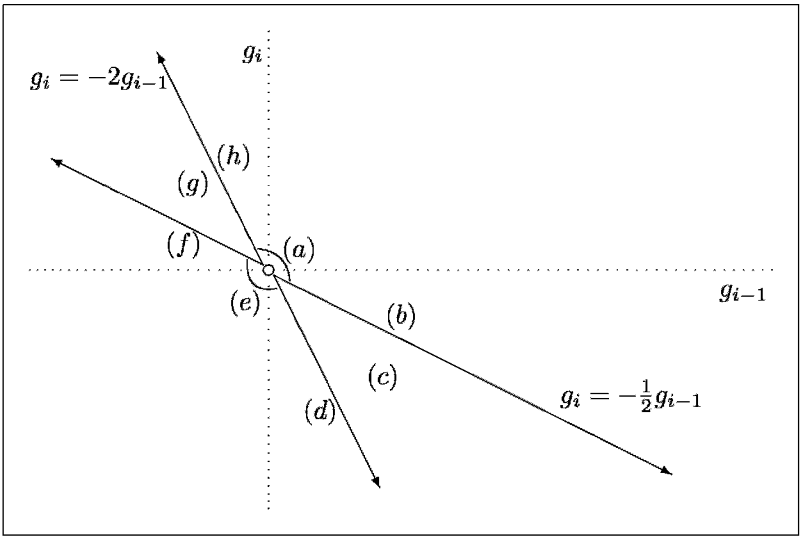

Figure 5. The possibilities for g

(e) If $g_{i-1} < -2g_i$ and $g_{i-1} < -\frac{1}{2}g_i$ then $g(x)$ has a maximum in $0 < x < 1$.
(f) If $g_{i-1} < 0$ and $g_i = -\frac{1}{2}g_{i-1}$ then $g(x)$ has a maximum at $x = 1$.
(g) If $g_{i-1} < 0$ and $-\frac{1}{2}g_{i-1} < g_i < -2g_{i-1}$ then $g(x)$ is monotone increasing.
(h) If $g_{i-1} < 0$ and $g_i = -2g_{i-1}$ then $g(x)$ has a minimum at $x = 0$.

The plane that has had the origin excluded is divided into eight new sectors (see Figure 6). We pair the sectors, leaving four named regions:

(i) $g_{i-1} > 0$, $-\frac{1}{2}g_{i-1} \geq g_i \geq -2g_{i-1}$ and $g_{i-1} < 0$, $-\frac{1}{2}g_{i-1} \leq g_i \leq -2g_{i-1}$

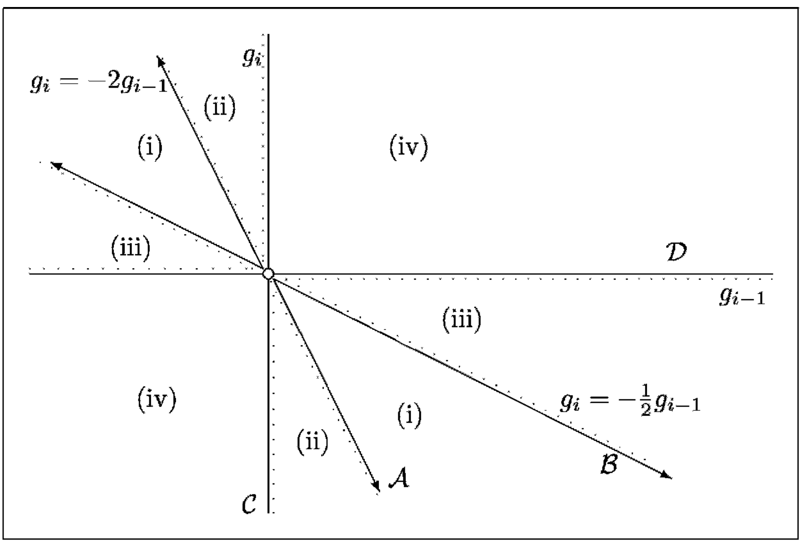

Figure 6. The reformulated possibilities for g

(ii) $g_{i-1}<0$, $g_i>-2g_{i-1}$ and $g_{i-1}>0$, $g_i<-2g_{i-1}$
(iii) $g_{i-1}>0$, $0 > g_i > -\frac{1}{2}g_{i-1}$ and $g_{i-1}<0$, $0<g_i<-\frac{1}{2}g_{i-1}$
(iv) $g_{i-1}\geqslant 0$, $g_i\geqslant 0$ and $g_{i-1}\leqslant 0$, $g_i\leqslant 0$

We now analyse these sectors in turn:[8]

(i) By (44) the interpolant needs to be decreasing, and the function already constructed can be used unaltered. Note that on the region boundary marked $\mathcal{A}$ we have $g(\tau)=g_{i-1}(1-3x^2)$ and on the region boundary marked $\mathcal{B}$ we have $g(\tau)=g_{i-1}\left(1-3x+\frac{3}{2}x^2\right)$.

(ii) By (43) we would like to arrange f to be non-decreasing, and the interpolant as defined so far fails this requirement (as we are in sector (a)). We wish to redefine our interpolant, ensuring the non-decreasing property, and retaining (46). The simplest and least intrusive fix is to insert a flat segment. So we take

$$g(\tau)=\begin{cases} g_{i-1} & \text{for } \quad 0\le x\le \eta \\ g_{i-1}+(g_i-g_{i-1})\left(\frac{x-\eta}{1-\eta}\right)^2 & \text{for } \quad \eta<x\le 1 \end{cases} \tag{49}$$

with η chosen to ensure that $g(\tau)$ averages to zero:

$$\eta=1+3\frac{g_{i-1}}{g_i-g_{i-1}}=\frac{g_i+2g_{i-1}}{g_i-g_{i-1}} \tag{50}$$

Note that $\eta\to 0$ as $g_i\to -2g_{i-1}$, so $g(\tau)$ is smooth across the boundary $\mathcal{A}$, as the interpolation formula reduces to $g(\tau)=g_{i-1}(1-3x^2)$ there.

(iii) Now by (44) we would like to arrange f to be non-increasing, and again the interpolant as defined so far fails this requirement. Here again we insert a flat segment. So we take

$$g(\tau)=\begin{cases} g_i+(g_{i-1}-g_i)\left(\frac{\eta-x}{\eta}\right)^2 & \text{for } \quad 0<x<\eta \\ g_i & \text{for } \quad \eta\le x<1 \end{cases} \tag{51}$$

with η again chosen to ensure that $g(\tau)$ averages zero:

$$\eta=3\frac{g_i}{g_i-g_{i-1}} \tag{52}$$

Note that $\eta\to 1$ as $g_i\to -\frac{1}{2}g_{i-1}$, so $g(\tau)$ is smooth across the boundary $\mathcal{B}$, as the interpolation formula reduces to $g(\tau)=g_{i-1}\left(1-3x+\frac{3}{2}x^2\right)$ there.

(iv) In this sector clearly $g(\tau)$ must have a minimum. At first glance the original quadratic suffices, but in fact we cannot use it for this sector because it does not reduce to (49) as $g_{i-1}\to 0$ (at $\mathcal{C}$) nor to (51) as $g_i\to 0$ (at $\mathcal{D}$). Nor does it make sense to patch in a flat segment. Instead we use two quadratics:

$$g(\tau)=\begin{cases} A+(g_{i-1}-A)\left(\frac{\eta-x}{\eta}\right)^2 & \text{for } \quad 0<x<\eta \\ A & \text{for } \quad x=\eta \\ A+(g_i-A)\left(\frac{x-\eta}{1-\eta}\right)^2 & \text{for } \quad \eta<x<1 \end{cases} \tag{53}$$

where we need to set

$$A = -\frac{1}{2}[\eta g_{i-1} + (1-\eta)g_i] \tag{54}$$

so that $g(\tau)$ averages zero as required. We now need to pick η. To match the preceding results, we need $A=0$ at $g_{i-1}=0$ and $g_i=0$. This means we need to choose

$$\eta = \begin{cases} 1 & \text{if} \quad g_{i-1} = 0 \\ 0 & \text{if} \quad g_i = 0 \end{cases}$$

We use the simplest choice

$$\eta = \frac{g_i}{g_i + g_{i-1}} \tag{55}$$

$$A = -\frac{g_{i-1}g_i}{g_{i-1} + g_i} \tag{56}$$

Enforcing Positivity of the Interpolant

If the application requires it (as is the case for yield curves, for example) we may guarantee that the instantaneous forward rates from the interpolation method are always positive.

In the eight sectors, the maximum and minimum values of $g(\tau)$ are:

$$\begin{array}{llll} g_{\min} = g_{i-1}, & g_{\max} = g_i, & \text{for} & g_{i-1} < 0, \quad g_i > 0 \\ g_{\min} = -\frac{g_{i-1}g_i}{g_{i-1}+g_i}, & g_{\max} = \max\{g_{i-1}, g_i\}, & \text{for} & g_{i-1} > 0, \quad g_i > 0 \\ g_{\min} = g_i, & g_{\max} = g_{i-1}, & \text{for} & g_{i-1} > 0, \quad g_i < 0 \\ g_{\min} = \min\{g_{i-1}, g_i\}, & g_{\max} = -\frac{g_{i-1}g_i}{g_{i-1}+g_i}, & \text{for} & g_{i-1} < 0, \quad g_i < 0 \end{array} \tag{57}$$

This is very promising. Whenever the data forces a maximum or minimum in the interval, the maximum deviation from the average value is $|g_{i-1}g_i/(g_{i-1}+g_i)|$, which is smaller than the smallest deviation of the endpoint.

For the applications of interest we may assume that all the average values f_i^d, are positive, for otherwise all interpolants $f(\tau)$ would be zero somewhere. By (38) all the f_i are positive, except possibly f_0 and f_n.

Thus $f(\tau)$ can only be negative if it has a negative local minimum within the inteval, which occurs in the quadrant $g_i>0$, $g_{i-1}>0$. Since $g_{\min}=-g_{i-1}g_i/(g_{i-1}+g_i)$, it suffices to require:

$$0 < f_{i-1} < 3f_i^d \text{ and } 0 < f_i < 3f_i^d \tag{58}$$

To remain a reasonable distance away from 0, we require the more stringent:

$$0 < f_{i-1} < 2f_i^d \text{ and } 0 < f_i < 2f_i^d \tag{59}$$

Thus, we apply where necessary the transformation[9]

$$f_0 \to \text{bound}\left(0, f_0, 2f_1^d\right) \tag{60}$$

$$f_i \to \text{bound}\left(0, f_i, 2\min\left(f_i^d, f_{i+1}^d\right)\right) \quad i = 1, 2, \ldots, n-1 \tag{61}$$

$$f_n \to \text{bound}\left(0, f_n, 2f_n^d\right) \tag{62}$$

If the application sets f_0, then we cannot apply the first shift.

Amelioration of the Interpolant

By shifting the f_i values we can make the interpolated curve smoother. The penalty is that the interpolated function will be less local; in some intervals $[\tau_{i-1}, \tau_i]$ the value of $f(\tau)$ might depend on f_j^d for $i-2 \leqslant j \leqslant i+2$. Thus in any particular application we must make a conscious decision as to whether we want the most locality or the best smoothness.

Let us consider the value $f_i \equiv f(\tau_i)$ between intervals i and $i+1$. Suppose first that $f_i^d > f_{i-1}^d$. If we also have $f_{i+1}^d > f_i^d$, then $f(\tau)$ is increasing in the interval i, and the smoothest results occur when f_i is in the range:

$$f_i^d + \frac{1}{2}\left(f_i^d - f_{i-1}\right) < f_i < f_i^d + 2\left(f_i^d - f_{i-1}\right) \tag{63}$$

This is our target range, the range in which we would prefer f_i to lie. Suppose now that $f_{i+1}^d < f_i^d$. Then $f(\tau)$ has a maximum in the interval. The maximum becomes steadily smaller as f_i increases towards f_i^d, but our interpolation function becomes increasingly asymmetric. In this case our target range is anything in

$$f_i^d - \frac{1}{2}\lambda\left(f_i^d - f_{i-1}\right) < f_i < f_i^d \tag{64}$$

where the parameter $0 \leqslant \lambda \leqslant 1$ determines the smoothness of the interpolated function. Experimentally $\lambda = 0.2$ seems to work well.

We cannot afford to have criteria for f_i which depend on values of $f(\tau)$ at other endpoints; this could lead to unpredictable non-locality and stability issues for marginal gains in smoothness. Instead we use the linear approximation to f_{i-1} as its proxy. Thus, to get good smoothness results for the interval i, we would like f_i to fall in the range

$$\begin{array}{ll} f_i^d + \frac{1}{2}\theta_i^- < f_i^- < f_i^d + \frac{1}{2}\theta_i^- & \text{if} \quad f_{i-1}^d < f_i^d < f_{i+1}^d \\ f_i^d - \frac{1}{2}\lambda\theta_i^- < f_i^- < f_i^d & \text{if} \quad f_{i-1}^d < f_i^d, f_i^d \geq f_{i+1}^d \end{array} \tag{65}$$

The targets for f_i if $f_i^d < f_{i-1}^d$ are obtained from similar considerations. Thus, considerations about the smoothness within interval i leads to the target range

$$f_{i,1}^{\min} \leq f_i \leq f_{i,1}^{\max} \tag{66}$$

$$\begin{array}{llll} f_{i,1}^{\min}=\min\left(f_i^d+\frac{1}{2}\theta_i^-, f_{i+1}^d\right), & f_{i,1}^{\max}=\min\left(f_i^d+2\theta_i^-, f_{i+1}^d\right) & \text{if} & f_{i-1}^d<f_i^d\leq f_{i+1}^d \\ f_{i,1}^{\min}=\max\left(f_i^d-\frac{1}{2}\lambda\theta_i^-, f_{i+1}^d\right), & f_{i,1}^{\max}=f_i^d & \text{if} & f_{i-1}^d<f_i^d, f_i^d>f_{i+1}^d \\ f_{i,1}^{\min}=f_i^d, & f_{i,1}^{\max}=\min\left(f_i^d-\frac{1}{2}\lambda\theta_i^-, f_{i+1}^d\right) & \text{if} & f_{i-1}^d\geq f_i^d, f_i^d\leq f_{i+1}^d \\ f_{i,1}^{\min}=\max\left(f_i^d+2\theta_i^-, f_{i+1}^d\right), & f_{i,1}^{\max}=\max\left(f_i^d+\frac{1}{2}\theta_i^-, f_{i+1}^d\right) & \text{if} & f_{i-1}^d\geq f_i^d>f_{i+1}^d \end{array} \tag{67}$$

where

$$\theta_i^- = \frac{\tau_i-\tau_{i-1}}{\tau_i-\tau_{i-2}}\left(f_i^d-f_{i-1}^d\right) \tag{68}$$

Similar considerations about the smoothness of $f(\tau)$ in the interval i+1 lead to the target ranges

$$f_{i,2}^{\min}\leq f_i\leq f_{i,2}^{\max} \tag{69}$$

$$\begin{array}{llll} f_{i,2}^{\min}=\max\left(f_{i+1}^d-2\theta_i^+, f_i^d\right), & f_{i,2}^{\max}=\max\left(f_{i+1}^d-\frac{1}{2}\theta_i^+, f_i^d\right) & \text{if} & f_i^d<f_{i+1}^d\leq f_{i+2}^d \\ f_{i,2}^{\min}=\max\left(f_{i+1}^d+\frac{1}{2}\lambda\theta_i^+, f_i^d\right), & f_{i,2}^{\max}=f_{i+1}^d & \text{if} & f_i^d<f_{i+1}^d, f_{i+1}^d>f_{i+2}^d \\ f_{i,2}^{\min}=f_{i+1}^d, & f_{i,2}^{\max}=\min\left(f_{i+1}^d+\frac{1}{2}\lambda\theta_i^+, f_i^d\right) & \text{if} & f_i^d\geq f_{i+1}^d, f_{i+1}^d<f_{i+2}^d \\ f_{i,2}^{\min}=\min\left(f_{i+1}^d-\frac{1}{2}\theta_i^+, f_i^d\right), & f_{i,2}^{\max}=\min\left(f_{i+1}^d-2\theta_i^+, f_i^d\right) & \text{if} & f_i^d\geq f_{i+1}^d\geq f_{i+2}^d \end{array} \tag{70}$$

where

$$\theta_i^+ = \frac{\tau_{i+1}-\tau_i}{\tau_{i+2}-\tau_i}\left(f_{i+2}^d-f_{i+1}^d\right) \tag{71}$$

To ameliorate the max's, min's, and general ugliness of the interpolant, we use the following procedure:

(a) add an additional interval at the beginning and the end:

$$\tau_{-1}=\tau_0-(\tau_1-\tau_0), \quad f_0^d=f_1^d-\frac{\tau_1-\tau_0}{\tau_2-\tau_0}\left(f_2^d-f_1^d\right) \tag{72}$$

$$\tau_{n+1}=\tau_n+(\tau_n-\tau_{n-1}), \quad f_{n+1}^d=f_n^d+\frac{\tau_n-\tau_{n-1}}{\tau_n-\tau_{n-2}}\left(f_n^d-f_{n-1}^d\right) \tag{73}$$

(b) Select the f_i's by linearly interpolating on the midpoints of the intervals:

$$f_i=\frac{\tau_i-\tau_{i-1}}{\tau_{i+1}-\tau_{i-1}}f_{i+1}^d+\frac{\tau_{i+1}-\tau_i}{\tau_{i+1}-\tau_{i-1}}f_i^d, \quad \text{for } i=0, 1, \ldots, n \tag{74}$$

Note that with the false intervals, this formula works for $i=0$ and $i=n$.

(c) For each $i=1, 2, \ldots, n-1$,

(i) if the target ranges overlap, define the common range

$$\max\left(f_{i,1}^{\min}, f_{i,2}^{\min}\right)\leq f_i\leq\min\left(f_{i,1}^{\max}, f_{i,2}^{\max}\right) \tag{75}$$

If f_i is outside this common range, make the minimum adjustment to f_i to place it in the common range:

$$\begin{aligned} &\text{if} \quad f_i < \max\left(f_{i,1}^{\min}, f_{i,2}^{\min}\right) \quad \text{set} \quad f_i = \max\left(f_{i,1}^{\min}, f_{i,2}^{\min}\right) \\ &\text{if} \quad f_i > \min\left(f_{i,1}^{\max}, f_{i,2}^{\max}\right) \quad \text{set} \quad f_i = \min\left(f_{i,1}^{\max}, f_{i,2}^{\max}\right) \end{aligned} \tag{76}$$

(ii) if the target ranges don't overlap, define the gap by

$$\min\left(f_{i,1}^{\max}, f_{i,2}^{\max}\right) \leq f_i \leq \max\left(f_{i,1}^{\min}, f_{i,2}^{\min}\right) \tag{77}$$

If f_i is below or above the gap, make the minimum adjustment to f_i to place it on the edge of the gap:

$$\begin{aligned} &\text{if} \quad f_i < \min\left(f_{i,1}^{\max}, f_{i,2}^{\max}\right) \quad \text{set} \quad f_i = \min\left(f_{i,1}^{\max}, f_{i,2}^{\max}\right) \\ &\text{if} \quad f_i > \max\left(f_{i,1}^{\min}, f_{i,2}^{\min}\right) \quad \text{set} \quad f_i = \max\left(f_{i,1}^{\min}, f_{i,2}^{\min}\right) \end{aligned} \tag{78}$$

(d) if now $\left|f_0 - f_0^d\right| > \frac{1}{2}\left|f_1 - f_0^d\right|$, replace f_0 with

$$f_0 = f_1^d - \frac{1}{2}\left(f_1 - f_0^d\right) \tag{79}$$

provided we don't know the value of f_0 (some markets explicitly quote f_0.)

(e) Similarly, if $\left|f_n - f_n^d\right| > \frac{1}{2}\left|f_{n-1} - f_n^d\right|$, replace f_n with

$$f_n = f_n^d + \frac{1}{2}\left(f_n^d - f_{n-1}\right) \tag{80}$$

(f) If the application requires $f(\tau)>0$ apply the transformations in (60), (61) and (62).

Stripping the BMA Basis Factor

As an example, consider stripping the Bond Market Association Municipal Swap Index basis factor. The BMA index is an index of tax-exempt Variable Rate Demand Notes issued by Municipalities. In a BMA basis swap, the BMA rate would be paid quarterly against a percentage of 3 month LIBOR for a specified number of years. In order to get the forward BMA one thus needs to bootstrap the LIBOR curve, fit this 'percentage' curve, and multiply. To fit this percentage curve is one of the toughest stripping problems around. Consider the input data for 9 May 2003, percentages for swaps of differing expiry dates, as follows:

15-May-03	0.88456260
7-Aug-03	0.88456260
6-Nov-03	0.92185792
6-May-04	0.94334882
12-May-05	0.86603225

11-May-06	0.80999250
10-May-07	0.81417719
8-May-08	0.83105534
7-May-09	0.80773549
6-May-10	0.79198677
12-May-11	0.82822284
10-May-12	0.82815909
9-May-13	0.82757452
7-May-15	0.83522826
10-May-18	0.83603242
11-May-23	0.84646560
12-May-33	0.86299257

The curve in Figure 7 results. After applying the amelioration step with $\lambda=0.20$, the curve in Figure 8 results.

This curve is clearly a smoother, better looking forward curve. However, by using amelioration the values of $f(\tau)$ in interval i depends on the average values f_k^d in intervals $i-2$, $i-1$, i, $i+1$, and $i+2$. The raw monotone convex spline without the amelioration step depends only on the average values in the neighbouring intervals $i-1$, i, and $i+1$. Whether the better smoothness properties make up for the poorer locality properties is an application-by-application judgement, which may depend on trader preferences. An interpolation application should offer both options.

Recovering the Rates from the Forwards

Suppose that the input rates were discrete forward rates for yield curve. We can now, given our functional form for f, derive the corresponding functional

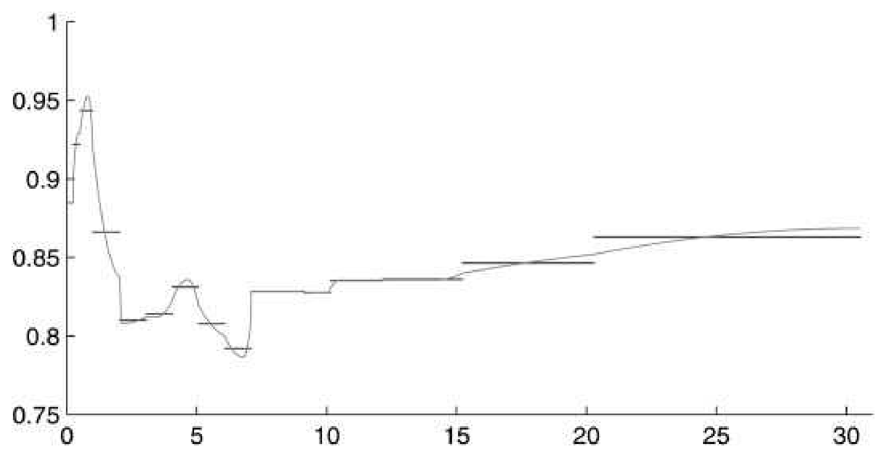

Figure 7. Monotone convex spline interpolation applied to BMA basis factor

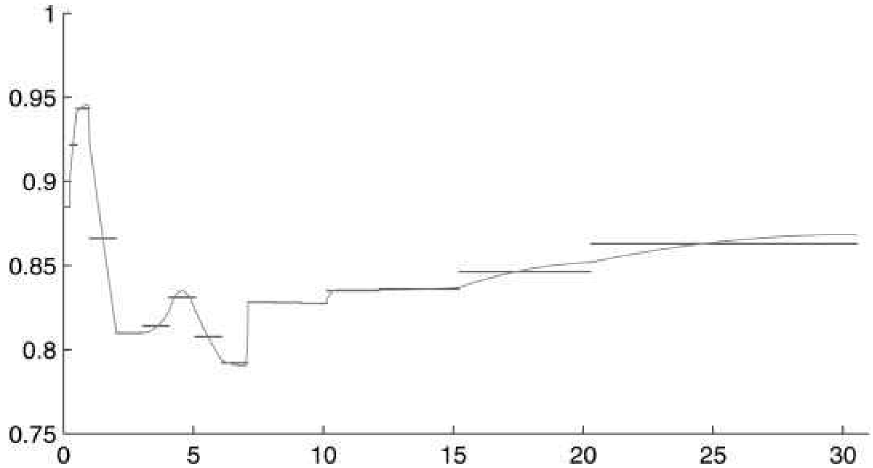

Figure 8. Monotone convex ameliorated spline interpolation applied to BMA basis factor

form for r:

$$\begin{aligned} r(\tau)\tau &= \int_0^\tau f(s)\,ds \\ &= \int_0^{\tau_{i-1}} f(s)\,ds + \int_{\tau_{i-1}}^{\tau} f(s)\,ds \\ &= r_{i-1}\tau_{i-1} + \int_{\tau_{i-1}}^{\tau} f(s)\,ds \\ &= r_{i-1}\tau_{i-1} + (\tau-\tau_{i-1})f_i^d + \int_{\tau_{i-1}}^{\tau} g(s)\,ds \end{aligned}$$

where i is found so that $\tau_{i-1} \leqslant \tau < \tau_i$ and $g(s) = f(s) + f_i^d$, as defined. We now apply the footnote equation (37). For example, if we are in region (i) or (v), then via (47) we have

$$\begin{aligned} \int_{\tau_{i-1}}^{\tau} g(s)\,ds &= \int_{\tau_{i-1}}^{\tau} g_{i-1}\left(1-4x(s)+3x(s)^2\right) + g_i\left(-2x(s)+3x(s)^2\right).\,ds \\ &= (\tau_i-\tau_{i-1})\left(g_{i-1}\left(x-2x^2+x^3\right)+g_i\left(-x^2+x^3\right)\right) \\ : &= I_\tau \end{aligned}$$

where $x = \frac{\tau-\tau_{i-1}}{\tau_i-\tau_{i-1}}$. The other cases are equally elementary.[10] Hence

$$r(\tau) = \frac{1}{t}\left(\tau_{i-1}r_{i-1} + f_i^d(\tau-\tau_{i-1}) + I_\tau\right) \tag{81}$$

The Performance of this Method on the Earlier Problematic Curve

How does this method perform on the curves which previously caused problems (with the cubic and quartic interpolation methods)? We show the bootstrapped forward curves for the monotone convex method and for the raw method.

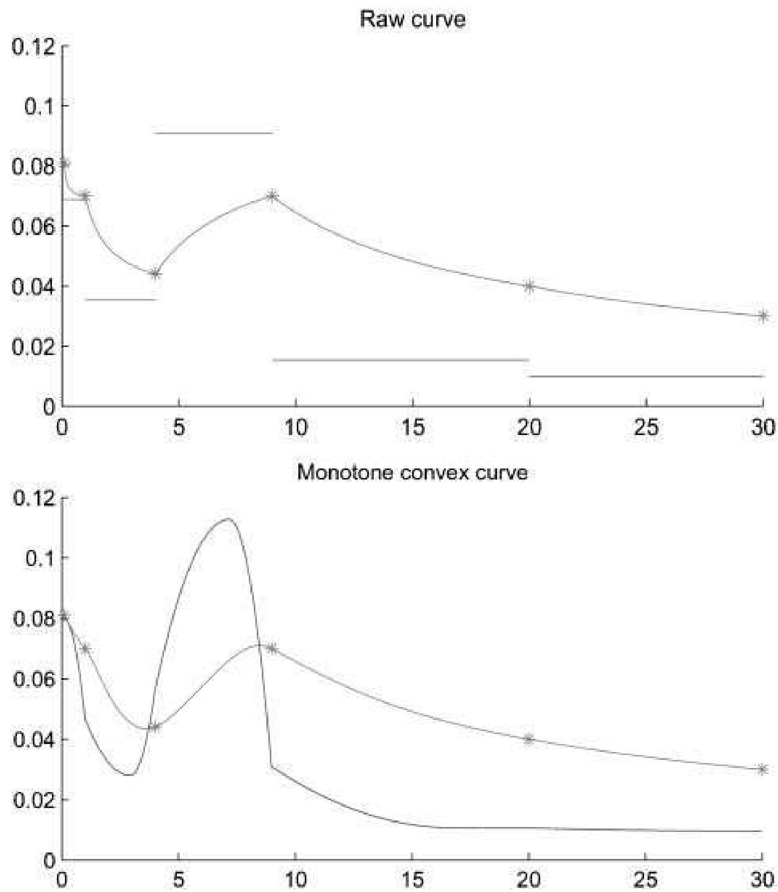

Figure 9. The forward raw and monotone convex splines applied to the curve seen earlier in Figure 2

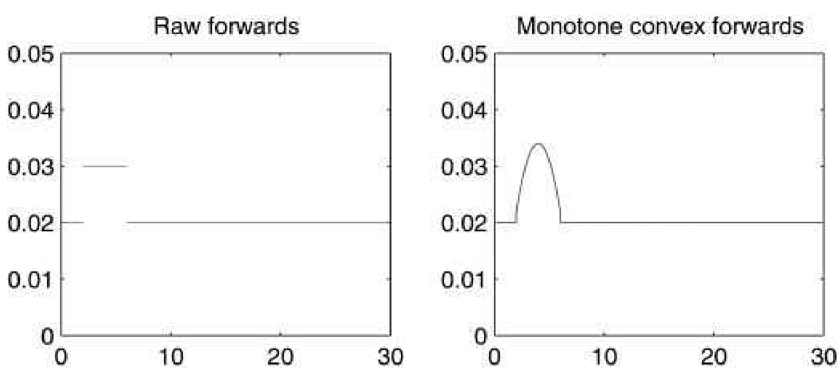

Figure 10. The forward raw and monotone convex splines applied to the forward curve seen Figure 3

A Minimalist Quadratic Interpolator

The Minimalist Interpolator

We introduce another new interpolation method. We seek an interpolator of the (forward) rates of the form

$$f(\tau) = a_i + b_i(\tau - \tau_{i-1}) + c_i(\tau - \tau_{i-1})^2 \quad \tau_{i-1} \leq \tau \leq \tau_i \tag{82}$$

for $1 \leqslant i \leqslant n$. The first requirement will be that

$$\frac{1}{\tau_i - \tau_{i-1}} \int_{\tau_{i-1}}^{\tau_i} f(\tau) d\tau = f_i^d \tag{83}$$

Let $h_i = \tau_i - \tau_{i-1}$ much as before. By elementary calculus, we find that this requirement is

$$f_i^d = a_i + \frac{1}{2} b_i h_i + \frac{1}{3} c_i h_i^2 \quad i = 1, 2, \ldots, n \tag{84}$$

This condition alone guarantees that the inputs to the curve are recovered by the interpolation scheme. The second requirement will be that the interpolant is continuous: clearly this simply means

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 \quad i = 1, 2, \ldots, n-1 \tag{85}$$

The Penalty Function

Thus we have $2n-1$ linear equations in $3n$ unknowns. The coefficients are finalised not by imposing a further $n+1$ linear constraints, as we have seen previously, but rather by minimizing an error function. Typical further linear constraints would be making the first or second derivatives continuous. Rather we will attempt to minimize a penalty function composed of sum of squares of the jumps in the first derivatives across the boundaries and sum of squares of the second derivatives, with a relative (user input) weighting between the two penalties.

Note that

$$f'(\tau) = b_i + 2c_i(\tau - \tau_{i-1}) \tag{86}$$

$$f''(\tau) = 2c_i \tag{87}$$

Let us now define the jumps in the first derivatives, and the weighted values in the second derivative, for attempted minimization:

$$J_{1,i} = b_{i+1} - b_i - 2c_i h_i \quad i = 1, 2, \ldots, n-1 \tag{88}$$

$$J_{2,i} = 2c_i h_i \quad i = 1, 2, \ldots, n \tag{89}$$

The penalty function, which we hope to minimize, is defined, for some prescribed $w \in (0,1)$, as

$$P_w = w \sum_{i=1}^{n-1} J_{1,i}^2 + (1-w) \sum_{i=1}^{n} J_{2,i}^2 \tag{90}$$

So we have a minimization problem subject to the constraints (84), (85), which suggests a Lagrange multiplier approach to the problem. We have $2n-1$ constraints in $3n$ variables, so $n+1$ degrees of freedom: thus, the setting is well formulated. So, for notational purposes, we define a $3n$ vector $\underline{x}=(a_1, b_1, c_1, a_2, \ldots, a_n, b_n, c_n)$; under this notational realignment we have

$$0 = x_{3i-2} + \frac{1}{2} x_{3i-1} h_i + \frac{1}{3} x_{3i} h_i^2 - f_i^d := g_i(\underline{x}) \quad i = 1, 2, \ldots, n \tag{91}$$

$$0 = x_{3i-2} + x_{3i-1} h_i + x_{3i} h_i^2 - x_{3i+1} := g_{n+i}(\underline{x}) \quad i = 1, 2, \ldots, n-1 \tag{92}$$

$$P_w = w \sum_{i=1}^{n-1} (x_{3i+2} - x_{3i-1} - 2x_{3i} h_i)^2 + 4(1-w) \sum_{i=1}^{n} x_{3i}^2 h_i^2$$

Now introducing Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_{2n-1}$ we have the solution

$$\nabla P_w = \sum_{k=1}^{2n-1} \lambda_k \nabla g_k$$

in other words

$$\frac{\partial P_w}{\partial x_i} = \sum_{k=1}^{2n-1} \lambda_k \frac{\partial g_k}{\partial x_i} \quad i = 1, 2, \ldots, 3n \tag{93}$$

Let $\underline{z}=(x_1, x_2, \ldots, x_{3n}, \lambda_1, \lambda_2, \ldots, \lambda_{2n-1})$ be the unknown and required vector. We have a system of $5n-1$ equations ((91), (92) and (93)) in the same number of unknowns $z_1, z_2, \ldots, z_{5n-1}$:

$$0 = [F]_i$$

$$= \begin{cases} -z_{3n+1} - z_{4n+1} & \text{if } i=1 \\ -z_{3n+(i+2)/3} + z_{4n-1+(i+2)/3} - z_{4n+(i+2)/3} & \text{if } i=4, 7, \ldots, 3n-5 \\ -z_{4n} + z_{5n-1} & \text{if } i=3n-2 \\ -2w(z_5 - z_2 - 2z_3 h_1) - z_{3n+1}\frac{1}{2}h_1 - z_{4n+1}h_1 & \text{if } i=2 \\ 2w\big(-z_{i-3} - 2z_{i-2}h_{(i-2)/3} + 2z_i + 2z_{i+1}h_{(i+1)/3} - z_{i+3}\big) & \\ \quad - z_{3n+(i+1)/3}\frac{1}{2}h_{(i+1)/3} - z_{4n+(i+1)/3}h_{(i+1)/3}^2 & \text{if } i=5, 8, \ldots, 3n-4 \\ 2w(z_{3n-1} - z_{3n-4} - 2z_{3n-3}h_{n-1}) - z_{4n}\frac{1}{2}h_n & \text{if } i=3n-1 \\ -4w\big(z_{i+2} - z_{i-1} - 2z_i h_{i/3}\big) + 8(1-w)z_i h_{i/3}^2 - z_{3n+i/3}\frac{1}{3}h_{i/3}^2 - z_{4n+i/3}h_{i/3}^2 & \text{if } i=3, 6, \ldots, 3n-3 \\ 8(1-w)z_{3n}h_n^2 - z_{4n}\frac{1}{3}h_n^2 & \text{if } i=3n \\ z_{3(i-3n)-2} + \frac{1}{2}z_{3(i-3n)-1}h_{i-3n} + \frac{1}{3}z_{3(i-3n)}h_{i-3n}^2 - f_{i-3n}^d & \text{if } 3n+1 \le i \le 4n \\ z_{3(i-4n)-2} + z_{3(i-4n)-1}h_{i-4n} + z_{3(i-4n)}h_{i-4n}^2 - z_{3(i-4n)+1} & \text{if } 4n+1 \le i \le 5n-1 \end{cases}$$

This can be written as a linear system $A\underline{z}=\underline{v}$ where

$$v_i = \begin{cases} f^d_{i-3n} & \text{if } 3n+1 \leq i \leq 4n \\ 0 & \text{otherwise} \end{cases}$$

This we solve with straightforward Gaussian elimination and back substitution, as seen in the algorithm in Press *et al.* (1992, Section 2.3).

The type of result one can obtain is as in Figure 11. However, the results are very sensitive to the choice of weight w.

First suppose we have curve with a sparse input set. Experimentally we find that $w<0.65$, say, results in curves which resemble piecewise linear forward curves, that is, dramatic zig-zags typically with negative forward rates. This is easily understood. In the one limit (as $w \downarrow 0$) we penalize the second derivatives very heavily, then we would have piecewise linear interpolation, which as we have seen is a disaster. In the other limit (as $w \uparrow 1$) we penalize the jumps in the first derivatives very heavily, so we have quadratic splines. Now quadratic splines have a bad reputation as interpolators, because after matching the endpoints one has one degree of freedom left, which is only assigned arbitrarily. (Compare this to cubic splines,where there are two degrees of freedom left, and these are used to match the derivatives at the endpoints.) However, our approach will be to assign that degree of freedom to ensure that (83) is satisfied. Hence, $w\approx1$ would be completely satisfactory. $w=1$ makes the matrix A singular.

On the other hand, if we have a dense input set, and choose a value of w close to 1, negative rates are very probable: now the zig-zag is in the curve itself. The splining is very smooth, in order to achieve this, some of the quadratics have enormous amplitude and the curve appears oscillatory where the input set is dense. Again, experimentally, we found this problem to emerge for $w>0.9$, say.

Thus, a default value of $w=0.8$ might be chosen. Communications with some practitioners since preprints of this work have started to appear suggest that this

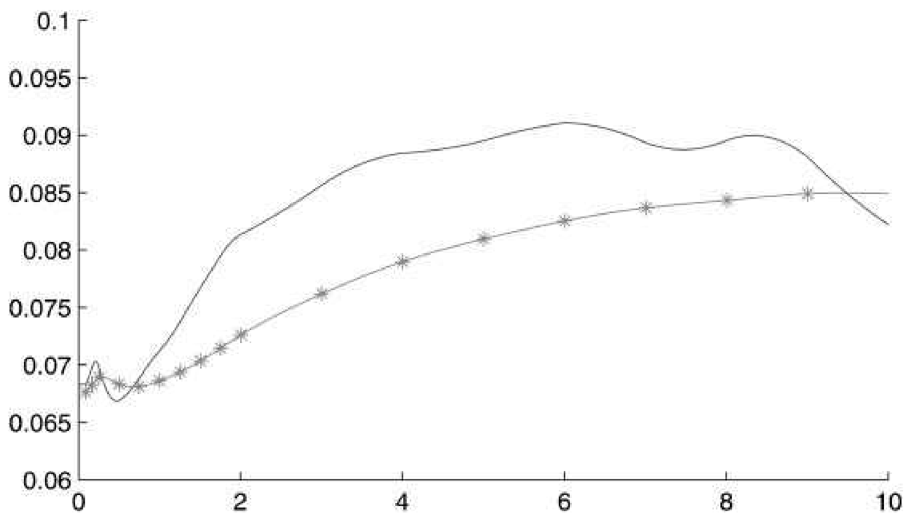

Figure 11. The minimal interpolator with $w=0.8$

method is very popular. However, this method does not guarantee positive forward rates, nor is it local. In fact, empirically its behaviour is often similar to the behaviour we have seen for some of the cubic or quartic spline algorithms. Nevertheless, this approach deserves attention, because it correctly interprets the information provided as information about intervals, not as information about the endpoint of the interval. Additional conditions to guarantee that each forward is positive on the domain of interest would need to be introduced, and penalty minimization under these conditions appears to be non-trivial.

Construction Quality Criteria

Our focus so far has mainly been on analysing if the curves generated 'look good' and if the forwards are positive and continuous. As promised earlier, we now consider the other three construction quality criteria we have isolated.

Localness of an Interpolation Method

Suppose we move the input at t_i. We wish to determine the interval (t_{i-l}, t_{i+u}) on which the yield curve values change. Clearly it is desirable that l and u be as small as possible. The results for yield curve interpolation are reported in Figure 12. For forward curve interpolation they are shown in Figure 13.

Some interesting observations arise. Among cubic and quartic splining methods, one of the Bessel methods or the monotone piecewise cubic method are the most attractive. The splines that use clamping at either end are all unatractive insofar as

Interpolation method	l	u
Linear on discount	1	1
Linear on rates	1	1
Raw	1	1
Linear on the log of rates	1	1
Natural cubic	$i-1$	$n-i$
Bessel	2	2
Financial	$i-1$	$n-i$
Quadratic natural	$i-1$	$n-i$
Bessel on cap functon	2	2
Monotone piecewise cubic	2	1
Quartic	$i-1$	$n-i$
Monotone convex (unameliorated)	2	2
Monotone convex (ameliorated)	3	3
Minimal	$i-1$	$n-i$

Figure 12. The localness indices for curve interpolation methods

Interpolation method	l	u
Raw	1	0
Quartic	$i-1$	$n-i$
Monotone convex (unameliorated)	2	1
Monotone convex (ameliorated)	3	2
Minimal	$i-1$	$n-i$

Figure 13. The localness indices for forward interpolation methods

this criterion is concerned. As expected, the simple methods described earlier are the most local. Finally, using amelioration in the monotone convex method compromises locality only very slightly.

Forward Stability

We want to measure how stable an interpolation method is – given a change in one of the inputs, how much can the output interpolated curve be changed? We measure this noise feature on the yield curve rates or on the forwards – both as inputs and as outputs. Appropriate norms for any interpolation method a given yield curve, or a given forward curve, would be

$$\|M(r)\| = \sup_t \max_i \left| \frac{\partial r(t)}{\partial r_i} \right|$$

$$\|M(f)\| = \sup_t \max_i \left| \frac{\partial f(t)}{\partial f_i^d} \right|$$

where the input rates are r_1, r_2, …, r_n or $f_1^d, f_2^d, \ldots, f_n^d$ respectively, and the bootstrapped curves are r and f respectively.

Except in some simple cases, these norms cannot be determined analytically, and would need to be estimated empirically. In those cases for any given curve this was achieved by measuring the maximum difference, in the supremum norm, between the original bootstrapped curve, and any of the $2n$ curves which arise when any of the n nodes are blipped up or down by one basis point. The difference was estimated by testing at discrete points along the entire curve, in steps of 0.01 of a year. These two measures (the theoretical derivative measure and the discrete measure) will be equivalent up to the scaling constant 10000 (since mathematically one basis point is 10000^{-1}), so in actual fact we will take the value found empirically and multiply it by 10000.

The second issue is to decide over what set of curves to calculate this norm. For this we simply considered a fairly large set of plausible curves that we have seen trading in various markets.

Clearly all of the earlier methods then have a norm of 1. The later arguments show that the monotone convex method initially has a norm of 1.5, although this will be compromised by the modifications which enforce monotonicity and convexity, and then by the amelioration of the interpolant. We found that, with an extensive set of plausible test curves, the norm for this method, unameliorated and with amelioration at $\lambda=0.2$, was never more than about 2.

All other methods, with the exception of the two quartic methods, had norms varying from about 1.2 to about 1.6. Perhaps the fact that the monotone convex method has a slightly worse norm is to be expected: more constraints are being imposed, so the norm should be higher. The two quartic methods had norms in excess of 100000 for our test set, and we have no reason to expect that, in the class of arbitrage free curves, these methods have finite norm. The instability observed was gross, and seemed to be the rule more than the exception.

Localness of Hedges

What we assume is that exactly the instruments which were used to bootstrap the yield curve are those that are available for creating hedge portfolios. Given a portfolio with certain risky cashflows, we wish to construct the portfolio of these underlying instruments that will hedge those cash flows.

We assume that the set of risky cash flows are all known, for example, swaps have been decomposed into a fixed coupon bond and a floating rate note, the latter of which has been discarded as riskless, and any flows with optionality have had some sort of Greek substitution as flows without optionality. Finally, we assume that we are in the world where our bootstrap prices all of the inputs exactly.

In order to proceed with this analysis, we first review the two classic approaches to hedging: bumping and waves.

In bumping, we form new curves indexed by j: to create the *jth* curve one bumps up the *jth* input rate by say 1 basis point, bootstraps the curve again, and reprices the risky portfolio with the new curve; the difference between the new and old price forms a vector ΔV, indexed by the bootstrapping instruments.

One now calculates the matrix P where P_{ij} is the change in price of the *jth* bootstrapping instrument under the *ith* curve. As each of the bootstraps prices the input set exactly to par, this matrix is diagonal: it is only the original instrument, which reprices now that that fair rate has changed. The hedge vector Q is chosen to satisfy $PQ=\Delta V$.

The portfolio Q perfectly hedges the given risky portfolio exactly under the case where the valuation curve moves from exactly one of the inputs being blipped by 1bp. One then hopes that the portfolio provides an adequate hedge against more general moves. Typically 1 bp is small enough to be in the linear regime, and so one is adequately protected against all small changes in the inputs.

In waves: one approach is to again form new curves, again indexed by i: the *ith* curve has the original yield curve incremented by a triangle, with left hand endpoint at t_{i-1}, height of say 1 or 10 basis points and apex at t_i, and right hand endpoint at t_{i+1}. (The first and last triangle will in fact be right angled, with their apex at the first and last time points respectively.) The matrix P is defined as before. This time we see that P is not diagonal, but is typically upper triangular – if there are no overlapping inputs then the *ith* curve is the same as the mark to market curve before time t_{i-1}, so if $i>j$, then the *jth* instrument does not change in value under this curve. (An example where we have no overlapping inputs would be where we have LIBOR instruments to 3 months, then 3×6, 6×9, 9×12, 12×15, 15×18 FRAs, and then annual swaps.

If we had in addition a 2×5 FRA say then the inputs would be overlapping. In this case, P will have a subdiagonal band of width 1; if we then added a 1×4 FRA the band width would increase to 2.) Also Q is defined as before. No problems arise with this setup, the matrix is invertible and has a low conditioning number: in the case the matrix is upper triangular, the eigenvalues are exactly the diagonal elements, and by inspection we found that the conditioning number for any interpolation method was typically as low as about 50. In the case where there were subdiagonal elements we used The MathWorks (2004) to test the conditioning number and again found it to be quite innocuous, never more than 100 say.

We will consider comparative tests of the algorithms developed here in some simple cases. We will suppose our risky portfolio is in fact a single fairly simple instrument. We consider a 30 year swap curve constructed with some LIBOR and FRA rates until 18 months, and then annual swap rates from 2 years onwards until 15 years, and then rates every 5 years. Our instrument to be hedged will be a $4\frac{1}{2}$ year swap. First, the hedge portfolio is constructed by bumping. Intuitively, one would expect the hedge to consist more or less of a position of half the nominal in a 4 year swap, and half in a 5 year swap. Typical results are in Figure 14. For clarity, we have truncated the graph to reflect only the region of interest; hedge leakage outside of the region shown was immaterial for all the methods displayed here. (Once again, the quartic methods showed grossly unreasonable behaviour, and have been eliminated from the analysis. Also, the performance of the minimal method is quite disappointing.)

Here we see a number of interesting features: all of the methods which involved clamping of any sort (cubic, financial, quadratic-normal, and as mentioned the quartic methods) have fairly significant leakage of hedge into buckets which are well away from 'the area of interest'. The local splining methods, and the Bessel methods, fair much better. Under raw interpolation the hedge portfolio consists entirely of 4

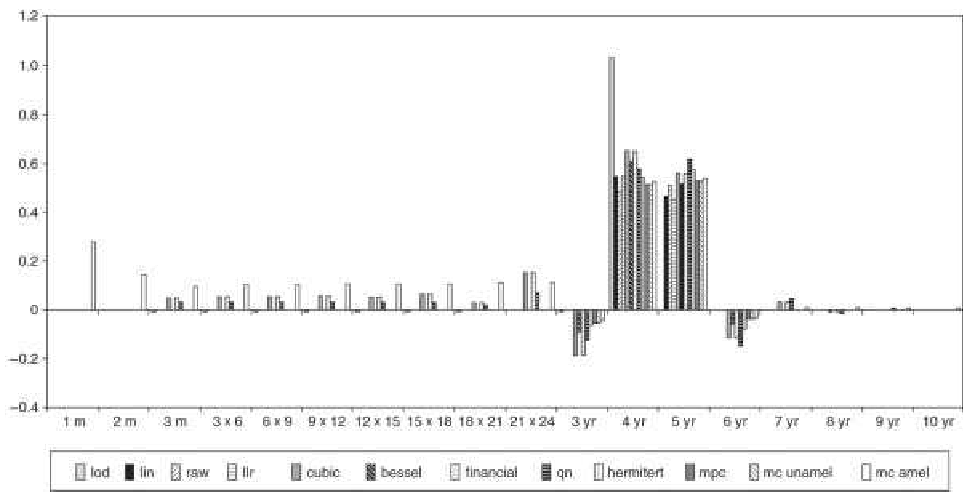

Figure 14. Hedge instruments required for hedging a 4.5 year swap under bumping

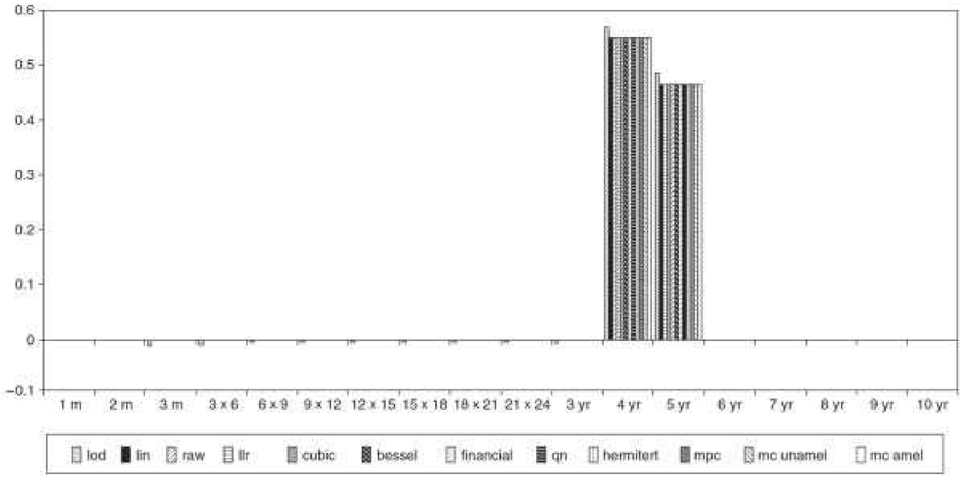

Figure 15. Hedge instruments required for hedging a 4.5 year swap under waves

and 5 year swaps; the unameliorated monotone convex method require a position in the 3 year swap, but nothing elsewhere. The ameliorated method shows significant hedge leakage.

Let us now construct the hedge portfolio with waves. Typical results are in Figure 15. Two points arise: in the first place, the hedging portfolios that arise under this approach are very intuitive and very stable. Secondly, no obvious criteria emerge to enable us to claim the superiority of one method over another.

Several questions should present themselves. For example, what happens to the hedge as we move into the swap, so the payment dates no longer coincide with the payment dates of instruments available in the market? Furthermore, what happens to hedge algorithms when we have overlapping input sets (say as the 1×4 or 2×5 FRAs mentioned earlier)? We intend to deal further with the issues of suitability and robustness of hedge algorithms in detail in forthcoming work.

Conclusions

We have reviewed many interpolation methods available and have introduced a couple of new methods. In the final analysis, the choice of which method to use will always be subjective, and needs to be decided on a case by case basis. But we hope to have provided some warning flags about many of the methods, and have outlined several qualitative and quantitative criteria for making the selection in which method to use.

Acknowledgement

We wish to thank an anonymous referee for several useful suggestions.

Notes

[1] By this we mean the intervals between the fixed payments of the swap, such as three or six months.

[2] For example, it is not reasonable to expect a bootstrapper to differentiate bonds according to tax status, rather, some value adjustment should be made *a priori* to the prices one set of bonds or the other, so that what is input can be considered to have the same tax status.

[3] To paraphrase the nomenclature of de Boor (1978, 2001): a cubic spline where the first derivative is known (in addition to the function values) is called a Hermite spline. The Bessel method is an intermediate method, where the derivatives are estimated from the function values, and then the Hermite method is applied.

[4] A matrix in a-1-b bandwidth form means that the entry in the *ith* row and *jth* column in this representation actually lies in the *ith* row and $i+j-a-1th$ column in the canonical matrix representation. Note that the entries in the top left and bottom right of the bandwidth matrix are redundant, they can be set to anything. This redundancy is denoted by a $\times$.

[5] Why not $f(r)=a_i+b_i(r-r_i)+c_i(r-r_i)^2+d_i(r-r_i)^3+e_i(r-r_i)^4$? We would prefer on principle this format. However, the bandwidth matrices which result are very unwieldy.

[6] This observation is consistent with Adams (2001) where, after interpolating the forward curves, one additional piece of information is needed to recover the interpolatory function on the yields i.e. the method of Adams (2001) has one remaining degree of freedom.

[7] If $g_{i-1}=0=g_i$, then the curve is flat, and $g=0$.

[8] It is advisable to ensure no problems with code execution ('division by zero') to trap the cases $x(r)=0$ and $x(r)=1$ up front: in these cases, $g(r)=g_{i-1}$ and $g(r)=g_i$ respectively.

[9] The bound function is given by bound(a, x, b)=min(max(a, x), b).

[10] The same trapping as before has $I_{\tau_{i-1}}=0=I_{\tau_i}$.

References

Adams, K. (2001) Smooth interpolation of zero curves, *Algo Research Quarterly*, 4(1/2), pp. 11–22.

Adams, K. J. and van Deventer, D. R. (1994) Fitting yield curves and forward rate curves with maximum smoothness, *Journal of Fixed Income*, June 4(1), pp. 52–62.

Burden, R. L. and Faires, J. D. (1997) *Numerical Analysis*, 6th edn (Brooks Cole).

Cox, J. C. *et al.* (1985) A theory of the term structure of interest rates, *Econometrica*, 53, pp. 385–407.

de Boor, C. (1978, 2001) *A Practical Guide to Splines: Revised Edition*, *Vol. 27 of Applied Mathematical Sciences* (New York: Springer-Verlag).

Fritsch, F. and Butland, J. (1980) An improved monotone piecewise cubic interpolation algorithm, *Lawrence Livermore National Laboratory Preprint UCRL.*

Heath, D. *et al.* (1990) Bond pricing and the term structure of interest rates: a discrete time approximation, *Journal of Financial and Quantitative Analysis*, 25, pp. 419–440.

Hull, J. (2002) *Options, Futures, and Other Derivatives*, 5th edn (Prentice Hall).

Hull, J. and White, A. (1990) Pricing interest rate derivative securities, *Review of Financial Studies*, 3.

Hyman, J. M. (1983) Accurate monotonicity preserving cubic interpolation, *SIAM Journal on Scientific and Statistical Computing*, 4(4), pp. 645–654.

Lim, K. and Xiao, Q. (2002) Computing maximum smoothness forward rate curves, *Statistics and Computing*, 12, pp. 275–279.

Lin, B.-H. (2002) Fitting term structure of interest rates using B-Splines: the case of Taiwanese government bonds, *Applied Financial Economics*, 12(1), pp. 57–75.

McCulloch, J. H. and Kochin, L. A. (2000) The inflation premium implicit in the US real and nominal term structures of interest rates, Technical Report 12, Ohio State University Economics Department. *http://www.econ.ohio-state.edu/jhm/jhm.html

Press, W. H. *et al.* (1992) *Numerical Recipes in C: the Art of Scientific Computing*, 2nd edn (Cambridge University Press).

Quant Financial Research (2003) The BEASSA zero coupon yield curves: technical specifications, Technical report, Bond Exchange of South Africa. *http://www.bondex.co.za/pricing/BEASSA_Zero_Curves_Tech_Specs_240504.pdf

Ron, U. (2000) A practical guide to swap curve construction. Technical Report 17, Bank of Canada. *http://www.bankofcanada.ca/en/res/wp00-17.htm

The MathWorks (2004) *Matlab Version 7*, The MathWorks, Inc.

van Deventer, D. R. and Inai, K. (1997) *Financial Risk Analytics - A Term Structure Model Approach for Banking* (Irwin: Insurance and Investment Managemen).

Vasicek, O. A. (1977) An equilibrium characterisation of the term structure, *Journal of Financial Economics*, 15.

Zangari, P. (1997) An investigation into term structure estimation methods for RiskMetrics, *RiskMetrics Monitor*, Third Quarter, pp. 3–48.