

Heath, Jarrow, and Morton Interest Rate Modelling Using

Principal Component Analysis¹

Cédreece Tamagushiku

March, 2006

¹ This finc590 is put forward as part of requirements of the Master of Business degree at the University of Otago. Many thanks must be given to Dr. Alan Stent for his guidance, patience and supervision of this research.

Heath, Jarrow, and Morton Interest Rate Modelling Using Principal Component Analysis

Cédreece Tamagushiku

Abstract

The purpose of this paper is to investigate the performance of three different models in the pricing of call options on ninety-day bank bill futures traded on the Sydney Futures Exchange between 1993 and 2000. The three models analysed are embedded into the Heath, Jarrow, and Morton framework namely; the one, two, and three factor models. Principal Components Analysis was applied in order to provide the forward rate volatility functions necessary to implement several popular multi-factor versions of the Heath, Jarrow, and Morton model. Results showed that the three-factor model consistently outperforms the one and two-factor models. Also the pricing errors are positively correlated with the time to maturity of the option and that no real relationship existed between the errors of one and two-factor models and the date and the moneyness of the options. Although three-factor models exhibited lower errors as time progressed.

Abstract	2
1. Introduction	4
2. Literature Review.....	6
3. Principal Component Analysis.....	7
3.1 Principal Component Properties	9
4. Heath, Jarrow, and Morton Framework	9
4.1 Parameter Estimation	10
4.2 One-Factor Model	10
4.3 Two and Three-Factor Models.....	10
5. The Sydney Futures Exchange.....	11
5.1 Futures Style Margining.....	11
5.2 Australian 90 Day Bank Accepted Bill Market	12
6. Data	12
7. Methodology	12
7.1 Exponential Smoothing.....	13
7.2 Building the HJM Tree.....	15
7.3 Option Valuation.....	16
7.4 Error Calculation.....	17
7.5 Error Analysis	17
8. Empirical Results	18
9. Conclusion.....	19
10. References	20
11. Figures.....	21
11.1 Figure 1	21
11.2 Figure 2	28

1. Introduction

There has been an accelerated growth in the use of interest rate instruments in fixed income portfolios and risk management systems over the past two decades. With this proliferation has emerged the necessity to formulate reliable models that accurately price and hedge such interest rate sensitive cash flows.

Single factor models have often been implemented due to their computational efficiency and reasonable approximation to actual historical data. Currently academics and practitioners alike are grappling empirically to determine the number of factors necessary in order to develop a parsimonious model for pricing and hedging interest rate contingent claims.

The aim of this study is to present a one, two, and three-factor Heath, Jarrow, and Morton model where each of the factors has been determined by Principal Component Analysis. Furthermore, it is to test and verify an algorithm to price the options on the 90-day bank bill futures using Heath, Jarrow, and Morton trees generated by the Matlab Financial Derivatives software. This software provides the functions to generate the trees and to access nodes, paths and probabilities thereon. The algorithm to price the specific derivative was being written for another purpose.²

We test the effectiveness of these three models through an analysis of their respective errors with an overall goal to examine whether the three-factor Heath, Jarrow, and Morton model adds any valuation capabilities to the pricing of these options over and above simpler one and two-factor models.

Essentially two branches of models exist in the literature tree of interest rate modelling. The first branch of models proposed to model the interest rate directly and they are known as the short rate models. The second branch of models encompasses the Heath, Jarrow, and Morton framework which attempts to model the evolution of the entire forward rate curve. This project has a direct interest in the latter of the two branches. The advantage of the Heath, Jarrow, and Morton type methodology over the

² A study described in Dempster, Chaput and Stent

short rate approach is that it achieves an automatic fit to the yield curve, whereas the short rate models require additional computation.

The Heath, Jarrow, and Morton framework requires an initial term structure of interest rates and a volatility specification of the volatility of the forward rates to build a tree representing the evolution of the forward interest rates. The Heath, Jarrow, and Morton class of models for pricing interest rate claims is based on the no arbitrage principle.³ These models require only the specification of the form of the volatility function of forward interest rates along with the initial term structure of interest rates as inputs.

The volatility specifications will be estimated from a time series of the term structures of interest rates. This approach adopted estimates the parameters historically rather than implicitly and follows the work carried out by Buhler et al. (1999). They undertake a “global” rather than a “local” test which is more appropriate for comparing the present models. The latter of the two are more relevant for measuring the success of model calibration and are important for the practical application of a model, but this is not the objective of the present project.

The structure of this paper is as follows: the following section provides some background on the previous research undertaken in this area. Section 3 describes Principal Component Analysis which is followed by a section on the Heath, Jarrow, and Morton Framework and a brief part on the Sydney Futures Exchange. Section 6 is a description of the data used in this analysis. The methodology is outlined and carried out in section 7. The paper then goes on to discuss the findings of the study and concludes in section 9 with the conclusions of the paper.

³ This constraint prohibits intertemporal arbitrage.

2. Literature Review

The literature on interest rate modelling is dominated by two main groups of models; namely the forward rate and short rate models. This research has a direct interest in the former of the two. The arbitrage free approach to pricing interest rate derivatives has its origin in Ho and Lee (1986) who in a binomial framework derived a model from the term structure of forward rates.

In this seminal work from Ho and Lee (1986) they showed how an interest rate model could be developed so that it is consistent with any specified initial term structure. They take the initial term structure as given and model subsequent term structure movements that are constrained to being arbitrage free. The basic Ho and Lee model is a one-factor which keeps the volatility specification constant through time.

Due to the poor performance of one-factor models in capturing the empirical dynamics of short rates, this gave rise to the ground-breaking work of Heath, Jarrow, and Morton (1992). Heath, Jarrow, and Morton extended and generalised this model to permit more than one factor to influence the forward rates. The interest rate volatility is permitted to change across time and so requires a specification of how the volatility changes in which they proposed several functional forms.

Buhler et al. (1999) carry out an empirical study of one and two-factor Heath, Jarrow, and Morton type models along with one and two-factor inversion type models. They analyse these models on German interest rate warrants over the period from 1989 to 1993. Buhler et al. (1999) estimated the parameters for their models from historical time series of observations from the previous nine months. Factor analysis was performed in order to determine the parameters for the two-factor models. The volatility parameters were estimated directly from the volatilities of the two factors and the corresponding factor loadings. Their study produced the surprising result that the one-factor Heath, Jarrow, and Morton outperformed the two-factor model. It was stated that this could be due to the incorrect estimation of the factor loadings of the second factor.

Amin and Morton (1994) analysed six specifications for the volatility function of the forward rates in a Heath, Jarrow, and Morton framework. Using Eurodollar futures and options during the period from 1987 to 1992 they found that single-factor Heath, Jarrow, and Morton models fared well in valuing short term options with an average absolute pricing error ranging from 1.57 to 2.23 basis points. To further validate our choice to pursue a global test, they found that the sensitivity of any particular models errors was greater with respect to the number of parameters than any other factor.

Flesaker (1993) tested the one-factor Ho and Lee (1986) model, which is the constant volatility version of the Heath, Jarrow, and Morton model. This analysis was carried out on Eurodollar futures options and found that call options had an average absolute pricing error of 3.33 basis points.

Allen and Chau (1999) analysed the Heath, Jarrow, and Morton model along with others on the Australian bank bill market during 1996. Their findings indicate that the Heath, Jarrow, and Morton model outperforms the rest based on their respective errors.

3. Principal Component Analysis

Principal Component Analysis (PCA, hereafter) is a multivariate procedure which combines two or more correlated variables into a smaller number of factors or principal components. Essentially, a set of correlated variables are transformed into a set of uncorrelated factors which are ordered by reducing variability. The main use of PCA is to reduce the dimensionality of a data set while retaining as much information as is possible. This is achieved as the factors are linear combinations of the original variables and so the last of these variables can be removed with minimum loss of real data.

As the interest rates in a yield curve tend to be highly collinear, only a few sources of information need to be considered. This allows the PCA to be utilised when modelling the term structure of interest rates. PCA is a recommended way of modelling the volatility structure of interest rates in the Heath, Jarrow, and Morton framework. The procedure can be seen as a rotation of the original axes to new positions in the space set by the original variables. There will now be no correlation

between the new variables defined by this rotation. The first new variable contains the maximum amount of variation; the second new variable contains the maximum amount of variation unexplained by the first and also orthogonal to the first, and so on.

Much empirical work has already been carried out on PCA given its introduction to financial modelling in the early 1980's. Avellaneda and Laurence (2000), Rebonato (1998), James and Weber (2000) have found that two factors explain about 95% of the variation in the term structure movements while three factors explain approximately 99%. The rest of the variation is generally regarded as noise

Litterman and Scheinkman (1991) use PCA on the unconditional covariance matrix of bond returns on U.S. treasury notes. They found that the first three principal components are significant and correspond to level, steepness, and curvature changes in the term structure of interest rates. Adding more factors to this model only slightly increases the explained variation, and the factor loadings for the extra factors were small and insignificant.

Furthermore, consistent with international findings Sherris (1995) established that three factors explain most of the historical yield curve variation in Australia. He carried out an empirical analysis on the nature of the factors that explain the amount of variation inherent in the Australian bond market yields.

To this end, the analysis deals with only the first three principal components. More principal components are not considered as this is not a study to examine whether three factors are sufficient in explaining the total variation in the term structure. If it were the purpose of this paper then four factors would need to be investigated to see if three were adequate. As stated earlier the intent is to evaluate the whether the three-factor Heath, Jarrow, and Morton model is superior to the simpler one and two-factor models.

A word of caution must be conditioned at this point. Although as much as 99% of yield curve variation can be explained with as little as three principal components, one must be aware that this is in reference to the historical term structure data and a reader should take heed in presuming a forecast of future variations.

3.1 Principal Component Properties

Principal components have several appealing features such as their intuitive interpretation. Principal components can be viewed as explaining different types of changes in the shape of the term structure. The first principal component affects interest rates at all maturities by a similar amount and in the same direction hence its interpretation as a parallel shift. This parallel shift contributes a majority of the variation in the term structure at approximately 80%.

The second principal component will have opposing affects on short and long end interest rates and so is described as a tilting of the yield curve. The third principal component has a similar affect on short and long end interest rates and will have an opposite affect on medium interest rates. It is for this reason that the third factor is described as having a curvature affect. The second and third factors together add a further 15-19% of explained variation. This highlights the importance of considering models with greater than one factor.

4. Heath, Jarrow, and Morton Framework

The Heath, Jarrow, and Morton (HJM, hereafter) framework refers to a class of models that are derived by directly modelling the dynamics of the instantaneous forward rates. The fundamental insight of this framework is to recognise that there exists a specific relationship between the drift and volatility parameters of the forward rate dynamics in a no arbitrage world (HJM, 1992). HJM were instrumental in allowing the forward rates become the state variables in the model as opposed to quantities derived from the spot rates.

Models of the HJM type are based on the dynamics of the instantaneous forward rates. The critical factor in determining the behaviour of the model is the form of the volatility specification. The model begins with a fixed number of unspecified factors which are driving the dynamics of the instantaneous forward rates. These multiple sources of uncertainty can be implemented with little difficulty and it has been shown already that all other models can be embedded within an HJM framework.

4.1 Parameter Estimation

We specify the forward rate volatility structure using PCA by estimating the parameters using historical series of interest rates rather than using implied volatility calibration. This removes a calibration process that may systematically favour one model over another and so indicates a global rather than a local test such as that in Buhler et al (1999).

4.2 One-Factor Model

The simplest of the three models is the one-factor HJM model. One-factor models are extremely popular due to their simplicity and mathematical tractability. The one factor model essentially assumes that the term structure is affected by a single source of uncertainty. This single source as represented by the first principal component captures a parallel shift in the term structure. This parallel shift makes up about 80% of all term structure variation. If the changes in the yield curve were perfectly correlated then only one factor would be required to explain the historical data and so is perhaps not surprising that the one-factor model fail in their ability to match the historical data.

4.3 Two and Three-Factor Models

Models that are driven by a single source of uncertainty can occasionally be insufficient in describing the complete evolution of term structure movements. Multi-factor models can provide a more realistic description of the transition behaviour of the term structure of interest rates and so we consider both two-factor and three-factor HJM models in this study. The two-factor HJM model is comprised of the first principal component along with the second that represents a tilting of the yield curve. In essence, any changes to the yield curve that correspond to either a parallel shift or a tilting will be captured by this particular model.

The three-factor HJM model is comprised of the first and second principal components along with the third that represents a convexity change of the yield curve.

In this case any changes to the yield curve that correspond to either a parallel shift, a tilting, or a convexity change will be captured by the three factor model. This model is able to account a far richer array of changes which is balanced by its complexity and implementation of the model.

5. The Sydney Futures Exchange

The Sydney Futures Exchange (SFE, hereafter) product under examination is the Australian 90-Day Bank Accepted Bill futures options. They are available in eight quarter months with strike prices set at 0.25% intervals. Exercising a futures option results in either a long or short futures position. When the holder exercises a futures call option, he buys the underlying futures contract at the strike price which is known as taking a long futures position. The writer of the call option must in turn sell which is known as taking a short futures position in the underlying futures contract. As they involve physical delivery, the options mature one week prior to the futures expiry to allow traders time to close out or adjust positions. Their primary advantages over the underlying cash market stems from the liquidity of futures market, the leveraging benefits, and the ability of investors to alter their terminal payoffs in their portfolios.

5.1 Futures Style Margining

All futures options traded at SFE are American in style and are margined by the SFE Clearing Corporation⁴. An initial margin is deposited and the option contract is marked to market at the end of each day in the same way that futures positions are maintained.

This futures style margining has fundamental implications on the pricing of these options. Firstly, Lieu (1990) found that when an option is subject to futures style margining it will never be optimal for an investor to early exercise either a put or a call option. Secondly, option premiums could possibly be higher under a futures style system because sellers of the option would demand a higher price to compensate them for the loss of interest on the full premium. Conversely, buyers of the option would be

⁴ The SPAN margining system has been in operation since 1994.

willing to pay this higher price because they would be gaining such interest income. Thirdly, the settlement has implications when building and folding back through the trinomial tree. Generally the node value is discounted back through the tree but because the entire premium is not paid up front this approach is therefore unnecessary.

5.2 Australian 90 Day Bank Accepted Bill Market

The underlying cash market is that of the Australian 90-day Bank Accepted Bills. They are Australia's benchmark indicator for short term interest rates. The bills are negotiable short term securities which represent a promise to pay the full face value of A\$1,000,000 at maturity.

6. Data

This study requires the use of two sets of data; interest rate and option price data. The interest rate data used is based on the Australian dollar yield curve and is comprised of discrete annualised spot rates of 22 different maturities ranging from 1-day through to 30-year maturities. They are zero coupon bond interest rates which were stripped from coupon bond prices and sourced by Westpac Institutional Bank. The data with approximately 2000 observations spans the period of 04 January 1993 to 15 November 2000. The option price data was collected from the Sydney Futures Exchange website ⁵ and spans the same time period. It is comprised of contemporaneous observations of call settlement values with their respective futures prices as their values must be observed in conjunction with each other. The strike prices are set at 0.25% intervals and range from 87.50 to 98.50 and options have standard quarter month expiration periods of March, June, September and December.

7. Methodology

This section presents the methodology undertaken in estimating the option values and carrying out the analysis on the associated errors which closely follows that of Buhler et al. (1999). The methodology began with the computation of the principal

⁵ <http://www.sfe.com.au>

components which then enabled the building of the tree. From here we were able to price the options and undertake further analysis on their respective errors.

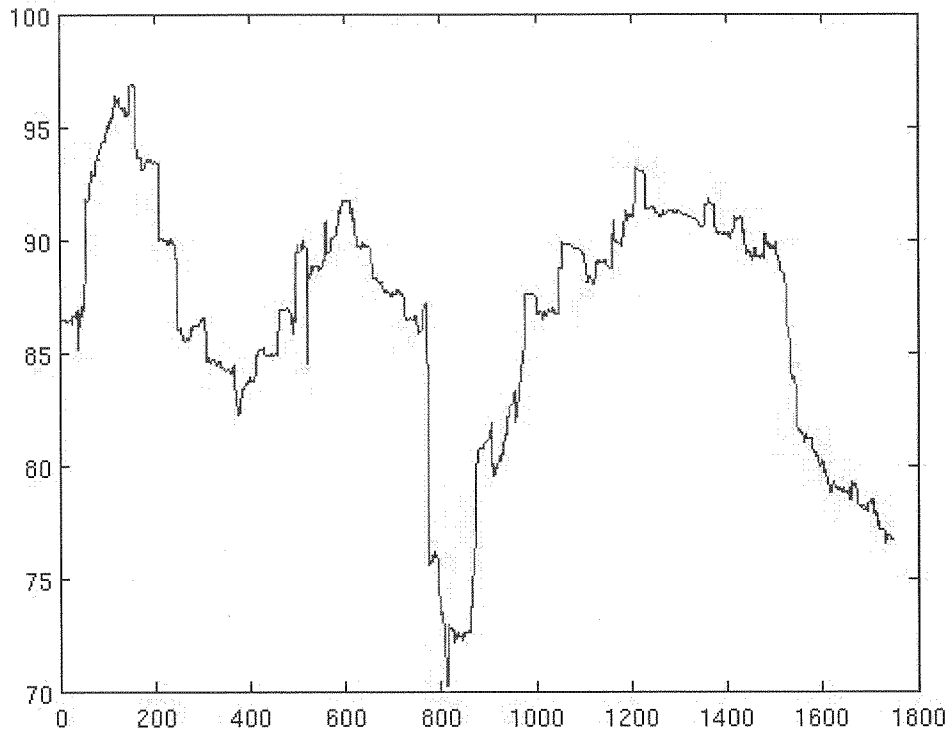
7.1 Exponential Smoothing

Principle components were fitted to a moving cross-section of interest rates, which is itself moved over time as a moving window. For year t the observations for year t and previous years were in the window. Two procedures were considered to obtain the length of the window – exponential smoothing and moving averages.

The exponentially weighted moving average approach to characterising volatility is an example of exponential smoothing. Exponential smoothing techniques employ one or more exponential smoothing parameters to give more weight to recent observations and less weight to older observations, in an attempt to respond dynamically to the changing value of the time series. The smoothing process is exponential because the weights employed are not arithmetic but, instead lie along an exponential curve.

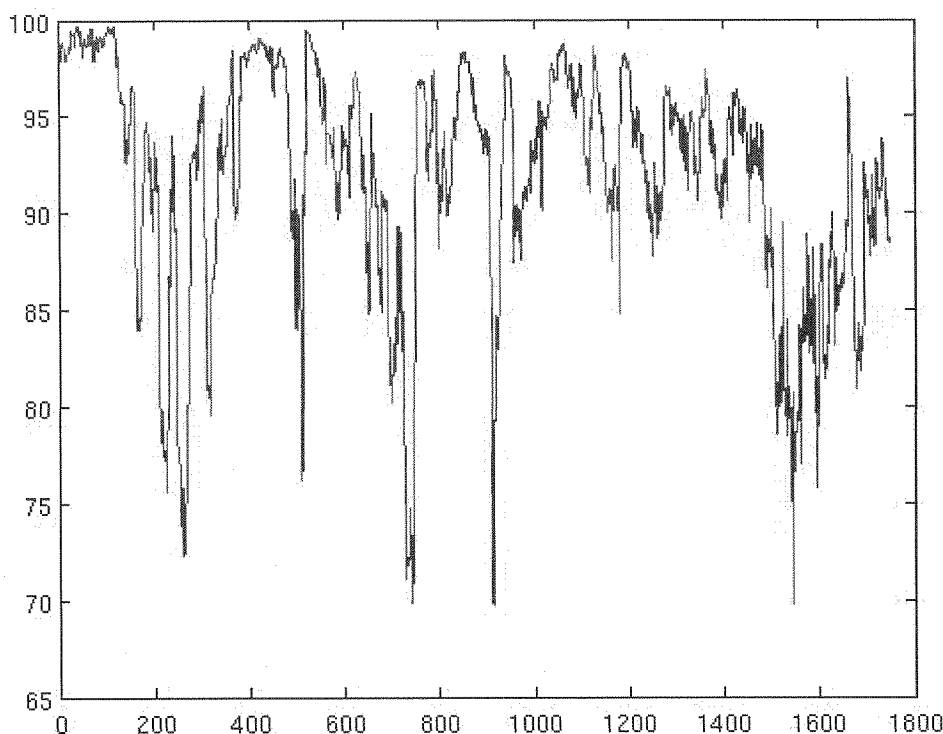
The speed at which older responses are dampened is a function of the smoothing constant alpha, α . When α is close to 1, dampening is quick and when α is close to 0, dampening is slow. The smoothing constant utilised was determined through a mix of theory and empirical testing such as that that minimised the mean square error. For this project the smoothing constant was found to be 0.9 which is in alignment with theory and prior results.

Graph 1 shows the percentage of variation explained in the term structure by the first two largest eigenvalues using an exponentially weighted moving average approach. The plot shows that around 85% variation is explained with as much a 97% and as low as 70% variation at times. This illustrates that occasionally one-factor models would be sufficient and occasionally two-factor models would not.



Graph 1: Plot of the variation explained of the first two eigenvalues over time.

In contrast, Graph 2 shows the percentage of variation explained in the term structure by the first two largest eigenvalues with a moving average window of approximately a year. The plot highlights the point that under this method, the variation explained varies greatly in relation to the exponential smoothing method. Because of this we decided to follow the former method when determining the principal components.



Graph 2: Plot of the variation explained of the first two eigenvalues over time.

7.2 Building the HJM Tree

Since many fixed income claims have American style features, we can use the tree method to correctly price in the distinguishing feature of American options which is that they can be exercised at any time before expiration. Therefore, we implement a backward induction function, which at each node of the tree compares the intrinsic value with the option value at that time and in that state if not exercised. A vector of forward rates must be considered at each node in the tree, rather than a single equity's price as in the case of equity options.

When the process is non-Markovian, the tree becomes bushy and the number of branches increases exponentially in size relative to the observation times and so the branches never recombine. However, it can be achieved within a reasonable time given the computing power and numerical techniques available today. Using a simple volatility specification, Heath, Jarrow, and Morton and Spindel (1992) show that beyond five steps the error is always within 0.5% when benchmarked to the 12 time step price. They show that we can compute the value of options using five time step

models in much less time than 12 time step models by sacrificing a relatively small amount of accuracy.

Amin and Morton (1994) and Buhler et al. (1999) both consider a tree with seven time steps as being sufficient with reasonable accuracy. A one-factor model has 254 nodes and 128 final nodes, whereas the two-factor model has 3,279 and 2,187 nodes and final nodes respectively.

The HJM trees used in this study for the one, two and three-factor models had a maximum of 18, 12 and 9 time steps respectively. The time steps were a function of the number of factors and the time to maturity of the option. The steps included the expiration points for the futures and the option, so the option was folded back at most two time steps less than the 18, 12 and 9 above.

The Matlab Financial Derivatives Toolbox provides functions to generate HJM trees. These were used for the study. It is necessary to write Matlab code to value particular derivatives.

7.3 Option Valuation

Once the code had been written, the HJM trinomial tree could be produced within Matlab.⁶ This code was applied to fit the one, two and three-factor models using PCA. To verify that the code was producing realistic values for the options, we “walked-through” an example on an excel spreadsheet.⁷ The valuation began at the final nodes of the tree and we worked backwards through the tree checking at each stage for the possibility of early exercise where we took the maximum of the intrinsic value and the value of the option if it was not exercised. As explained earlier, the node values were not discounted since the options were subject to futures style margining requirements.

Each option was then priced under all three HJM models and values were entered into a spreadsheet ready for the calculation of their errors followed by an error analysis.

⁶ See figure 1 for the Matlab code.

⁷ The spreadsheet model used can be obtained from Alan Stent if required.

7.4 Error Calculation

Once all option values from the one, two and three-factor HJM models had been calculated then they were compared with actual call option prices to determine their errors. As our analysis deals with a global test rather than calibrating our model parameters we must accept greater absolute errors than those whose models' parameters were found implicitly through optimisation techniques. The formula that was used in calculating the absolute pricing error is below:

$$\text{Absolute Pricing Error} = | \text{Actual Price} - \text{Model Value} |$$

7.5 Error Analysis

The final step is to assess any systematic pricing errors by regressing the absolute pricing errors on time to expiration, moneyness, current date along with dummy variables that specify the type of model used. Below is the regression equation that was used when analysing the errors associated with the valuation of the call options.⁸

$$\text{Error} = \alpha + \beta \text{Time to Maturity} + \gamma \text{Moneyness} + \delta \text{Date} + \sum \zeta_i D_i + \varepsilon$$

Error is calculated as the absolute value of the actual option value minus model price, time to maturity is simply the options maturity in years and moneyness is the number of basis points that the options underlying is away from the strike. The date variable is a vector of time points and D is a dummy variable used to classify the three models have been tested (one factor, two factor and three factor).

Time to maturity and moneyness were considered to examine whether the models systematically over or underpriced short lived or long lived call options that were either in or out of the money. The date was used as an explanatory variable to

⁸ See figure 3 for the R regression code.

examine whether as time progresses the models become more accurate as so would be associated with smaller errors.

The final explanatory variable was a dummy variable which measured the different models. This was of particular interest as it would allow us to measure each model's absolute error and determine which model had the lowest absolute pricing errors.

8. Empirical Results

This section presents our empirical results. The most significant result was that the three-factor Heath, Jarrow, and Morton model consistently outperformed the simpler one and two-factor models, the differences between the dummy variables being statistically significant at the 5% level. This improved performance was seen through lower pricing errors. The three-factor model outperformed the two-factor model which outperformed the one-factor model. This result highlights the point that by implementing models that are able to account for a richer array of term structure movements, the modeller is able to capture more variation which results in a more accurate pricing model.

When assessing the effects that time to maturity has on the pricing errors, the results showed consistent positive estimate which indicates that options with longer maturities have greater errors. This finding, although intuitive in the sense that greater uncertainty is associated with larger errors, was not found in the Dempster, Chaput and Stent (2002) paper.

Another result from the analysis of the errors is to assess whether the errors of the models are related to the date of the option contract. The idea was to check if, as time went on (maturity closer), the model was able to price the call options with less error and so may indicate a growing preference for the models. The results showed no clear pattern except for the three-factor HJM model which had lower errors as time progressed. This supports the use of the more complicated pricing model over longer times to maturity. One can rationalise this by looking at an interest rate environment that is becoming more and more volatile and so one-factor models are no longer sufficient.

The final empirical result from the study was with respect to the degree of moneyness of the option. This was measured as the number of basis points the call option was in (positive), at (zero) and out (negative) of the money. The results showed no visible consistent pattern with respect to the errors.

9. Conclusion

The study investigated Principal Component Analysis to model the volatility structure of three forward rate models in a Heath, Jarrow, and Morton framework. The analysis was carried out on the Australian 90-day bank accepted bill futures options that trade on the Sydney Futures Exchange over the period of January 1993 to November 2000. An outcome of the study was that the pricing algorithm was tested and the errors were corrected to an acceptable confidence in the code for its use in the Dempster, Chaput, and Stent (2002) paper.

The most prominent result was that the three-factor HJM model outperformed the simpler one and two-factor models. Finally, with this model the errors trended downwards with the time remaining until maturity, and were uncorrelated with the date and the strike price of the options.

10. References

Allen, D.E., and Chau, I., 1999, "A test of various pricing models on options on Australian bank and bill futures", *School of Finance and Economics*, Cowan University.

Amin, K., and Morton, A., 1994, "Implied volatility functions in arbitrage free term structure models", *Journal of Financial Economics*, 35, 141-180.

Black, F., E. Derman, and W. Toy, 1990, "A One-Factor Model of Interest Rates and Its Application to Treasury Bond Options," *Financial Analysts Journal*, Jan-Feb. 33-39.

Buhler, W., Uhrig-Homburg, M., Walter, U., and Weber, T., 1999, "An empirical comparison of forward rate and spot rate models for valuing interest rate options", *The Journal of Finance*, 54, 269-305.

Dempster, P., 2001, "An empirical examination of single factor interest rate models: An Australian perspective", Unpublished Master's thesis, Department of Finance and Quantitative Analysis, University of Otago.

Dempster, P., J.S. Chaput, and A.F. Stent, "A comparison of interest rate option models on Australian Bank Bill Futures", Working Paper, October, 2004, Department of Finance and Quantitative Analysis, University of Otago.

Flesaker, B., 1993, "Testing the Heath-Jarrow-Morton/Ho-Lee model of interest rate contingent claims pricing", *Journal of Financial and Quantitative Analysis*, 28, 483-495.

Heath, D., Jarrow, R., and Morton, A., 1992, "Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuations", *Econometrica*, 60, 77-105.

Heath, D., Jarrow, R., Morton, A., and Spindle M., 1992, "Easier done than said", *Risk*, 5, 77-80.

Ho, T. S. Y., and S.B. Lee, 1986, "Term Structure Movements and Pricing of Interest Rate Claims", *Journal of Finance*, 41, 1011-1029.

Jarrow, R.A., Modeling Fixed-Income Securities and Interest Rate Options, Stanford, 2002.

Lieu, D., 1990, "Option pricing with futures-style margining", *Journal of Futures Markets*, 10, 327-338.

Litterman, R., Scheinkman, J., 1991, "Common factors affecting bond returns", *Journal of Fixed Income*, 1, 54-61.

Sherris, M., 1998, "Interest rate risk factors in the Australian bond market", *Unpublished paper*.

Vasicek, O. A., 1977, "An Equilibrium Characterization of the Term Structure," *Journal of Financial Economics*, 5, 177-188.

Computer programs

Matlab Release 13 v6.5, 2005, The Mathworks, Inc.

R, 2005, A Language and Environment for Statistical Computing

11. Figures

11.1 Figure 1

```

function callRuns(fread,leaveSpot,forecastDay)
% Pass leaveSpot the number of columns of spotAll to leave out; e.g. 2 for 1 day, 2
day
% Pass forecastDay the day of the week (1=Sun, etc) to make the forecast.

global face colsSpot
global spot irdd irmm iryyyy leavespot colsSpot nspot spotDate ydDD ydMM ydYY
face = 1000000; % Value of the bank bill, 90 days after expiration of the Future
% colsSpot is the number of columns in spot =number of spot rates)

% If not already loaded, set up the Tx22 matrix of interest rates, spot.
if ~size(spot,1)
    load ../../yieldCurve/spotAll.dat;
    irdd = spotAll(:,1);
    irmm = spotAll(:,2);
    iryyyy = spotAll(:,3);
    % leaveSpot = 2; % First two columns of spot rates (1 day, 2 day) left out
    spot = spotAll(:,4+leaveSpot:end)/100;
    clear spotAll;
    colsSpot = size(spot,2);
    nspot = size(irdd,1);
    spotDate = datenum(iryyyy,irmm,irdd);
    ydDD = [1, 2, repmat(0,1,20)]; % Increments for end dates
    ydDD = ydDD(leaveSpot+1:end);
    ydMM = [0, 0, 1, 2, 3, 4, 5, 6, 9, 0, 3, 6, 9, repmat(0,1,9)];
    ydMM = ydMM(leaveSpot+1:end);
    ydYY = [repmat(0,1,9), 1, 1, 1, 1, 2, 3, 4, 5, 7, 10, 15, 20, 30];
    ydYY = ydYY(leaveSpot+1:end);
end

% Set up the option file to analyse, eg ssMar1993.dat
% The file name must be prestored in fread
disp(sprintf('From file %s',fread));
dirname='~/tc/d/irDec04/ssMmmmyyyy8750To9850/';
inName = strcat(dirname,fread);
ss = load(inName);
nss = size(ss,1);
ncolss = size(ss,2);
ssDate = zeros(nss,1);
for i=1:nss
    ssDate(i) = datenum(ss(i,3),ss(i,2),ss(i,1));
end
optionExpiryDate = datenum(ss(1,7),ss(1,6),ss(1,5)); % yyyyymmdd, constant
throughout the file

% Load the file of parameters for the hjm volatility structures
load ../../yieldCurve/hjmVols.dat;

% Open the file of eigenvalues and eigenvectors
eigname = 'eigvalues.dat';

```

```

fidEigs=fopen(strcat(' ../../cedreece05/m/',eigname),'r');
curveTerm = (ydYY + ydMM/12 + ydDD/365.25);
curveTerm = curveTerm(1:13);

% Output file:
foutName = {strcat(strcat(fread(1:end-4),'Calls'))};
foutFid = fopen(strvcat(foutName),'w');
% Write a header record:
fprintf(foutFid,...
'FutExpDate OpExpDate Date Model Strike ObdPrice ModelPrice\n');

futureExpiryDate = datenum(ss(1,7),ss(1,6),ss(1,4)); % Constant for file
compounding = 1; % The compounding period of the annual rates in spot

iDate = 1; % Runs through ssDate (the dates in this option file)
ispot = 1; % Updates position in spotDate interest rate file
eigenMissing = 0; % Set to 1 if a date is missing from fidEigs
while iDate<=nss
    siDate = ssDate(iDate);
    iwd = weekday(siDate); % Day of the week
    if iwd==forecastDay
        disp(sprintf('Processing %s',datedisp(ssDate(iDate))))
        % Get first position in spot:
        while (spotDate(ispot+1) <= siDate)
            ispot = ispot + 1;
            if ispot>=nspot
                disp('Error. ispot to exceed size of spotAll.dat');
                return
            end
        end
        if hjmVols(ispot,1) ~= siDate
            disp('Error. Date in hjmVols.dat does not match that in spotAll.dat');
            disp(sprintf('hjmVols date at position %d is %d, or %s',...
            ispot,hjmVols(ispot,1),datedisp(hjmVols(ispot,1))));
            disp(sprintf('spotAll date is %d, or %s',siDate,datedisp(siDate)));
            return;
        end
        tEig=1; % Set to zero when the correct date is found in the file eigname
        while tEig
            if eigenMissing
                % The next record has already been read
                eigenMissing = 0;
            else
                % Read the next record
                eigLine=fgetl(fidEigs);
                if eigLine<0
                    disp(sprintf('No data left in %s\n',eigname));
                    disp(sprintf('spotAll date is %d, or %s',siDate,datedisp(siDate)));
                    fclose(fidEigs);
                    fclose(foutFid);
                end
            end
        end
    end
    iDate = iDate + 1;
end

```

```

        return;
    end
    [eigDate,eigLine]=getf(eigLine);
end
if eigDate>siDate
    % There is no observation on eigenvalues yet
    eigenMissing = 1;
    tEig = 0;
else
    if eigDate==siDate
        tEig = 0;
    else
        % Skip next three records
        fgetl(fidEigs);
        fgetl(fidEigs);
        fgetl(fidEigs);
    end
end
end

% Set up volatility specifications, one for each model:
%C volSpecHolee = hjmvolspec('Constant',hjmVols(ispot,2));
%C volSpecExpon = hjmvolspec('Exponential',hjmVols(ispot,3),hjmVols(ispot,4));
if ~eigenMissing
    % For eigenvalue specifications:
    ev = sscanf(eigLine,'%f');
    % 1 factor HJM
    eigLine=fgetl(fidEigs);
    curveProp1 = sscanf(eigLine,'%f')*sqrt(ev(1));
    % volSpecEig1 = hjmvolspec('Proportional',curveProp1',curveTerm,1e6);
    volSpecEig1 = hjmvolspec('Stationary',curveProp1',curveTerm);
    % 2 factor HJM
    eigLine=fgetl(fidEigs);
    curveProp2 = sscanf(eigLine,'%f')*sqrt(ev(2));
    % volSpecEig2 =
    hjmvolspec('Proportional',curveProp1',curveTerm,1e6,'Proportional',curveProp2,curveTerm,1e6);
    volSpecEig2 =
    hjmvolspec('Stationary',curveProp1',curveTerm,'Stationary',curveProp2,curveTerm);
    % 3 factor HJM
    eigLine=fgetl(fidEigs);
    curveProp3 = sscanf(eigLine,'%f')*sqrt(ev(3));
    volSpecEig3 =
    hjmvolspec('Stationary',curveProp1',curveTerm,'Stationary',curveProp2,curveTerm,...
    'Stationary',curveProp3,curveTerm);
end

% Set up spot (rate) specification:
endDatev = datestr(datenum([iryyyy(ispot)+ydYY' irmm(ispot)+ydMM'
irdd(ispot)+ydDD']));

```

```

spotSpec = intenvset('compounding',compounding,'StartDates',...
spotDate(ispot),'EndDates',endDatev,'Rates',spot(ispot,:));

% Calibrate level of interest rates to match modelled and actual Futures values
if ~eigenMissing %C
actualFuturePrice = sfe90Value(ss(iDate,8));
bpAdj =
calibrateZerosForHjmSingle(spotSpec,actualFuturePrice,futureExpiryDate);
spotSpec = intenvset(spotSpec,'Rates',(spot(ispot,:)+bpAdj));
end %C

% disp(sprintf('Actual Futures price is %12.4f
(%7.4f)',actualFuturePrice,ss(iDate,8)))
% pv = cfbyzero(spotSpec,face,futureExpiryDate+90,spotDate(ispot)); % pv at t=0
of face dollars
% calibratedFuturePrice = cfbyzero(spotSpec,pv,spotDate(ispot),futureExpiryDate);
% calibratedQuotedFuturePrice = sfe90Quoted(calibratedFuturePrice);
% disp(sprintf(...
% 'Calibrated value is %16.4f
(%7.4f)',calibratedFuturePrice,calibratedQuotedFuturePrice))

% Set up time specification for the 1-factor hjm trees
clear maturity;
nmat = max(3,round((futureExpiryDate-spotDate(ispot))/16)); % To limit number
of nodes
% The first node is not included in maturity
maturity(1) = min(spotDate(ispot)+nmat,optionExpiryDate);
if maturity(1)<optionExpiryDate
im = 1;
while maturity(im)+nmat < optionExpiryDate
im = im + 1;
maturity(im) = maturity(im-1)+nmat;
end
maturity(im+1) = optionExpiryDate;
else
% Too close to the option expiry date
break;
end
maturity(im+2) = futureExpiryDate;
maturity(im+3) = maturity(im+2)+90;
timeSpec = hjmtimespec(spotDate(ispot),maturity,1);

if ~eigenMissing
% Set up time specification for the 2-factor hjm tree
clear maturity2;
nmat2 = max(3,round((futureExpiryDate-spotDate(ispot))/10)); % To limit
number of nodes
% The first node is not included in maturity
maturity2(1) = min(spotDate(ispot)+nmat2,optionExpiryDate);
if maturity2(1)<optionExpiryDate

```

```

    im = 1;
    while maturity2(im)+nmat2 < optionExpiryDate
        im = im + 1;
        maturity2(im) = maturity2(im-1)+nmat2;
    end
    maturity2(im+1) = optionExpiryDate;
else
    % Too close to the option expiry date
    break;
end
maturity2(im+2) = futureExpiryDate;
maturity2(im+3) = maturity2(im+2)+90;
timeSpec2 = hjmtimespec(spotDate(ispot),maturity2,1);

% Set up time specification for the 3-factor hjm tree
clear maturity3;
nmat3 = max(3,round((futureExpiryDate-spotDate(ispot))/7)); % To limit number
of nodes
% The first node is not included in maturity
maturity3(1) = min(spotDate(ispot)+nmat3,optionExpiryDate);
if maturity3(1)<optionExpiryDate
    im = 1;
    while maturity3(im)+nmat3 < optionExpiryDate
        im = im + 1;
        maturity3(im) = maturity3(im-1)+nmat3;
    end
    maturity3(im+1) = optionExpiryDate;
else
    % Too close to the option expiry date
    break;
end
maturity3(im+2) = futureExpiryDate;
maturity3(im+3) = maturity3(im+2)+90;
timeSpec3 = hjmtimespec(spotDate(ispot),maturity3,1);
end

% Set up an HJM tree for each model
%C hjmtHolee = hjmtree(volSpecHolee,spotSpec,timeSpec);
%C hjmtExpon = hjmtree(volSpecExpon,spotSpec,timeSpec);
if ~eigenMissing
    hjmtEig1 = hjmtree(volSpecEig1,spotSpec,timeSpec);
    hjmtEig2 = hjmtree(volSpecEig2,spotSpec,timeSpec2);
    hjmtEig3 = hjmtree(volSpecEig3,spotSpec,timeSpec3);
end

% Do forecast
% disp(sprintf('\nForecasting %s',datedisp(ssDate(iDate))))

% For each strike price:
iCallPos = 0; % Index for following loop

```

```

for callPos=10:5:ncolss-5
    iCallPos = iCallPos+1;
    callStrike = sfe90Value(ss(iDate,callPos));
    if ss(iDate,callPos+1)<0
        % disp(sprintf('%14.4f %5.2f Missing',callStrike,ss(iDate,callPos)));
    else
        callPremium=calcPremium90(ss(iDate,callPos));
        actualCall=callPremium*ss(iDate,callPos+1)*100;

        % Model the price, using each model in turn:
        %C holeePrice=valueCall(hjmtHolee,callStrike);
        %C exponPrice=valueCall(hjmtExpon,callStrike);
        if ~eigenMissing
            eig1Price=valueCall(hjmtEig1,callStrike);
            eig2Price=valueCall(hjmtEig2,callStrike);
            eig3Price=valueCall(hjmtEig3,callStrike);
        end

        % Write the data, one record for each model:
        % (1) Future expiry date
        % (2) Option expiry date
        % (3) Observation date
        % (4) Model
        % (5) Strike price
        % (6) Observed price
        % (7) Modelled price
        % Ho-Lee model:
        %C fprintf(foutFid,'%d %d %d %s %11.4f %11.4f %11.4f\n',...
        %C
        futureExpiryDate,optionExpiryDate,siDate,'holee',ss(iDate,callPos),actualCall,...
        %C holeePrice);
        % Exponential model:
        %C fprintf(foutFid,'%d %d %d %s %11.4f %11.4f %11.4f\n',...
        %C
        futureExpiryDate,optionExpiryDate,siDate,'expon',ss(iDate,callPos),actualCall,...
        %C exponPrice);
        if ~eigenMissing
            % 1 factor eigenector model:
            fprintf(foutFid,'%d %d %d %s %11.4f %11.4f %11.4f\n',...
            futureExpiryDate,optionExpiryDate,siDate,'eig1 ',ss(iDate,callPos),actualCall,...
            eig1Price);
            % 2 factor eigenector model:
            fprintf(foutFid,'%d %d %d %s %11.4f %11.4f %11.4f\n',...
            futureExpiryDate,optionExpiryDate,siDate,'eig2 ',ss(iDate,callPos),actualCall,...
            eig2Price);
            % 3 factor eigenector model:
            fprintf(foutFid,'%d %d %d %s %11.4f %11.4f %11.4f\n',...
            futureExpiryDate,optionExpiryDate,siDate,'eig3 ',ss(iDate,callPos),actualCall,...
            eig3Price);
        end
    end
end

```

```

        end % End of loop over strike prices
    end
end
iDate = iDate + 1;
end

fclose(fidEigs);
fclose(foutFid);
return

% Controls runs NOVEMBER 2005
% Prepare file names
fileName = {...
'ssMar1993.dat'; 'ssJun1993.dat'; 'ssSep1993.dat'; 'ssDec1993.dat';...
'ssMar1994.dat'; 'ssJun1994.dat'; 'ssSep1994.dat'; 'ssDec1994.dat';...
'ssMar1995.dat'; 'ssJun1995.dat'; 'ssSep1995.dat'; 'ssDec1995.dat';...
'ssMar1996.dat'; 'ssJun1996.dat'; 'ssSep1996.dat'; 'ssDec1996.dat';...
'ssMar1997.dat'; 'ssJun1997.dat'; 'ssSep1997.dat'; 'ssDec1997.dat';...
'ssMar1998.dat'; 'ssJun1998.dat'; 'ssSep1998.dat'; 'ssDec1998.dat';...
'ssMar1999.dat'; 'ssJun1999.dat'; 'ssSep1999.dat'; 'ssDec1999.dat';...
'ssMar2000.dat'; 'ssJun2000.dat'; 'ssSep2000.dat'; 'ssDec2000.dat';...
'ssMar2001.dat'; 'ssJun2001.dat'; 'ssSep2001.dat'; 'ssDec2001.dat';...
};
nf = size(fileName,1);

startAt = 'ssMar2001.dat';
istart = 1;
while istart < nf & ~strcmp(startAt, fileName(istart))
    istart = istart + 1;
end

for ic = istart:nf
    fread = strvcat(fileName(ic));
    callRuns(fread, 2, 4); % Leave 0 spot rates, forecast day is Wed
end

```

11.2 Figure 2

Analysis of errors

```

toAnalyse=c(
"Mar1995Calls","Jun1995Calls","Sep1995Calls","Dec1995Calls",
"Mar1996Calls","Jun1996Calls","Sep1996Calls","Dec1996Calls",
"Mar1997Calls","Jun1997Calls","Sep1997Calls","Dec1997Calls",
"Mar1998Calls","Jun1998Calls","Sep1998Calls","Dec1998Calls",
"Mar1999Calls","Jun1999Calls","Sep1999Calls","Dec1999Calls",
"Mar2000Calls","Jun2000Calls","Sep2000Calls","Dec2000Calls",
"Mar2001Calls","Jun2001Calls","Sep2001Calls","Dec2001Calls",
)

nAn=length(toAnalyse)
tauEsts=matrix(rep(0,nAn),nAn,1) # Matrix to store estimates of tau
nMod=3 # Number of model effects
modEffects=matrix(rep(0,nMod*nAn),nAn,nMod) # Matrix to store model effects
nStrikes=11*4 # Maximum number of different strike values collected
strikeEffects=matrix(rep(NA,nStrikes*nAn),nAn,nStrikes)

fid=file('rOut.dat','w');
sink(fid)

na=0;
for (ia in toAnalyse){
  na=na+1;
  fname=paste("ss",ia,sep="")
  cat("\n\nFrom file",fname)
  ia=read.table(file=fname,header=T)
  ia.tau = (ia$OpExpDate-ia$Date)/365

  ia.absPriceErr = abs(ia$ObdPrice - ia$ModelPrice)
  ia.priceErrorModel = lm(ia.absPriceErr~0+ia.tau+
factor(ia$Model)+factor(ia$Strike)+factor(ia$Date))
  print(summary(ia.priceErrorModel))
  tauEsts[na]=coefficients(ia.priceErrorModel)[1]
  modEffects[na,1]=coefficients(ia.priceErrorModel)[2]
  modEffects[na,2]=coefficients(ia.priceErrorModel)[3]
  modEffects[na,3]=coefficients(ia.priceErrorModel)[4]

  cat("TEST H0: Constant model effects, from file",fname,"\n");
  submodel = lm(ia.absPriceErr~0+ia.tau+factor(ia$Date)+factor(ia$Strike))
  print(anova(submodel,ia.priceErrorModel))

  cat("TEST H0: No eig3 effect, from file",fname,"\n");
  f2 = ia$Model
  for (i in 1:length(f2)) {
    if (f2[i]=='eig3') f2[i]='eig2'
  }
  submodel = lm(ia.absPriceErr~0+ia.tau+f2+factor(ia$Date)+factor(ia$Strike))
  print(anova(submodel,ia.priceErrorModel))
  cat("\f")
}

```

```
sink(type="output") # Redivert output to screen
close(fid)

fidCoeffs=file('rCoeffs.dat','w');
sink(fidCoeffs)
for (ia in 1:na)
  cat(tauEsts[ia],modEffects[ia,1],modEffects[ia,2],modEffects[ia,3],"\\n")
sink(type="output")
close(fidCoeffs)
```