Term Structure Literature

Barton Baker

August 29, 2012

1 Introduction

The relationship between the term premium on bond returns and macroeconomic fluctuations has long been a subject of study. The causality runs in two directions. Movements in consumption, output, inflation and monetary policy tools impact both present and expectations of future interest rates, leading to changes in yields all along the yield curve. Movements in interest rates on bonds of all maturities have an impact on investment activity through their effect on the rates that banks charge on loans to businesses and consumers. In order to track down the theoretical underpinnings of co-movements between macroeconomic variables and the term structure, research has begun in consumption-based finance models and led to increasingly stylized general equilibrium models.

Before discussing a brief history of the term structure within modern macroeconomic economics, it is important to solidify what we mean by movements in the term structure. The term structure of interest rates on government bonds is often summarized by the yield curve. The yield curve is a graphical snapshot of bond yields at a certain time, plotting yield on the vertical axis versus maturity on the horizontal axis, with a sample of at least a few bonds. Movements in the yield curve, come in three main forms: “level”, “slope”, and “curvature”. The level of the yield curve can be thought of as the average yield of all the bonds. The slope of the yield curve is the difference between the yield on any two bonds of different maturity. It common practice to choose a very short maturity (3-month or 6-month) and a longer maturity (greater than five years) yield to calculate the slope, as this minimizes movements in the slope measure due to peculiarities among particular spans of the yield curve. The curvature is the convexity of the yield curve and must be summarized with at least 3 selected yields from the yield curve. Any significant movements in the yield curve can be a result of the change in the level, the slope, the curvature, or (most likely) some combination of the three.

2 Affine models of the term structure

Affine models of the term structure are one of the primary ways through which movements in the term structure are explained. Affine models begin with a simple specification of the time series process governing the inter-temporal discount factor. Adding the assumptions of no-arbitrage and the expectations hypothesis, the entire yield curve can be derived from an affine transformation of the process governing the discount factor[1]. The use of no-arbitrage affine models to identify movements in the term structure was first primarily introduced in the work of Vasicek (1977) and Cox et al. (1985), who proposed single factor models to estimate the term structure.

With the yield curve as the empirical reference of the term structure, empirically motivating movements in the term structure began in its modern expression through empirical models and principal component analysis. The first step in this direction came with Nelson and Siegel (1987), who proposed a three parameter model to explain most movement in the term structure. The authors derive a second-order differential equation that can be summarized using three free parameters. A linear combination of these three estimated factors estimated over simple functions of bond maturity and a fit parameter could explain more than 90% of the variation in 75% of the samples tested. Litterman and Scheinkman (1991) reached a similar conclusion using an empirical model of the term structure, rather than the differential equation in Nelson and Siegel (1987). Litterman and Scheinkman (1991) showed that 96% of the movement in the term structure could be explained by three principal components. They related these components to concepts of "level", "slope", and "curvature", but remained mute on the subject of what determined these factors and what relationship they had to observed macroeconomic variables. Explaining what these components actually represent has been a natural jump-off point for an entire area of research. Both papers, one from the finance literature, one from the macro literature, while offering little in the way of a causal explanation of the term premium, were both important catalysts for future theoretical investigations. Specifically, these papers offered two critical contributions to much of literature moving forward: 1) subtle hints regarding the relationship between unobserved factors and observed yields and 2) the number of factors needed to generate empirically meaningful yield curves.

Duffie and Kan (1996) built-upon the affine term structure model literature by offering a complete framework within which single-factor and multi-factor models could be studied. Not only did this offer an opportunity to combine the modeling framework of Vasicek (1977) and Cox et al. (1985) with the empirical specification of Nelson and Siegel (1987) and Litterman and Scheinkman (1991), but it also served as the

[1] A generalized model is derived equation by equation later in this paper.

framework utilized by most affine term structure model papers in its wake. Duffie and Kan (1996) precisely derived the conditions under which a unique solution exists to affine models and defined the general class of factor models that could be expanded to include many factors. These models took the general form (Duffie and Kan, 1996)[382]:

$$R(x) = \lim_{r \downarrow 0} \frac{-\log f(x, \tau)}{\tau} \tag{2.1}$$

where R is the short-rate, x is a time-homogeneous Markov process of length n where n is the number of factors influencing the term structure, τ is the maturity of a zero-coupon bond maturing at time $t + \tau$, and $f(x, \tau)$ is the price or market value of a single zero-coupon bond of maturity τ. To solidify terminology, the short-rate in any period t is defined as:

$$R(X_t) = \lim_{r \downarrow 0} \frac{-\log f(X_t, \tau)}{\tau} \tag{2.2}$$

Duffie and Kan (1996) assume that X satisfies a stochastic differential equation (SDE) of the form:

$$dX_t = \nu(X_t)dt + \sigma(X_t)dW_t^* \tag{2.3}$$

which under additional technical regularity can be expressed as a standard Brownian motion W in $\mathbb{R}^n$:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t \tag{2.4}$$

Equations 2.1, 2.2, and 2.4 define what Duffie and Kan (1996) call "General Factor Models" of the term structure. Beyond this broad categorization of factor models, Duffie and Kan (1996) show affine factor models satisfy the requirements of the system outlined above. They define a class of compatible models by specifying the functional form of $f(x, \tau)$:

$$f(x, \tau) = \exp[A(\tau) + B(\tau) \cdot x] \tag{2.5}$$

where A and B are C^1 functions on $[0, \infty]$. This is the form that affine models have taken in the main thread of affine term structure model literature.

In addition to laying out this formative framework, the self-proclaimed primary purpose of Duffie and Kan (1996), Duffie and Kan (1996) also offered a suggestion for a solution method to these models, particularly pertinent in the case of multiple factors. This method involved first solving for elements of B from Eq. 2.5

using the fourth-order Runge-Kutta method. Following this initial parametrization of B a combination of Newton-Raphson and ADI solution algorithms are applied to match an increasing number of consistency conditions. While this exact method is not always used in modern affine model term structure models, the approach of iteratively estimating different parameters in different rounds has become very popular [2]. The exact method prescribed by Duffie and Kan (1996) becomes increasingly untractable with higher numbers of factors. Estimation methods for affine term structure models are treated as a whole in Duffee and Stanton (2004). Given that a number of estimation methods had become popular in the early 2000's, the authors examined Maximum Likelihodd (ML), Estimated Method of Moments (EMM), and a Kalman filter method. They find that the ML and EMM method does not perform well under highly-persistent, small samples common in bond data. Specifically, the ML method results in biased parameter estimates in the speed of mean reversion Duffee and Stanton (2004, 19), but overall efficient use of the information when feasible. At the time of writing, ML methods were computationally intensive for affine term structure models. The EMM method performs much worse than the ML method, resulting mostly from the differences in the curvature of the auxiliary function between the highly-persistent small sample and the true parameter function of the hypothetical infinite sample. The Kalman filter method results in less precision than ML estimates (higher standard deviations of the estimates under Monte Carlo techniques), but is more feasable than ML techniques, where searches over a very large parameter space can become computationally intensive.

By building the sandbox within which the affine-model term structure literature could grow, Duffie and Kan (1996) spurred a flurry of papers integrating both latent and observed factors to estimate the term structure. Because these models relied on no-arbitrage assumptions and a single pricing kernel, the implied term premium could easily be computed post-estimation by attributing any yield on longer-maturity bonds above and beyond the implied risk-free yield to the term premium. Much of this work has exclusively used unobserved latent factors to estimate the term structure while others have combined observed macro variables with latent variables, with even others using exclusively observed macro variables as factors. Latent variables are unobserved components of a statistical process that are derived using regression methods, unlike the components the estimated components in Principal Component Analysis (PCA) which are always orthogonal. Major investigations using exclusively latent factors include Dai and Singleton (2000) and Kim and Wright (2005). While Dai and Singleton (2000) are able to fit the observed moments of the term structure quite well, their use of latent factors offer little in the way of theoretical interpretation. The model proposed in Kim and Wright (2005), a three latent factor affine model of the term structure has became the workhorse

[2] In fact, the Kalman filter is the only other solution method that is widely used to solve these models

model for generating a reliable estimate of the term premium. Kim and Wright (2005)'s main advantage over Dai and Singleton (2000) is that they take into account the examination of estimation methods performed in Duffee and Stanton (2004), using a small sample consistent Kalman filtering to estimate their model.Dai and Singleton (2000) instead use a simulated method of moments method requires more identifying restrictions on the model parameters.

In response to criticisms that unobserved latent factors offer little in the way of causal relevance, the Ang and Piazzesi (2003) extension of Dai and Singleton (2000) uses two macro variables, inflation and real activity, and three latent factors, per the suggestion of Litterman and Scheinkman (1991) and Nelson and Siegel (1987). They find that the addition of observed macro factors significantly increases the ability of the model to forecast term structure moments. This paper showed that observed macro variables could be used to summarize at least some of the information entering expectations formation related to movement in the term structure. Bernanke et al. (2005) took this approach one step further in their attempt to estimate the effects of zero-lower bound restricted alternative monetary policy. Bernanke et al. (2005) used five observable macro variables[3] as their factors, arguing that their five variables summarized the relevant information set entering bond pricing decisions. They find that their model is able to fit the yield data relatively well, with pricing errors increasing with maturity (see Table 1 for details) and realistic tracking of the paths of yields at all maturities included in the analysis (see Figure 1). Rudebusch and Wu (2008) (originally appearing in 2004 as Rudebusch and Wu (2004)) also integrated both latent and observed factors into an affine model as in Ang and Piazzesi (2003), but instead related the observed factors to the latent factors through New Keynesian structural relationships rather than direct inclusion in the information set. Rudebusch and Wu (2008) showed that including the macro variables added to the model fit, but including the observed factors did not change the estimates of the unobserved factors by much. Diebold et al. (2006) provided an interesting link between non structural affine models of the term structure and the New Keynesian general equilibrium framework. With the exception of Rudebusch and Wu (2008), the other affine models mentioned only allowed for causality in the direction of macro and latent factors to term structure yields. Diebold et al. (2006) relates the first two latent factors to observed macroeonomic outcomes. The first latent factor or "level" factor is related to inflation, while the second factor or "slope" factor is related to capacity utilization or cyclical fluctuations. Diebold et al. (2006) attempts to focus on this issue of bi-directional causality, finding that while both directions are significant, the effect of macro and latent factors on term structure moments are much more important than the effect of term structure moments on macro and latent factors.

[3] The details of which are discussed in section 3.

In addition to combining observed and unobserved factors to increase ability of affine models to explain the term structure, other modeling methods have been attempted to supplement or refine the information available to price-determining bond investors. Kim and Orphanides (2005) proposed that the three-factor model can be supplemented with Blue-Chip survey estimates of 6-month- and 12-month-ahead expected 3-month T-bill yield in order increase the ability of the model to fit the data. The authors show that the survey forecasts add important information to the model solution algorithm, decreasing the required sample size for stable parameter estimates, allowing estimation to take place using shorter periods where the condition of stable "true" parameters is more likely to hold. Orphanides and Wei (2010) took a different approach, echoing the approach of Orphanides (2001) to increase the fit of a Taylor-rule using real-time data and adding the possibility of time-varying parameters. Instead of beginning with an unobserved latent factor model, Orphanides and Wei compare several versions of affine term structure models, but their final proposed model is where affine model state-space parameters are iteratively estimated over time using VAR(2) with real-time data available at time t. While the authors show that a 3 factor affine model still outperforms their proposed model, the model iteratively estimated with real-time data outperforms their baseline comparison model, which is estimated using final data and without the possibility of time-varying parameters. Little time was spent in this paper on examining the advantage of using real-time data to estimate structural models, since the focus was more on using an iterative VAR to inform the pricing kernel.

All affine models of the term structure can be built up from a basic general setup. The rest of this section examines the set-up and estimation strategies.

Affine models of the term structure begin with the assumption that the pricing kernel for all government bond yields is completely determined by an VAR process, most often summarized as a VAR(1) for simplicity[4]:

$$X_t = \mu + \Phi X_{t-1} + \Sigma \varepsilon_t \tag{2.6}$$

where X_t is an $n \times 1$ vector of latent unobserved and/or observed macro factors. Eq. 2.6 fully identifies the time series of the information entering bond pricing decisions and ε is assumed $\mathcal{N}(0, 1)$.

The price of any n-period zero-coupon bond in period t can be recursively defined as the expected product of the pricing kernel in period t, m_t and the price of the same security matured one period in $t+1$:

$$p_t^n = E_t[m_{t+1} p_{t+1}^{n-1}] \tag{2.7}$$

[4] This notation is taken from Bernanke et al. (2005) and Ang and Piazzesi (2003), but expanded to refer to a broader class of models

We assume that the period-ahead pricing kernel is conditionally log-normal, a function only of the current risk-free rate, $i_t^{(1)}$ and the prices of risk, λ_t:

$$m_{t+1} = \exp\left(-i_t^{(1)} - \frac{1}{2}\lambda_t'\lambda_t - \lambda_t'\varepsilon_{t+1}\right) \tag{2.8}$$

Without perfect foresight, agents price risk attributed to each macro factor given the current state-space, X_t:

$$\lambda_t = \lambda_0 + \lambda_1 X_t \tag{2.9}$$

where λ_0 is $n \times 1$ and λ_1 is $n \times n$.

We can then define the price of any zero-coupon bond of maturity n in period t as a function of the pricing kernel, combining Eqs. 2.7 and 2.9:

$$p_t^n = \exp\left(\bar{A}_n + \bar{B}_n' X_t\right) \tag{2.10}$$

where $\bar{A}_n$ and $\bar{B}_n$ are recursively defined as follows:

$$\begin{aligned} \bar{A}_{n+1} &= \bar{A}_n + \bar{B}_n'(\mu - \Sigma\lambda_0) + \frac{1}{2}\bar{B}_n'\Sigma\Sigma'\bar{B}_n' - \delta_0 \\ \bar{B}_{n+1}' &= \bar{B}_n'(\Phi - \Sigma\lambda_1) - \delta_1' \end{aligned} \tag{2.11}$$

with $\bar{A}_1 = \delta_0$ and $\bar{B}_1 = \delta_1$ and δ_0 and δ_1 relate the macro factors to the one-period risk free:

$$p_t^1 = \exp \delta_0 + \delta_1 X_t \tag{2.12}$$

In the same way, the yield can be expressed as:

$$y_t^n = A_n + B_n' X_t \tag{2.13}$$

where $A_n = -\bar{A}_n/n$ and $B_n = -\bar{B}_n/n$.

In the case where X_t contains only observed macro-variables, μ, Φ and Σ are easily calculated using ML in first stage of the estimation process. In the second stage, Eq. 2.13 is fit to various points along the yield curve also using ML or Least Squares. This second stage estimation becomes especially trivial if the risk free rate is included in the set of macro-variables informing bond pricing decisions, X_t. This is the method used (with Least Squares) in Bernanke et al. (2005).

When unobserved latent variables are included in X_t, the solution algorithm becomes more complicated. One method to solve for the unknown factors has been to assume that a number of points on the yield curve, equal to the number of unobserved latent factors, are observed without error. The factors are first solved for using the yields measured without error and the likelihood function of the VAR system and then the remainder of the yield curve is estimated by maximizing the likelihood of the yield curve relationship, Eq. 2.13. Practitioners usually choose well-spaced points along the yield curve (3-month, 12-month, and 36-month yields), but these choices are mostly arbitrary. Ang and Piazzesi (2003) utilizes this method.

Another method of estimating the affine system is by using the Kalman filter. The VAR(1) system defined by Eq. 2.6 and the yield/macro factor relationship of Eq. 2.13 together naturally define a state-space system. The system can then be iteratively solved using standard Kalman filter solution algorithms. This method has become more popular in affine term structure model papers in recent time, since it was first compared to ML methods in Duffee and Stanton (2004) and utilized in the benchmark latent factor model of Kim and Wright (2005).

3 Replication of Bernanke et al. (2005)

This section is an attempt to replicate the results of Bernanke et al. (2005), an investigation into the effects of alternative monetary policy when a zero-lower bound for the short-rate is a binding condition. The second section of their paper involves the estimation of an affine model of the term structure using five observed factors: an HP-filtered employment gap, inflation over the past year (measured as PCE excluding food and energy), expected inflation over the subsequent year from the Blue Chip survey, the federal funds rate, and the year-ahead Eurodollar rate. This is the section that this small paper will attempt to replicate. They utilize a VAR directly estimated on the five macro variables to generate a pricing kernel to fit the implied zero-coupon bonds (CRSP Fama files) of maturities six-months and one, two, three, four, five, seven and ten years. They use monthly data from 1982 to 2004. While the authors do not present the exact parameters estimates, they do present graphs of the two-year and ten-year actual, predicted, and risk neutral yields on bonds. They also present prediction errors for all of the yields used to estimate the model, using an estimated model with the Eurodollar and without.

Presented in Figure 1 are the two graphs presented as they appear in Bernanke et al. (2005)[46]. Each graph shows three lines: the actual yield, the model-predicted yield, and the risk-neutral yield. The model-predicted and risk-neutral yield are both generated from the estimated parameters, the difference between which is the implied term premium.

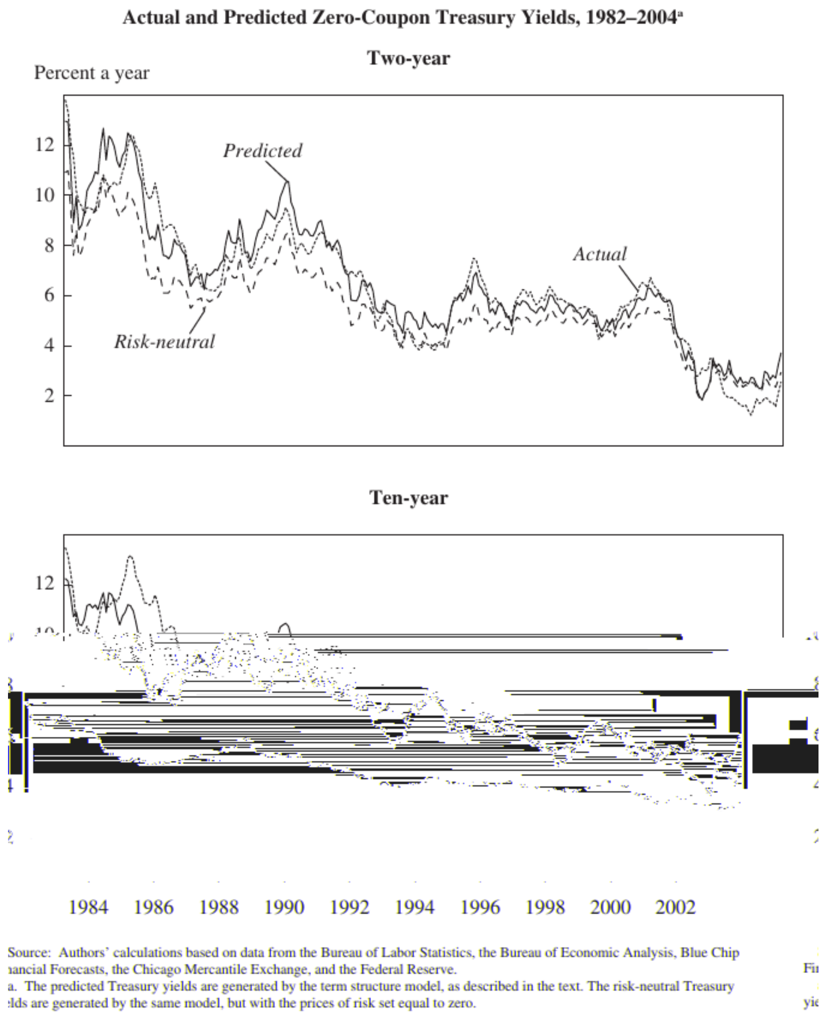

Figure 1: Page 46 from Bernanke et al. (2005)

Table 1 is also taken from Bernanke et al. (2005)[47], presenting the pricing errors for all of the yields used for estimating using two models, with and without Eurodollar futures as a macro factor.

As a proof of concept, this paper attempts to replicate the results presented in Bernanke et al. (2005) (BSR) using a custom-written solution method in Python (see accompanying code-base) and using data as close as possible to data used in BSR. Instead of 1982 to 2004, this replication used data from December 1987 to December 2004 because of data availability. Instead of Blue Chip survey inflation year-ahead forecasts,

Prediction Errors for Treasury Yields in the Term Structure Model

Basis points

	Standard deviation of predicted yield	
Maturity	*VAR with Eurodollar shocks*	*VAR without Eurodollar shocks*
6 months	33.0	62.1
1 year	50.3	78.9
2 years	73.3	97.4
3 years	81.2	100.7
4 years	82.5	98.3
5 years	81.5	95.0
7 years	83.3	93.3
10 years	80.8	87.8

Source: Authors' calculations based on data from the Bureau of Labor Statistics, the Bureau of Economic Analysis, Blue Chip Financial Forecasts, the Chicago Mercantile Exchange, and the Federal Reserve.

Table 1: Page 47 from Bernanke et al. (2005)

expected annual inflation from the Survey of Consumers Thomson Reuters and University of Michigan (2010) is used as an indicator of inflation expectations. The other VAR factors are as in BSR. Also, because of current unavailability of Fama-CRSP implied zero-coupon yields on Tresuries, Treasury Constant Maturities available from FRED are used as a second-best replacement Federal Reserve Bank of St. Louis (2011) and the 4 year maturity Tresuries are not included.

The parameters governing the prices of risk are estimated by minimizing the sum of squared predictions errors. Using these parameter estimates, the model-predicted yields are generated by feeding the VAR elements, X_t for each t into Equation 2.9 using the estimated λ_0 and λ_1. By setting the prices of risk to zero in λ_0 and λ_1, the implied risk-neutral yields can be generated. These two time series, along with the actual yields are presented graphically in figures 3 and 2, echoing their presentation Figure 1 from BSR.

As you can see, the estimates do not match exactly, but this can most likely be attributed to non-matching data sources. The results qualitatively match in a few ways. The estimation results in smaller pricing error for the two year maturity yields compared to the ten year. This is probably a result of the fact that the federal funds rate is included as a factor in the VAR and the shorter-term maturity bonds are more likely to be directly effected by movements in the federal funds rate. Also, as in BSR, the replication obtains the appealing result that the term premium, the distance between the predicted and risk-free yields, increases with maturity.

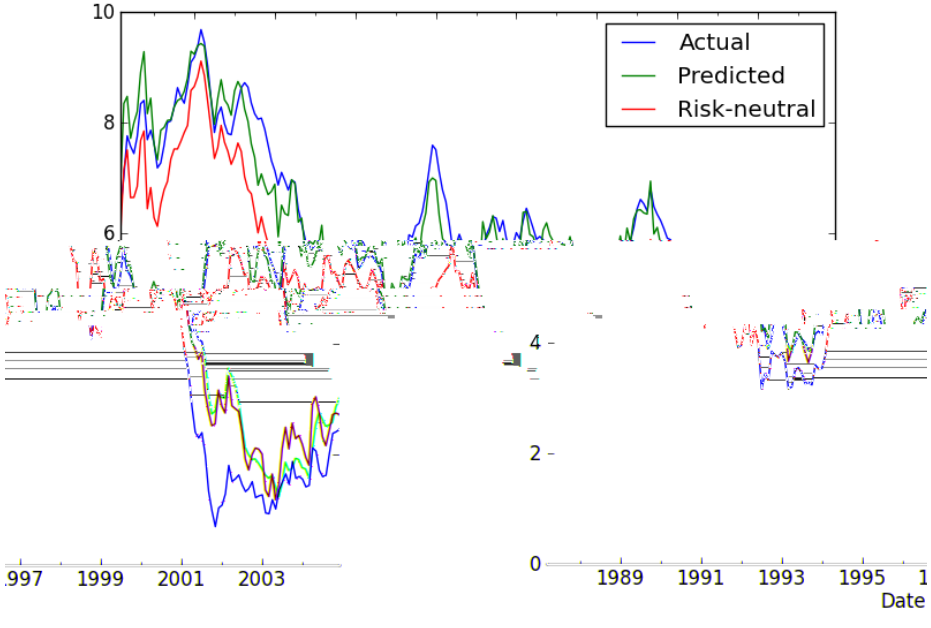

Figure 2: Author's estimation results of 2-year Treasury Constant Maturity

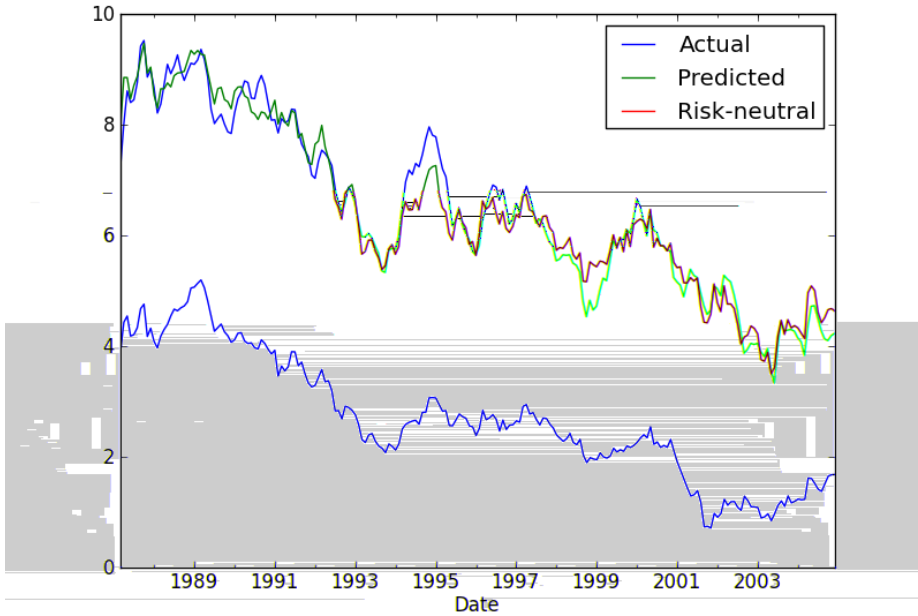

Figure 3: Author's estimation results of 10-year Treasury Constant Maturity

Maturity	VAR with ED shocks	VAR without ED shocks
6 months	17.1	18.3
1 year	21.8	25.1
2 years	24.6	34.6
3 years	23.3	36.7
5 years	21.4	37.3
7 years	19.8	34.4
10 years	20.0	33.4

Table 2: Pricing error in basis points

	λ_0	λ_1				
		Empl. Gap	Inflation	Exp. Inflation	Fed. Funds	Euro-Dollar Fut.
Empl. Gap	13605.3199	-19532.194	-777.9469	592.7662	36547.055	-3989.245
	(87417.1363)	(916243.4202)	(3444.2113)	(2989.3648)	(53123.5917)	(3704.0872)
Inflation	148.2248	3429.2952	26.4911	18.8169	-290.9901	23.1623
	(553.6984)	(8548.1197)	(43.1522)	(23.2349)	(225.6018)	(50.48)
Exp. Inflation	4.8861	-200.1545	-1.9348	-2.2384	86.8642	-9.2543
	(192.732)	(2199.7329)	(9.499)	(6.6299)	(90.9963)	(9.466)
Fed. Funds	-16.1298	-89.8569	-2.1435	0.1829	3.4266	0.3604
	(13.289)	(406.1849)	(2.185)	(1.4263)	(26.4616)	(2.7115)
EuroDollar Fut.	-30.9481	-187.8645	0.3971	-1.9902	-44.3954	4.215
	(116.5332)	(1191.8426)	(4.2237)	(4.1643)	(86.4603)	(4.9276)

Table 3: Parameter estimates with standard errors in parenthesis.

The pricing errors for the replication models with and without Eurodollar (ED) futures are presented in Table 3. While the pricing errors do not exactly match the results from BSR presented in Table 1, they do emulate two of the generate characteristics present in the BSR results. Firstly, the smallest pricing error in both the no-ED and ED model is with for the 6 month maturity yield. This is not surprising as the VAR system has more direct information to inform the pricing kernel at this end of maturity spectrum given its inclusion of the federal funds rate. Secondly, the inclusion of Eurodollar futures does reduce the pricing error for Treasury yields of all maturities in both the original BSR model and this replication.

Parameters

4 moving forward

4.1 real-time data and affine models of the term structure

At this point, my dissertation would break into three chapters, all related to affine models of the term structure.

Chapter 1 will first replicate the results of Bernanke et al. (2005) (BSR), using the identical data sources to approach their results as close as possible. After these results are replicated, the data will be extended to up the financial crisis and after the financial crisis to see if the pricing error increases or decreases. It is likely that the pricing error would deteriorate by extending the data into the crisis period, since there is no inclusion of a high-frequency indicator of uncertainty. To respond to this deterioration, one of the factors (likely the Eurodollar futures) will be replaced with an aggregate measure of credit default swaps. My hypothesis is that including this factor will lead to lower pricing error for affine models during crisis periods. This analysis for this chapter will largely build off of the analysis presented in the above section.

Chapter 2 will focus on improving the information set feeding bond pricing by using real-time rather than final macro-data. The aim of this chapter will be to answer the question: Does real-time data increase the fit of affine models of the term structure that use observed factors? While real-time data has been integrated into an affine model framework in Orphanides and Wei (2010), the focus of the paper was not on the value-added of using real-time data. Because the definition of the pricing kernel at time t is supposed to reflect all information informing bond pricing decisions at time t, the necessity of adding unobserved latent factors to an affine model with observed factors is most often to increase the fit of the model, because the observed factors alone cannot match all of the variation. This may simply be an artifact of incorrectly attributing available information regarding macroeconomic fundamentals at time t. Final versions of data, especially GDP and inflation, are often not available until quarters after the period that they refer to, making final data insuitable as a proxy for available information at time t influencing bond pricing decisions at time t.

This will require some theoretical derivation of the exact advantage of using real-time data vs. final-data in estimating term structure models. After deriving this advantage, I will estimate models using both real-time and final data from a variety of samples of U.S data and compare the two to estimate where the gain in explanatory value is statistically significant. Orphanides and Wei (2010) show that estimating the model using real-time data and time-varying parameters results in overall lower estimates of the term premium, because a higher proportion of the premium can be attributed to future short-rate expectations. I will compare my results to the results presented in Orphanides and Wei (2010) in order to place this chapter in the affine term structure model literature.

Chapter 3 will be a computational extension of the Python model solving class. In this chapter, I will write a C++ extension to the python class that I have already written that solves latent factor affine models of the terms structure. After reading through the solution methods suggested in Dai and Singleton (2000), Ang and Piazzesi (2003) and Duffee and Stanton (2004), it seems as though some of the assumptions needed to solve these models are largely arbitrary. For example, because of the large number of free parameters in the model presented in Ang and Piazzesi (2003), the authors have to assume that certain parameters are equal to 0 after they are insignificant at the 5% level and iterate over multiple steps where structural and price of risk parameters are estimated one after the other. Because of how arbitrary this seems, I'd like to investigate the implications of slightly altering their approach on the qualitative results.

It may be interesting to see if the affine model produced measure of the term premium referenced in many of the post-crisis papers results in a much lower proportion of the term premium explained by the portfolio balance channel rather than through expectations of future short-term interest rates (as in Bauer and Rudebusch (2011)). because using final data may result in a mis-specified affine model, the percentage of the change in yields on longer-maturity bonds may be incorrectly split between a term premium component and expected future short-rates.

I would also like to address the conclusion of Diebold et al. (2006) using real-time data. because Diebold et al. (2006) uses the same data on both sides of the causality (term structure causing macroeconomic fluctuations and vice versa), it would be interesting to see if their model would generate the same conclusions of 1) interest rates $\rightarrow$ macro factors is not as important as 2) macro factors $\rightarrow$ interest rates if the macro factors in 1) are replaced with real-time macro data.

4.2 Work that I have done

I have already written a maximum likelihood solver in Python for models and have used it for a previous term paper. I would like to fully utilize it and also move onto a Kalman filter approach to estimating the system. I've also worked through solving some of the DSGE models of the term premium addressed in section 5 using the permutation solution algorithm proposed by Swanson et al. (2005).

5 Structural models of the term structure

This section is not as relevant as before. Moving forward, I will be focusing exclusively on affine models of the term structure for my own investigation, but will use the results in reached in DSGE models of the term structure as a reference for how affine models of the term structure fare computationally, statistically, and structurally within the broader realm of models of the term structure.

While affine models of the term structure offer a rigorous way of explaining movements in the term structure, the framework offers little in the way of microeconomic foundations or structural interpretations. Given this, an entire parallel literature attempting to model movements in the term structure of interest rates with structural macroeconomic models has grown. This literature has progressed in a way where a series of issues are dealt with in matching structural models to the data. Even though Mehra and Prescott (1985) focused on the equity premium, rather than the term premium, the insight of their paper held just as important to the term premium literature: the risk aversion parameter plays a primary role in determining the differential yields among securities, and in order to fit the yields on equity, estimates of this parameter are way beyond any reasonable microeconomic estimates. This observation became a vital crux of the consumption based models of the term structure in 1990s. Donaldson et al. (1990), using a popular consumption-based model with power preferences, showed that the simple model even with huge shocks to consumption was incapable of generating the real term structure. Expected utility in this model, along with the other consumption-based models of the 1990s, took the following form:

$$E_t(\sum_{t=0}^{\infty}\beta^t\frac{c_t^{\gamma}-1}{\gamma}) \tag{5.1}$$

where β is the inter-temporal discount factor, c_t is consumption in period t, and γ is a parameter.

The functional form of expected utility ends up playing a key role in determining the ability of structural models' to generate a sizeable term premium. Reinforcing Donaldson, Johnsen, and Mehra conclusion, den Haan (1995) showed in order to generate a sizeable term premium in this simple consumption model, persistent expected growth rates of consumption and negative autocorrelation of consumption growth are required. Neither of these are present in the U.S data. Without these characteristics, the standard consumption model results in a negative sloping yield curve.

One of the first papers to offer an innovation as a possible way of generating a sufficient asset price premium was Campbell and Cochrane (1999). While this paper intended to explain the equity premium, the same function form was adopted by much of the following term premium literature. The driving innovation in this paper was the introduction of a habit to the utility specification:

$$E_t(\sum_{t=0}^{\infty}\beta^t \frac{(c_t - X_t)^(1-\gamma) - 1}{1-\gamma}) \tag{5.2}$$

where X_t is the habit.

Because the representative agent derives utility from some value over and above some baseline value, the habit, agents aim to smooth consumption, resulting in a more persistent consumption stream compared to the standard power utility specification (Eq. 5.1). Smoother consumption results in equity prices that are correlated with movements in consumption relative to the habit parameter, making them more risky, thus requiring a higher return in compensation for that risk. This lessens the necessity of an excessively high risk aversion parameter. Even though Campbell and Cochrane were able to match the empirical equity premium using this method, they still required a very high steady state risk aversion.

Jermann (1998) moves the innovations of the above mentioned finance-consumption models into a general equilibrium, real business cycle (RBC) framework. Attempting to price both equity and bonds, Jermann utilizes both a utility specification with habit and capital adjustment costs to fit movements in equity and bond yields. Although Jermann's model results in a short-term interest rate that is much too volatile, risk-premia on bonds that are too high relative to the equity premium, and very low consumption volatility, his integration of structural rigidities into a general equilibrium framework was an important step in the development of empirically binding micro-founded term structure models.

The importance of habit is further investigated in a complete general equilibrium model in Lettau and Uhlig (2000). On top of consumption habit to generate meaningful term premia, Lettau and Uhlig attempt to overcome empirically unsupported steady consumption by introducing labor habit. Lettau and Uhlig hypothesize that volatility in labor is drawing volatility away from consumption, but applying the habit in labor habit on top of habit in consumption results in much smoother labor paths, but still relatively steady consumption paths. Altering other parameters are unable to increase consumption volatility without lessening model fit of other macroeconomic variables. Lettau and Uhlig does not attempt to fit the model to asset yield data (bonds or equity), but the purpose of the paper echoes the importance of consumption in the general equilibrium asset price literature, leaving many remaining questions for habits empirical tenability. Boldrin et al. (2001) takes a similar approach in a general equilbrium framework, introducing inter-sectoral rigidities rather than labor habit, but is still left with consumption processes that are too persistent to match to the data.

Following the seeming inability of general equilibrium models with consumption habit utility specifications to generate empirically tenable consumption streams, the structural term premium literature turned to other utility specifications. The most fruitful of these forays has been the introduction of preferences a la Epstein and Zin (1989) (EZ). In both the power utility (Eq. 5.1) and habit-adjusted power utility (Eq. 5.2) cases, a single parameter (γ) determines both the elasticity of inter-temporal substitution (EIS) and the within period level relative risk aversion (RRA). EZ preferences enter these DSGE models in the form:

$$V_t = u(c_t, l_t) + \beta(E_t V_{t+1}^{1-\alpha})^{1/(1-\alpha)} \tag{5.3}$$

with:

$$u(c_t, l_t) \equiv \frac{c_t^{1-\phi}}{1-\varphi}$$

where V_t is utility in period t and α, φ, χ_0, and χ are parameters.

The key difference, both theoretically and for matching to empirical distributions, is that the parameter governing EIS, α, is distinct from the parameter governing relative risk aversion, φ. Intuitively, this allows the model to calibrate higher term premia generated by higher RRA without smoothing consumption to empirically untenable values through lower EIS.

This important innovation in utility specification not only sparked a more fruitful investigation of DSGE term premium literature, but also inspired a revisitation of the finance-based exogenous consumption stream models. Piazzesi and Schneider (2007) show the importance of using Epstein-Zin (EZ) preferences in these models. In this simplified model, Piazzesi and Schneider show that there is an important distinction between the real and nominal risk embedded in the term premium and monopolizing on this distiction is very much a by-product of EZ preferences. Real risk is positive if, as noted above, there is positive correlation between expected real output and real bond yields. Nominal risk exists if there is a positive correlation between expected nominal output and nominal bond yields. Allowing for a time-varying real pricing kernel (Euler equation), Piazzesi and Schneider back out the very different effects that real versus nominal shocks have on the real and nominal variation in the term structure. Utilizing actual yield curve data allows the authors to analyze whether the economic activity was driven by real or nominal shocks in the post-war era, leading them to conclude that inflation shocks only played a dominant role in the 1970s and 1980s, while real shocks dominated in all other time periods.

Along with important theoretical innovations in the structural term premium models, innovations in solution algorithms have also played an important role in driving the literature forward. While linearization serves as a sufficient model estimation method for most DSGE dynamics, linearization does not allow for any differences in ex-ante returns on different securities. As emphasized in Jermann (1998), this makes first-order Taylor series approximations an ineffective way of studying the term premium, a stylized fact relying entirely on differences in asset (bond) returns of different maturities. In order to generate differing returns on bonds of different maturities, authors have resorted to other methods in order to study the term premium in these models.

Jermann (1998) proposed a two-step solution method relying on assumptions initially suggested by Hansen and Singleton (1983) in order to generate differences in yields of bonds of different maturity. First, the first order conditions of the RBC model with habit persistence and capital adjustment costs are log-linearized and re-written as a VAR(1) process. Then, assuming that all future marginal utilities and all future asset returns payments are conditionally log-normal, the returns on the assets in the model (bonds and equity) can be derived individually.

Wu (2006) makes an interesting connection between the bond pricing method proposed by Jermann (1998), the New-Keynesian DSGE framework, and affine term structure models summarized in the above section. Wu's method of solving his model, a New-Keynesian model with price adjustment costs and capital adjustment costs, first involves log-linearizing the model and transforming it. Wu then uses a state space system to summarize the system:

$$z_t = \mu_z + \Psi_z s_t \tag{5.4}$$

Where the following law of motion governs s_t:

$$s_t = \mu_s + \Psi_s s_{t-1} + \Sigma_s \varepsilon_t \tag{5.5}$$

Wu uses s_t as the summary of all information entering bond pricing decisions at time t. In the language of the affine term structure literature, s_t becomes the sole determinant of the pricing kernel of all securities. By assuming that the pricing kernel is conditionally log-normal, he overlays an entire affine model of the term structure onto the pricing kernel generated from the log-linearized state-space system of the DSGE model. Instead of relying on latent variables as in most of the affine term structure model literature, Wu generates all of the information entering the pricing decision from the DSGE model. All implied bond prices are then solved forward as in all affine models of the term structure.

While Wu's model generates realistic term structure dynamics, the solution method does not allow for time-varying term premia. Because all bond prices are generated from a lognormal approximation of the model generated bond pricing equation where variances are constant, bond prices will always react to the same shocks in the same way. In this case, a term premium exists, but this solution method is not the best route for studying the determinants of variation in the term premium.

Bekaert et al. (2010) use a similar to method to solve a NK model and arrive at similar conclusions to Wu (2006), but also help to unravel the need for persistence in structural variables for sizeable term premium. Because structural models allow for bi-directional causality between bond returns and "real" variables, Bekaert et al. show that including the term structure in a structural model increases the persistence of macro variables, making the model match the "empirical facts" of a non-theoretical VAR closely. Specifically, $VAR(n)$ with $n >= 2$ often exhibit high persitence in the macroeconomic variables detrended output and inflation that is lacking in NK models. Even though the time series representation of Bekaert et al.'s model is $VAR(1)$, the model is able to generate the persistence seen in the longer lag empirical VAR's through the identification and interaction between the term structure and the macro variables. By generating term premia without a large increase in consumption volatility, the additional structural specifications provide an important theoretical overlay to the consumption-based finance models of the term structure. In short, the additional interactions between the equations in Bekaert et al. (2010) are able to generate empirically meaningful term premia without increasing the volatility of aggregate consumption beyond reality, as required in models ala Donaldson et al. (1990).

Both Wu (2006) and Bekaert et al. (2010) spend considerable time linking the model back to the "level" and "slope" addressed in section 1. Wu finds that monetary shocks in the model have a much larger effect on short-maturity rather than longer-maturity returns, while technology shocks have more uniform effect on returns of bonds of all maturities. Wu hypothesizes that real shocks drive the "level" factor, while monetary shocks drive the "slope" factor. Bekaert et al. (2010) instead finds that the "level" factor is primarily driven by the changes in the long-run inflation target of the central bank, while the "slope" factor is driven by cyclical adjustments by the monetary authority. While the exact driving factor is different in these two cases, the "level" factor is always related to a more long-run, real determinant as opposed to the "slope" factor that is related to a nominal, business-cycle adjustments.

Second order Taylor-series approximations have also been a method through which term premium models have been solved. Hördahl et al. (2008) uses a second order Taylor-series approximation to price bonds with a New Keynesian model with habit persistence and Calvo (1983) pricing. Measuring the term premium as the difference between the yield-to-maturity of a 20 period bond and a one period bond, Hördahl et al. (2008) are able to decompose the importance of the different determinants of the term premium, finding that the real determinant of the term premium, the correlation between consumption and yields, dominates the nominal determinant of the term premium, inflation risk on longer maturity bonds. This method, while interesting for investigating the differential determinants of the term premium, cannot generate time-varying term premia, because time-variation in the term premium is generated by solution moments that are only reached when considering third-order moments.

6 LSAP events

Following the economic turmoil of the financial crisis of 2008, a slew of papers have emerged attempting to explain the method through which alternative monetary policy actions have lowered the returns on longer-maturity bonds. These investigations have focused on large scale asset purchases (LSAPs) performed by the Federal Reserve Bank. In an LSAP event, the Fed purchased large amounts of longer-maturity securities in an attempt to raise the aggregate percentage of the securities held by the government, lowering their supply, raising the bid price on these bonds and lowering the yield. Using the implied term premia generated by Kim and Orphanides (2005), Gagnon et al. (2010) attributes the majority of the decrease in yields on longer-maturity bonds to decreases in the term premium rather than lower expectations of future short-term rates. Gagnon et al. (2010) finds that even though a liquidity effect may have played a significant role in lowering term premia in the immediate aftermath of the crisis, the major contributing factor to the lower term premia was through the portfolio balance effect, first proposed by Tobin (1956). The portfolio balance effect, in summary, is where, by lowering the aggregate amount of longer-term assets available through direct purchase, the demand for these assets relative to the supply will rise, increasing the price and decreasing the yield.

Bauer and Rudebusch (2011), using a Bayesian estimation method, instead find that the LSAP events had a much larger impact through future short-rate expectations than attributed in Gagnon et al. (2010). Bauer and Rudebusch (2011) instead focuses on the signaling channel through which LSAP events lowered longer-maturity yields, rather than the portfolio balance effect. The signalling channel is in effect where the increase purchase of longer term securities decreases signals to the public that the federal funds rate target will be low for a number of years, leading to lower interest rates on longer term securities, the rates of which are largely determined by expectations of future yields. In either case, the jury is out on what the primary channel is through which these LSAP events have an impact on long-maturity interest rate.

References

Ang, A. and M. Piazzesi (2003). A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. *Journal of Monetary economics 50*(4), 745–787.

Bauer, M. D. and G. D. Rudebusch (2011). The signaling channel for federal reserve bond purchases. Working Paper Series 2011-21, Federal Reserve Bank of San Francisco.

Bekaert, G., S. Cho, and A. Moreno (2010, 02). New keynesian macroeconomics and the term structure. *Journal of Money, Credit and Banking 42*(1), 33–62.

Bernanke, B., B. Sack, and V. Reinhart (2005). Monetary policy alternatives at the zero bound: An empirical assessment. *Brookings Papers on Economic Activity 2004*(2), 1–100.

Boldrin, M., L. J. Christiano, and J. D. M. Fisher (2001, March). Habit persistence, asset returns, and the business cycle. *American Economic Review 91*(1), 149–166.

Calvo, G. (1983). Staggered prices in a utility-maximizing framework. *Journal of Monetary Economics 12*(3), 383–398.

Campbell, J. Y. and J. Cochrane (1999, April). Force of habit: A consumption-based explanation of aggregate stock market behavior. *Journal of Political Economy 107*(2), 205–251.

Cox, J. C., J. Ingersoll, Jonathan E, and S. A. Ross (1985, March). A theory of the term structure of interest rates. *Econometrica 53*(2), 385–407.

Dai, Q. and K. J. Singleton (2000, October). Specification analysis of affine term structure models. *Journal of Finance 55*(5), 1943–1978.

den Haan, W. J. (1995). The term structure of interest rates in real and monetary economies. *Journal of Economic Dynamics and Control 19*(5-7), 909–940.

Diebold, F. X., G. D. Rudebusch, and S. Borag[caron]an Aruoba (2006). The macroeconomy and the yield curve: a dynamic latent factor approach. *Journal of Econometrics 131*(1-2), 309–338.

Donaldson, J. B., T. Johnsen, and R. Mehra (1990, October). On the term structure of interest rates. *Journal of Economic Dynamics and Control 14*(3-4), 571–596.

Duffee, G. and R. Stanton (2004). Estimation of dynamic term structure models.

Duffie, D. and R. Kan (1996). A yield-factor model of interest rates. *Mathematical Finance 6*(4), 379–406.

Epstein, L. G. and S. E. Zin (1989, July). Substitution, risk aversion, and the temporal behavior of consumption and asset returns: A theoretical framework. *Econometrica 57*(4), 937–69.

Federal Reserve Bank of St. Louis (2011, 9). Federal reserve economic data. `http://research.stlouisfed.org/fred2/`.

Gagnon, J., M. Raskin, J. Remache, and B. Sack (2010). Large-scale asset purchases by the Federal Reserve: Did they work? *Federal Reserve Bank of New York Sta Report 441*.

Hansen, L. P. and K. J. Singleton (1983, April). Stochastic consumption, risk aversion, and the temporal behavior of asset returns. *Journal of Political Economy 91*(2), 249–65.

Hördahl, P., O. Tristani, and D. Vestin (2008). The yield curve and macroeconomic dynamics*. *The Economic Journal 118*(533), 1937–1970.

Jermann, U. J. (1998, April). Asset pricing in production economies. *Journal of Monetary Economics 41*(2), 257–275.

Kim, D. H. and A. Orphanides (2005, November). Term structure estimation with survey data on interest rate forecasts. CEPR Discussion Papers 5341, C.E.P.R. Discussion Papers.

Kim, D. H. and J. H. Wright (2005). An arbitrage-free three-factor term structure model and the recent behavior of long-term yields and distant-horizon forward rates. Finance and Economics Discussion Series 2005-33, Board of Governors of the Federal Reserve System (U.S.).

Lettau, M. and H. Uhlig (2000, January). Can habit formation be reconciled with business cycle facts? *Review of Economic Dynamics 3*(1), 79–99.

Litterman, R. and J. Scheinkman (1991). Common factors affecting bond returns. *The Journal of Fixed Income 1*(1), 54–61.

Mehra, R. and E. C. Prescott (1985, March). The equity premium: A puzzle. *Journal of Monetary Economics 15*(2), 145–161.

Nelson, C. R. and A. F. Siegel (1987, October). Parsimonious modeling of yield curves. *The Journal of Business 60*(4), 473–89.

Orphanides, A. (2001). Monetary policy rules based on real-time data. *American Economic Review 91* (4), 964–985.

Orphanides, A. and M. Wei (2010). Evolving macroeconomic perceptions and the term structure of interest rates. Finance and Economics Discussion Series 2010-01, Board of Governors of the Federal Reserve System (U.S.).

Piazzesi, M. and M. Schneider (2007, June). Equilibrium yield curves. In *NBER Macroeconomics Annual 2006, Volume 21*, NBER Chapters, pp. 389–472. National Bureau of Economic Research, Inc.

Rudebusch, G. and T. Wu (2004). A macro-finance model of the term structure, monetary policy, and the economy. *Proceedings* (Mar).

Rudebusch, G. and T. Wu (2008). A macro-finance model of the Term Structure, Monetary Policy and the Economy*. *The Economic Journal 118* (530), 906–926.

Swanson, E., G. Anderson, and A. Levin (2005, November). Higher-order perturbation solutions to dynamic, discrete-time rational expectations models: Methods and an application to optimal monetary policy. Computing in Economics and Finance 2005 146, Society for Computational Economics.

Thomson Reuters and University of Michigan (2010, 10). Survey of consumers. `http://www.bea.gov/national/nipaweb/SelectTable.asp`.

Tobin, J. (1956). Liquidity preference as behavior towards risk. Cowles Foundation Discussion Papers 14, Cowles Foundation for Research in Economics, Yale University.

Vasicek, O. (1977, November). An equilibrium characterization of the term structure. *Journal of Financial Economics 5* (2), 177–188.

Wu, T. (2006, October). Macro factors and the affine term structure of interest rates. *Journal of Money, Credit and Banking 38* (7), 1847–1875.

7 Code

```
"""
This defines the class objection Affine, intended to solve affine models of the
```

```python
term structure
This class inherits from statsmodels LikelihoodModel class
"""

import numpy as np
import statsmodels.api as sm
import pandas as px
import scipy.linalg as la

from statsmodels.tsa.api import VAR
from statsmodels.base.model import LikelihoodModel
from statsmodels.tools.numdiff import (approx_hess, approx_fprime)
from operator import itemgetter
from scipy import optimize

#debugging
#import pdb

#############################################
# Create affine class system                #
#############################################

class Affine(LikelihoodModel):
    """
    This class defines an affine model of the term structure
    """
    def __init__(self, yc_data, var_data, rf_rate=None, maxlags=4,
                 freq='M', latent=0, no_err=None):
        """
        Attempts to solve affine model
        yc_data : yield curve data
```

```
        var_data : data for var model
        rf_rate : rf_rate for short_rate, used in latent factor case
        max_lags: number of lags for VAR system
        freq : frequency of data
        latent : # number of latent variables
        no_err : list of the yields that are estimated without error
        """
        self.yc_data = yc_data

        #gen VAR instance
        mod = VAR(var_data, freq=freq)
        vreg = mod.fit(maxlags=maxlags)

        #generates mths and mth_only
        self._proc_to_mth()

        #number of latent variables to include
        lat = self.latent = latent
        self.no_err = no_err

        self.k_ar = k_ar = vreg.k_ar
        self.neqs = neqs = vreg.neqs
        self.params = params = vreg.params.values
        sigma_u = vreg.sigma_u

        if lat:
            assert len(no_err) >= lat, "One yield estimated with no err"\
                                       + "for each latent variable"

            #order is lam_0, lam_1, delt_1, phi, sig
            len_lst = [neqs+lat, (neqs + lat)**2, lat, lat**2, lat**2]
```

```
        pos_lst = []
        acc = 0
        for lengths in len_lst:
            pos_lst.append(lengths+acc)
            acc += lengths
        self.pos_lst = pos_lst
        yc_data_cols = yc_data.columns.tolist()
        self.noerr_indx = list(set(yc_data_cols).intersection(no_err))
        self.err_indx = list(set(yc_data_cols).difference(no_err))

    mu = np.zeros([k_ar*neqs+lat, 1])
    mu[:neqs] = params[0, None].T
    self.mu = mu

    phi = np.zeros([k_ar*neqs, k_ar*neqs])
    phi[:neqs] = params[1:].T
    phi[neqs:, :(k_ar-1)*neqs] = np.identity((k_ar-1)*neqs)
    self.phi = phi

    sig = np.zeros([k_ar*neqs, k_ar*neqs])
    sig[:neqs, :neqs] = sigma_u
    self.sig = sig

    if lat == 0:
        self.delta_0 = 0
        delta_1 = np.zeros([neqs*k_ar, 1])
        #delta_1 is vector of zeros, with one grabbing fed_funds rate
        delta_1[np.argmax(var_data.columns == 'fed_funds')] = 1
        self.delta_1 = delta_1

    else:
```

```
        #this is the method outlined by Ang and Piazzesi (2003)
        reg_data = var_data.copy()
        reg_data['intercept'] = 1
        par = sm.OLS(rf_rate, reg_data).fit().params
        self.delta_0 = par.values[-1]
        delta_1 = np.zeros([neqs*k_ar+lat, 1])
        delta_1[:neqs, 0] = par.values[:neqs]
        self.delta_1 = delta_1

    #get VAR input data ready
    x_t_na = var_data.copy()
    for lag in range(k_ar-1):
        for var in var_data.columns:
            x_t_na[var + '_m' + str(lag+1)] = px.Series(var_data[var].
                    values[:-(lag+1)], index=var_data.index[lag+1:])

    #check this, looks fine
    self.var_data = x_t_na.dropna(axis=0)

    super(Affine, self).__init__(var_data)


def solve(self, lam_0_g, lam_1_g, method="ls", delt_1_g=None, phi_g=None,
        sig_g=None, maxfev=10000, ftol=1e-100, xtol=1e-100,
        full_output=False):
    """
    Attempt to solve affine model

    method : string
        ls = linear least squares
        cf = nonlinear least squares
```

```
lam_0_g : array (n x 1),
    guess for elements of lam_0
lam_1_g : array (n x n),
    guess for elements of lam_1
delt_1_g : array
    guess for elements of delt_1
phi_g : array
    guess for elements of phi
sig_g : array
    guess for elements of sigma
scipy.optimize.leastsq params
maxfev : int
    maximum number of calls to the function for solution alg
ftol : float
    relative error desired in sum of squares
xtol : float
    relative error desired in the approximate solution
full_output : bool
    non_zero to return all optional outputs
"""
lat = self.latent
neqs = self.neqs
lam = []
x_t = self.var_data
mth_only = self.mth_only

#assert np.shape(lam_0_g) == neqs + lat, "Length of lam_0_g not correct"
#assert len(lam_1_g) == (neqs + lat)**2, "Length of lam_1_g not correct"
#creates single input vector for params to solve
if lat:
    assert len(delt_1_g) == lat, "Length of delt_1_g not correct"
```

```
    assert len(phi_g) == lat**2, "Length of phi_g not correct"
    assert len(sig_g) == lat**2, "Length of sig_g not correct"
    lam = np.asarray(lam_0_g + lam_1_g + delt_1_g + phi_g + sig_g)
else:
    lam_0_list = flatten(lam_0_g[:neqs])
    lam_1_list = flatten(lam_1_g[:neqs, :neqs])
    for param in range(len(lam_0_list)):
        lam.append(lam_0_list[param])
    for param in range(len(lam_1_list)):
        lam.append(lam_1_list[param])

#this should be specified in function call
if method == "ls":
    func = self._affine_nsum_errs
    reslt = optimize.leastsq(func, lam, maxfev=maxfev,
                             xtol=xtol, full_output=full_output)
    lam_solv = reslt[0]
    output = reslt[1:]
    func = self._affine_pred
elif method == "cf":
    func = self._affine_pred
    #need to stack
    yield_stack = self._stack_yields(mth_only)
    #run optmization
    reslt = optimize.curve_fit(func, x_t, yield_stack, p0=lam,
                                   maxfev=maxfev, xtol=xtol,
                                   full_output=full_output)

    lam_solv = reslt[0]
    lam_cov = reslt[1]
```

```
        lam_0, lam_1, delta_1, phi, sig = self._proc_lam(*lam_solv)

        a_solve, b_solve = self.gen_pred_coef(lam_0, lam_1, delta_1, phi, sig)

        #if full_output:
            #return lam_0, lam_1, delta_1, phi, sig, a_solve, b_solve, output
        if method == "cf":
            return lam_0, lam_1, delta_1, phi, sig, a_solve, b_solve, lam_cov
        elif method == "ls":
            return lam_0, lam_1, delta_1, phi, sig, a_solve, b_solve, output

    def score(self, lam):
        """
        Return the gradient of the loglike at AB_mask.

        Parameters
        ----------
        AB_mask : unknown values of A and B matrix concatenated

        Notes
        -----
        Return numerical gradient
        """
        loglike = self._affine_nsum_errs
        return approx_fprime(lam, loglike, epsilon=1e-8)

    def hessian(self, lam):
        """
        Returns numerical hessian.
        """
        loglike = self._affine_nsum_errs
```

```
    return approx_hess(lam, loglike)[0]

#def loglike(self, params):
#    """
#    Loglikelihood used in latent factor models
#    """
#    # here is the likelihood that needs to be used
#    # sig is implied VAR sig
#    # use two matrices to take the difference
#    like = -(T - 1) * np.logdet(J) - (T - 1) * 1.0 / 2 * \
#            np.logdet(np.dot(sig, sig.T)) - 1.0 / 2 * \
#            np.sum(np.dot(np.dot(errors.T, np.inv(np.dot(sig, sig.T))),\
#                          err)) - (T - 1) / 2.0 * \
#            np.log(np.sum(np.var(meas_err, axis=1))) - 1.0 / 2 * \
#            np.sum(meas_err/np.var(meas_err, axis=1))

def gen_pred_coef(self, lam_0_ab, lam_1_ab, delta_1, phi, sig):
    """
    Generates prediction coefficient vectors A and B
    lam_0_ab : array
    lam_1_ab : array
    delta_1 : array
    phi : array
    sig : array
    """
    mths = self.mths
    delta_0 = self.delta_0
    mu = self.mu
    max_mth = max(mths)
    #generate predictions
    a_pre = np.zeros((max_mth, 1))
```

```
        a_pre[0] = -delta_0
        b_pre = []
        b_pre.append(-delta_1)
        for mth in range(max_mth-1):
            a_pre[mth+1] = (a_pre[mth] + np.dot(b_pre[mth].T, \
                            (mu - np.dot(sig, lam_0_ab))) + \
                            (1.0/2)*np.dot(np.dot(np.dot(b_pre[mth].T, sig), \
                            sig.T), b_pre[mth]) - delta_0)[0][0]
            b_pre.append(np.dot((phi - np.dot(sig, lam_1_ab)).T, \
                                b_pre[mth]) - delta_1)
        n_inv = 1.0/np.add(range(max_mth), 1).reshape((max_mth, 1))
        a_solve = -(a_pre*n_inv)
        b_solve = np.zeros_like(b_pre)
        for mths in range(max_mth):
            b_solve[mths] = np.multiply(-b_pre[mths], n_inv[mths])
        return a_solve, b_solve

    def _affine_nsum_errs(self, lam):
        """
        This function generates the sum of the prediction errors
        """
        lat = self.latent
        mths = self.mths
        mth_only = self.mth_only
        x_t = self.var_data

        lam_0, lam_1, delta_1, phi, sig = self._proc_lam(*lam)

        a_solve, b_solve = self.gen_pred_coef(lam_0, lam_1, delta_1, phi, sig)

        #this is explosive
```

```
        if lat:
            x_t = self._solve_x_t_unkn(a_solve, b_solve)

        errs = []

        for i in mths:
            act = np.flipud(mth_only['l_tr_m' + str(i)].values)
            pred = a_solve[i-1] + np.dot(b_solve[i-1].T, np.fliplr(x_t.T))[0]
            errs = errs + (act-pred).tolist()
        return errs

    def _solve_x_t_unkn(self, a_in, b_in, x_t = None):
        """
        This is still under development
        It should solve for the unobserved factors in the x_t VAR data
        """
        lat = self.latent
        no_err = self.no_err
        mth_only = self.mth_only
        yc_data = self.yc_data
        x_t_new = np.append(x_t, np.zeros((x_t.shape[0], lat)), axis=1)
        errors = x_t[1:] - mu - np.dot(phi, x_t[:-1])
        if x_t is None:
            x_t = self.var_data
        T = x_t.shape[0]

        # solve for unknown factors
        noerr_indx = self.noerr_indx
        a_noerr = select_rows(noerr_indx, a_in)
        b_0_noerr = select_rows(noerr_indx, b_in)
```

```
        # this is the right hand for solving for the unobserved latent
        # factors
        r_hs = yc_data[no_err] - a_noerr[None].T - np.dot(b_0_noerr, x_t)
        lat = la.solve(b_u, r_hs)

        #solve for pricing error on other yields
        err_indx = self.err_indx
        a_err = select_rows(err_indx, a_in)
        b_0_err = select_rows(err_indx, b_in)
        r_hs = yc_data[no_err] - a_noerr[None].T - np.dot(b_0_noerr, x_t)
        meas_err = la.solve(b_m, r_hs)

        #create Jacobian (J) here

        #this taken out for test run, need to be added back in
        #J =

    def _proc_to_mth(self):
        """
        This function transforms the yield curve data so that the names are all
        in months
        (not sure if this is necessary)
        """
        frame = self.yc_data
        mths = []
        fnd = 0
        n_cols = len(frame.columns)
        for col in frame.columns:
            if 'm' in col:
                mths.append(int(col[6]))
                if fnd == 0:
```

```
                mth_only = px.DataFrame(frame[col],
                        columns = [col],
                        index=frame.index)
                fnd = 1
            else:
                mth_only[col] = frame[col]
        elif 'y' in col:
            mth = int(col[6:])*12
            mths.append(mth)
            mth_only[('l_tr_m' + str(mth))] = frame[col]
    col_dict = dict([( mth_only.columns[x], mths[x]) for x in
                range(n_cols)])
    cols = np.asarray(sorted(col_dict.iteritems(),
                    key=itemgetter(1)))[:,0].tolist()
    mth_only = mth_only.reindex(columns = cols)
    mths.sort()
    self.mths = mths
    self.mth_only = mth_only

#def _unk_likl(self):
#     likl = -(T-1)*np.logdet(J) - (T-1)*1.0/2*np.logdet(np.dot(sig,\
#             sig.T)) - 1.0/2*

def _proc_lam(self, *lam):
    """
    Process lam input into appropriate parameters
    """
    lat = self.latent
    neqs = self.neqs
    k_ar = self.k_ar
```

```
if lat:

    pos_lst = self.pos_lst

    lam_0_est = lam[:pos_lst[0]]
    lam_1_est = lam[pos_lst[0]:pos_lst[1]]
    delt_1_g = lam[pos_lst[1]:pos_lst[2]]
    phi_g = lam[pos_lst[2]:pos_lst[3]]
    sig_g = lam[pos_lst[3]:]

    lam_0 = np.zeros([k_ar*neqs+lat, 1])
    lam_0[:neqs, 0] = np.asarray(lam_0_est[:neqs]).T
    lam_0[-lat:, 0] = np.asarray(lam_0_est[-lat:]).T

    lam_1 = np.zeros([k_ar*neqs+lat, k_ar*neqs+lat])
    lam_1[:neqs, :neqs] = np.reshape(lam_1_est[:neqs**2], (neqs, neqs))
    nxt = neqs*lat
    lam_1[:neqs, -lat:] = np.reshape(lam_1_est[neqs**2:\
                                    neqs**2 + nxt],(neqs,lat))
    nxt = nxt + neqs**2
    lam_1[-lat:, :neqs] = np.reshape(lam_1_est[nxt: \
                                    nxt+lat*neqs], (lat, neqs))
    nxt = nxt + lat*neqs
    lam_1[-lat:, -lat:] = np.reshape(lam_1_est[nxt: \
                                    nxt + lat**2], (lat, lat))
    delta_1 = self.delta_1.copy()
    delta_1[-lat:, 0] = np.asarray(delt_1_g)

    #add rows/columns for unk params
    phi_n = self.phi.copy()
    add = np.zeros([lat, np.shape(phi_n)[1]])
```

```
            phi_n = np.append(phi_n, add, axis=0)
            add = np.zeros([np.shape(phi_n)[0], lat])
            phi = np.append(phi_n, add, axis=1)
            #fill in parm guesses
            phi[-lat:, -lat:] = np.reshape(phi_g, (lat, lat))

            #add rows/columns for unk params
            sig_n = self.sig.copy()
            add = np.zeros([lat, np.shape(sig_n)[1]])
            sig_n = np.append(sig_n, add, axis=0)
            add = np.zeros([np.shape(sig_n)[0], lat])
            sig = np.append(sig_n, add, axis=1)
            sig[-lat:, -lat:] = np.reshape(sig_g, (lat, lat))

        else:
            lam_0_est = lam[:neqs]
            lam_1_est = lam[neqs:]

            lam_0 = np.zeros([k_ar*neqs, 1])
            lam_0[:neqs] = np.asarray([lam_0_est]).T

            lam_1 = np.zeros([k_ar*neqs, k_ar*neqs])
            lam_1[:neqs, :neqs] = np.reshape(lam_1_est, (neqs, neqs))

            delta_1 = self.delta_1
            phi = self.phi
            sig = self.sig

        return lam_0, lam_1, delta_1, phi, sig

    def _affine_pred(self, x_t, *lam):
```

```
    """
    Function based on lambda and x_t that generates predicted yields
    x_t : X_inforionat
    """
    mths = self.mths
    mth_only = self.mth_only

    lam_0, lam_1, delta_1, phi, sig = self._proc_lam(*lam)

    a_test, b_test = self.gen_pred_coef(lam_0, lam_1, delta_1, phi, sig)

    pred = px.DataFrame(index=mth_only.index)

    for i in mths:
        pred["l_tr_m" + str(i)] = a_test[i-1] + np.dot(b_test[i-1].T,
                                            x_t.T).T[:,0]

    pred = self._stack_yields(pred)

    return pred

def _stack_yields(self, orig):
    """
    Stacks yields into single column ndarray
    """
    mths = self.mths
    obs = len(orig)
    new = np.zeros((len(mths)*obs))
    for col, mth in enumerate(orig.columns):
        new[col*obs:(col+1)*obs] = orig[mth].values
    return new
```

```python
def flatten(array):
    """
    Flattens array to list values
    """
    a_list = []
    if array.ndim == 1:
        for index in range(np.shape(array)[0]):
            a_list.append(array[index])
        return a_list
    elif array.ndim == 2:
        rshape = np.reshape(array, np.size(array))
        for index in range(np.shape(rshape)[0]):
            a_list.append(rshape[index])
        return a_list


def select_rows(rows, array):
    """
    Creates 2-dim submatrix only of rows from list rows
    array must be 2-dim
    """
    if array.ndim == 1:
        new_array = array[rows[0]]
        if len(rows) > 1:
            for row in rows[1:]:
                new_array = np.append(new_array, array[row])
    elif array.ndim == 2:
        new_array = array[rows[0], :]
        if len(rows) > 1:
            for row in enumerate(rows[1:]):
                new_array = np.append(new_array, array[row, :], axis=0)
```

```
return new_array
```