# ROBUST ESTIMATION OF TERM STRUCTURE OF INTEREST RATES AND IMPLIED VOLATILITY IN EMERGING MARKETS

A Thesis

by

Emrah Ahi

Submitted to the
The Graduate School Of Business
In Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in the
Department of Business Administration

Özyeğin University
July 2016

*To my family*

# ABSTRACT

Despite powerful advances in interest rate curve modeling for data-rich countries in the last 30 years, comparatively little attention has been paid to the key practical problem of estimation of the term structure of interest rates for emerging markets. This may be partly due to limited data availability. However, emerging bond markets are becoming increasingly important and liquid. It is, therefore, important to understand whether conclusions drawn from developed countries carry over to emerging markets. I estimate model parameters of fully flexible Nelson Siegel Svensson term structures model which has become one of the most popular term structure model among academics, practitioners, and central bankers. I investigate four sets of bond data: U.S. Treasuries, and three major emerging market government bond data-sets (Brazil, Mexico and Turkey). I found that gradient and direct search methods perform poorly in estimating term structures of interest rates, while global optimization methods, particularly the hybrid particle swarm optimization, do well. Results are consistent across countries, both in- and out-of-sample, and for perturbations in prices and starting values. Another asset class I used the Nelson Siegel model is FX options where volatility smile for both emerging and developed markets is consistent with factor analysis in which three factor explains almost 100 % of the variation. I examine a number of models from literature to test whether they are consistent on the trading of options on the currencies from the over-the-counter market. I examine the in-sample and out-of-sample performance of the Nelson-Siegel model and found it has a superior performance when compared with benchmark models on FX options data set.

# ÖZETÇE

Son 30 yılda gelişmiş ülkelerde faiz eğrisi modellemesinde çok önemli gelişmeler olmasına rağmen, gelişmekte olan ülkelerde faiz eğrisi modellemesine çok az önem verilmiştir. Kısmen bunun nedeni sınırlı veriye sahip olunmasıdır. öte yandan, gelişmekte olan ülkelerin tahvil piyasaları önem kazanmakta ve daha likit hale gelmektedir. Dolayısıyla, gelişmiş ülke piyasalarındaki sonuçların gelişmekte olan ülke piyasalarına taşınıp taşınmadığının anlaşılması önemlidir. Bu çalışmada akademisyenler, pratisyenler ve merkez bankacıları arasında en popüler faiz eğrisi modellerinden biri olan Nelson-Siegel-Svensson modelinin parametrelerini hesaplıyorum. Dört farklı veri setini inceliyorum : Amerikan hazine tahvilleri ve üç ana gelişmekte olan ülkenin devlet tahvilleri (Brezilya, Meksika ve Türkiye). Gradyan tabanlı yöntemler ve doğrudan arama yöntemlerin faiz eğrisi modeli hesaplamasında zayıf performans gösterdiğini, öte yandan global optimizasyon yöntemlerinin, özellikle hibrid parçacık sürü optimizasyonun daha iyi sonuç verdiği saptanmıştır. örneklem içi ve dışı ve perturbasyon sonuçlarının ülkeler arasında tutarlı olduğu gözlemlenmiştir. Nelson-Siegel modelini kullandığım diğer bir varlık sınıfı da döviz kuru opsiyonlarıdır. Burada hem gelişmiş hem de gelişmekte olan ülkelerde oynalık eğrisinin tüm varyasyonunun neredeyse % 100'ünün üç faktörle açıklandığı faktör analizi ile tutarlı olduğu gözlemlenmiştir. Literatürdeki birçok modelin tezgah üstü döviz kuru opsiyonlarının alım satımında tutarlı olup olmadığı incelenmiştir. Nelson-Siegel modelinin döviz kuru opsiyon verisi ile örneklem içi ve dışı performanslarını inceledim ve benzer modellere göre üstün performansı olduğunu tespit edilmiştir.

# ACKNOWLEDGEMENTS

I am enormously grateful to my supervisor, Dr. Emrah Sener, for his precious suggestions and recommendations; this work owns an important intellectual debt to him. I am also glad to acknowledge my gratitude to the Graduate School of Business at Ozyegin University. I wish to thank all PhD students of the Istanbul Risk Management Laboratory for constructive debates; I am also grateful to them for profound friendship and kind help. Last, but not least, I wish to express my gratitude to my parents and my brother for their patience and precious support.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER I

# TERM STRUCTURE ESTIMATION : EMERGING VS DEVELOPED MARKETS

## *1.1 Introduction*

The term structure of interest rates is one of the most important empirical constructs in financial economics and in the practice of finance. It is a fundamental concept not only in economic and financial theory, but also to pricing and risk management of fixed income securities and interest rate contingent claims, and long-dated liabilities such as pensions and life insurance. The term structure of interest rates, and forecasts derived from the term structure are critical to monetary policy. It is also the basis for all theoretical models of interest rates and bond pricing. Such theoretical models are validated using empirical term structures.

However, the term structure of interest rates is not usually directly observable.[1] It must be estimated from coupon bond prices. Unfortunately, rarely in financial economics is the contrast between theory and reality more troublesome than in the problem of the term structure estimation. While conceptually straightforward, the estimation of zero coupon or spot rates from observed coupon bond prices creates various theoretical and numerical obstacles. This is important as slight differences in the estimated term structures may result in significant differences when pricing bonds or fixed income portfolios, in forecasts, or hedging solutions.

[1]Zero coupon bonds are available in the form of Treasury Bills with maturities limited to one year, and Treasury STRIPS (Separate Trading of Register Interest and Principal Securities) out to 30 years.

The problems inherent in estimating term structures of interest rates from coupon bond prices include the distribution of maturities of bonds used in the estimation process, and the resulting distribution of cash flows to be discounted. These give rise to optimization problems with objective functions with multiple local minima (leading to false convergence). Numerical problems may also arise from the high sensitivity of the optimization algorithm to the initial starting values which can cause great variability in the estimated parameters. Another potential source of difficulty can be the properties of the functions used to fit the term structure. The Nelson-Siegel (NS) and Nelson-Siegel-Svensson (NSS) functional forms have become two of the most popular models among academics, practitioners, and central bankers. [2] , [3] These functional forms are widely used in the management of pension liabilities and life insurance policies, both to discount liabilities and to fit longevity curves. [4] However, these functions, which I employ in this study, are known to be difficult to estimate. All these factors combine to make the fitted term structure sensitive to the numerical methods applied in the estimation process. These issues are not unique to estimating the term structure model employed in this study. My exploration of these problems in the context of the NSS model may therefore shed light on dealing with similar difficulties in other contexts.

The objective of this study is to understand why the NS and NSS functions, which

[2]Central banks report their term structure estimates and methodologies to the Bank for International Settlements (BIS). Most of them use the NS or NSS functional forms to estimate term structure. (BIS, 2005)

[3]Term structure estimation (or fitting) models are to be distinguished from models of interest rate dynamics. The latter begin with assumptions as to the stochastic processes driving interest rates and then frequently derive analytic formulas for the theoretical term structure. These theoretical models are then fitted to zero coupon curves to derive parameter estimates for the underlying term structure process. See for example Brown and Dybvig (1986).

[4]Valuation of such long dated liabilities is particularly sensitive to small variations in the discount rates employed. For example, the present value of a $1 billion, 20-year defined benefit liability valued at five percent would be $377 million. At 5.10 percent the present value would be $370 million, a 1.8 percent change. Ten basis points is well within the measurement error I find for some optimization methods in my tests.

otherwise have many desirable properties, are so difficult to employ in practice, and to systematically document the link between optimization methods employed and the resulting problems in the estimated spot rates. I explore a number of alternative optimization algorithms, including a new algorithm, which I call the "Hybrid Particle Swarm Optimization (Hybrid PSO)" method.[5]

I investigate four sets of bond data: U.S. Treasuries, and three major emerging market sovereign bond data-sets (Brazil, Mexico and Turkey). These emerging sovereign bond markets share three important features: (i) they have large and liquid Eurobond markets in which search and trading costs are low; (ii) they belong to JPMorgan EMBI Global Index, an investable index for emerging market bonds and account 30% of the index which is quite representative of the emerging market as a whole; (iii) they are among the most frequently traded external debt markets (according to volume surveys conducted by the Emerging Market Trading Association). My data set covers the time period between July 2005 and December 2015. The advantage of using this longer sample period for the fixed income markets is that it covers three major financial crises (the U.S. subprime crisis, the European sovereign debt crisis and FED tapering tantrum) rather than just the relatively uneventful period. This provides us an unusual laboratory to study relative performance of optimization algorithms where interest rates fluctuate so widely that large fluctuations in the parameters are most likely. By including both the very dense U.S. data and the comparatively sparse emerging market data, it is ensured that are results are not specific to a particular data-set.

[5]Manousopoulos and Michalopoulosa (2007) also tests a variety of optimization methods to estimate the NSS functional form and so is most similar to my study in the existing literature. My study differs from theirs in using multiple data-sources, extensive robustness tests, and a new optimization method which I find to be superior the alternatives. I also reach a different conclusion as to which method is recommended for estimating NS and NSS term structures.

I began by showing why the NS and NSS functional forms were attractive and widely used for estimating the term structure of interest rates from bond prices. I then noted that the NS and in particular the NSS functions could lead to potential degeneracy that would make parameter estimation challenging. However, not all optimization methods are equally susceptible to these problems. The types of degeneracy that the NSS function is prone to create greater problems for gradient and direct search methods then for methods based on global optimization methods. The relative immunity of global optimization algorithms to problems inherent in the NS and NSS functions was confirmed repeatedly in my in-sample, out-of-sample, and robustness tests. I find that the global optimization methods, and in particular Hybrid PSO algorithm, generally showed the best in and out-of-sample goodness-of-fits. My findings also reveal that Hybrid PSO is quite robust to perturbations of initial starting values and bond prices, validating that the superior performance of the Hybrid PSO methods lies in their using multiple starting values, employing a randomized search process, and keeping track of past best solutions (i.e., having memory). Thus, I demonstrate that with the appropriate choice of algorithm the NS and NSS models can be reliably employed.

It is found that for this important computational problem, gradient-based methods are generally unsatisfactory. Among the particle swarm or global optimization algorithms, I find that Hybrid PSO achieves a better in-sample solution to the optimization problem than other methods, while producing stable solutions that are computationally efficient. In out-of-sample fit tests, the best test of whether the estimated term structure is reliable, the Hybrid PSO also outperformed the alternatives. PSO is also quite robust to perturbations of initial starting values and bond prices. Thus, it is demonstrated that with the appropriate choice of algorithm the NS and NSS models can be reliably employed.

## *1.2 Literature Survey*

There are two distinct approaches for modeling the yield curve. The first approach is based on *curve-fitting* techniques – a direct specification of the bond prices as a function of some parameters and the time to maturity (the cross-sectional dimension). The second approach is based on models which make explicit assumptions about the dynamics of state variables (the time-series dimension) and asset pricing methods (using either equilibrium or arbitrage arguments), which in turn result in cross-sectional models for bond prices. Furthermore, curve-fitting techniques can be classified into two groups: (i) *spline-based* models and (ii) *function-based* models. In this study I focus on function-based models. Discussion of spline-based and function-based models can be found in Appendix 4.1.

Within the class of the function based models, Nelson and Siegel (1987) were the first to use an exponential polynomial functional form. Svensson (1994) extended Nelson and Siegel's function by adding a fourth term with two additional parameters, aiming at increasing the flexibility of the model and improving the fit. This latter model, called as the NSS yield curve, has become one of the most popular yield curve models among the central bankers (BIS, 2005) and practitioners, due to the fact that it gives a good estimate of the yield curve and its parameters provide estimates for a number of factors which are interpreted as level, slope, and curvature. Studies by Bliss (1997a),Bliss (1997b),Seppala and Viertio (1996) suggested that practitioners requiring a reliable and parsimonious representation of the yield curve should use an exponential polynomial approach, in preference to the spline-based approaches.

There is a vast selection in the curve fitting techniques. A wide number of various techniques caused by two reasons: (i) lack of theoretical foundations and (ii) the trade off between smoothness and goodness of fit. Since there is no theoretical explanations for huge gaps between the spot rates, discount factors or forward rates of close maturities, the smoothness of the curve is a important criterion for a model. Such fluctuations especially in long end of the curves are completely unreasonable. The closer the fit to data, the less smooth the curve. Trying a better fit for prices always result in fluctuating forward or spot rates. An ideal model should keep balance between these to factors(Bolder and Gusba, 2002).

Spline-base models are adaptations of the spline methods used in statistics to the yield curve concepts. In McCulloch (1971), McCulloch proposed a spline based approach to yield curve modeling. He fitted a linear combination of basis functions to the discount factor. As a result of the linear relationship, the parameters are simply estimated by linear regression. As basis functions he used piecewise quadratic functions. However it results in "knuckles" in the forward rate curve. To overcome such a problem, he used cubic basis functions in McCulloch (1975).

After McCulloch's pioneering works, other spline based models has been developed by Vasicek and Fong (1981), Shea (1984). In general spline based models results a better goodness of fit, however they lack smoothness. In most instances, the implying spot rate and forward rate curves are not stable (Fisher et al., 1994). To overcome such adversities, Fisher et al. (1994) developed a technique which adds a penalty for "excess roughness". They tried to fit their model to discount factor, spot rate, forward rate and capitalization function. Experiments shows the model fitted to forward rates results best. However Bliss (1997b) found out the model performs poorly. Details of the spline-based models can be found in the Appendix 4.1.

## *1.3 The Estimation Procedure*

In this study I will be comparing eight different optimization methods in the context of an identical term structure estimation problem. I will be using the same starting values (where appropriate), the same objective function, the same constraints, and the same metrics for evaluating the results. Before introducing the eight optimization methods in the next section I will cover the common elements just enumerated of the estimation problem.

### 1.3.1 Bond math

The term structure of interest rates, also referred to as the spot rate or the zero coupon yield term structure, is defined as a function, $r(m)$, mapping the interest rate used a discount any single cash flow from time, $m$, of that cash flow back to the present, giving the present value of that cash flow under the current term structure of interest rates. The theoretical price for a bond, $P$, is then the sum of the individual cash flows, $c(t)$, each discounted at the interest rate, $r(m)$, applicable to the time, $m$, when the cash flow occurs:

$$P = \sum_{t=1}^{T} c(m)e^{-r(m)m}. \tag{1}$$

Estimating the term structure of interest rates reverses the process. Observing a series of bond prices I wish to infer the underlying values of the term structure of interest rates that gives rise to those prices. To do this the functional form $r(m, \Theta)$ is used that the term structure conforms to. Then, using a nonlinear optimization process, arrive at the parameter values for the term structure that best fits the observed

data. If I know the observed bond prices on a given observation date, $P_i$, and the fitted bond prices, $\hat{P}_i(\Theta)$ conditional on particular set parameter values, $\Theta$, are given by

$$\hat{P}_i(\Theta) = \sum_{m=1}^{M_i} c_i(m) e^{-r(m,\Theta)m}, i = 1, ..., N. \tag{2}$$

then the term structure estimation problem becomes one of minimizing some function of the aggregate fitted price errors, $P_i - \hat{P}_i(\Theta)$ :

$$\min_{\Theta} \sum_{i=1}^{N} f(P_i - \hat{P}_i(\Theta)). \tag{3}$$

A number of functions have been used for aggregating fitted price errors including weighted and unweighted squared and absolute values, and several different weighting approaches. Given the choices of function to be used in the fitting and weighting scheme, the term structure estimation problem then becomes a non-linear optimization problem.

### 1.3.2 Nelson-Siegel and Svensson functions

Term structure estimation models can be classified into two groups: (i) spline-based models and (ii) parametric functional form-based models. The choice between them will be dictated by the trade-off between the goodness-of-fit to the set of observed government coupon prices and the smoothness of the corresponding zero-coupon rate function. A number of functional forms have been used in the past. [6] These include piecewise linear splines, cubic splines, and polynomials. [7] However, the form first proposed by Nelson and Siegel (1987) and extended by Svensson (1994) has been widely adopted by practitioners because of its parsimonious and intuitively appealing

[6] For prior comparisons of yield curve estimation techniques using alternative specifications of the functions to be fitted and the methods used, see Bliss (1997b) and Ioannides (2003)

[7] More Details can be found in Appendix 4.1

structure.

Not only do the NS class of models provide plausible estimates of the yield curve, they also provide estimates for the parameters which can be interpreted as corresponding to the level, slope, and curvature of the spot rate curve. This breakdown of the term structure and the estimated parameters are invaluable for practitioners hedging the risk of complex bond portfolios such as butterfly and ladder portfolios.[8]

The original Nelson and Siegel (1987) model, in its spot rate form, is given by:

$$r(m,\Theta) = \beta_0 + \beta_1 \frac{1-e^{(-m/\tau)}}{m/\tau} + \beta_2 \left( \frac{1-e^{(-m/\tau)}}{m/\tau} - e^{(-m/\tau)} \right). \tag{4}$$

For ease of discussion, I will rewrite this in terms of its component functions as follows:

$$r(m,\Theta) = \beta_0 + \beta_1 g(m|\tau) + \beta_2 h(m|\tau) \tag{5}$$

$$g(m|\tau) = \left( \frac{1-e^{-m/\tau}}{m/\tau} \right) \tag{6}$$

$$h(m|\tau) = \left( \frac{1-e^{-m/\tau}}{m/\tau} - e^{-m/\tau} \right) \tag{7}$$

Both $g(m|\tau)$ and $h(m|\tau)$ are defined for $m \geq 0$ and $\tau \geq 0$. As $m$ and $\tau$ always appear as a ratio, $\frac{m}{\tau}$ can be thought of as a time-scaling parameter. The function $g(m|\tau)$ has a fixed range $(0, 1]$ and declines monotonically as time horizon increases. It is associated with the overall "slope" of the term structure. The parameter $\beta_1$ controls both the size of the rise or fall of interest rates between the short ($m = 0$) and long ($m \to \infty$) ends of the term structure and whether the term structure is rising

[8]NSS based fixed income relative value models are increasingly popular in the investment community because they are more suitable for cheap/rich analysis since this class of models focuses on the actual cash flows of the underlying securities rather than using the yield-to-maturity measure, which is subject to a number of shortcomings. (Martellini et al. (2003))

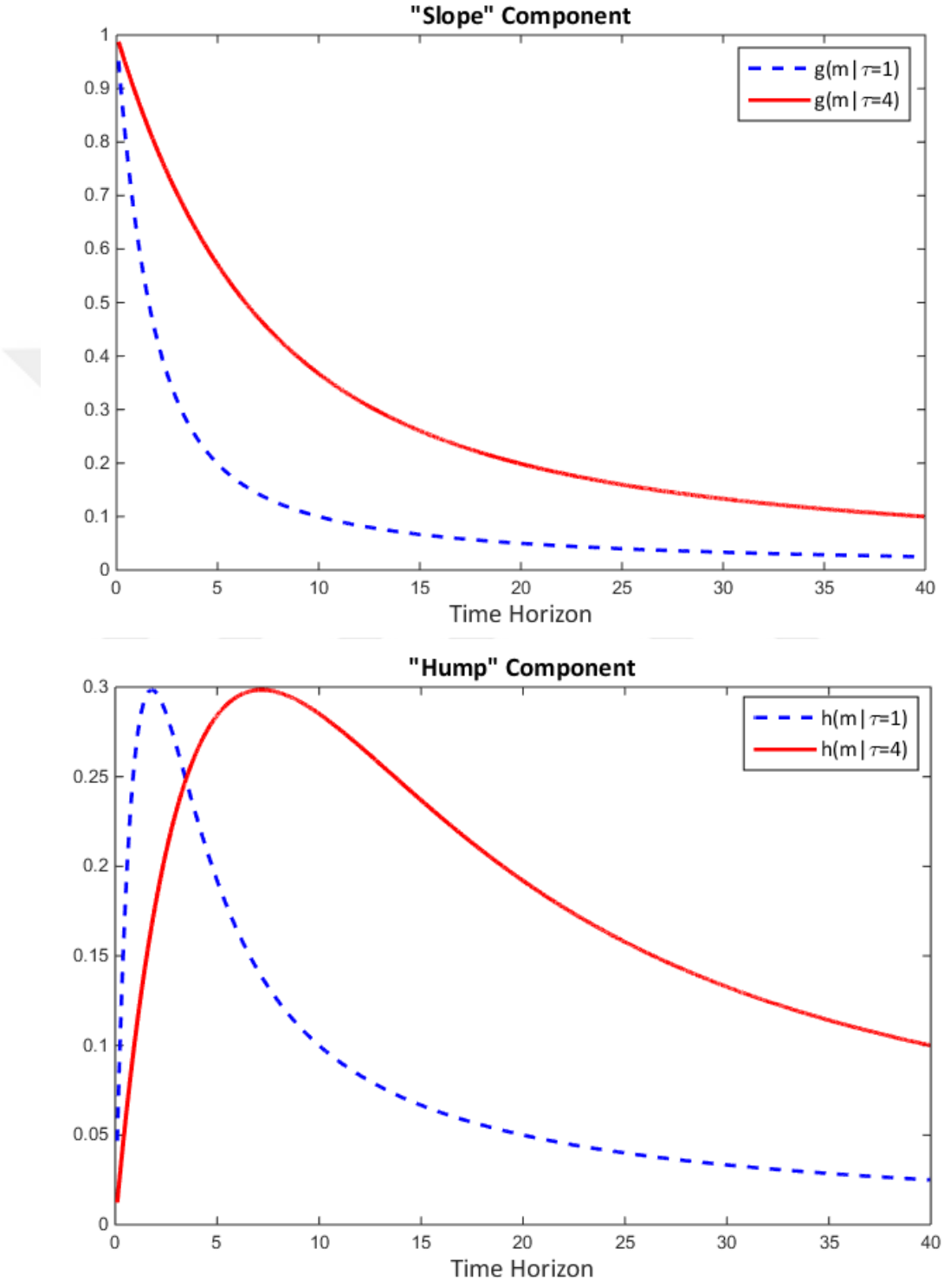


**Figure 1:** Slope and hump component functions of the Nelson-Siegel function, excepting the constant term.

($\beta_1 > 0$) or falling ($\beta_1 < 0$). The function $h(m|\tau)$ also has a fixed range $[0, \approx 0.298]$. The point where the maximum occurs varies with $\tau$ in the time dimension, but occurs at a constant value of the ratio $\frac{m}{\tau}$ ($\approx 1.793$). The function $h(m|\tau)$ rises from zero and then falls exponentially back down to zero. For this reason, it fits humps or dips in the term structure. The parameter $\beta_2$ scales the range of $h(m|\tau)$ and determines whether it contributes a concave "hump" to the term structure ($\beta_2 > 0$) or a convex "dip" ($\beta_2 < 0$).

The Nelson Siegel function has several attractive properties. It is parsimonious while still being flexible. The flexibility is greatest at the short end of the term structure where I observe the greatest empirical variation in interest rates. Most importantly it is able to capture, through its "hump" component, the occasional non-monotonicity of the term structure. It is also asymptotically flat. This means that forward rates implied by the term structure do not become unbounded.[9]

The level, slope, and hump (curvature) breakdown of the Nelson-Siegel functional form coincides with an observed empirical regularity of changes in the term structure of interest rates. Litterman and Scheinkman (1991, 1994), using a factor decomposition of movements in U.S. money market and U.S. Treasury zero coupon term structures, observed that the factors were characterized by 1) approximately equal shifts in interest rates across all maturities, this became known as the "level" factor; 2) a "twist" with short rates moving in one direction and long rates in the other, this became known as the "slope" factor; and 3) a factor that had little impact on very short or very long maturities, but did effect intermediate values, this became known as the "curvature" factor. Numerous studies have confirmed the persistence of

[9]This condition and imposed on spline-based methods through appropriate constraints, but occurs naturally with the Nelson-Siegel functional form. It cannot be achieved using polynomial forms.

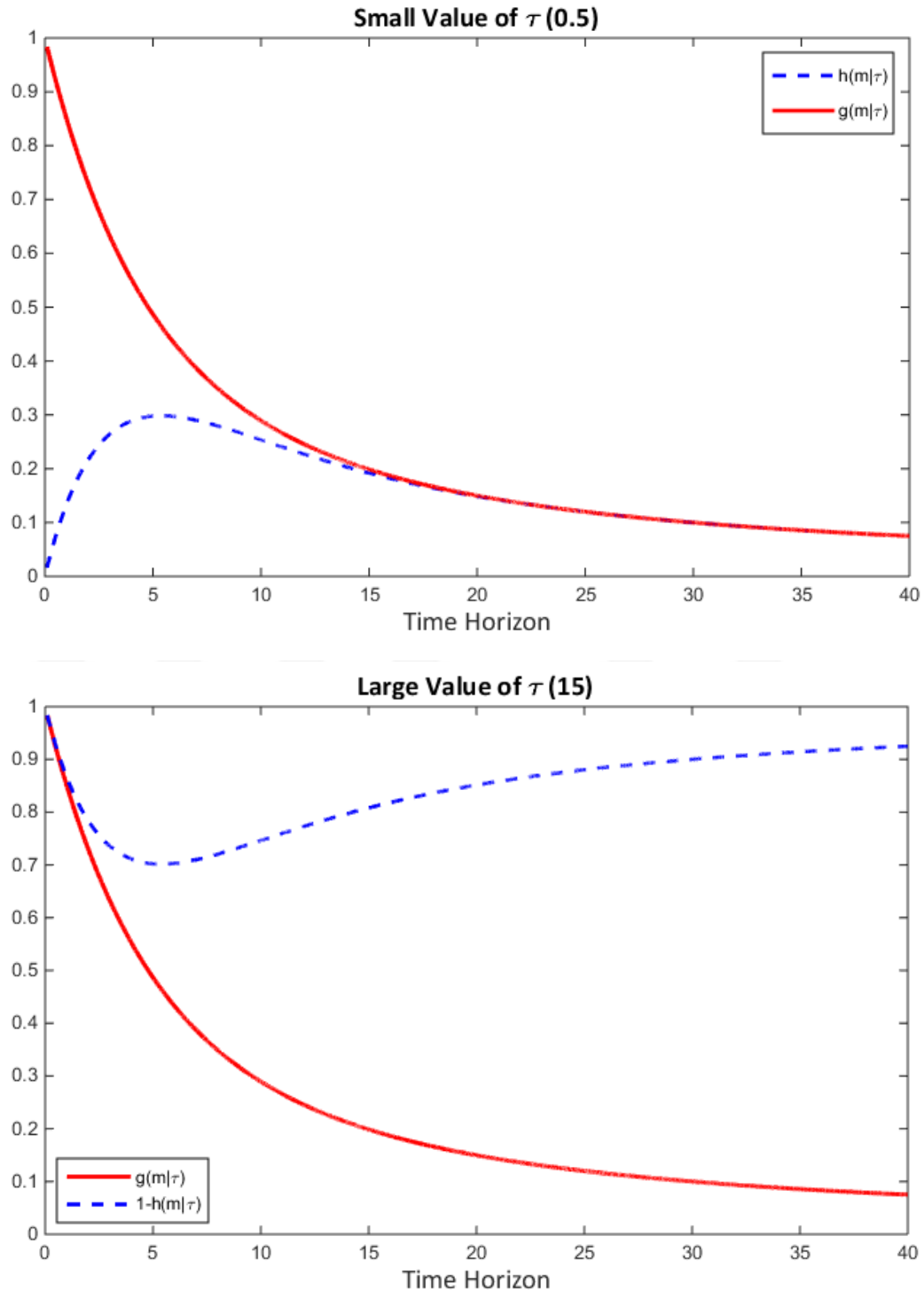


**Figure 2:** The effect of small and large values of $\tau$ on the two non-linear components of the NS function.

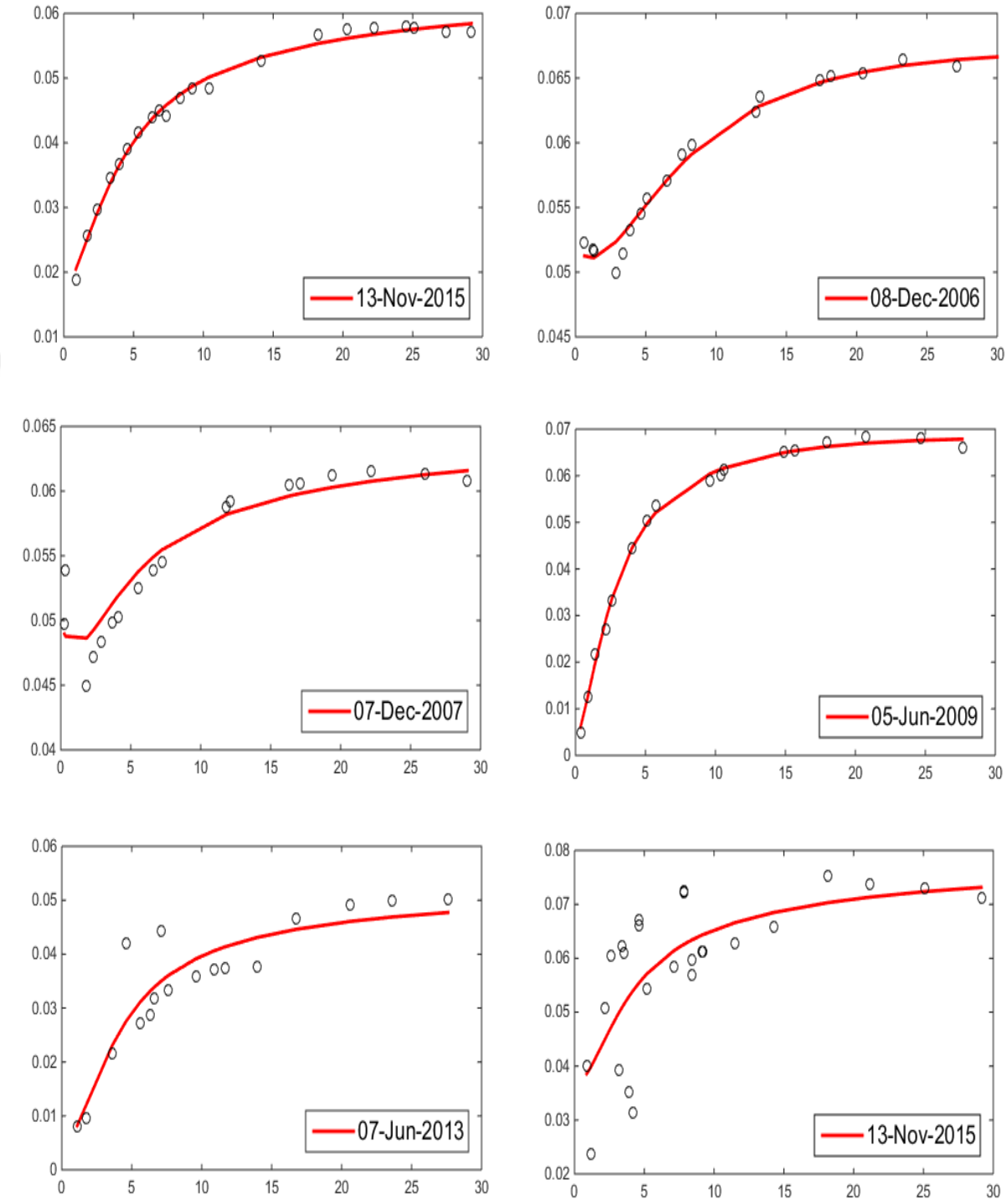


**Figure 3:** Selected fitted yield curves fitted to Brazilian eurobond data using the hybrid PSO optimization method. Circles indicate spot rates obtained using a simple bootstrap applied to the same data.

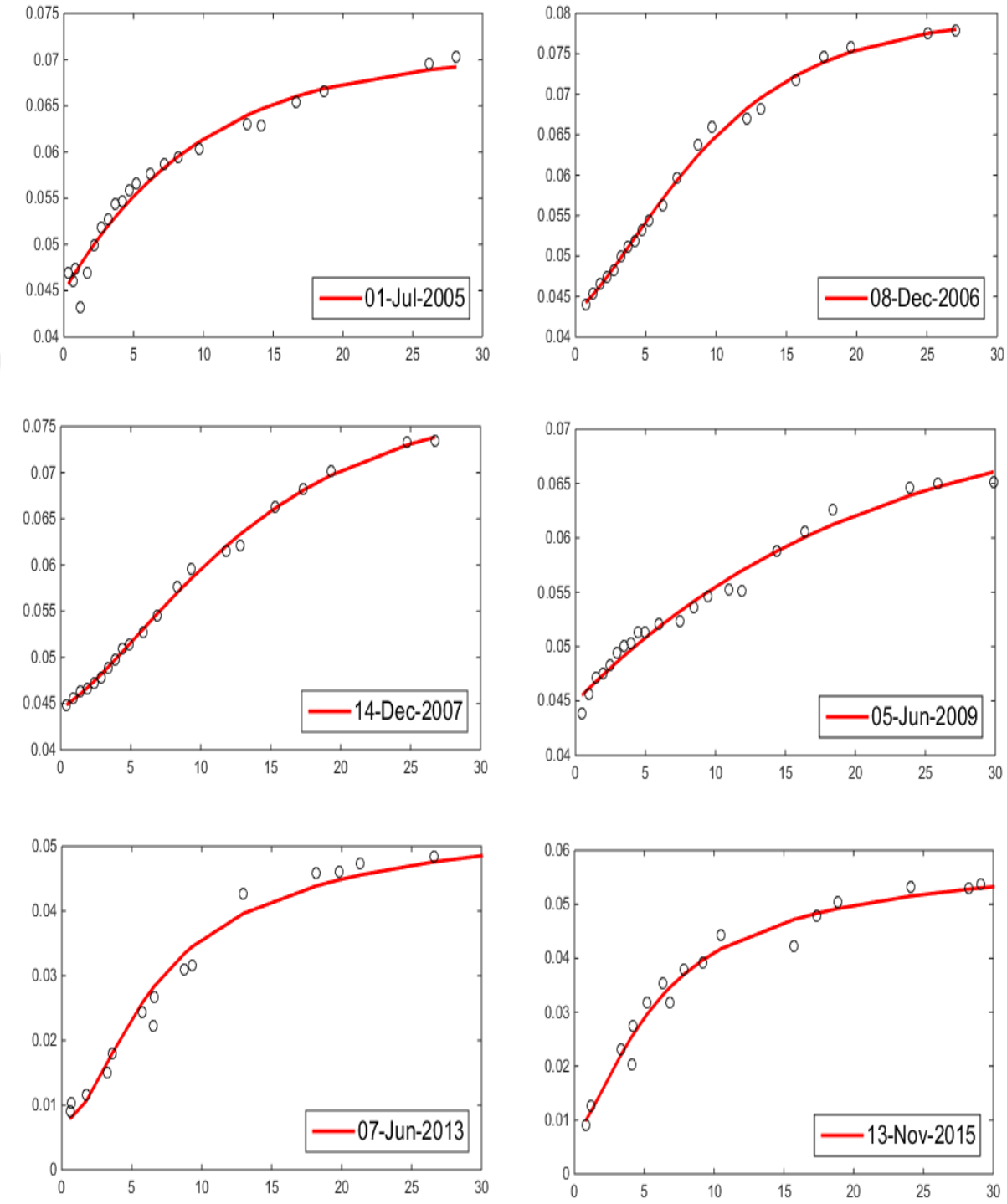


**Figure 4:** Selected fitted yield curves fitted to Mexican eurobond data using the hybrid PSO optimization method. Circles indicate spot rates obtained using a simple bootstrap applied to the same data.

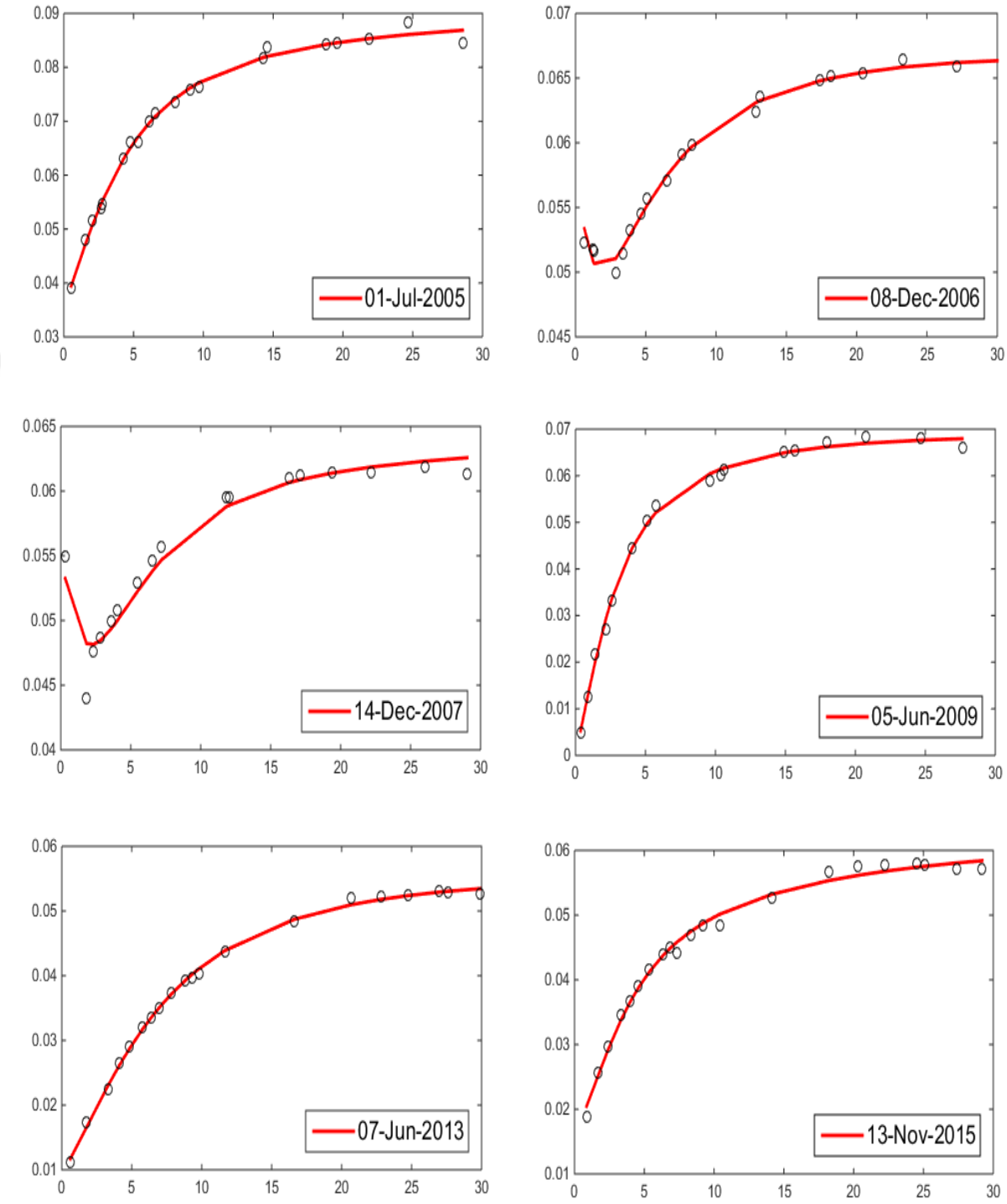


**Figure 5:** Selected fitted yield curves fitted to Turkish eurobond data using the hybrid PSO optimization method. Circles indicate spot rates obtained using a simple bootstrap applied to the same data.

this factor structure, both through time and across currencies.[10] Other authors have linked the factors to macro-economic variables or characteristics of interest rates, such as volatility.[11] Krippner (2012) has shown that the NS functional form can arise from the popular affine class of theoretical term structure models.

While intuitively appealing and consistent with both the observed shapes of the term structure and its changes, the NS functional form has not always been easy to fit to the data. A number of studies have reported numerical difficulties when working with the NSS model.[12] This is not surprising. As Figure 2 shows, when $\tau$ is very small the $g(m|\tau)$ and $h(m|\tau)$ functions are nearly identical over most of their span. This makes it difficult for certain optimization methods to resolve $\beta_1$ and $\beta_2$.[13] This problem increases as the smaller $\tau$ becomes. When $\tau$ becomes large then $g(m|\tau)$ is very close to the $1 - h(m|\tau)$ at shorter maturities where most of the cash flows occur. When the term structure is monotonically increasing over the range of available data, it is difficult for optimization routines to decide whether to set $\beta_2$ to zero or to use extreme values of $\tau$ to improve fit if the monotonic curvature is not quite exponential. If $\tau_1 \approx \tau_2$, then only $\beta_2 + \beta_3$ is meaningful and their individual values cannot be determined.

When estimating zero-coupon spot rates within the range of underlying bond maturities, the estimates obtained for zero-coupon spot rates are usually reasonable, though not necessarily accurate. I show the numerical problems can translate into

[10] Some authors, such as Lekkos (2001), have shown that this factor structure can be replicated with random forward rates and a fairly simple correlation structure. Be that as it may, the level–slope–curvature view of the term structure has now become standard.

[11] See, for example, Afonso and Martins (2012); Diebold et al. (2005a,b).

[12] See, for example, Bolder and Streliski (1999), Gimeno and Nave (2006), Gurkaynak et al. (2007), De Pooter (2007), and Annaert et al. (2010).

[13] Gilli et al. (2010) also commented on this potential degeneracy. They argue, I believe correctly, that the NSS function is ill-conditioned; a point I shall return to later.

unstable, that is noisy, estimated values for the zero-coupon rates depending on the optimization algorithm employed. Noisy estimates of values later used as inputs for other purposes necessarily degrades the reliability of subsequent analysis.

The traditional optimization methods such as direct search, gradient-based, and quasi-Newton algorithms, which may be used for solving the associated optimization problem, carry the risk of numerical problems of false convergence and severe sub-optimality (see Bolder and Streliski, 1999). Nelson and Siegel (1987), Diebold and Li (2006), Annaert et al. (2010) finesse these problems by linearizing the NS functional form.[14] They do this by simply fixing $\tau$ at a reasonable level before estimating the $\beta$ coefficients by ordinary least squares. This solves the numerical problems, but at the expense of an *ad hoc* procedure for selecting $\tau$, which does not lead to a globally optimum set of parameter values. The alternative of a full grid search is not feasible. The time required to refine the solution and increase the confidence of having an accurate global optimum increases exponentially with the number of different starting values chosen for each parameter. For example, estimating the six parameters of the NSS model, with five different starting values for each parameter, requires a grid size of $5^6 = 15,625$ for the first pass alone.[15]

One problem that cannot be resolved by judicious choice of empirical methods, arises in extrapolating beyond the range of available bond maturities. This is, of course a general problem in estimation, but because the parameters of the NS and NSS lend themselves to economic interpretations, the danger may be particularly acute.

[14] Nelson and Siegel (1987), Diebold and Li (2006), and Gilli et al. (2010) fit the NS or NSS functions to zero coupon rates instead of coupon bond prices. This is a numerically easier problem. Nelson and Siegel use Treasury Bill yields; Diebold and Li and Gilli et al. use the Unsmoothed Fama-Bliss term structures, which are available from Robert Bliss.

[15] Annaert et al. (2012) linearize the NSS function by performing a 2-dimensional grid search over values for $\tau_1$ and $\tau_2$.

Fitting a monotonically increasing term structure with NS or NSS can produce a fitted term structure that either asymptotes or peaks well beyond the range of maturities in the sample and at levels not observed in the data. For this reason, one has to be careful interpreting fitted NS functions when they are extrapolated beyond the range of maturities from which they are derived.

The Nelson-Siegel-Svensson (NSS) variant adds a second $h(m|\tau)$ with an additional scaling parameter and a distinct $\tau$ resulting in a function with six parameters, $\Theta \in \{\beta_0, \beta_1, \beta_2, \beta_3, \tau_1, \tau_2\}$:

$$r(m|\Theta) = \beta_0 + \beta_1 g(m|\tau_1) + \beta_2 h(m|\tau_1) + \beta_3 h(m|\tau_2). \tag{8}$$

This addition is *ad hoc* and does not correspond to observed factor decompositions. It does however, add an additional degree of flexibility and allows for two inflection points along the term structure so that it may rise, fall and then rise again before approaching its long term level.

In summary, both the NS and NSS functions have attractive properties in terms of shape flexibility, congruence with observed behavior of interest rate changes (NS only), and being norms in industry and academia. Their downside lies in the numerical problems its empirical implementations are prone to.

### 1.3.3 The estimation framework

In this study I minimize the sum of inverse-duration-weighted squared fitted-price errors.

$$\min_{\Theta} J(\Theta) = \sum_{i=1}^{N} w_i (P_i - \hat{P}_i(\Theta))^2 \tag{9}$$

where $w_i = \frac{\frac{1}{D_i}}{\sum_{i=1}^{N} \frac{1}{D_i}}$, $D_i$ is the $i^{th}$ bond's Macaulay duration computed at its yield-to-maturity, and $\hat{P}_i(\Theta)$ is defined by Eq. (2) and (8). Inverse duration weighting has the effect of down-weighting fitted price errors for long maturity bonds where large fitted price errors are more frequently observed. Such large errors can become influential and distort the estimated term structure function at short and intermediate maturities, resulting in systematic mispricing of short and intermediate maturity bonds.[16]

Bliss (1997b) reports tests of several weighting schemes, using out-of-sample fit. Empirically, both inverse duration and inverse maturity weighting are superior to equal and spread weighting. However, inverse duration weighting is theoretically more correct as price sensitivity to interest rates changes is related to duration, not maturity. Bolder and Streliski (1999) have also employed inverse duration weighting.

#### *1.3.3.1 Constraints*

The NS and NSS functions give rise to several constraints on the parameters that must be imposed if the resulting fitted term structure is to make economic sense. The term structure has to be non-negative throughout. This means that the short ($m = 0$) and long ($m \to \infty$) rates must also be non-negative. As $g(0|\tau) = 1$, $h(0|\tau) = 0$ and $g(\infty|\tau) = h(\infty|\tau) = 0$, the short rate $r(0, \Theta) = \beta_0 + \beta_1$, the long rate $r(\infty, \Theta) = \beta_0$. Therefore, my first two constraints are:

$$c_1(\beta_0) = \beta_0 \geq 0 \tag{10}$$

$$c_2(\beta_0, \beta_1) = (\beta_0 + \beta_1) \geq 0. \tag{11}$$

[16] By systematic mispricing I mean long runs of over- and under-pricing for adjacent maturities by the fitted term structure.

The $\tau_1$ and $\tau_2$ cannot be negative as this would completely change the behavior of the component functions. Also, $\tau_1$ and $\tau_2$ have to be bounded away from zero else $\frac{m}{\tau}$ would be undefined. Therefore, my second pair of constrains are:

$$c_3(\tau_1) = \tau_1 > 0 \tag{12}$$

$$c_4(\tau_2) = \tau_2 > 0 \tag{13}$$

These constraints apply to both the NS and NSS functional forms.

#### *1.3.3.2 Starting values*

Identical starting values were used for the six optimization methods that begin with a single starting point. To obtain these starting values, I adapt the procedure proposed in Diebold and Li (2006). As Diebold and Li (2006) were working with the Unsmoothed Fama-Bliss yields, rather than coupon bond prices, I first apply a simple bootstrap to the data to generate similar set of zero coupon yields using the zbtprice function in the MATLAB Financial Toolkit. Then following Yallup (2012), I calculate $\tau_1$ as the maturity that maximizes the loading on the curvature factor which is the numerical second derivative of the yield curve to estimate initial values of $\beta_0, \beta_1$, and $\beta_2$ with ordinary least squares. Since I are deriving these four parameter values from an approximation of the yield curve, I expect the starting values to be close to the final solution, an unusual advantage when conducting non-linear optimizations.[17] Finally, to avoid collinearity I set $\tau_2 = \tau_1 + 1$ and $\beta_3 = \beta_2 + 1$ . I denote this vector of starting values, derived from Diebold and Li (2006), as $\Theta^{DL}$. [18]

[17]The MATLAB bootstrap routine lacks the internal filter of the Fama-Bliss method (Fama and Bliss (1987)), and neither produce yield curves that are smooth or lend themselves to economic interpretation, as does the NS model.

[18]As starting point may be biased against gradient methods, alternatively I search for different $\tau_2$ values with $\tau_2 = \tau_1 + \gamma_1$ where $\gamma_1$ set from 1 to 10. I find that the results hold robust with respect to choice of $\gamma_1$.

### 1.3.4 Performance metrics

I analyze the performance of the different optimization algorithms in four dimensions: goodness-of-fit, computational time, distance traveled, and stability of parameters. My first performance measure is goodness-of-fit, for which I use the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). Clearly, the method that generates smaller fitted-price errors is to be preferred. Secondly, I calculate the CPU time taken by each of the algorithms. An algorithm that yields smaller errors with less computational time would be preferable over one that takes greater time to achieve comparable results. Thirdly, I approximate the amount of parameter space scanned. Algorithms that are able to scan more space will have an obvious advantage searching for the global optimum. Finally, I present graphical illustrations of how the NSS parameters evolution over the time, to observe their stability, my fourth measure.

#### *1.3.4.1 Goodness-of-fit*

The performance statistics MAE, and RMSE can be calculated as: MAE $= \frac{1}{N}\sum_{i=1}^{N}|P_i - \hat{P}_i|$, RMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \hat{P}_i)^2}$.

where $N$ represents the number of bonds. RMSE places a greater weight upon larger errors and, therefore, gives a better indication as to how well the models fit the data at each particular observation. A low value for the RMSE indicates that the model is flexible and, on the average, is able to fit the yield curve fairly accurately. MAE is the average distance between the theoretical bond prices and observed bond prices in absolute value terms. This measure is not as easily influenced by extreme observations as RMSE. Therefore, these two measures are complementary.

#### *1.3.4.2 Computation time and distance traveled*

For day-to-day estimation of NSS parameters in an applied setting, computation time could be important as large numbers of term structures repeatedly need to be generated. Total CPU time taken to estimate term structures for all observation periods is recorded for each optimization algorithm.

Average distance traveled from the starting parameter values to the final solution provides an approximation of the size of the space scanned.[19] Distance traveled needs to be evaluated in conjunction with goodness-of-fit. A large distance traveled that is associated with a poor goodness-of-fit may indicate that the optimization method is jumping too far from the starting values and then getting hung up on a local minimum. Similarly, a small distance traveled combined with a poor goodness-of-fit might indicate an optimization algorithm that converges to readily to the first local minimum it finds. For each observation period, the average distance traveled from the starting values to the final solution is calculated as $||\Theta_{min} - \Theta_0||$ where $\Theta_{min}$ is the solution obtained in the optimization process.

#### *1.3.4.3 Inter-temporal stability of parameters*

Past researchers employing the NSS model have mainly focused on the in-sample goodness-of-fit fit and have paid little attention to parameter stability. I examine the stability of the estimated term structures in three ways. The $\beta_0$, $\beta_1$, and $\beta_2$ parameters correspond to components of the shape of the term structure. As these are estimated at weekly intervals, it is reasonable to expect that these will not show great variation from week to week. Consider for instance, the first parameter of NSS model,

[19]For the optimization methods which begin with multiple starting values, the distance traveled was measured from the mean of the multiple starting values, computed parameter by parameter, to the final solution.

$\beta_0$. This is interpreted as the long-run level of interest. From one day to the next, jumps of several percentage points for the estimates of this parameter would suggest that the results are economically implausible, even if the goodness-of-fit is good. To this end, I plot these parameters as time series, together with their "empirical proxies" computed according to the study of Diebold and Li (2006) and compare them across optimization methods.[20] The literature does not suggest empirical proxies for $\beta_3$, and $\tau_2$.

#### *1.3.4.4 Robustness tests*

I test the sensitivity of the optimization methods to the starting values and to potential measurement error in the. I first randomly perturb the starting values and reestimate the fitted parameters. This process is repeated 100 times. For each iteration, I then compute the three-month, two-year and 10-year estimated spot rates. I then plot their distributions and compare across optimization methods.

Returning to the original fixed starting values, I perturbed the observed prices with in the quoted bid-ask spreads using uniformly distributed random numbers and re-estimate, again generating distributions of three-month, two-year and 10-year estimated spot rates for each optimization method.

The in-sample goodness-of-fit statistics were generated using all available data for each observation. For my robustness tests, I generated out-of-sample goodness-of-fit statistics by taking the available bonds in an observation period, ordering them by maturity, and then alternately assigning them to estimation- and test-subsamples.

[20] As described in Diebold and Li (2006), the empirical proxy for $\beta_0$ (level) is defined as the 10-year yield, the empirical proxy for $\beta_1$ (slope) as the difference between the 10-year and 3-month yields, and the empirical proxy for $\beta_2$ (curvature) as the twice the 2-year yield minus the sum of the 3-month and 10-year yields.

The optimization algorithms were then applied to the estimation-subsample and the estimated parameters were used to obtain the goodness-of-fit for bonds in the test-subsample. The two subsamples were then reversed in the process repeated.

For the price perturbation and starting value perturbation tests the random number generator was reset before each optimization method was applied to ensure that the results were comparable. The out-of-sample tests were structured so that all optimization methods were applied to identical subsamples.

## *1.4 Optimization Algorithms*

To comparatively evaluate the performance of the PSO algorithm, I implemented several well-known numerical optimization algorithms from the literature. My focus in this paper is not only which optimization algorithm 'wins the horse race,' but also how one could solve the numerical problems reported in the literature. The selected algorithms are classified into three categories: global optimization algorithms, direct search algorithms and gradient based algorithms.[21] I can also distinguish optimization methods on two other dimensions: those that follow a single point from starting value to solution and those that begin with a population of many starting values and then converge these to a single solution point (denoted "single point" or "multiple point" methods); and those that do not or do employ random numbers during the search (denoted "deterministic" or "randomized" methods). The multi-point methods begin by randomly selecting a population of starting points, but then can proceed either deterministically or employing random numbers during the process.

The gradient-based class of algorithms require computation of the gradient of

[21]The most common approach by far is to use gradient-based methods. Authors using solely gradient-based methods include Csajbók (1999) and Gurkaynak et al. (2007) who used Gauss-Newton; and Bliss (1997b) and Ioannides (2003) use Broyden-Fletcher-Goldfarb-Shanno (BFGS).

the objective function with respect to each parameter at each step in the search. Some gradient-based methods require computation or approximation of the Hessian as well. I use two popular and efficient gradient based optimization algorithms, namely Broyden-Fletcher-Goldfarb-Shanno Algorithm (BFGS) and the Gauss-Newton. All gradient-based methods are deterministic, single point algorithms.

Our direct search methods include Powell's Method which is a single point, deterministic algorithm, and the Nelder-Mead Method which begins with a population of randomly selected starting points, and then proceeds deterministically to a single solution. Global optimization algorithms include Simulated Annealing which follows a single point with random perturbations along the way to a solution, and the PSO, which begins with a population (swarm) of randomly selected starting points and employs random values in determining how the population converges toward a common solution.[22]

Finally, all of these methods I investigate, except PSO, only use information from the current position value(s). That is they have no "memory" of where they have been, how previous points in the search have performed, and no means to "go back" if they advance to a worse solution from a better one, unless their algorithm happens to do so based on information at the new, inferior, point. The PSO algorithm on the other hand retains "best solutions so far" for individual particles as they evolve and the global best across all particles as they converge to the solution.

In the following subsections, I describe in greater detail than is usual the optimization methods I are testing. I do this in order to lay the groundwork for the discussion of why I believe these methods perform as they do, and to provide transparency as to how results were obtained. [23]

[22]Gimeno and Nave (2006) use a genetic algorithm to fit the NSS function to Spanish government bond prices. They then compare this method with what they call the "traditional" method.

[23]The code for the algorithms is available upon request from authors.

### 1.4.1 Global Optimization Algorithms

The objective of global optimization is to find the globally best solution of (possibly non-linear) models, in the presence of multiple local optima.

#### *1.4.1.1 Particle Swarm Optimization*

PSO is a population-based metaheuristic technique.[24] It is a fast converging algorithm, is easy to implement, and has been successfully applied to optimizing various continuous nonlinear functions in other applications. [25] PSO has not previously been applied to term structure estimation. PSO is essentially inspired from the social behaviour of the individuals. In a simple social setting, decision process of each individual is affected by his own experiences and other individual's experiences. Every individual keeps a memory of their best choice, as well as the overall best choice of the population. In my PSO setting, for each observation day, dataset a set of particles search for good solutions to the NSS term structure fitting optimization problem as described in Sections 1.3.2 and 1.3.3. Each particle is a solution of the NSS optimization problem and uses its own experience and also the experience of neighbour particles to choose how to move in the search space to find a better fitting term structure.

The PSO algorithm is initialized with a population of random candidate solutions, called *particles*. Each particle is assigned a random location and a random velocity, and is iteratively moved through the problem space. Every particle is attracted towards the location of the best solution achieved by the particle itself and towards the location of the best solution achieved across the whole population. At each iteration,

[24] A metaheuristic is a method that make few assumptions about the structure of the problem and combines heuristic techniques in an efficient way to solve general classes of problems. These techniques cannot guarantee convergence to an optimal solution, but then neither can more formal approaches when their underlying assumptions are violated.

[25] See Clerc and Kennedy (2002); Pedersen and A.J. (2010); Trelea (2003).

velocity vector ($v_i$) and position (parameter) vector ($\Theta_i$) of a particle is updated as follows:

$$v_k \leftarrow v_{k-1} + U(0, \phi_1) \times (pb_{k-1} - \Theta_{k-1}) + U(0, \phi_2) \times (gb - \Theta_{k-1}) \tag{14}$$

$$\Theta_k \leftarrow \Theta_{k-1} + v_{k-1} \tag{15}$$

where the $U(0, \phi_i)$ are scalars uniformly distributed in the interval $[0, \phi_i]$, $pb_i$ is the best known position of particle $i$, and $gb$ is the best known position across the entire population. The parameters $\phi_1$ and $\phi_2$ denote the magnitude of the random forces in the direction of personal best $pb_i$ and swarm best $gb$. The components $U(0, \phi_1) \otimes (pb_i - \Theta_i)$ and $U(0, \phi_2) \otimes (gb - \Theta_i)$ can be interpreted as attractive forces produced by springs of random stiffness.

PSO is an unconstrained optimization algorithm, which required us to handle the constraints in the objective function. To this end I added a penalty function that is a scalar value times the square of the violation of constraint $c_i$ to the objective i.e

$$D_i(\Theta) = \begin{cases} 0 & , c_k(\Theta) \leq 0, \\ C(c_k(\Theta))^2, & c_k(\Theta) > 0. \end{cases} \tag{16}$$

where $C$ is large scalar value. Thus, the optimization problem becomes:

$$\min_{\Theta} f(\Theta) = J(\Theta) + \sum_{k=1}^{4} D_k(\Theta) \tag{17}$$

There are several variants of the original PSO algorithm. Shi and Eberhart (1998) developed a variant which I call PSO-W. They modify the updating of the velocity vector to place decreasing weight on the last value, and thereby increasing weight on

the individual and global bests as the iteration count increases:

$$v_k \leftarrow \omega v_{k-1} + U(0, \phi_1) \times (pb_{k-1} - \Theta_{k-1}) + U(0, \phi_2) \times (gb - \Theta_{k-1}) \quad (18)$$

$$\Theta_k \leftarrow \Theta_{k-1} + v_k \quad (19)$$

$$\omega \leftarrow \omega_p \times \omega \quad (20)$$

where $\omega$ is termed the inertia weight, and $0 < \omega_p < 1$ is the decay factor. Effectively, the inertia weight preserves the initial random search direction, the magnitude of which decreases over iterations.

## PSO-W Algorithm

*Initialization.* Choose parameters $\omega_p$, $\phi_1$, and $\phi_2$, and stopping criteria $k_{max}$ and $\epsilon$. Initialize $\omega$, particle population $i = 1, ..., i^{max}$, $\Theta_0^{(i)} \sim U(\Theta^{Min}, \Theta^{Max})$ and corresponding velocity vectors $v_0^{(i)} \sim U(\Theta^{Min}, \Theta^{Max})$.[a] Set $pb_i = \Theta_0^{(i)}$ and $gb = pb_k$, where $k = \text{argmin}_i f(pb_i)$.

*Step 1.* Update $\Theta_k^{(i)}$ and $v_k^{(i)}$:

$$v_k^{(i)} \leftarrow \omega v_{k-1}^{(i)} + (U(0, \phi_1) \times (pb_i - \Theta_i)) + U(0, \phi_2) \times (gb - \Theta_i)$$

$$\Theta_k^{(i)} \leftarrow \Theta_{k-1}^{(i)} + v_k^{(i)}$$

$$\omega \leftarrow \omega_p \times \omega$$

*Step 2.* For all i, if $f(\Theta_k^{(i)}) < f(pb_i)$, then $pb_i \leftarrow \Theta_k^{(i)}$. If $f(pb_i) < f(gb)$, then $gb \leftarrow pb_i$

*Step 3.* If $k > k_{max}$ or $||\Theta_k^{(i)} - \Theta_{k+1}^{(i)}|| < \epsilon(1 + ||\Theta_k^{(i)}||)$, then RETURN $gb$ and $f(gb)$, and STOP; else increment $k$ and go to step 1.

[a] The elements of $\Theta$ correspond to the parameters $[\beta_0, \beta_1, \beta_2, \beta_3, \tau_1, \tau_2]'$ of the NSS function. The upper and lower bounds for the uniformly drawn initial values are $\Theta^{Min} = [0, -1, -1, -1, 0, 0]'$ and $\Theta^{Max} = [1, 1, 1, 1, 30, 30]'$

Another variant of PSO was proposed by Pedersen and A.J. (2010). This variant, named as PSO-G, disregards the personal best values and focuses on the neighborhood of the population best.

Our computational experiments show that the best results were obtained when PSO-W and PSO-G were applied sequentially. This method will be referred to as

Hybrid PSO for the rest of the study.

## PSO-G Algorithm

*Initialization.* Choose parameters $\omega_p$, $\phi_2$, and stopping criteria $k_{max}$ and $\epsilon$. Initialize $\omega$, particle population $i = 1, ..., i^{max}$, $\Theta_0^{(i)} \sim U(\Theta^{Min}, \Theta^{Max})$ and corresponding velocity vectors $v_0^{(i)} \sim U(\Theta^{Min}, \Theta^{Max})$. Set $pb_i = \Theta_0^{(i)}$ and $gb = pb_k$, where $k = \text{argmin}_i f(pb_i)$.

*Step 1.* Update $\Theta_k^{(i)}$ and $v_k^{(i)}$:

$$v_k^{(i)} \leftarrow \omega v_{k-1}^{(i)} + U(0, \phi_2) \times (gb - \Theta_i)$$
$$\Theta_k^{(i)} \leftarrow \Theta_{k-1}^{(i)} + v_k^{(i)}$$
$$\omega \leftarrow \omega_p \times \omega$$

*Step 2.* For all i, if $f(\Theta_k^{(i)}) < f(pb_i)$, then $pb_i \leftarrow \Theta_k^{(i)}$. If $f(pb_i) < f(gb)$, then $gb \leftarrow pb_i$.

*Step 3.* If $k > k_{max}$ or $||\Theta_k^{(i)} - \Theta_{k+1}^{(i)}|| < \epsilon(1 + ||\Theta_k^{(i)}||)$, then RETURN $gb$ and $f(gb)$, and STOP; else increment $k$ and go to step 1.

## Hybrid PSO Algorithm

*Initialization.* Choose parameters $\omega_p$, $\phi_1$, $\phi_2$, and stopping criteria $k_{max}$ and $\epsilon$. Initialize $\omega$, particle population $i = 1, ..., i^{max}$, $\Theta_0^{(i)} \sim U(\Theta^{Min}, \Theta^{Max})$ and corresponding velocity vectors $v_0^{(i)} \sim U(\Theta^{Min}, \Theta^{Max})$.Set $pb_i = \Theta_0^{(i)}$ and $gb = pb_k$, where $k = \text{argmin}_i f(pb_i)$.

*Step 1.* Update $\Theta_k^{(i)}$ and $v_k^{(i)}$:

If $k \leq k_g$, then the algorithm is PSO-W,

$$v_k^{(i)} \leftarrow \omega v_{k-1}^{(i)} + U(0, \phi_1) \times (pb_i - \Theta_i) + U(0, \phi_2) \times (gb - \Theta_i)$$

If $k > k_g$, then the algorithm is PSO-G,

$$v_k^{(i)} \leftarrow \omega v_{k-1}^{(i)} + U(0, \phi_2) \times (gb - \Theta_i)$$

$$\Theta_k^{(i)} \leftarrow \Theta_{k-1}^{(i)} + v_k^{(i)}$$

$$\omega \leftarrow \omega_p \times \omega$$

*Step 2.* If $f(\Theta_k^{(i)}) < f(pb_i)$, then $pb_i \leftarrow \Theta_k^{(i)}$. If $f(pb_i) < f(gb)$, then $gb \leftarrow pb_i$.

*Step 3.* If $k > k_{max}$ or $||\Theta_k^{(i)} - \Theta_{k+1}^{(i)}|| < \epsilon(1 + ||\Theta_k^{(i)}||)$, then RETURN $gb$ and $f(gb)$, and STOP; else increment $k$ and go to step 1.

#### *1.4.1.2 Simulated Annealing*

Simulated Annealing (SA) algorithm, proposed by Kirkpatrick et al. (1983), is a probabilistic local search algorithm that looks for the minimum of an objective function using the neighborhood information of a point in the search space. The name and

inspiration of the algorithm come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends both on the difference between the corresponding function values and also on a global parameter $T$ (called the temperature), that is gradually decreased during the process.

Local search algorithms usually start with a random initial solution. A neighbour of this solution is then generated by some suitable mechanism and the change in cost is calculated. Neighbors are chosen as a network topology where the local optima are shallow or a similar topology where there are many deep local minima. If a reduction in cost is found, the current solution is replaced by the generated neighbour, otherwise the current solution is retained. The process is then repeated until no further improvement can be found in the neighbourhood of the current solution and the algorithm terminates at a local minimum. In SA, sometimes a neighbour that increases the cost is accepted, to avoid becoming trapped in a local optimum. A move that does not improve the solution is accepted with a probability that decreases with iterations. Usually, the probability is selected as $e^{-\delta/T}$ where $\delta$ is the increase in the objective function value at each iteration and $T$ is a control parameter.

**Simulated Annealing Algorithm** *Initialization.* Initialize $\Theta_0 = \Theta_0^{DL}$ and $T$. Set $k$=1 and $f_{best} = f(\Theta_0)$.

*Step 1.* Set $\Theta' = trim(\Theta_{k-1} + \bar{\phi}(0,1))$.[a]

*Step 2.* If $f(\Theta') < f_{best}$, then set $f_{best} = f(\Theta')$, $\Theta_{best} = \Theta'$, $\Theta_k = \Theta'$, $k = 1$, and go to step 5.

*Step 3.* Set $\sigma = f(\Theta') - f(\Theta_{k-1})$ and $p = U(0,1)$. If $p \leq e^{-\sigma/T}$, then set $\Theta_k = \Theta'$; else set $\Theta_k = \Theta_{k-1}$.

*Step 4.* If $k > k_{max}$, then RETURN $\Theta_{best}$ and $f_{best}$, and STOP; else increment $k$ and go to step 1.

[a]Where $\bar{\phi}(0,1)$ is a vector of standard normal random numbers and $trim()$ truncates the elements of resulting vector sum to lie in the range $(\Theta^{Min}, \Theta^{Max})$ where $\Theta^{Min} = [0, -1, -1, -1, 0, 0]'$ and $\Theta^{Max} = [1, 1, 1, 1, 30, 30]'$

### 1.4.2 Direct Search Algorithms

Direct search is a class of methods for solving optimization problems that does not require any information about the gradient of the objective function. Direct search algorithms search a set of points around the current point, looking for one where the value of the objective function is lower than the value at the current point.

#### *1.4.2.1 Nelder-Mead Method*

Nelder-Mead method is based on the idea of a "simplex", which refers to the generalized form of a triangle in $n \geq 2$ dimensions. The algorithm is initialized with $n+1$ solutions, which form a simplex in $R^n$. The midpoint of all initial solutions excluding the worst one, is computed first. The *reflection* of the worst solution with respect to this midpoint is then determined as a candidate solution. Based on the parameters of the algorithm, *expanded* and *contracted* points on the line connecting the worst point and the midpoint are also determined as candidate solutions. Finally, $n$ more candidates are generated by *shrinking* the simplex towards the best solution, and determining the new corners of the simplex. The best solution among all candidates then replaces the worst solution, and the algorithm is iterated until the worst solution cannot be improved.

The Nelder-Mead method (Lagarias et al. (1965); Nelder and Mead (1965)) uses four scalar parameters: $\rho$ (reflection), $\chi$ (expansion), $\gamma$ (contraction), $\sigma$ (shrinkage). Let the vertices are denoted as $\Theta^{(1)}, \Theta^{(2)}, ...\Theta^{(n+1)}$ where $n$ is the number of parameters to be estimated.

**Algorithm for Nelder-Mead's method**

*Initialization.* Determine vertices $\Theta^{(1)}, \Theta^{(2)}, ..., \Theta^{(n+1)}$ and parameters $\rho$, $\chi$, $\gamma$, $\sigma$ which satisfy $\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1, \quad 0 < \sigma < 1$, and $\epsilon \geq 0$. Set $k = 1$.

*Step 1.* Sort the vertices as $f(\Theta^{(1)}) < f(\Theta^{(2)}) < ... < f(\Theta^{(n+1)})$.

*Step 2.* $\Theta_k^r = \bar{\Theta}_k + \rho\bar{\Theta}_k - \Theta_k^{(n+1)}$ where $\bar{\Theta} = \sum_{i=1}^{N} \frac{\Theta^{(i)}}{n}$ If $f(\Theta_k^{(1)}) \leq f(\Theta_k^r) \leq f(\Theta_k^{(n)})$, then $\Theta_k^{(n+1)} \leftarrow \Theta_k^r$, and go to step 1

*Step 3.* $\Theta_k^e = \bar{\Theta}_k + \chi(\Theta_k^r - \Theta_k^{(n+1)})$. If $f_e < f_r$, then $\Theta_k^{(n+1)} \leftarrow \Theta_k^e$, and go to step 4. Else $\Theta_k^{(n+1)} \leftarrow \Theta_k^r$, and go to step 1.

*Step 4.* $\Theta_k^c = \bar{\Theta}_k + \gamma(\Theta_k^r - \bar{\Theta}_k)$. If $f_c \leq f_r$, then $\Theta_k^{(n+1)} \leftarrow \Theta_k^c$, and go to step 1. Else go to step 5.

*Step 5.* $\left\{\Theta_k^{(1)}, \Theta_k^{(2)}, ..., \Theta_k^{(n+1)}\right\} = \left\{\Theta_k^{(1)}, v_k^{(2)}, ..., v_k^{(n+1)}\right\}$ where $v_k^{(i)} = \Theta_k^{(1)} + \sigma(\Theta_k^{(i)} - \Theta_k^{(1)})$

*Step 6.* If $k > k_{max}$ or $||\Theta_k^{(i)} - \Theta_{k+1}^{(i)}|| < \epsilon(1 + ||\Theta_k^{(i)}||)$, then STOP; else increment $k$ and go to step 1.

#### *1.4.2.2 Powell's Method*

Powell's algorithm (Powell (1964)) is a generalization of a straightforward and intuitive optimization algorithm, called the "taxi-cab" algorithm. The taxi-cab algorithm has $n$ search directions, usually the unit vectors in every dimension. The algorithm starts with an initial solution and search directions are evaluated sequentially. The best possible improvement is found in a search direction (through line minimization methods such as Golden section search), and is applied before moving on to the next search direction. The innovation of the Powell's algorithm over the taxi-cab algorithm

is the idea of updating the search directions based on the history of the search. Every time an improving solution is found, one of the search directions is replaced by the vector connecting the current solution and the improving solution, as it proved to be a good search direction.

## Algorithm for Powell's method

*Initialization.* Choose stopping parameters $k_{max}$ and $\epsilon$. Initialize $\Theta_0 = \Theta_0^{DL}$. Set $k = 1$.

*Step 1.* For $i = 1, ..., n$ find $\lambda_i$ that minimizes $f(\Theta_k + \lambda_i u_i)$ where $u_i = (0, ..., 1_i, ..., 0)$.

*Step 2.* Set $u_{j+1} = u_j$ for $j = 1, ..., n-1$ and $u_n = \Theta_n - \Theta_0$.

*Step 3.* $\Theta_{k+1} = \Theta_k + \lambda_i u_k$ for $i = 1, ..., n$.

*Step 4.* $u_{k+1} = \lambda u_k$.

*Step 5.* If $k > k_{max}$ or $||\Theta_k - \Theta_{k+1}|| < \epsilon(1 + ||\Theta_k||)$, then STOP; else increment $k$ and go to step 1.

### 1.4.3 Gradient Based Algorithms

Gradient based optimization algorithms iteratively search for a minimum by computing (or approximating) the gradient of the objective function. I use two popular and efficient gradient based optimization algorithms, namely Broyden-Fletcher-Goldfarb-Shanno Algorithm (BFGS) and Generalized Reduced Gradient (GRG). One of the key advantage of these algorithms compared to global and direct search algorithms is their theoretical capability of using the geometry of NSS parameters space via its function gradient information.

#### *1.4.3.1 Broyden-Fletcher-Goldfarb-Shanno Algorithm*

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (see Judd (1998)) is a quasi-Newton algorithm which is an iterative procedure with a local quadratic approximation of the function. An approximation of the Hessian of the objective function is used instead of the function itself, which decreases the complexity of the algorithm. As all variants of Newton's methods, the idea behind them is to start at a point $\Theta_0$ and find the quadratic polynomial $J^0(\Theta)$ that approximates $J(\Theta)$ to its second degree Taylor expansion evaluated at $\Theta^0$:

$$J^0(\Theta) \equiv J(\Theta_0) + \nabla J(\Theta_0)'(\Theta - \Theta_0) + \tfrac{1}{2}(\Theta - \Theta_0)'\nabla^2 J(\Theta_0)(\Theta - \Theta_0), \tag{21}$$

where $\nabla$ is the gradient operator, such that:

$$\nabla J(\Theta^0) = \left(\frac{\partial J}{\partial \beta_0}(\Theta_0), \frac{\partial J}{\partial \beta_1}(\Theta_0), \frac{\partial J}{\partial \beta_2}(\Theta_0), \frac{\partial J}{\partial \beta_3}(\Theta_0), \frac{\partial J}{\partial \tau_1}(\Theta_0), \frac{\partial J}{\partial \tau_2}(\Theta_0)\right) \tag{22}$$

and $H = \nabla^2 J(\Theta_0)$ is the Hessian matrix of $J(\Theta_0)$, that is the symmetric matrix containing the second-order derivatives of $J$:

$$H = \left(\frac{\partial^2 J}{\partial \Theta \partial \Theta'}(\Theta_0)\right). \tag{23}$$

Quasi-Newton methods specify $d_k$ (the direction vector) as:

$$d_k = -H_k^{-1} \nabla J(\Theta_k)^{\mathsf{T}}, \tag{24}$$

The step size $\alpha_k$ is obtained by line minimization, where the direction $H_k$ is a positive definite matrix, $H_k$ may be adjusted from one iteration to the next so that $d_k$ tends to approximate the Newton direction.

#### *1.4.3.2 Generalized Reduced Gradient Algorithm*

The idea behind the Generalized Reduced Gradient (GRG) algorithm (Lasdon et al. (1978)) is to build a derivative matrix that contains the derivatives of the function with respect to each variable. After the first iteration, the algorithm modifies the values of the initial guess, as the optimization process goes on the variable values are updated after each iteration, until the algorithm reaches a satisfactory value (from the point of view of the operator) or the error reaches its pre-defined limit. One of the necessity of the GRG requires the function derivative and its calculation. In the NSS estimation process, the functional form is suitable for calculation of the derivative.

GRG formulates a local linearization of the nonlinear constraints and performs variable elimination. The optimization problem is solved by line minimization along a direction obtained by the gradient of the reduced objective function. Gradient is computed in the same manner as explained in Section 1.4.3.1 (BFGS). At each iteration, a search direction $d$ is generated and a one dimensional search begins. If the search finds an improved point, a new iteration is started. If the search fails and $d_k \neq P\nabla J(\Theta_k)$, then $d_k$ is set to $P\nabla J(\Theta_k)$ and a new search is started. Otherwise the program stops.

**Algorithm for BFGS method.** *Initialization.* Initialize $\Theta^{(0)} = \Theta_0^{DL}$, initial positive definite Hessian guess $H_0$ (usually the identity matrix), and stopping parameters $\delta$ and $\epsilon > 0$.

*Step 1.* Solve $H_k d_k = -\nabla J(\Theta_k)$ for the search direction.

*Step 2.* Solve $\alpha_k = \arg\min_\alpha J(\Theta_k k + \alpha d_k)$.

*Step 3.* $\Theta_{k+1} = \Theta_k + \alpha_k d_k$.

*Step 4.* Update $H_k$ :

$$\begin{aligned} z_{k+1} &= \Theta_{k+1} - \Theta_k, \\ s_{k+1} &= \nabla J(\Theta_{k+1}) - \nabla J(\Theta_k), \\ u_k &= s_k H_k z_k, \\ v_k &= \frac{z_k}{z_k' s_k} - \frac{H_k s_k}{u_k}, \\ H_{k+1} &= H_k - \frac{z_k z_k'}{z_k' s_k} - \frac{H_k s_k s_k' H_k}{s_k' H_k s_k} \end{aligned}$$

*Step 5.* If $||\Theta_k - \Theta_{k+1}|| < \epsilon(1 + ||\Theta_k||)$, then go to step 6, else increment $k$ go to step 1.

*Step 6.* If $||f(\Theta_k)|| < \delta(1 + ||f(\Theta_k)||)$, then STOP; else increment $k$ and go to step 1.

**Algorithm for GRG method** *Initialization.* Initialize $\Theta^{(0)} = \Theta_0^{DL}$, and stopping parameters $\delta$ and $\epsilon > 0$.

*Step 1.* Compute $d_k = P\nabla J(\Theta_k)$: $P = Q_2^T Q_2$ satisfying $N^T P w = 0$ and $Q_2^T = \begin{bmatrix} -N_1^{-1}N_2 \\ I \end{bmatrix}$. [a]

*Step 2.* Solve $\alpha_k = \arg\min_\alpha J(\Theta_k + \alpha d_k)$.

*Step 3.* $\Theta_{k+1} = \Theta_k + \alpha_k d_k$.

*Step 4.* If $||\Theta_k - \Theta_{k+1}|| < \epsilon(1 + ||\Theta_k||)$, then go to step 5, else go to step 1.

*Step 5.* If $||f(\Theta_k)|| < \delta(1 + ||f(\Theta_k)||)$, then STOP; else increment $k$ and go to step 1.

[a] $N^T = [N_1 \quad N_2]$ (Columns of $N$ are the gradients of constraints) $N_1$ is the transpose of $r$ linearly independent rows of $N$. The matrix $Q_2$ consists of the last $n - r$ rows of the $Q$ factor in the $QR$ factorization of $N$ and $w$ is an arbitrary vector. Namely,

$$QN = \begin{bmatrix} Q_1 N \\ Q_2 N \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{25}$$

where $R$ is an $r \times r$ upper triangular matrix.

### 1.4.4 Implementation details

The algorithms were coded in MATLAB, except for the GRG algorithm, for which I used the Solver module that is bundled with Microsoft Excel Fylstra (1998). For Nelder-Mead, Powell, SA and PSO algorithms the constraints have been embedded in objective function with a penalty constant $C = 1000$, whereas they have been explicitly stated for BFGS. Stopping criterion for the algorithms has been chosen as $\epsilon = 10^{-8}$. Parameter set for the Nelder-Mead algorithm is chosen as $\rho = 1, \chi = 2, \gamma = 0.5, \sigma = 0.5$. For the SA algorithm, initial value of $T$ is chosen as 1.

## *1.5 Empirical Results*

In this section, I present the properties of my dataset, performance measures for my algorithms and details of my computational experiments.

### 1.5.1 Data

I collected week end (i.e., Friday) quotations for government bonds from three emerging countries—Brazil, Mexico and Turkey—and the U.S. The emerging market sovereign debt was in the form of U.S. dollar Eurobonds. [26] I exclude all the bonds with special characteristics (e.g. callable, puttable, structured, convertible, and Brady Bonds) in order to ensure that a homogeneous and reliable sample is used in my analysis. The three emerging market countries were selected because of the relative liquidity of the markets for these bonds and their increasing popularity among investors. The emerging market bond bid- and asked prices data were obtained from Bloomberg and Datastream and the characteristics of emerging market bonds were collected from Reuters 3000 Extra. The U.S. bond bid and asked prices and bond characteristics were obtained from the CRSP Government Bond files. My data set covers the time period between July 2005 and December 2015. The advantage of using this longer sample period for the fixed income markets is that it covers three major financial crises (the U.S. subprime crisis, the European sovereign debt crisis and FED tapering tantrum) rather than just the relatively uneventful period. This enables us to see how robust is the optimization algorithms during the most turbulent markets conditions.

The number of issues included in the market varies considerably across countries as shown in Table 1. This allows us to test the optimization methods on both dense

[26] I consider only dollar denominated Eurobonds issued by these countries because U.S. dollar denominated bonds have longer maturity and more liquidity relative to Euro denominated bonds for the same countries.

and sparse data sets.

**Table 1:** Numbers of quotations per observation period

| | **Brazil** | **Mexico** | **Turkey** | **U.S.** |
|---|---|---|---|---|
| Min | 18 | 8 | 23 | 156 |
| Mean | 23.2 | 11.5 | 24.5 | 199.4 |
| Max | 26 | 14 | 26 | 262 |

Figure 6 plots the actual maturities of bonds in four country datasets for each observation period. All countries, provide a number of bonds of more than 10 years maturity. While there are not many such bonds, they do help to anchor the long rate, $\beta_0$.

**Table 2:** Brazil, Mexico and Turkey bonds in the dataset

| | | | | |
|---|---|---|---|---|
| | | **Turkey** | | |
| **# of bonds per day** | **Maturity : 0-3** | **Maturity : 3-5** | **Maturity : 5** | **All maturities** |
| **Average** | 12 | 4 | 16 | 31 |
| **Min** | 8 | 1 | 13 | 25 |
| **Max** | 14 | 5 | 20 | 36 |
| | | **Brazil** | | |
| **# of bonds per day** | **Maturity : 0-3** | **Maturity : 3-5** | **Maturity : 5** | **All maturities** |
| **Average** | 9 | 3 | 11 | 22 |
| **Min** | 6 | 1 | 2 | 16 |
| **Max** | 13 | 5 | 16 | 29 |
| | | **Mexico** | | |
| **# of bonds per day** | **Maturity : 0-3** | **Maturity : 3-5** | **Maturity : 5** | **All maturities** |
| **Average** | 10 | 3 | 13 | 26 |
| **Min** | 6 | 1 | 5 | 14 |
| **Max** | 15 | 5 | 18 | 33 |

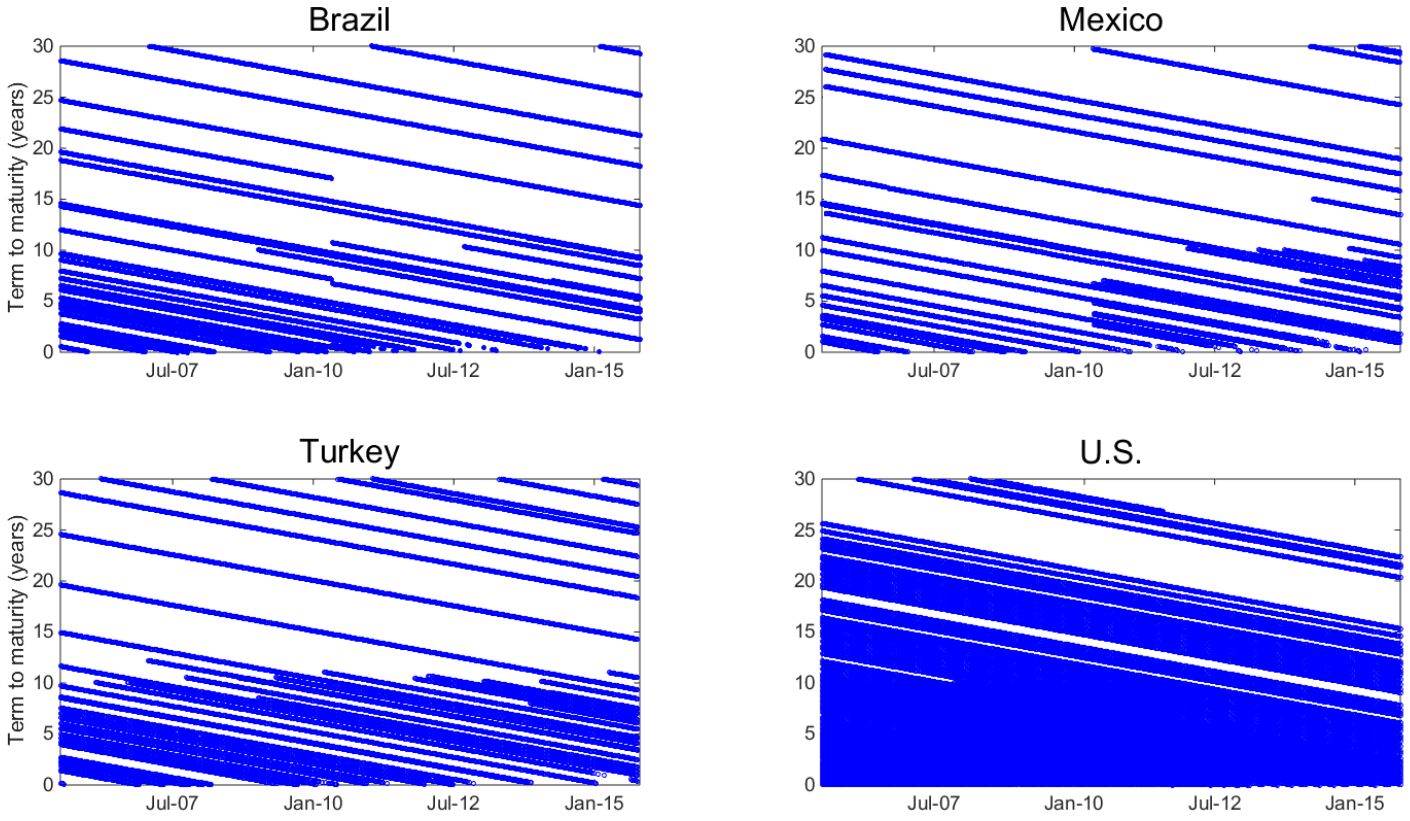


**Figure 6:** Maturities of bonds used by observation data and country

### 1.5.2 In-sample results

The in-sample goodness-of-fit statistics are given in Table 3.[27] The most striking result is the poor performance of the BFGS optimization method. This is consistently true across countries and whether I measure goodness-of-fit with RMSE and MAE. Differences among the remaining methods are less striking. However, while the Hybrid PSO method is only marginally better than the remaining methods, it consistently has the smallest RMSE and MAE, which suggests that this result is not random. This suggests that the global optimization algorithms may be better at finding the global minimum of the NSS function in the presence of multiple local minima.

Computation time results are mixed across the optimization algorithms. Table 4 clearly shows that the Nelder-Mead is by and far the fastest algorithm. As can be seen from description in Section 1.4.2.1, the Nelder-Mead algorithm applies simple operations to the simplex (reflection, contraction, expansion) and also does not evaluate gradient information, which makes it quite fast. This might be expected due to nature of the direct search algorithms, however, the Powell method does not do nearly as well. The BFGS method is far and away the slowest. The PSO variants are materially quicker than all but the Nelder-Mead method. Interestingly, the remaining global optimization algorithm, the SA, is much slower than the PSO-based methods. I note that Hybrid PSO is slowest of the PSO-based methods for all countries.

The solution quality also depends on the initial starting point, as I argued in Section 1.4.4, I believe the starting values I use are reasonably good ones. However, the distance traveled statistics reported in Table 4 have to be evaluated in conjunction

[27] The lowest values for each column are shown in bold face in Tables 3 through 7.

with the goodness-of-fit achieved. I find that the BFGS algorithm travels much further than the other methods, with one notable exception, the Powell method applied to the Mexican data. However, the BFGS goodness-of-fit statistics are the worst of any of the methods. So instead of traveling a greater distance to find a better solution, the BFGS method shows a tendency to lose its way and move away from the better solutions nearer to the starting point that the other methods are able to find. As I shall show in the next section, the BFGS algorithm also has severe problems converging to a solution on many observation days. When combined with the slowness of the BFGS, these other results suggest that the BFGS is unsuited to this type of problem, or at least to estimating a functional form with the problems that I noted in Section 1.3.2.

In contrast to the BFGS algorithm, the SA algorithm does not travel very far to find a solution for the emerging markets data. However, the solutions obtained were slightly worse than those obtained by the PSO-W and Hybrid PSO algorithms that have traveled much farther. For the U.Ss data, I see the SA algorithm traveling much farther only to find a worse solution, particularly when looking at MAE, then, the Hybrid PSO algorithm. The very large distance traveled by the Powell algorithm when applied to the Mexican data, taken together with the only moderately worse goodness-of-fit for that case, is indicative of the flatness of the objective function surface and, therefore, of the difficulty that this estimation problem presents.

Table 5 presents the deviations of individual parameter estimates from their starting values for the different countries and different optimization methods. The statistics reported are the root mean squared differences. This allows us to see whether the distance traveled results in Table 4, which apply to the entire 6-parameter starting

point and solution vectors, carry over to the individual parameters in which I are primarily interested when interpreting the term structure. I can see that the individual parameter distances traveled are roughly consistent with the explanatory power of the corresponding factors; the distance traveled is largest when the corresponding factor explanatory power is lowest. The level factor is greatest power for explaining shifts in the term structure, varies between 60 and 90 percent (see Bliss (1997a)) and the corresponding $\beta_0$ has the least individual parameter distance traveled. The explanatory power of the slope factor varies between six and 20 percent. The corresponding $\beta_1$ parameter shows a slightly further individual distance traveled than $\beta_3$. The curvature factor explains only between two and 10 percent of term structure movements. Where a factor has little explanatory power, changes in the corresponding parameter may have less effect on the objective function value. This may be the reason for the pattern I observe in the individual parameters distances traveled, observed by optimization method and country while comparing the values for the three parameters.

Finally, I plot the time series of the estimated $\beta_0$, $\beta_1$, and $\beta_2$ parameters, which capture the size of level, slope and curvature factors, together with their empirical proxies. To save space, I present the results only from one country, Brazil, and for the four optimization algorithms that produced the lowest RMSE statistics for that country. [28]

Figures 7, 8, and 9 first show that the empirical proxies for these three parameters do show some variation from week to week.[29,30] However, they do not show evidence

[28]The results for other countries are similar, and for the remaining optimization algorithms are worse than those shown here, with the exception of the PSO-G algorithm, which behaves much like the other global optimization algorithms in this respect. The results omitted here are available from the authors on request.

[29]To save the space, I only provide the results for Brazil. Results are very similar for other countries as well and are available from the authors on request.

[30] I have performed this analysis with daily observations as well. The results were similar.

of instability as I see, for instance, in the three Powell algorithm-estimated parameter time series. The Hybrid PSO- and SA-estimated $\beta_0$ and $\beta_1$ parameters follow their empirical proxies very closely, while the PSO-W-based estimates show some slight problems for these parameters. Only the Hybrid PSO is free of evident problems for $\beta_2$. The PSO-W- and SA-estimated parameters show large deviations from their empirical values, while Powell algorithm is clearly having problems estimating this parameter.

Hybrid PSO algorithm's superior performance might be due to the uniquely distinguishing characteristic of PSO—that knowledge of good solutions is retained by all the particles. Each particle has memory and keeps track of its previous best position and the corresponding value of the objective function. Although this leads to a number of local optima for the respective particles in the swarm, the global best of the swarm is given special weight in converging the individual particles towards the algorithm's solution. This constructive cooperation between particles; that is, particles in the swarm share information among themselves, leads to stable parameters for yield curve. This study demonstrates that PSO algorithm can differ markedly from other optimization algorithms in terms of smoothness of estimated NSS parameters.

**Table 3:** Comparison of optimization algorithms in terms of total RMSE and MAE

| | **Total RMSE** | | | | **Total MAE** | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **Brazil** | **Mexico** | **Turkey** | **U.S.** | **Brazil** | **Mexico** | **Turkey** | **U.S.** |
| Gradient Based Algorithms | | | | | | | | |
| BFGS | 0.507 | 0.523 | 0.491 | 0.559 | 0.440 | 0.407 | 0.810 | 0.407 |
| GRG | 0.239 | 0.234 | 0.248 | 0.234 | 0.281 | 0.146 | 0.187 | 0.146 |
| Direct Search Algorithms | | | | | | | | |
| Nelder-Mead | 0.282 | 0.241 | 0.271 | 0.246 | 0.275 | 0.198 | 0.222 | 0.198 |
| Powell | 0.234 | 0.198 | 0.236 | 0.224 | 0.282 | 0.232 | 0.220 | 0.232 |
| Global Optimization Algorithms | | | | | | | | |
| SA | 0.224 | 0.215 | 0.230 | 0.235 | 0.171 | 0.171 | 0.178 | 0.171 |
| PSO-W | 0.214 | 0.179 | 0.204 | 0.217 | 0.166 | 0.145 | 0.153 | 0.145 |
| PSO-G | 0.236 | 0.201 | 0.238 | 0.227 | 0.186 | 0.164 | 0.185 | 0.164 |
| Hybrid PSO | **0.212** | **0.178** | **0.203** | **0.196** | **0.165** | **0.143** | **0.149** | **0.107** |

**Table 4:** Comparison of optimization algorithms in terms of CPU time and average distance of the solution from the starting point

| | CPU Time (sec) | | | | Avg. Distance | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **Brazil** | **Mexico** | **Turkey** | **U.S.** | **Brazil** | **Mexico** | **Turkey** | **U.S.** |
| **Gradient Based Algorithms** | | | | | | | | |
| BFGS | 3156.2 | 2450.3 | 2747.9 | 2450.3 | 2.842 | 7.112 | **5.210** | **8.574** |
| GRG | 312.7 | 320.7 | 290.6 | 783.7 | 2.012 | 2.563 | 2.466 | 3.773 |
| **Direct Search Algorithms** | | | | | | | | |
| Nelder-Mead | **6.5** | **5.6** | **6.4** | **12.5** | 1.732 | 1.950 | 1.916 | 3.012 |
| Powell | 207.8 | 212.5 | 269.7 | 343.2 | 1.139 | **19.505** | 2.420 | 3.558 |
| **Global Optimization Algorithms** | | | | | | | | |
| SA | 451.4 | 398.5 | 438.5 | 562.6 | 0.263 | 0.249 | 0.238 | 4.275 |
| PSO-W | 117.1 | 76.9 | 107.3 | 115.8 | 2.665 | 3.122 | 2.810 | 4.200 |
| PSO-G | 106.4 | 68.8 | 97.5 | 103.5 | **3.065** | 3.248 | 2.588 | 4.439 |
| Hybrid PSO | 164.7 | 107.2 | 151.7 | 172.8 | 2.722 | 3.217 | 3.024 | 1.260 |

**Table 5:** Deviation of three factors from their empirical proxies

| $\beta_0$ | | | | |
|---|---|---|---|---|
| **Method** | **Brazil** | **Mexico** | **Turkey** | **U.S.** |
| Gradient Based Algorithms | | | | |
| BFGS | 2.596 | 69.609 | 11.593 | 3.941 |
| GRG | **0.283** | 1.529 | **0.252** | 0.329 |
| Direct Search Algorithms | | | | |
| Nelder-Mead | 1.120 | 1.263 | 0.818 | 1.026 |
| Powell | 3.108 | 6.235 | 3.559 | 3.402 |
| Global Optimization Algorithms | | | | |
| SA | 0.915 | **0.455** | 0.403 | 0.408 |
| PSO-W | 1.037 | 1.117 | 0.879 | 0.325 |
| PSO-G | 1.143 | 1.061 | 0.863 | 0.288 |
| Hybrid PSO | 0.949 | 1.110 | 0.911 | **0.255** |
| $\beta_1$ | | | | |
| Gradient Based Algorithms | | | | |
| BFGS | 76.137 | 76.491 | 46.624 | 6.575 |
| GRG | 2.309 | 3.050 | 3.877 | 2.050 |
| Direct Search Algorithms | | | | |
| Nelder-Mead | 2.414 | 2.491 | 2.629 | 3.261 |
| Powell | 5.656 | 7.785 | 43.478 | 5.082 |
| Global Optimization Algorithms | | | | |
| SA | **1.261** | **1.565** | **1.200** | 0.722 |
| PSO-W | 2.398 | 1.795 | 2.035 | 0.829 |
| PSO-G | 1.746 | 1.708 | 1.628 | 0.808 |
| Hybrid PSO | 1.676 | 1.705 | 1.682 | **0.650** |
| $\beta_2$ | | | | |
| Gradient Based Algorithms | | | | |
| BFGS | 66.093 | 114.659 | 73.855 | 31.057 |
| GRG | 849.774 | 5.379 | 220.904 | 6.379 |
| Direct Search Algorithms | | | | |
| Nelder-Mead | 30.748 | 34.618 | 35.094 | 26.582 |
| Powell | 122.262 | 11312.762 | 160.978 | 80.857 |
| Global Optimization Algorithms | | | | |
| SA | 6.461 | **3.883** | 4.755 | 3.792 |
| PSO-W | 4.450 | 5.178 | 6.261 | 3.787 |
| PSO-G | 4.258 | 4.254 | 4.290 | 4.305 |
| Hybrid PSO | **4.232** | 4.104 | **4.341** | **3.229** |

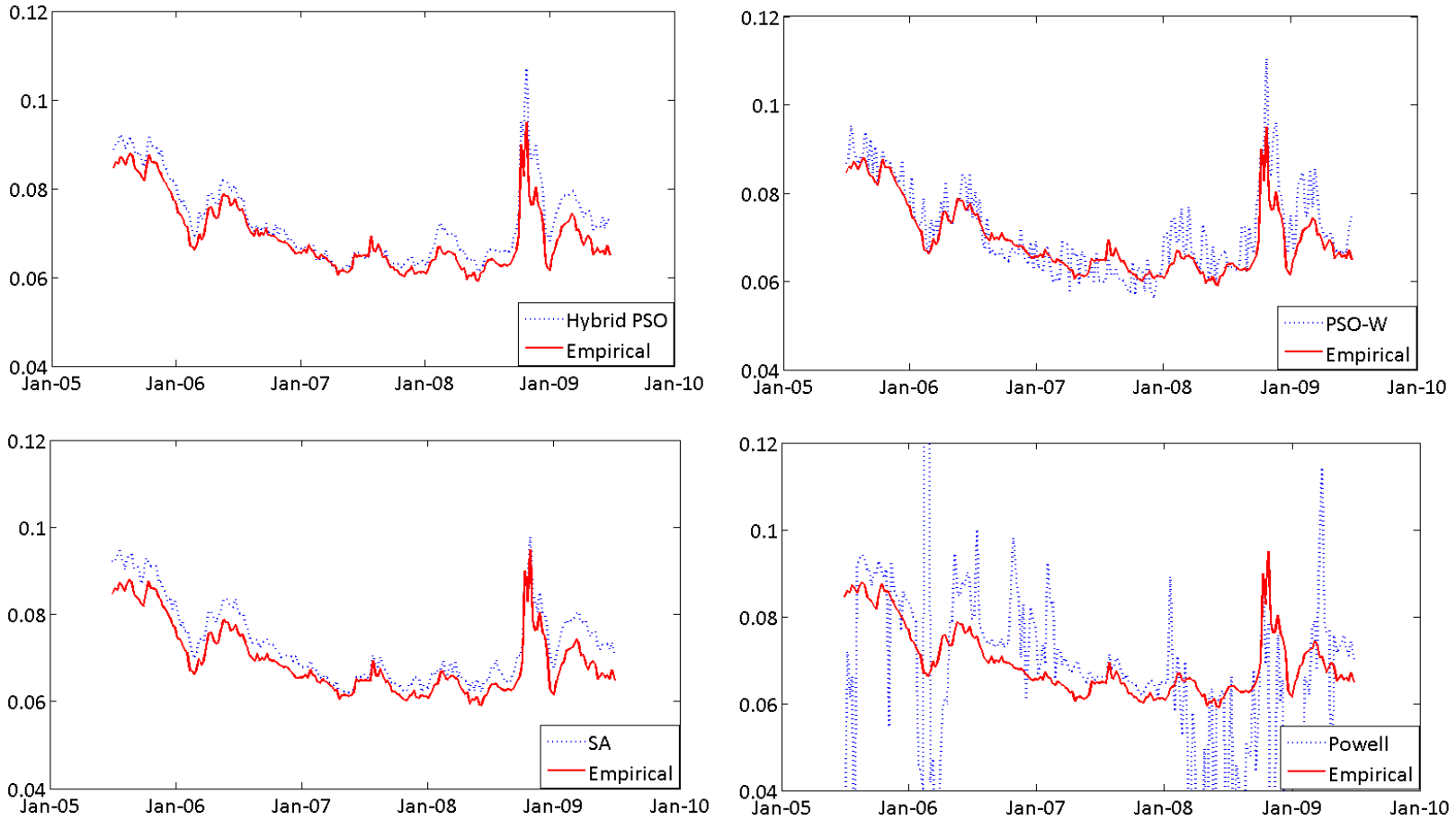


**Figure 7:** Evolution for level ($\beta_0$) in Brazil

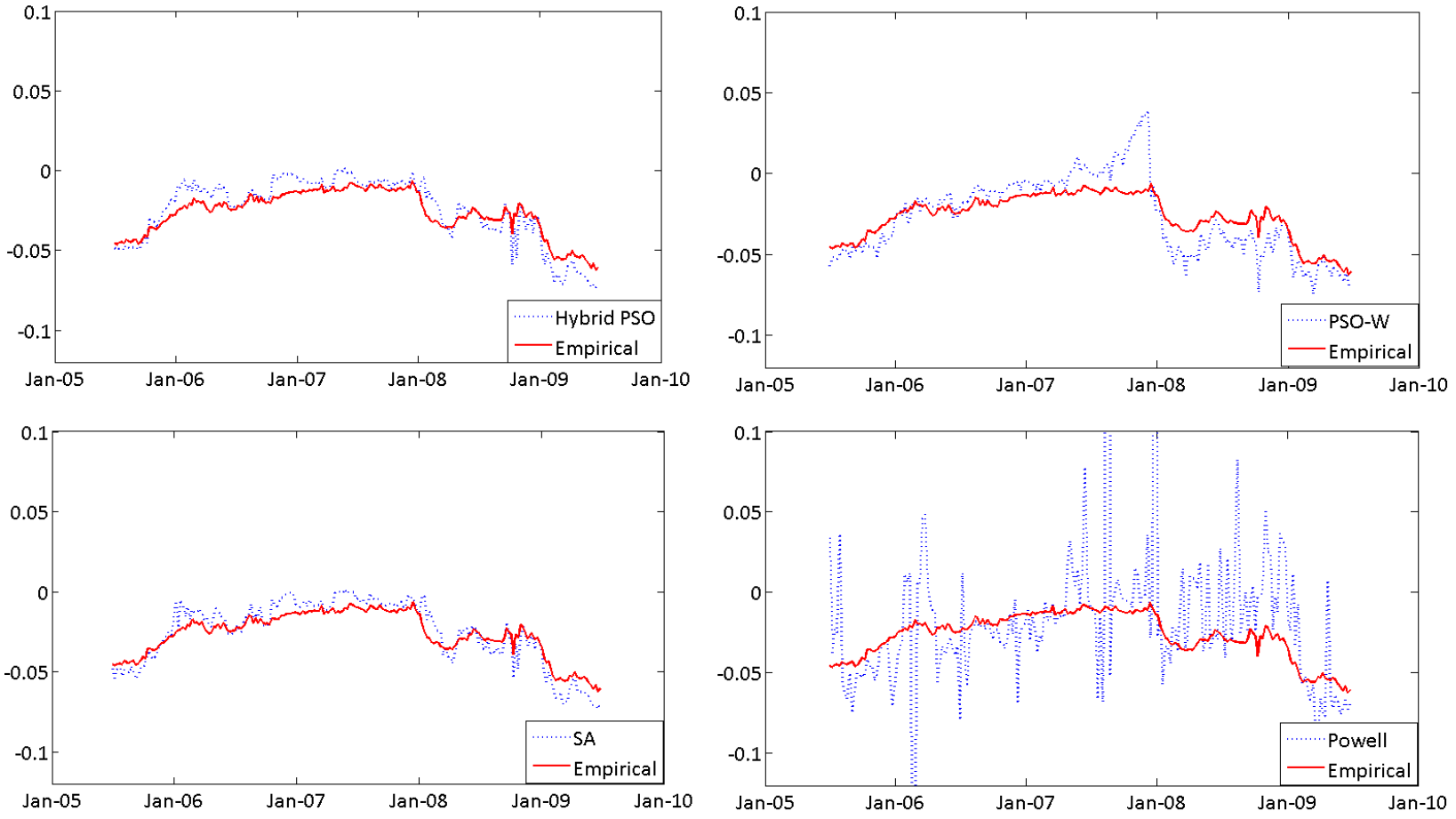


**Figure 8:** Evolution for slope ($\beta_1$) in Brazil

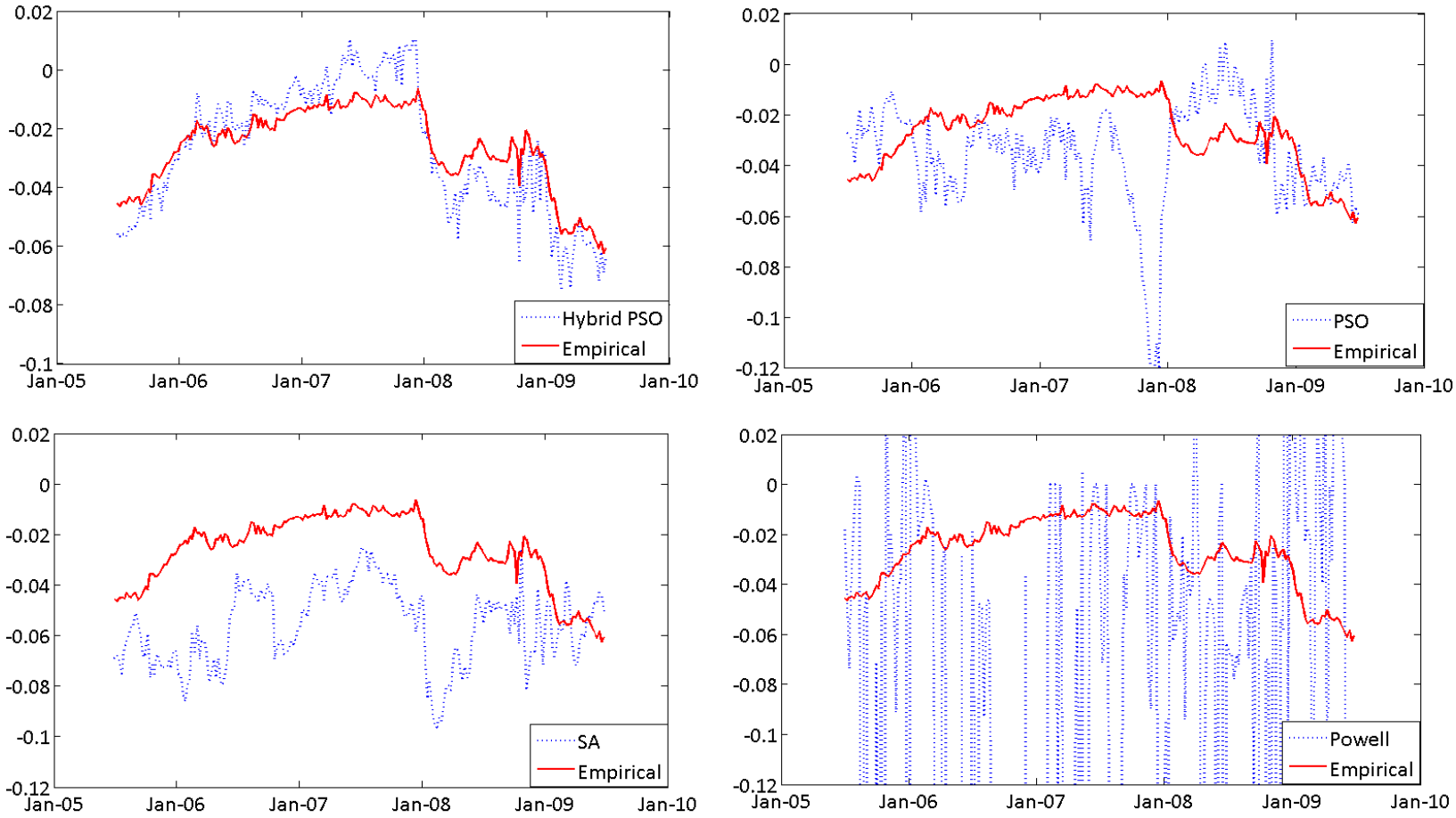


**Figure 9:** Evolution for curvature ($\beta_2$) in Brazil

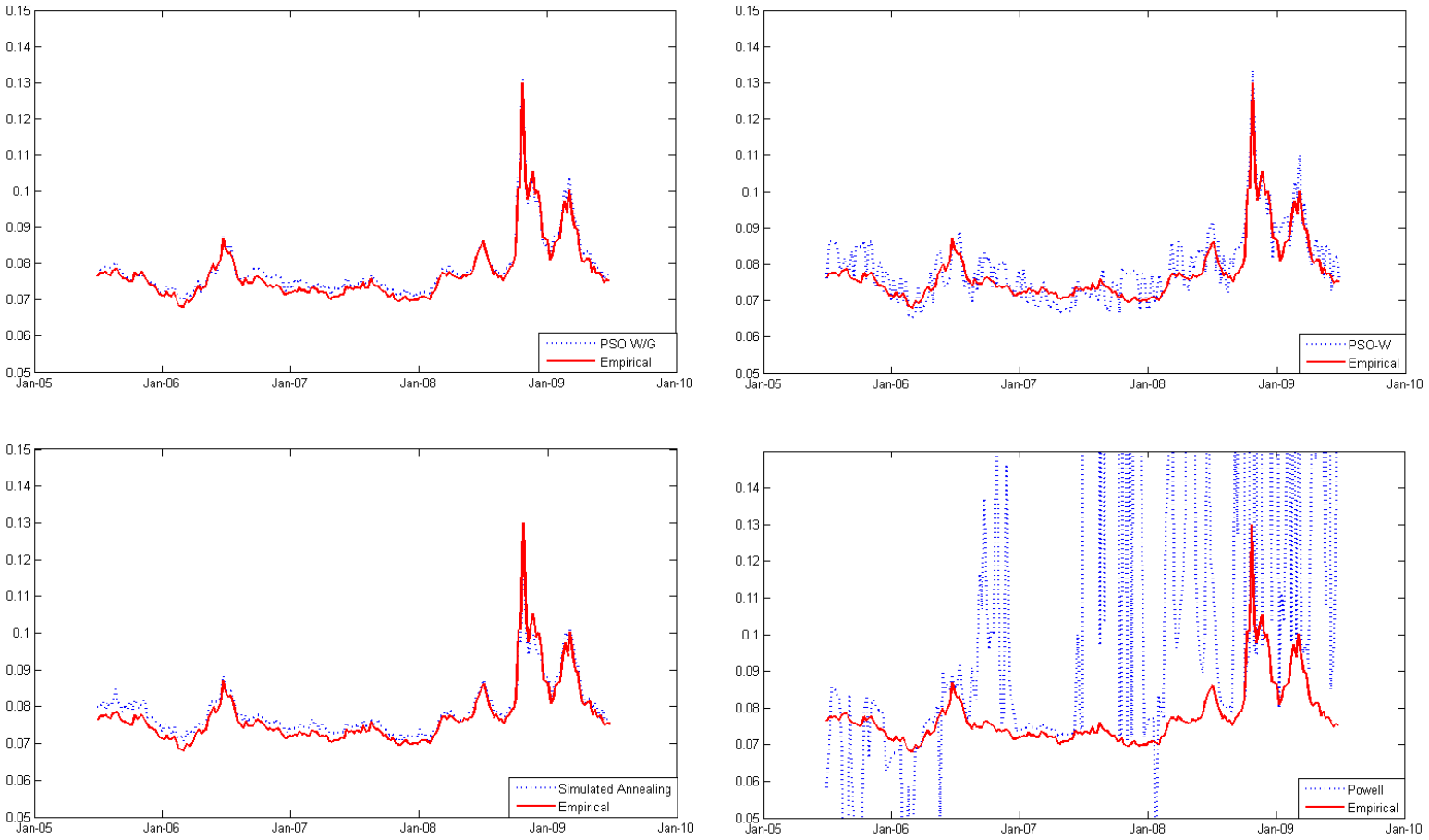


**Figure 10:** Evolution for level ($\beta_0$) in Turkey

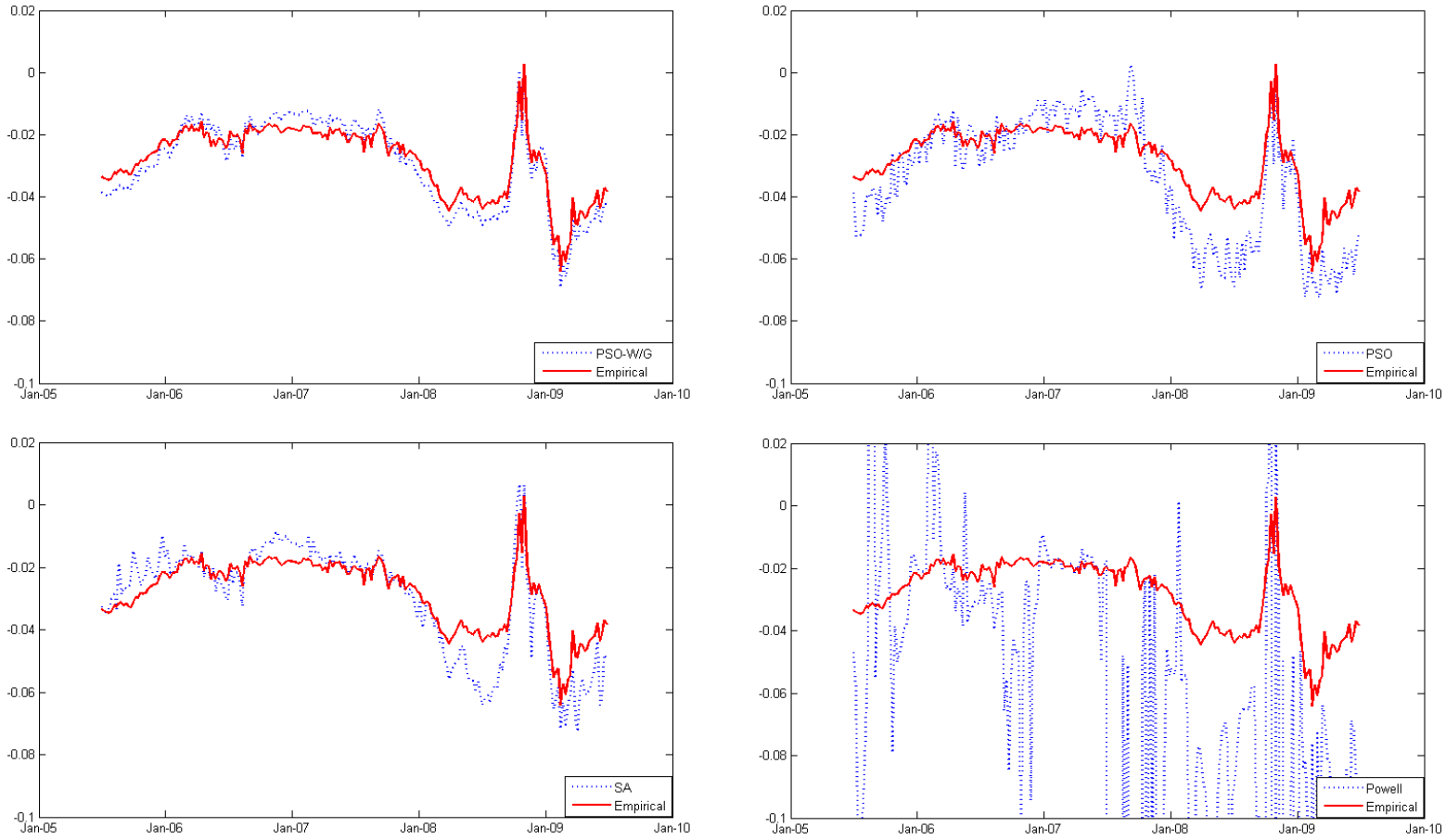


**Figure 11:** Evolution for slope ($\beta_1$) in Turkey

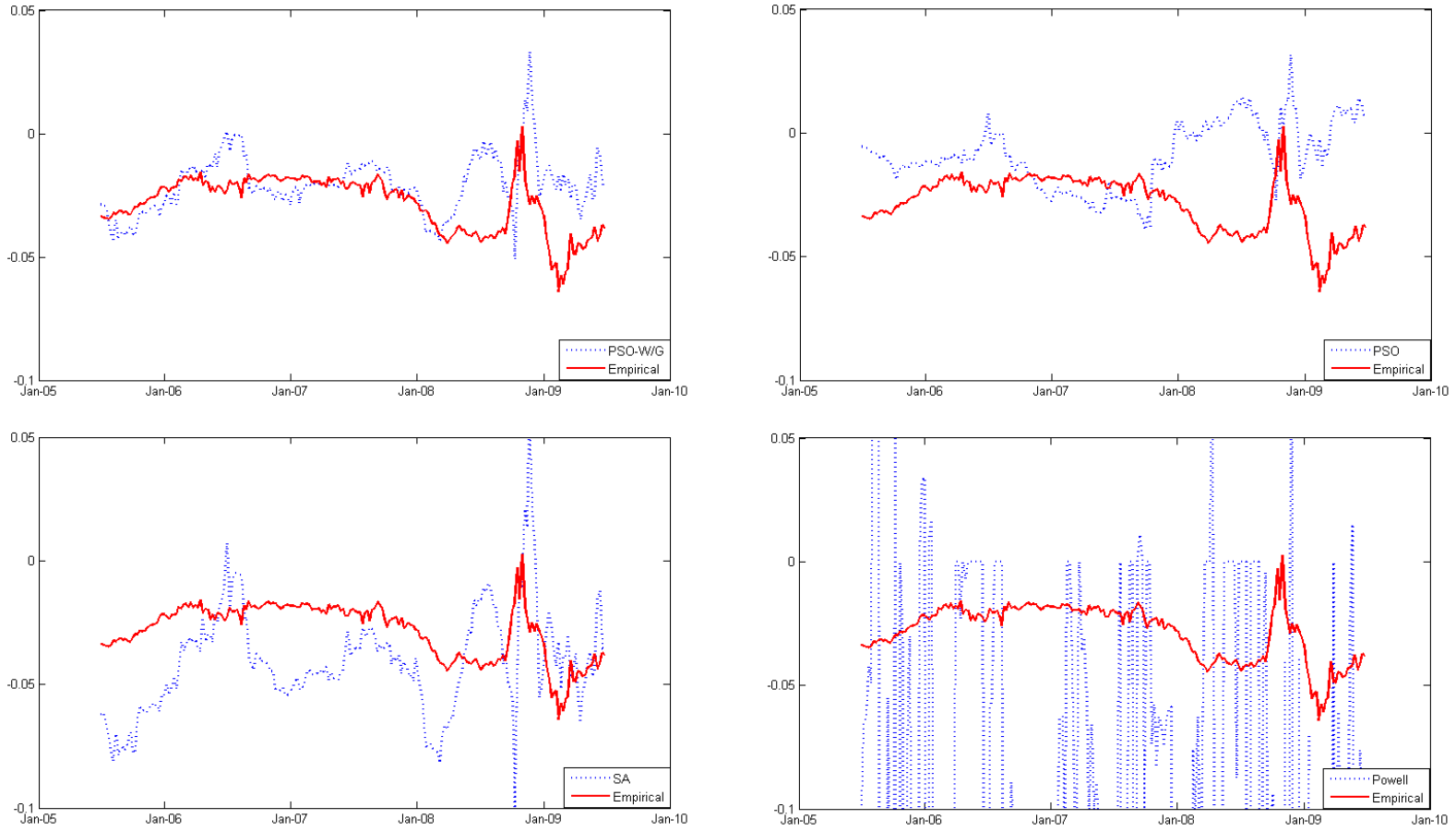


**Figure 12:** Evolution for curvature ($\beta_2$) in Turkey

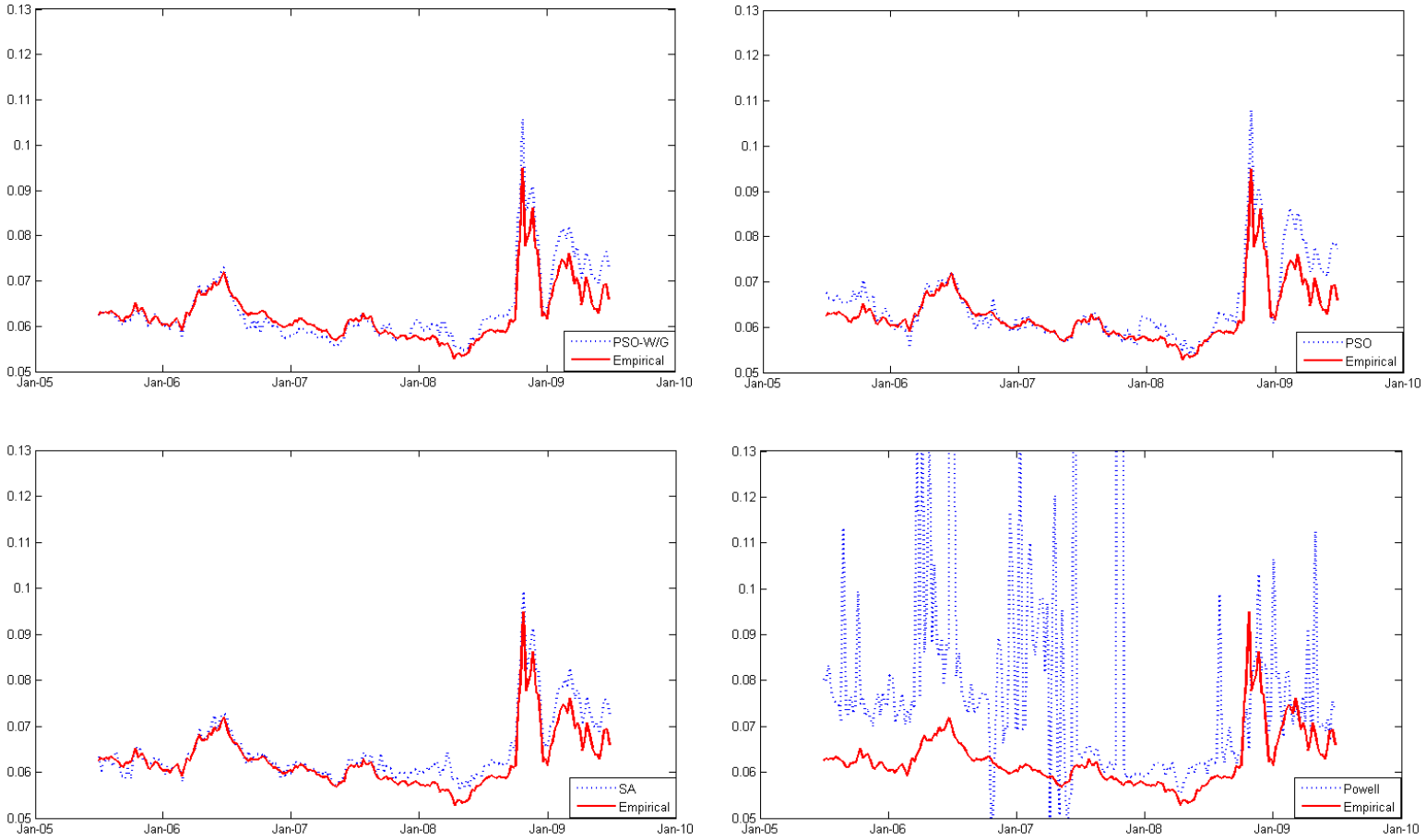


**Figure 13:** Evolution for level ($\beta_0$) in Mexico

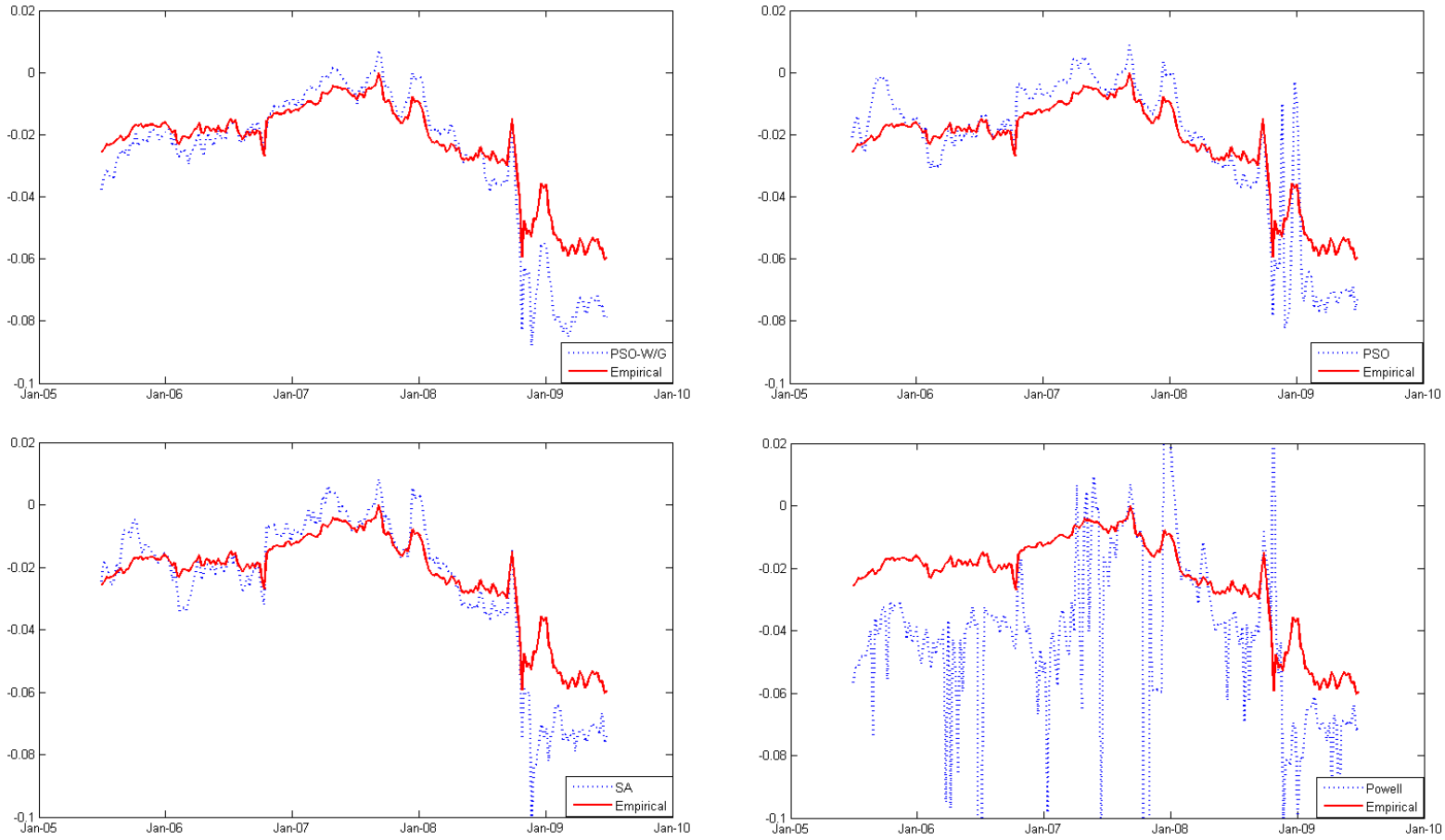


**Figure 14:** Evolution for slope ($\beta_1$) in Mexico

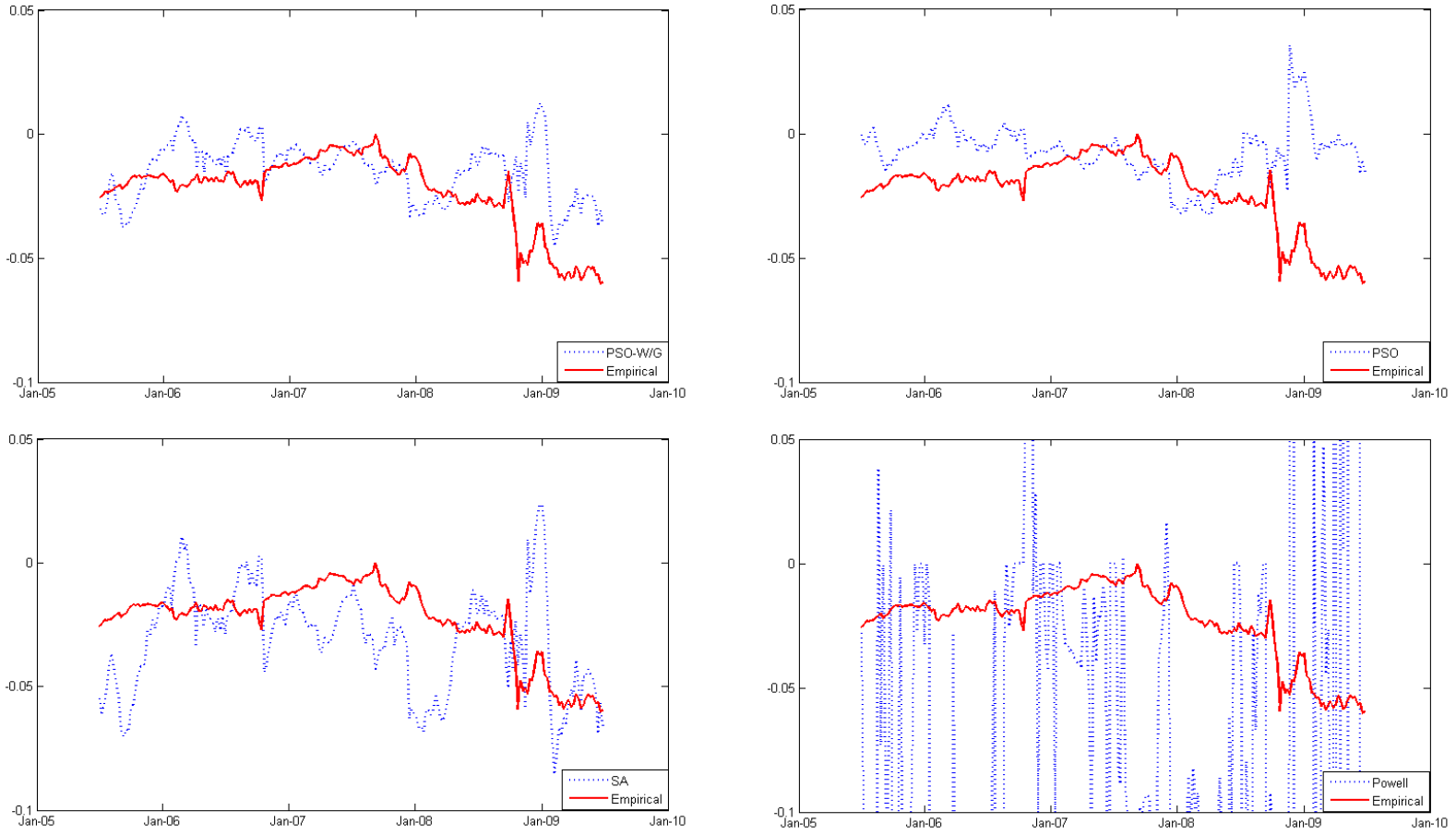


**Figure 15:** Evolution for curvature ($\beta_2$) in Mexico

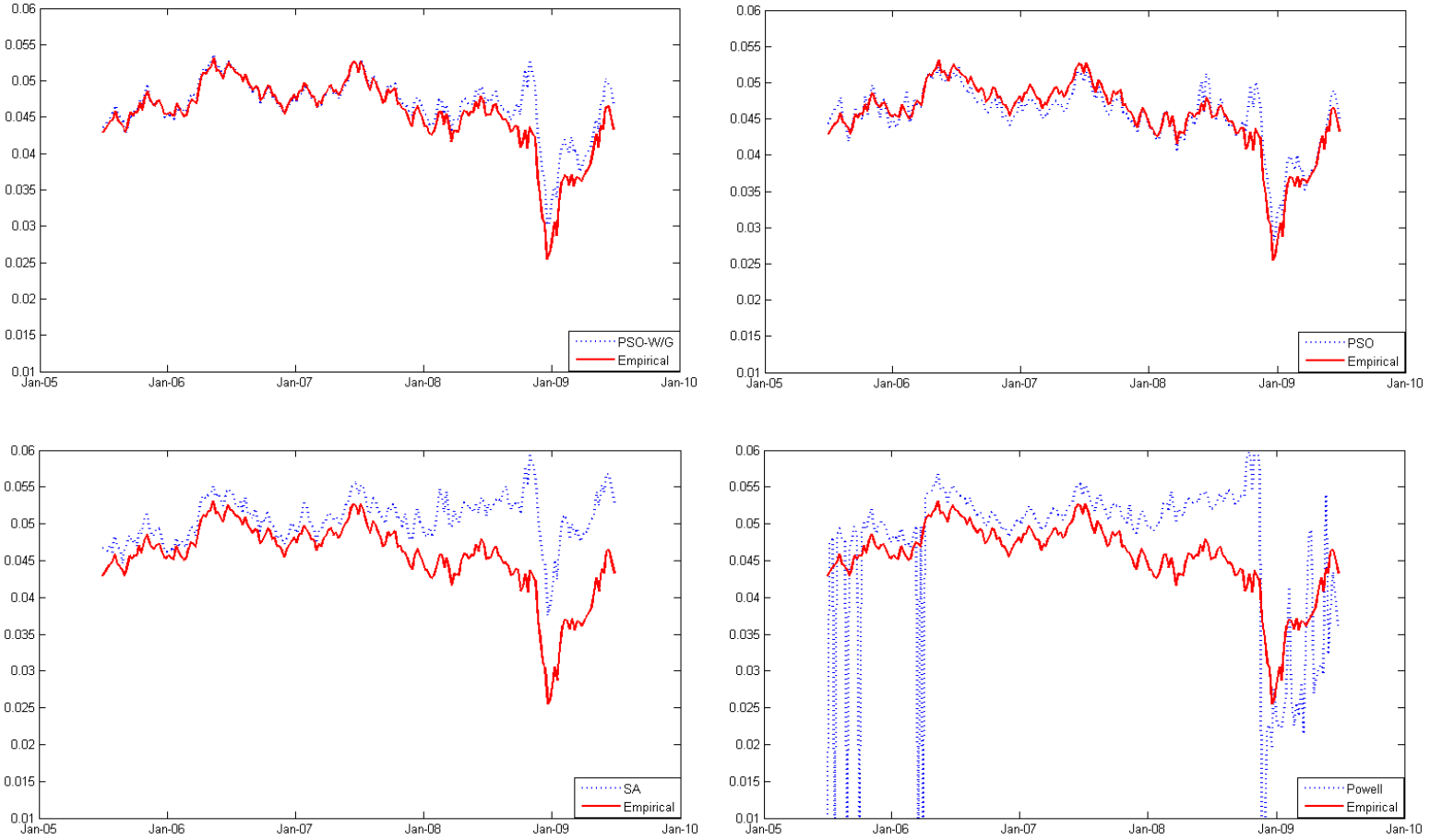


**Figure 16:** Evolution for level ($\beta_0$) in US

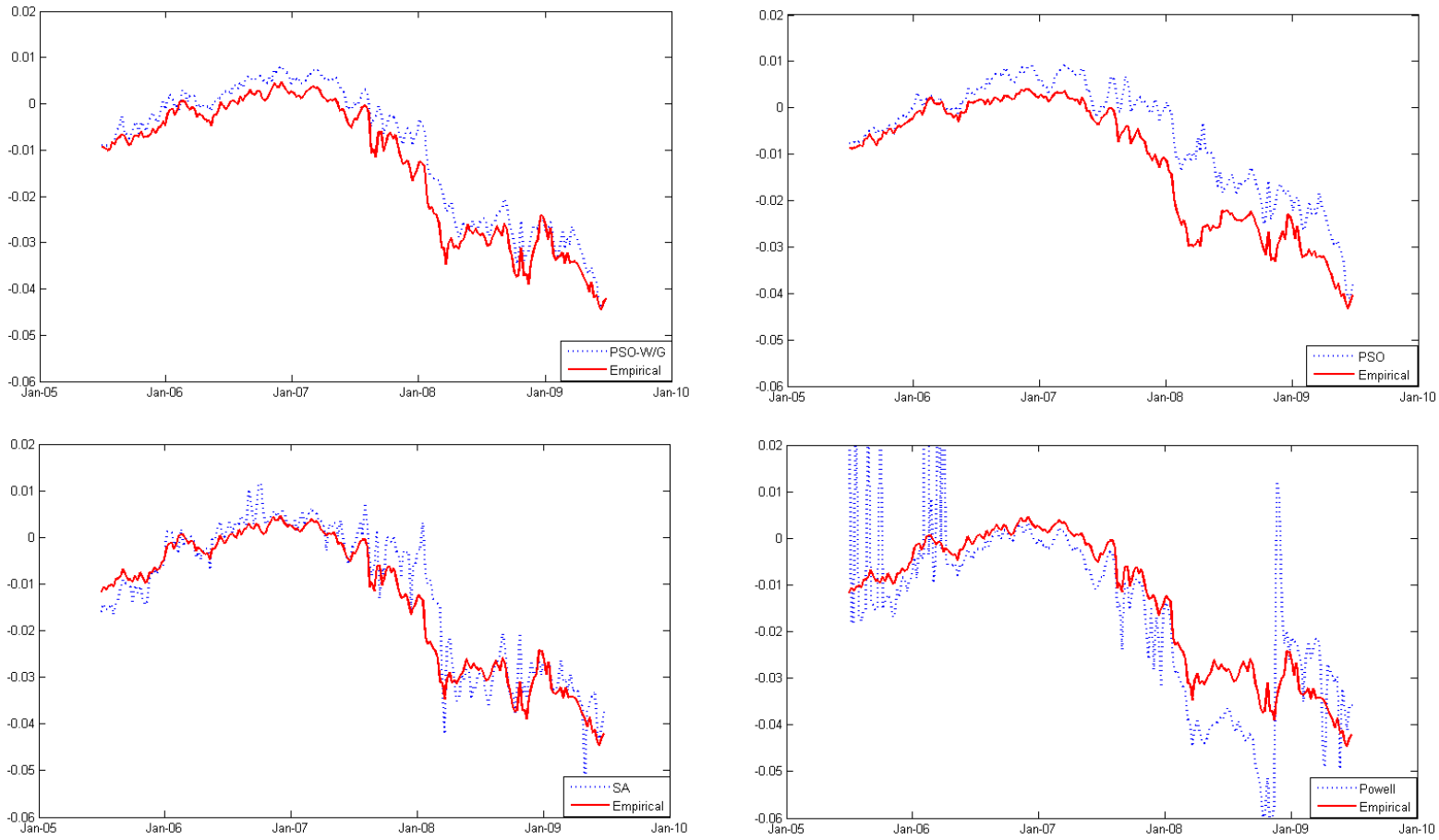


**Figure 17:** Evolution for slope ($\beta_1$) in US

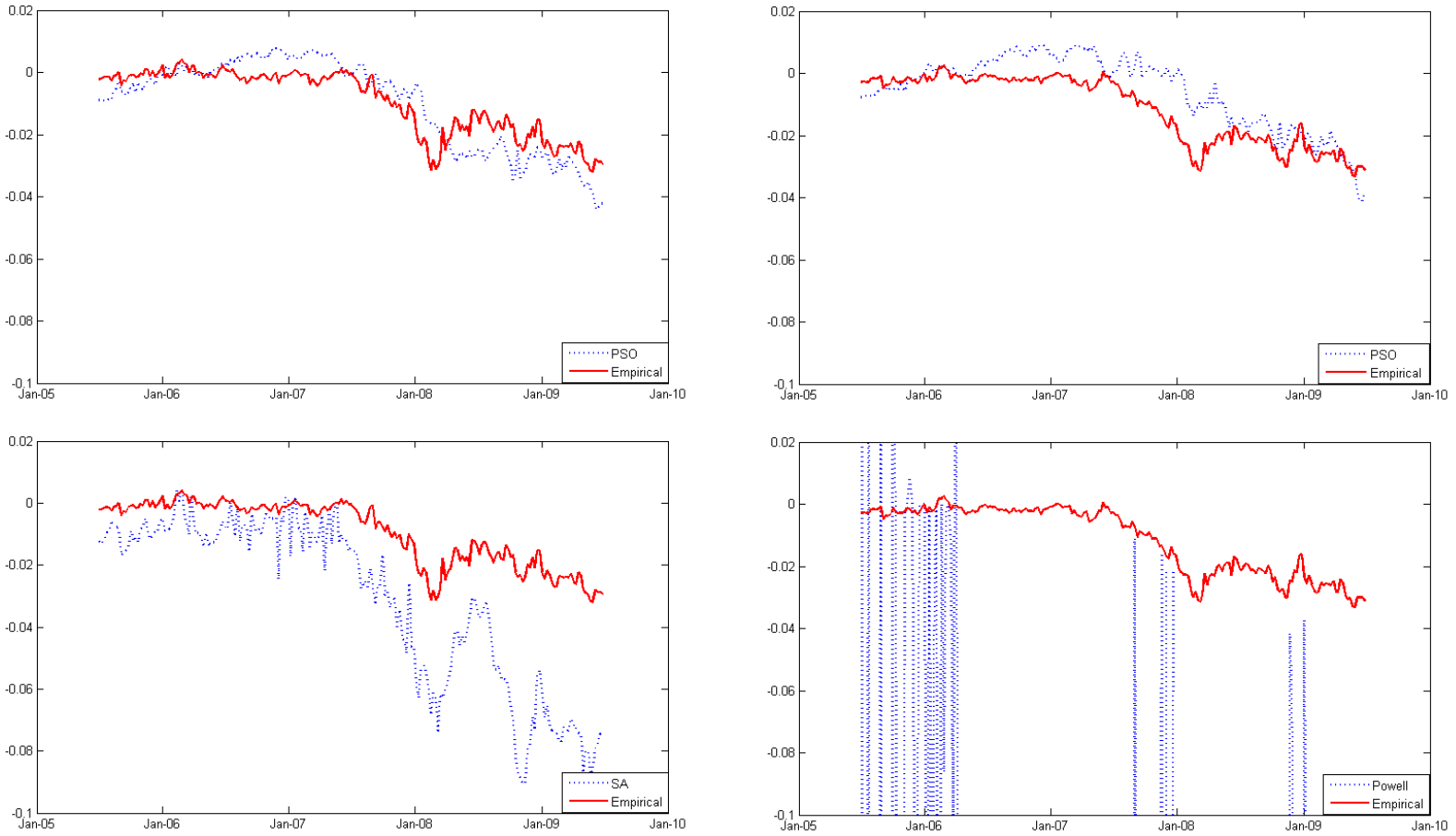


**Figure 18:** Evolution for curvature ($\beta_2$) in US

## *1.6 Robustness*

As discussed in Section 1.3.4.4, I apply four different checks to diagnose the robustness of the nonlinear optimization algorithms in the context of this term structure estimation problem: out-of-sample goodness-of-fit, to ensure that the in-sample results are reliable; sensitivity to starting values; sensitivity to pricing errors and application of the algorithms to local currency bonds.

While conducting the out-of-sample tests, I tracked the number of convergence failures for each method. The reported results are computed across both the original and reversed division of the data into estimation and hold-out sub-samples using the Brazil data. The relatively small number of data points in the Brazilian data sub-samples increases the difficulty in converting to a solution, and therefore highlights the relative performance of the methods. As I can see, the BFGS and Nelder-Mead have particular difficulty finding a solution in this situation. Only the PSO variants invariably converge to a solution.

**Table 6:** Percentage of convergence failures

| **Method** | **%Failures** |
|---|---|
| **Gradient Based Algorithms** | |
| BFGS | 52.1% |
| GRG | 39.1% |
| **Direct Search Algorithms** | |
| Nelder-Mead | 54.5% |
| Powell | 46.3% |
| **Global Optimization Algorithms** | |
| SA | 21.2% |
| PSO-W | 0.0% |
| PSO-G | 0.0% |
| Hybrid PSO | 0.0% |

### 1.6.1 Perturbation of bond prices

To test the sensitivity of the optimization algorithms to possible measurement error in the bond prices, I generate bond prices (uniformly) randomly selected from within the bid-asked range rather than the mean of the bid and asked quotes. I next fit the term structures using each optimization method and then generate three-month, two-year, and 10-year spot rates from the fitted term structures. The spot rates generated by the unperturbed data (using bid-asked mean prices) are then subtracted from the perturbed-price values. The experiment is then repeated 100 times to generate a distribution of the fitted spot rate deviations that result from the price perturbations. Figures 19 to 22 presents these distributions each of the three spot rates and for each of the eight optimization methods using box-and-whiskers (Tukey) plots. [31] Results are very similar for other countries as well and are available from the authors on request.

### 1.6.2 Out-of-sample tests

Table 7 presents the out-of-sample goodness-of-fit results. It is to be compared with Table 3 in Section 1.5.2. In general, the out-of-sample goodness-of-fit statistics are slightly worse than their in-sample equivalents.

The intra-quartile ranges for all the optimization methods are reasonably similar and small. Where the methods differ significantly is in the number and size of the outliers. The Powell method appears to be particularly sensitive to price perturbations in a large number of instances. The Hybrid PSO shows the smallest distributions.

[31] Each "box" shows the inter-quartile range of the data. The line inside the box is the median value. The "whiskers" are 1.5 times the inter-quartile range in length. The individual points outside of the whiskers are individual "outliers" defined as points greater than $q_3 + 1.5(q_3 - q_1)$ or less than $q_1 - 1.5(q_3 - q_1)$.

These results need to be qualified because I have the distribution only in those cases where the algorithms converge. Since I know that some algorithms have a high rate of convergence failure, the relatively small interquartile ranges that I see see here are not representative of the true distribution of outcomes.

### 1.6.3 Sensitivity to initial values

Another robustness measure is the sensitivity of the optimization algorithm for the changes in the initial values. I randomly changed initial value of each NSS parameter using using normally distributed perturbations with mean zero and the standard deviation of the observed week-to-week changes of these parameters observed in full sample tests. I next fit the term structures using each optimization method and then generate three-month, two-year, and 10-year spot rates from the fitted term structures. The spot rates generated with the original starting values are then subtracted from the perturbed starting value results. Figure 19 presents these distributions each of the three spot rates and for each of the eight optimization methods. The results are similar to price perturbation tests.

**Table 7:** Total RMSE and MAE when one half of the data is chosen randomly and NSS model is fitted to the data and the same process is applied to other half of the data, and vice versa. Goodness-of-fit statistics are combined for the two hold-out samples.

| | **Total RMSE** | | | | **Total MAE** | | | |
|---|---|---|---|---|---|---|---|---|
| **Gradient Based Algorithms** | | | | | | | | |
| **Method** | **Brazil** | **Mexico** | **Turkey** | **U.S.** | **Brazil** | **Mexico** | **Turkey** | **U.S.** |
| BFGS | 0.994 | 0.903 | 0.912 | 0.878 | 0.591 | 0.617 | 0.540 | 0.639 |
| GRG | 0.299 | 0.247 | 0.309 | 0.281 | 0.272 | **0.134** | 0.182 | 0.175 |
| **Direct Search Algorithms** | | | | | | | | |
| Nelder-Mead | 0.293 | 0.278 | 0.302 | 0.272 | 0.256 | 0.210 | 0.239 | 0.219 |
| Powell | 0.347 | 0.218 | 0.259 | 0.283 | 0.292 | 0.244 | 0.287 | 0.293 |
| **Global Optimization Algorithms** | | | | | | | | |
| SA | 0.234 | 0.237 | 0.248 | 0.271 | 0.171 | 0.188 | 0.199 | 0.197 |
| PSO-W | 0.227 | 0.184 | 0.213 | 0.259 | 0.176 | 0.155 | 0.162 | 0.173 |
| PSO-G | 0.255 | 0.215 | 0.249 | 0.261 | 0.196 | 0.172 | 0.193 | 0.189 |
| Hybrid PSO | **0.219** | **0.180** | **0.199** | **0.208** | **0.144** | 0.149 | **0.160** | **0.152** |

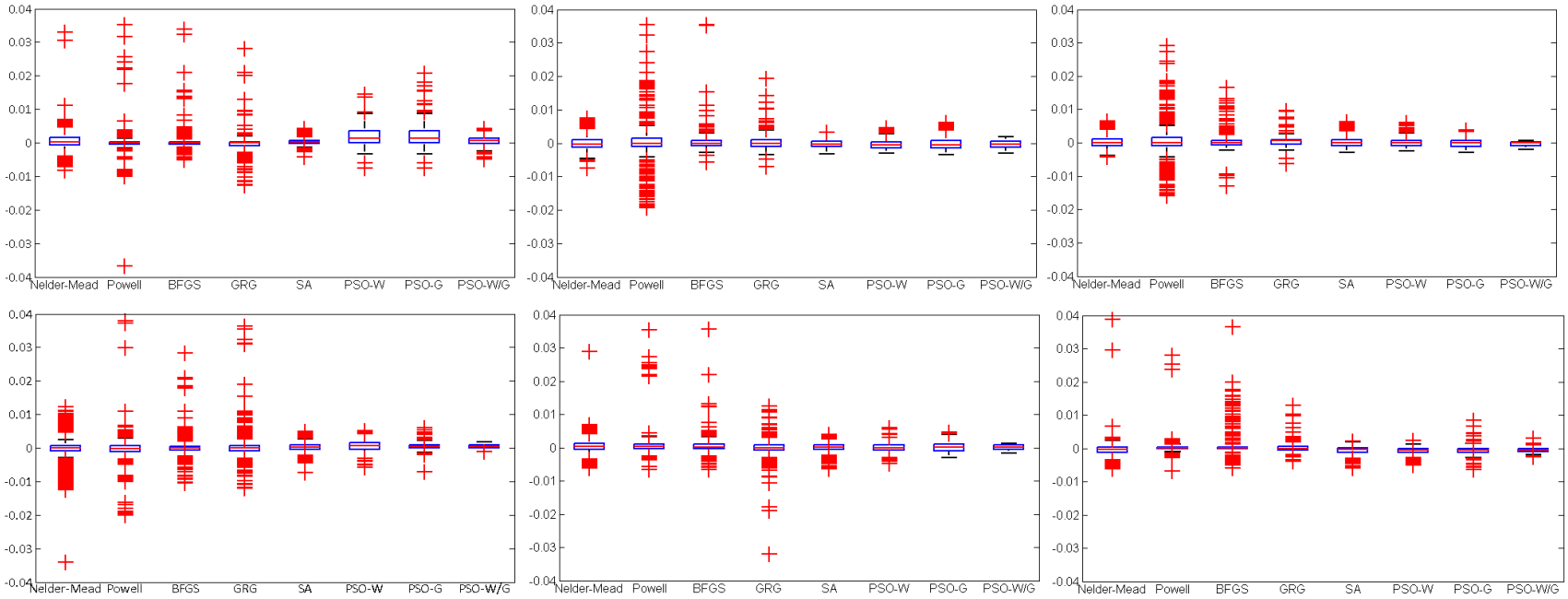


**Figure 19:** Price and starting values perturbation results of Brazil for 3 month, 2 year and 10 year spot rate respectively.

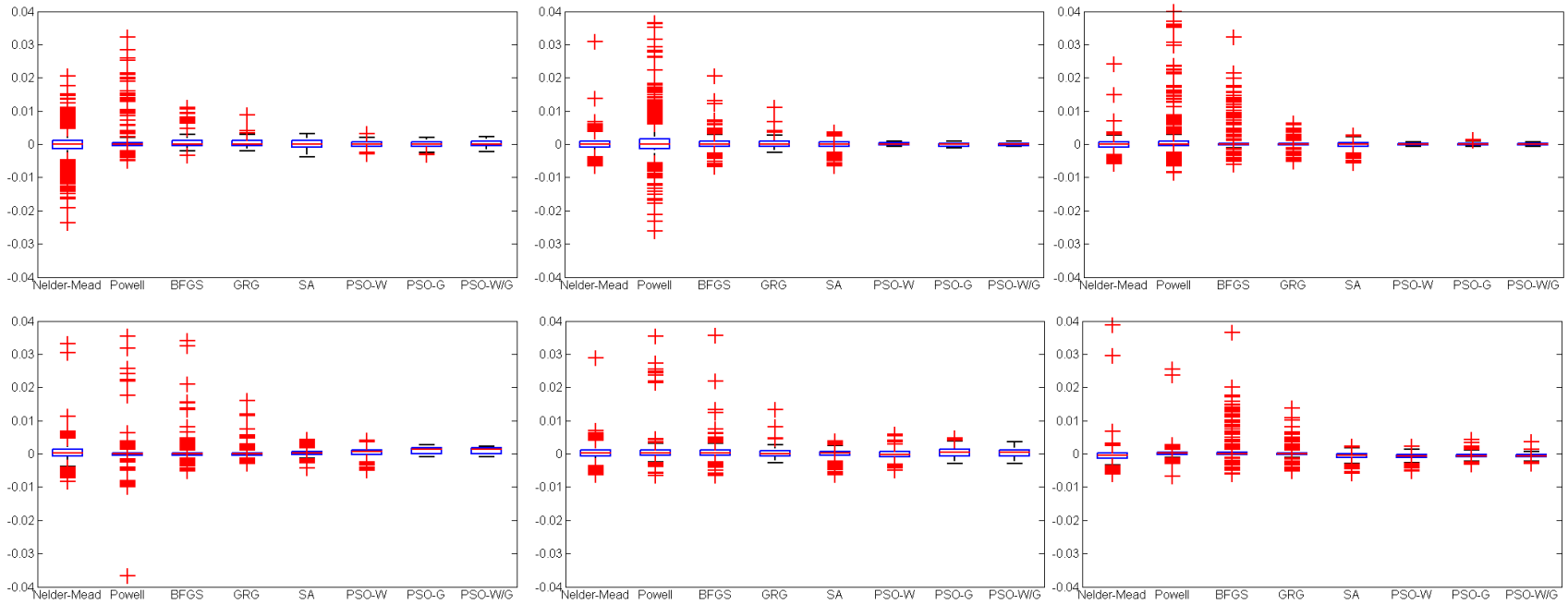


**Figure 20:** Price and starting values perturbation results of Turkey for 3 month, 2 year and 10 year spot rate respectively.

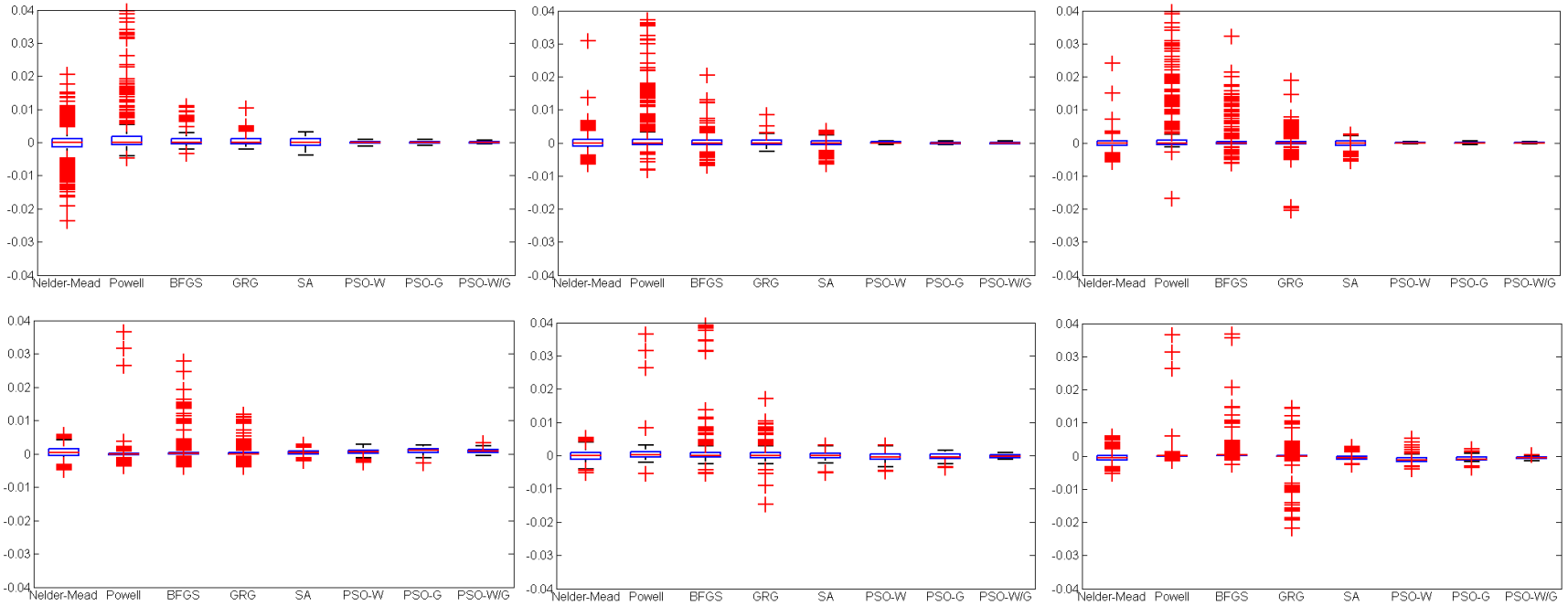


**Figure 21:** Price and starting values perturbation results of Mexico for 3 month, 2 year and 10 year spot rate respectively.

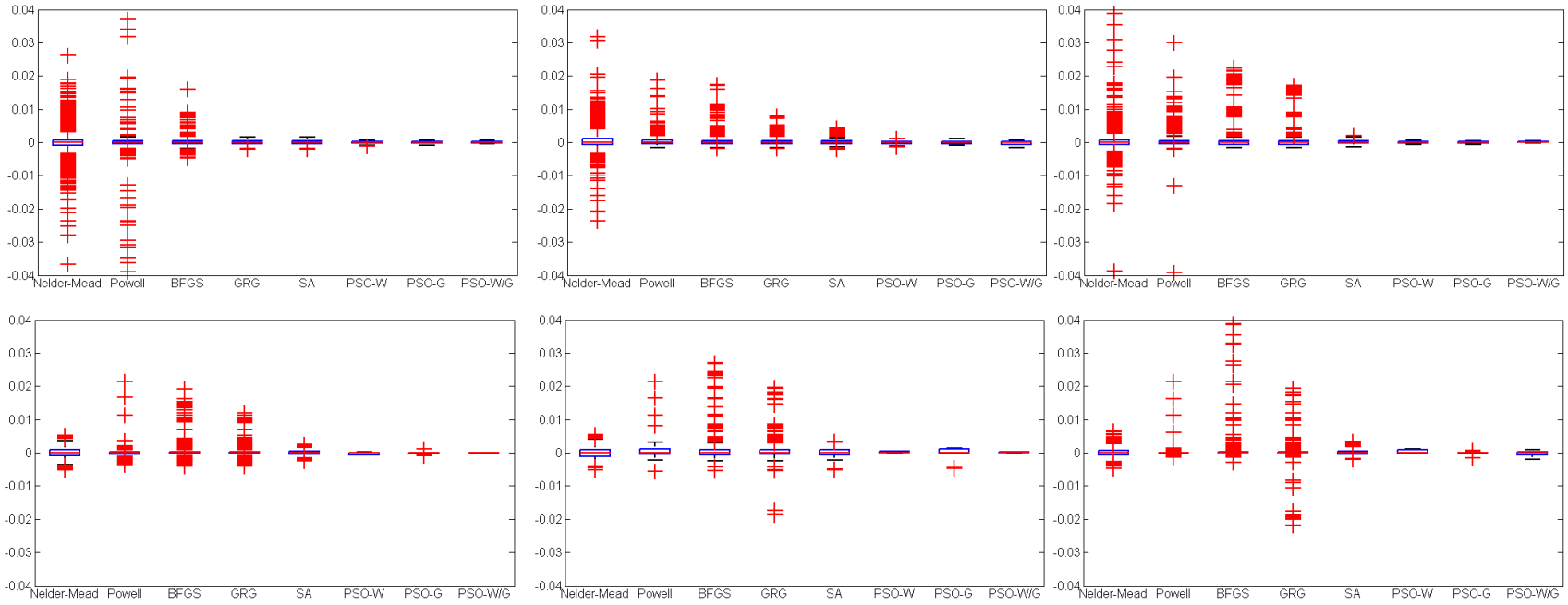


**Figure 22:** Price and starting values perturbation results of U.S. for 3 month, 2 year and 10 year spot rate respectively.

### 1.6.4 Local Currency Bonds

A foreign investor has two options when it comes to investing in emerging market debt. The first option is to invest in dollar-denominated debt (eurobonds as explained in the previous sections) and the second type of emerging debt is bonds that are denominated in local currencies. During the last decade, local currency bonds in emerging market economies were looked over by investors. Nowadays, these bonds are getting more attention as time passes. However, the liquidity of these local currency bonds are still low and they have a more sparse structure which can be depicted in 23. Being more challenging to estimate, it would be a perfect laboratory for us to test the optimization algorithms. I have used weekly mid-price data for the fixed coupon paying local currency bonds of Brazil, Mexico, and Turkey in the time period from July 2005 to December 2015, retrieved from Bloomberg (2010) and Datastream.

Similar to the eurobond results, goodness-of-fit statistics given in 8 show that global optimization algorithms clearly outperform gradient based and direct search algorithms. Among global optimization algorithms, PSO-W/G algorithm achieves the smallest error both in terms of RMSE and MAE in comparison to the other algorithms. In Figures 24, 25, and 26 I have also provided sample yield curve fits for Turkey, Brazil and Mexico local currency yield curves which are estimated with the PSO-W/G algorithm. As it can be observed from the figures, the NSS model is also capable of replicating a variety of yield curve shapes for local currency bonds despite their sparse behavior.

Graphical representation of level, slope and curvature factors ($\beta_0$, $\beta_1$, $\beta_2$) of the best four optimization algorithms for Turkey, Brazil and Mexico local bonds are depicted in Figures 27, 28, and 29. Potential problem of parameter instability for highly

parametrized NSS functional form is obvious from the figures. Deviation from empirical proxies (RMSE) for three factors are given in table 8. The optimization algorithms tested differ greatly in their degree of parsimony. Results hold robust for Mexico, and Turkey local currency bond portfolios. Evolution of parameters for Brazil is given in Figures 30, 31, and 32. Parameter evolution for Mexico is depicted in Figures 33, 34, and 35.

As it can be observed from the results, global optimization algorithms are capable of generating smoother and more realistic level, slope, and curvature factors when compared to gradient based and direct search algorithms similar to eurobond results. I also tested the robustness of the algorithms for the local bond data. First, I accounted for the perturbation of bond prices by adding a random term to each bond price as a uniform random value in the interval of bid-ask spread. Second, I compared in-sample and out-of sample results. Third, I tested the sensitivity of the optimization algorithms to the initial values by randomly selecting the initial values. They are shown in 36, 38, and 37.

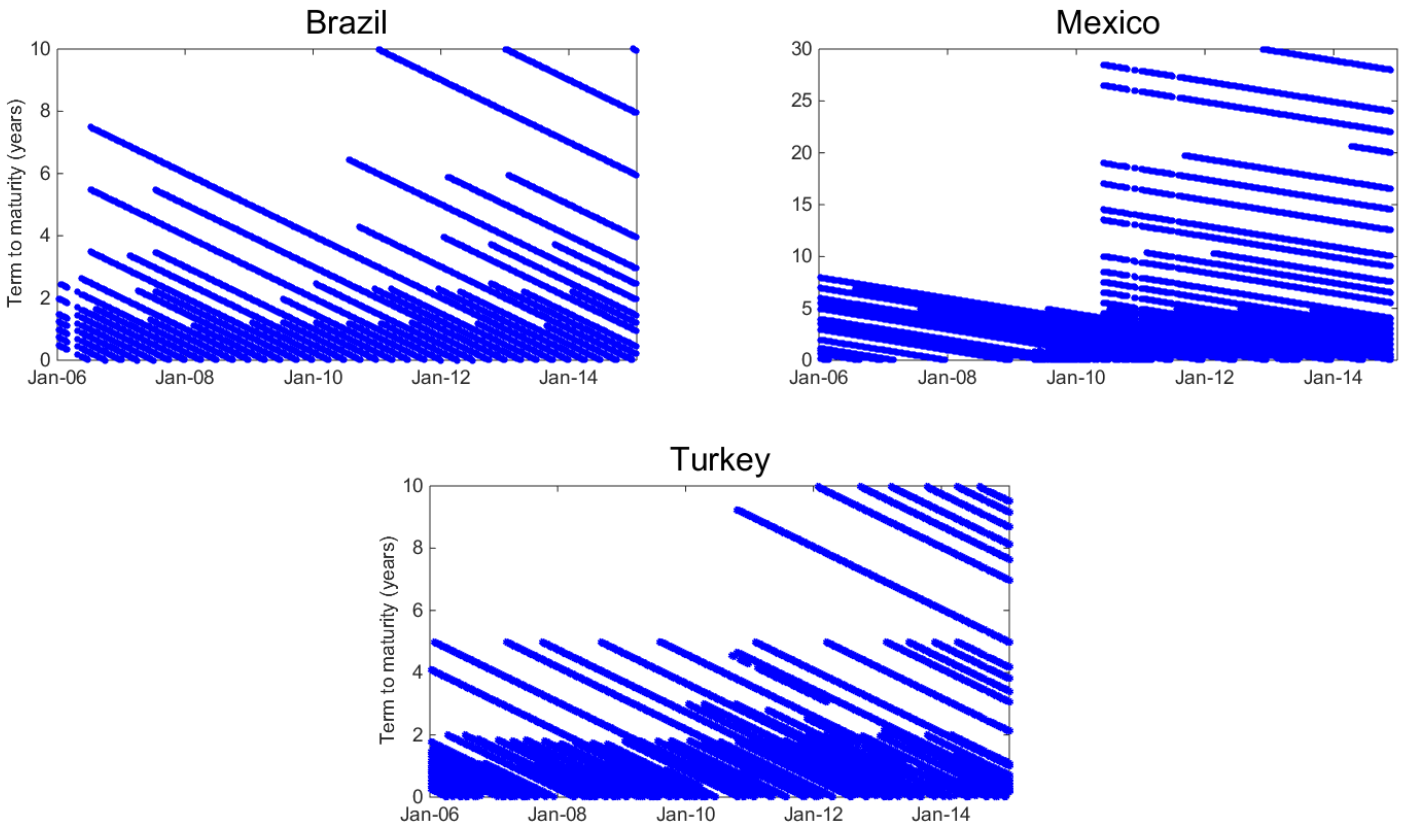


**Figure 23:** Maturities of local currency bonds used by observation data and country

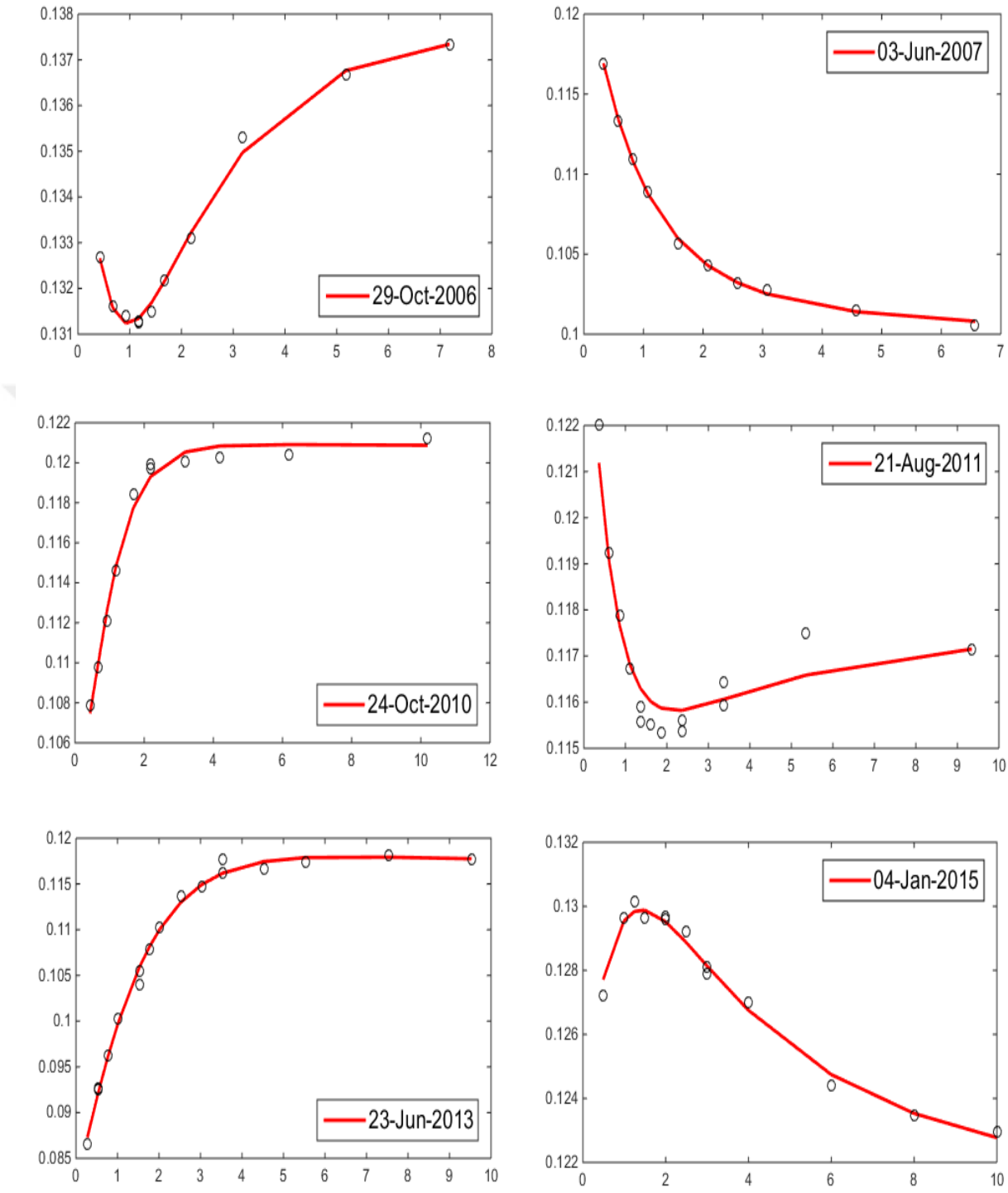


**Figure 24:** Selected fitted yield curves fitted to Brazilian local bond data using the hybrid PSO optimization method. Circles indicate spot rates obtained using a simple bootstrap applied to the same data.

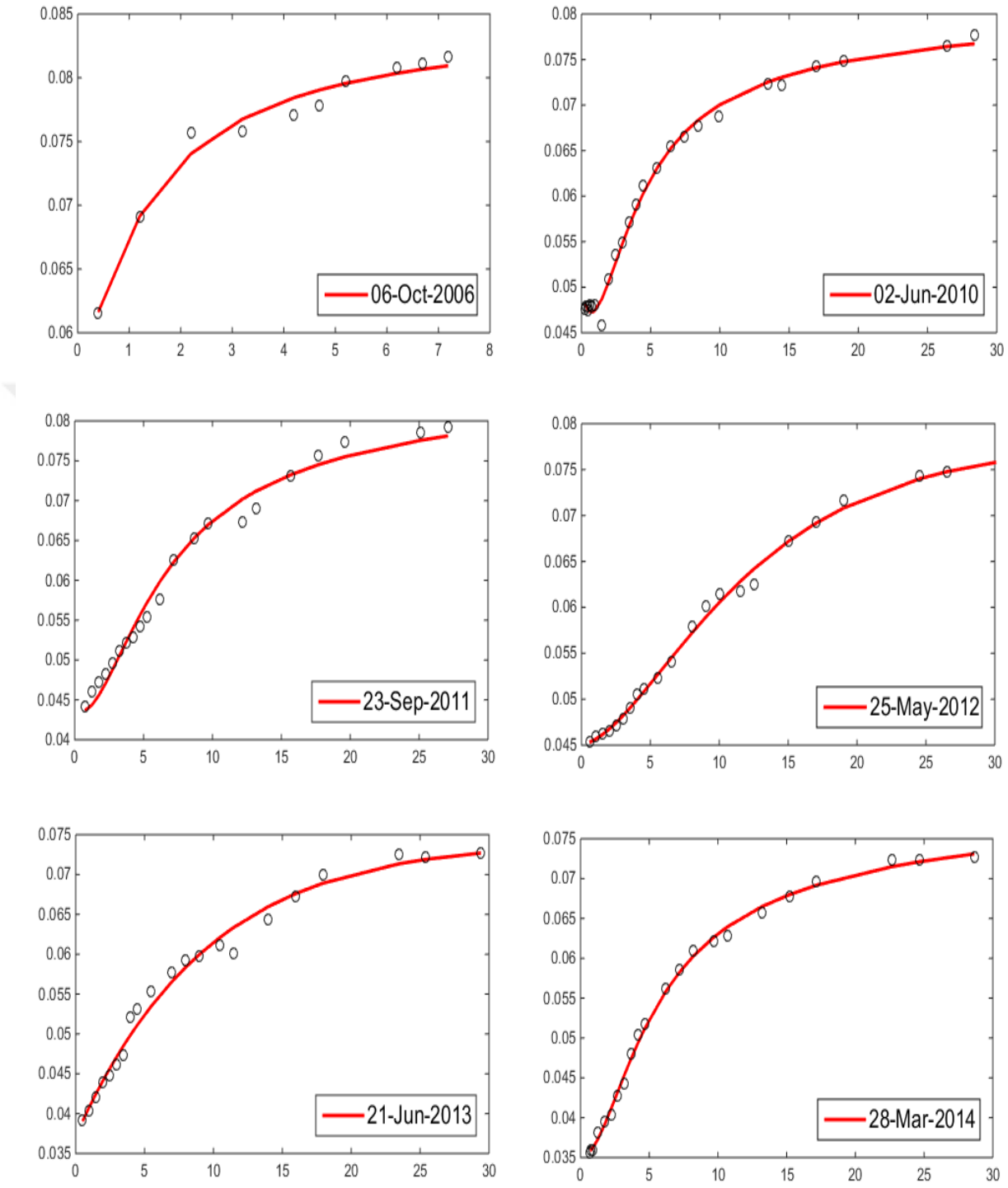


**Figure 25:** Selected fitted yield curves fitted to Mexican local bond data using the hybrid PSO optimization method. Circles indicate spot rates obtained using a simple bootstrap applied to the same data.

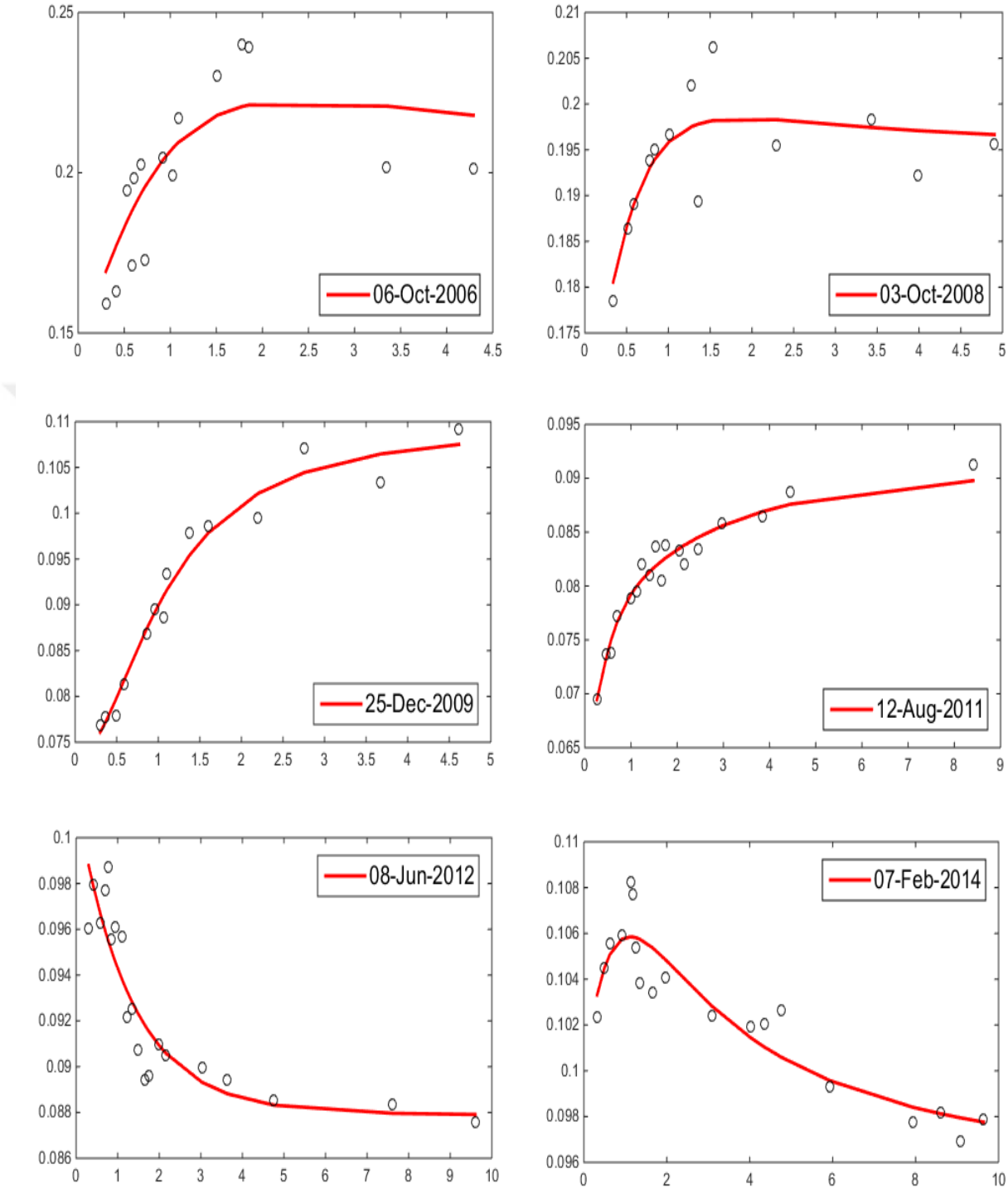


**Figure 26:** Selected fitted yield curves fitted to Turkish local bond data using the hybrid PSO optimization method. Circles indicate spot rates obtained using a simple bootstrap applied to the same data.

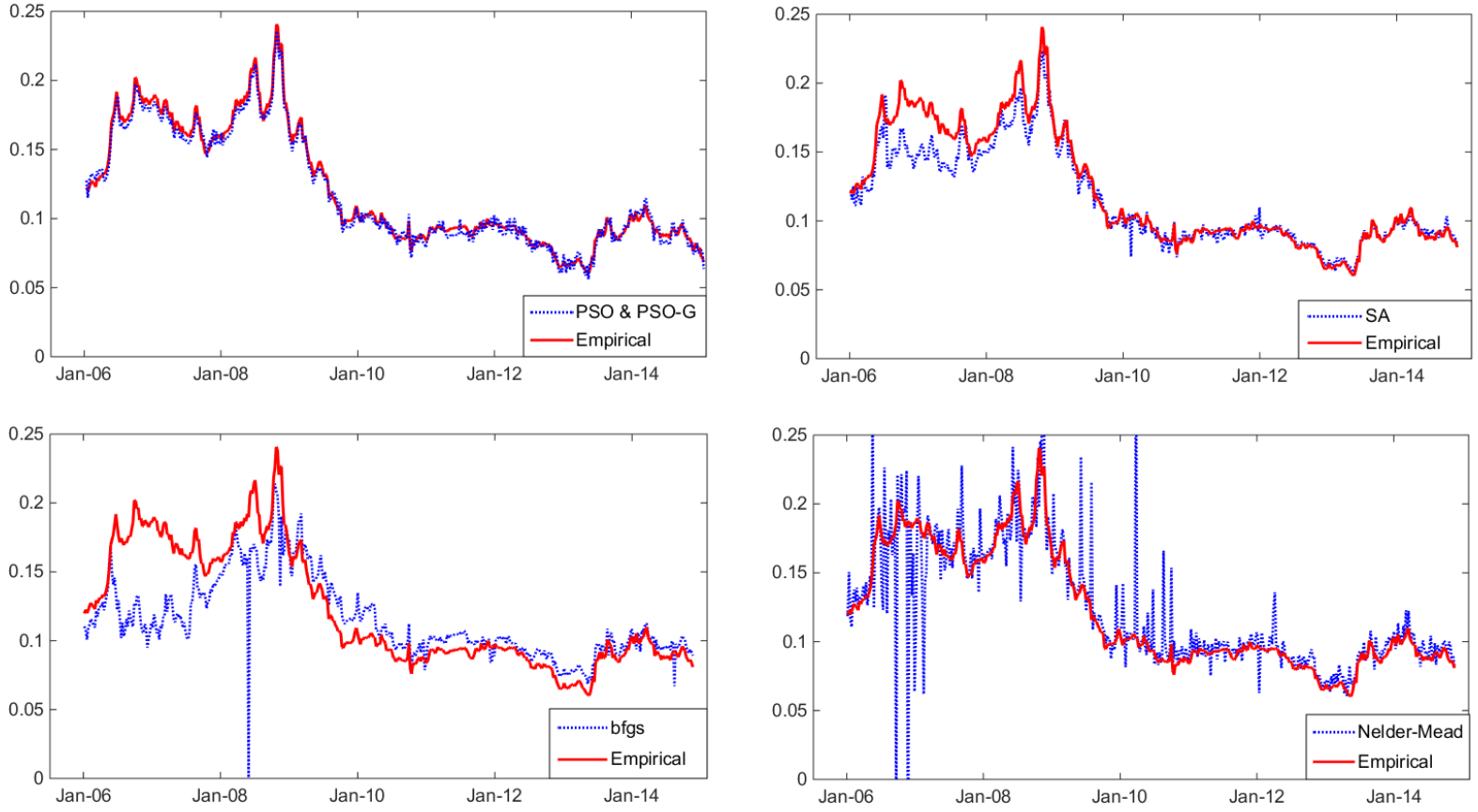


**Figure 27:** Evolution for level ($\beta_0$) in Turkey local bonds

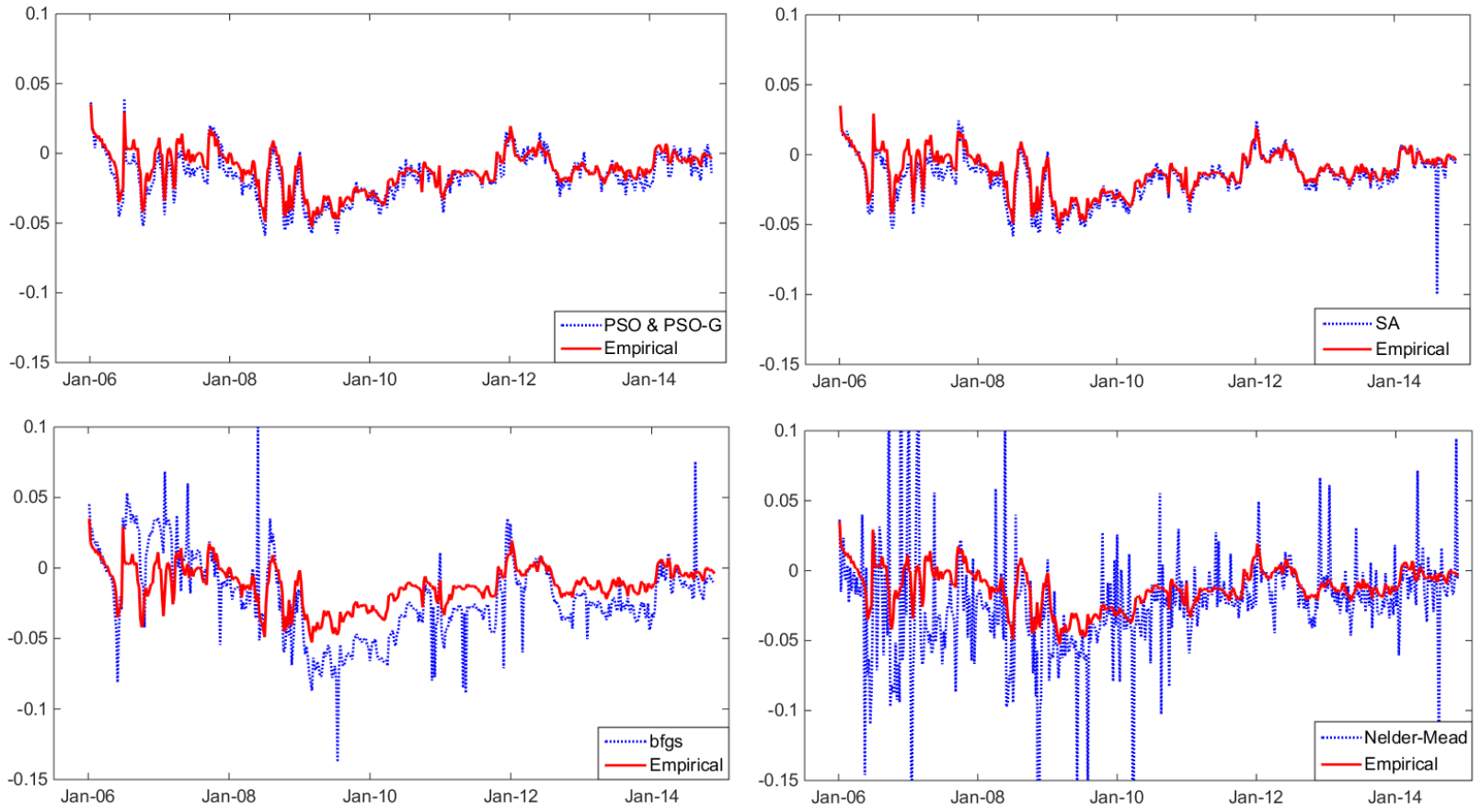


**Figure 28:** Evolution for slope ($\beta_1$) in Turkey local bonds

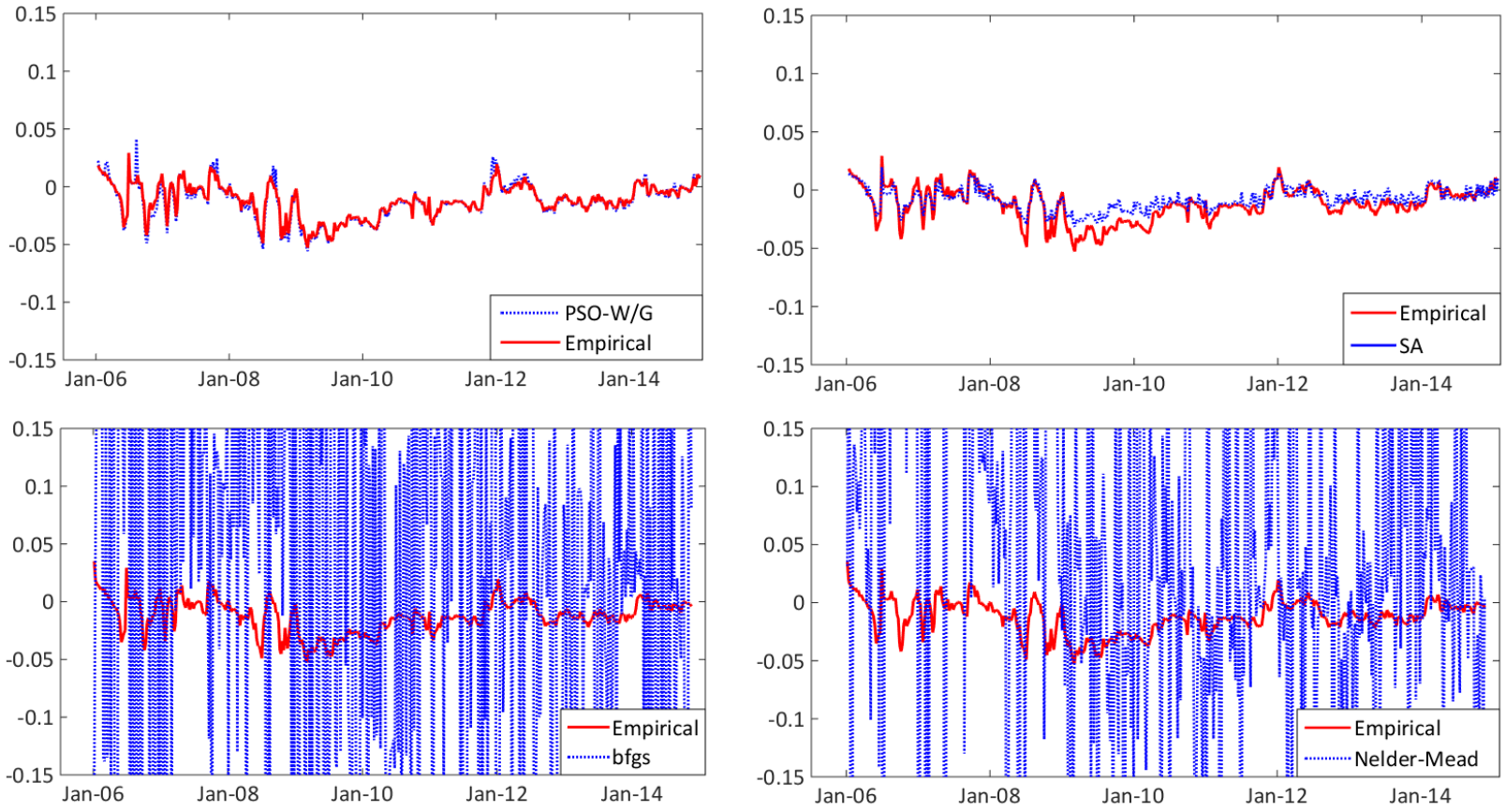


**Figure 29:** Evolution for curvature ($\beta_2$) in Turkey local bonds

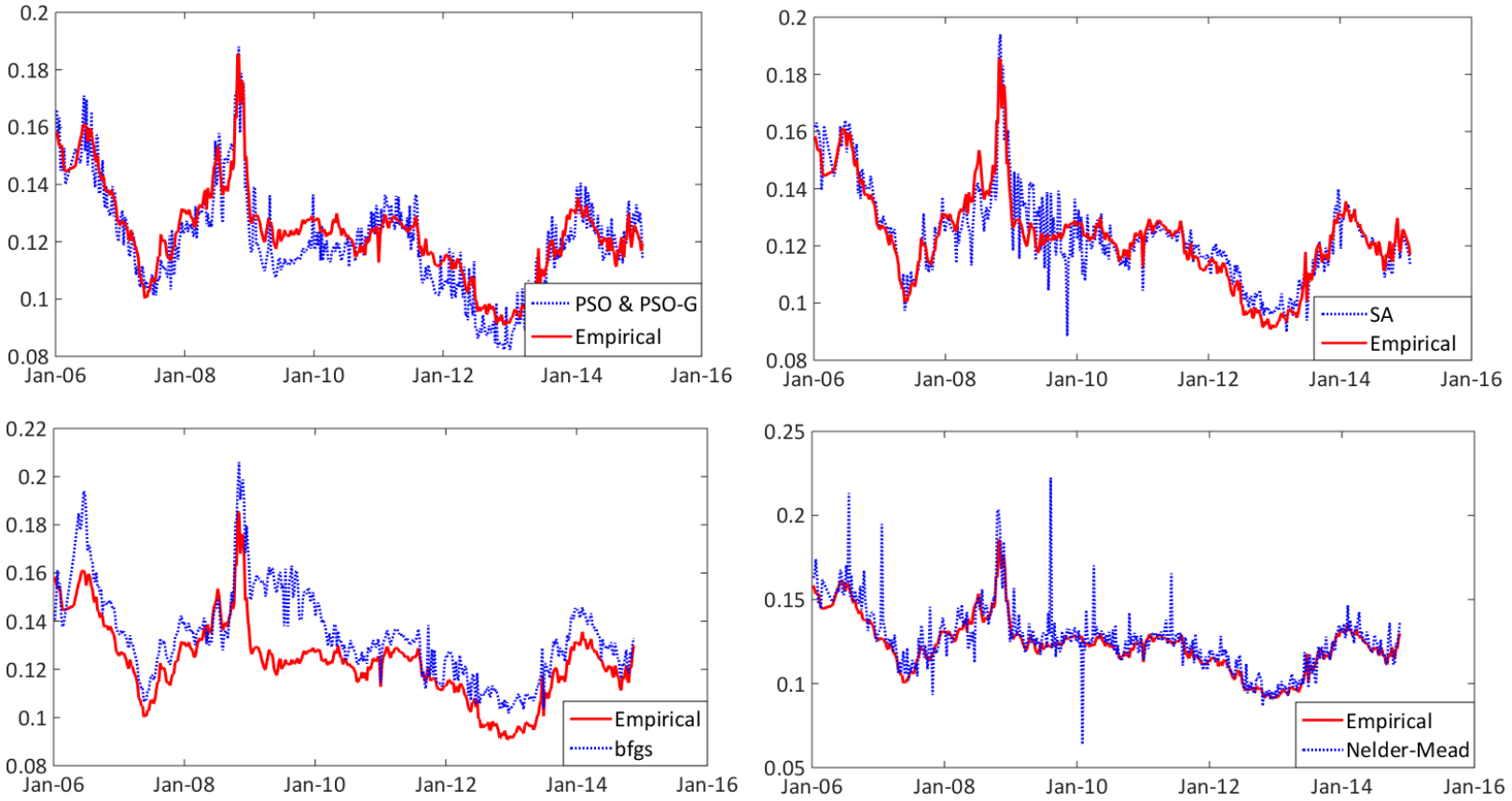


**Figure 30:** Evolution for level ($\beta_0$) in Brazil local bonds

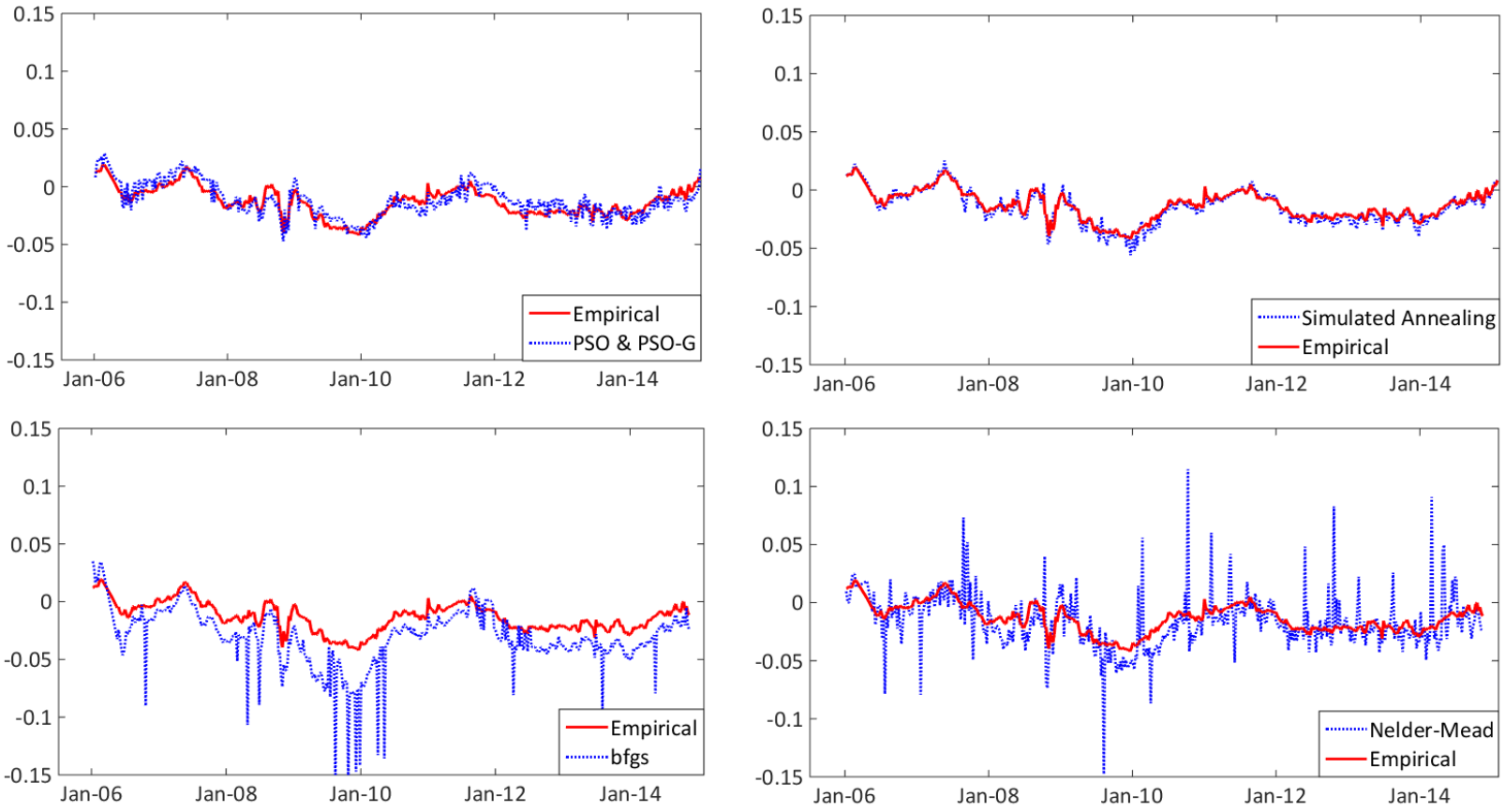


**Figure 31:** Evolution for slope ($\beta_1$) in Brazil local bonds

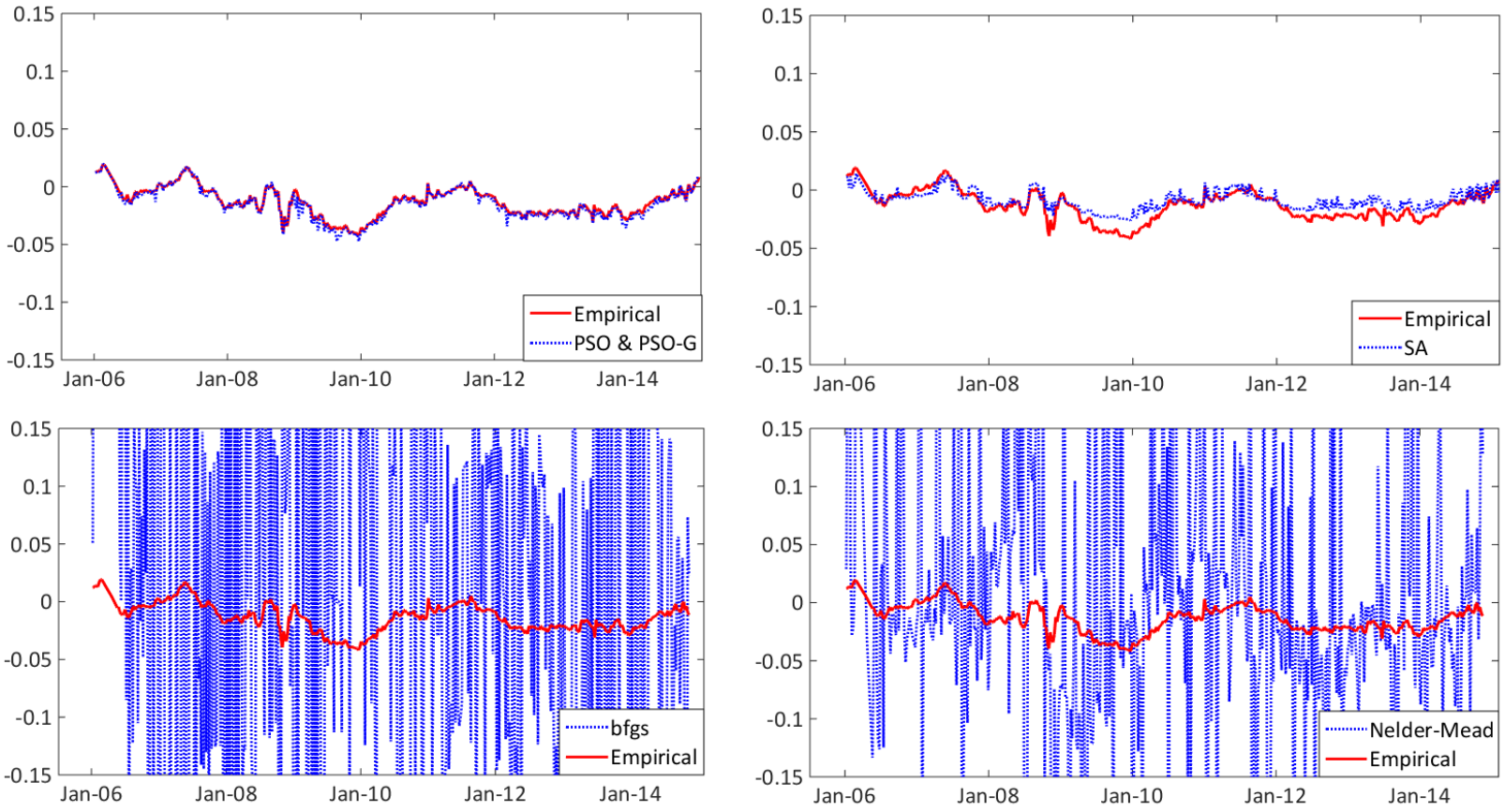


**Figure 32:** Evolution for curvature ($\beta_2$) in Brazil local bonds

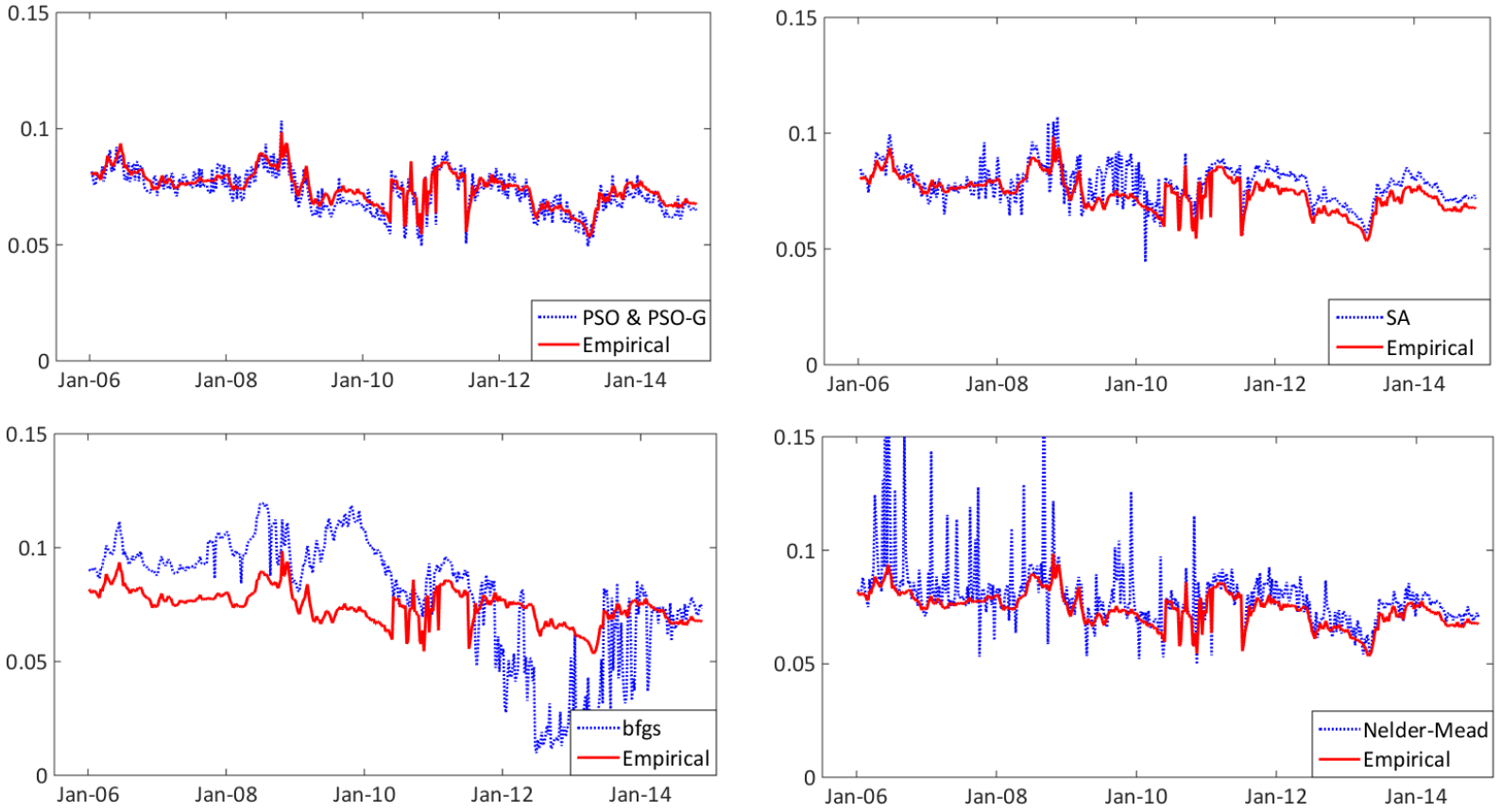


**Figure 33:** Evolution for level ($\beta_0$) in Mexico local bonds

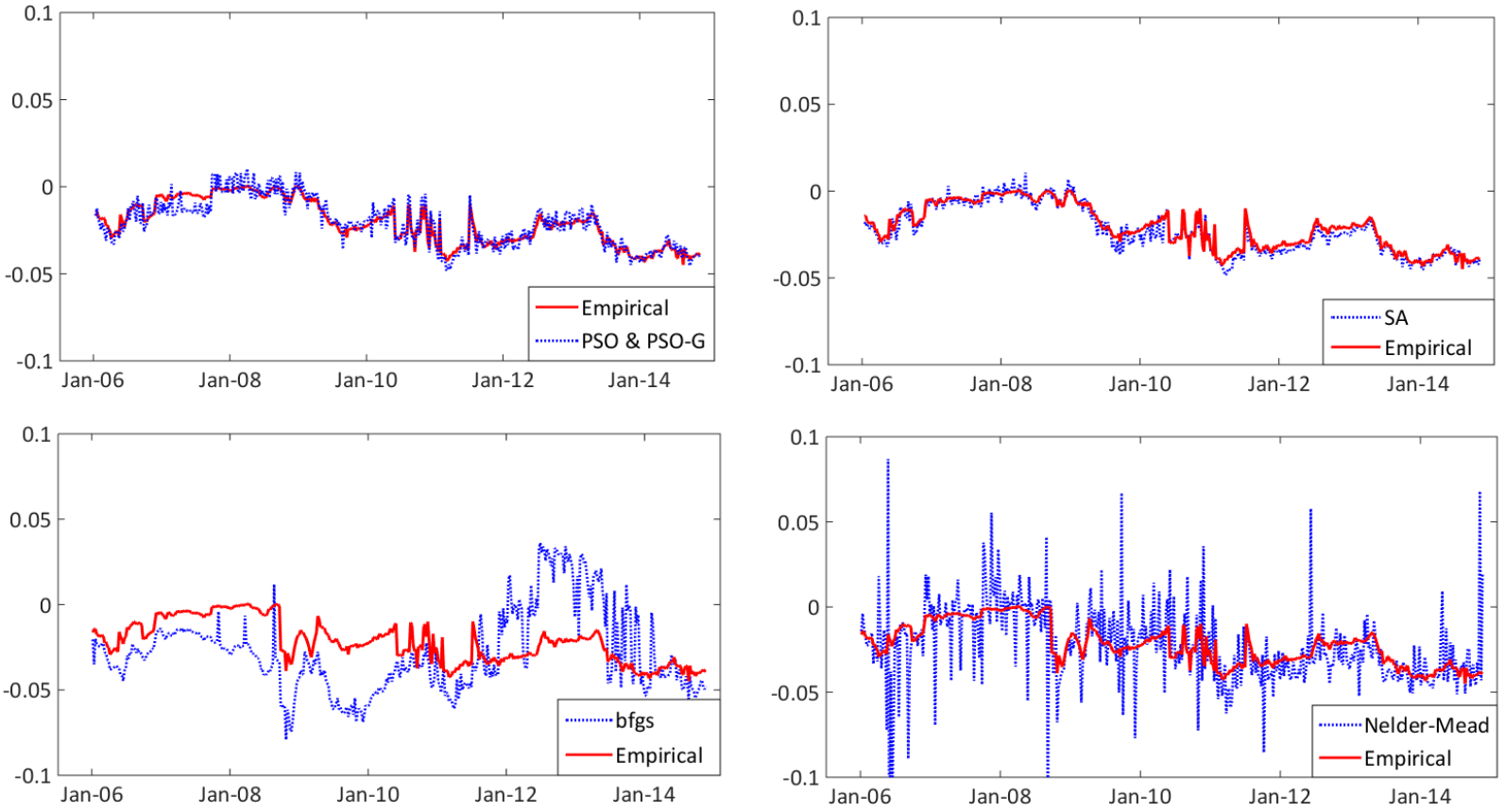


**Figure 34:** Evolution for slope ($\beta_1$) in Mexico local bonds

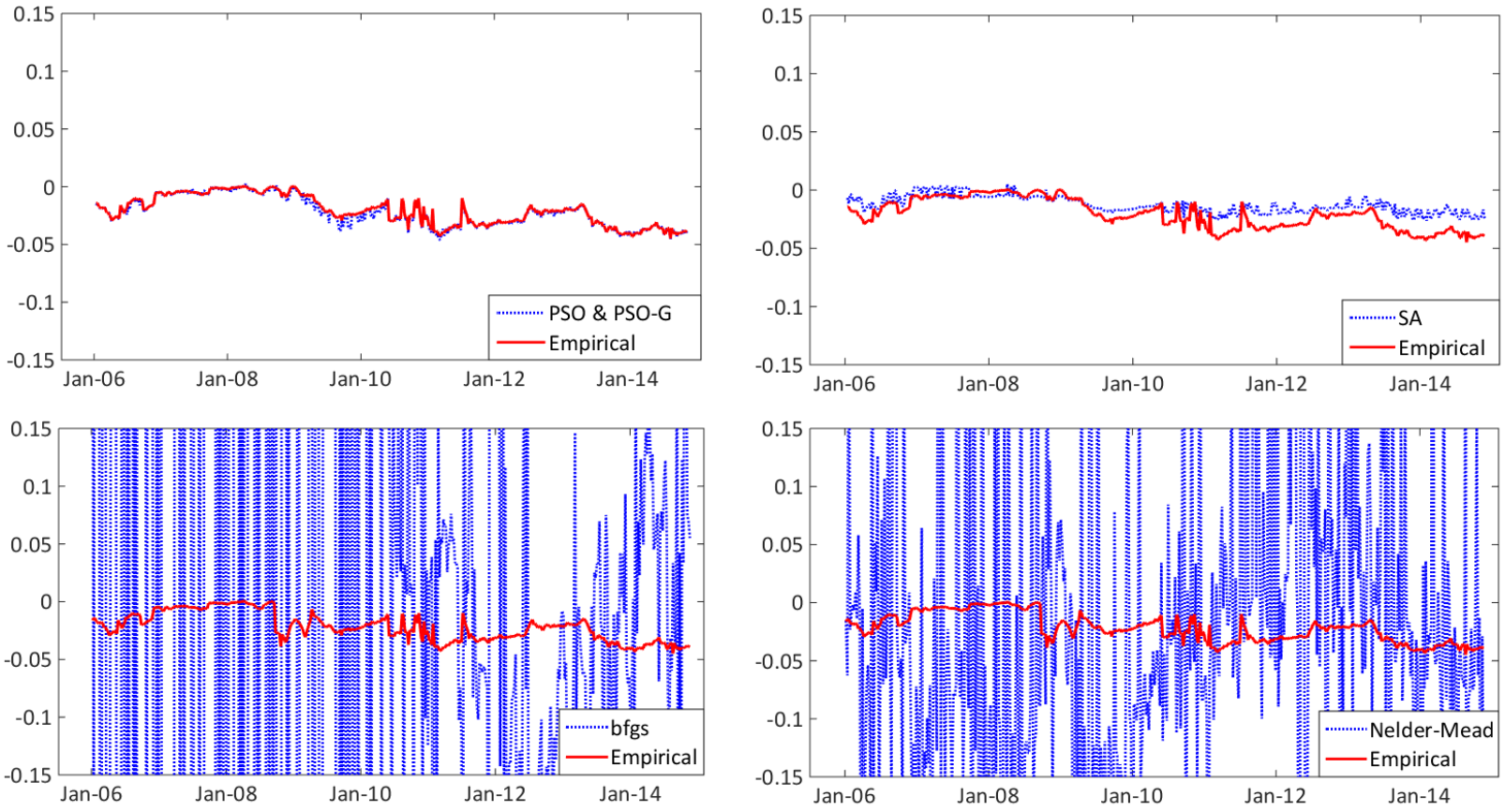


**Figure 35:** Evolution for curvature ($\beta_2$) in Mexico local bonds

**Table 8:** Table summarizes the comparison of optimization algorithms in terms of total RMSE and MAE using local bond data.

| | Total RMSE | | | | | Total MAE | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Gradient Based Algorithms** | | | | | | | | | | |
| **Method** | **Brazil** | **India** | **Mexico** | **SOAF** | **Turkey** | **Brazil** | **India** | **Mexico** | **SOAF** | **Turkey** |
| BFGS | 1,49 | 0,81 | 1,48 | 1,33 | 2,29 | 1,13 | 0,66 | 1,17 | 1,08 | 1,64 |
| GRG | 2,38 | 1,13 | 2,31 | 1,50 | 3,80 | 0,87 | 0,98 | 2,28 | 0,61 | 0,44 |
| **Direct Search Algorithms** | | | | | | | | | | |
| NM | 0,40 | 0,62 | 1,51 | 0,77 | 1,46 | 0,33 | 0,49 | 1,15 | 0,55 | 1,11 |
| Powell | 1,20 | 0,93 | 4,80 | 4,35 | 6,38 | 1,09 | 0,81 | 4,38 | 4,21 | 6,15 |
| **Global Optimization Algorithms** | | | | | | | | | | |
| PSO-W | 0,27 | 0,55 | 1,71 | 0,65 | 1,52 | 0,22 | 0,44 | 1,34 | 0,44 | 1,14 |
| PSO-G | 0,30 | 55,45 | 1,62 | 23,01 | 1,58 | 0,24 | 54,29 | 1,27 | 22,33 | 1,19 |
| H-PSO | 0,32 | 0,51 | 1,47 | 0,63 | 1,67 | 0,26 | 0,40 | 1,12 | 0,42 | 1,30 |
| SA | 0,34 | 0,53 | 1,44 | 0,67 | 1,45 | 0,28 | 0,42 | 1,11 | 0,45 | 1,09 |

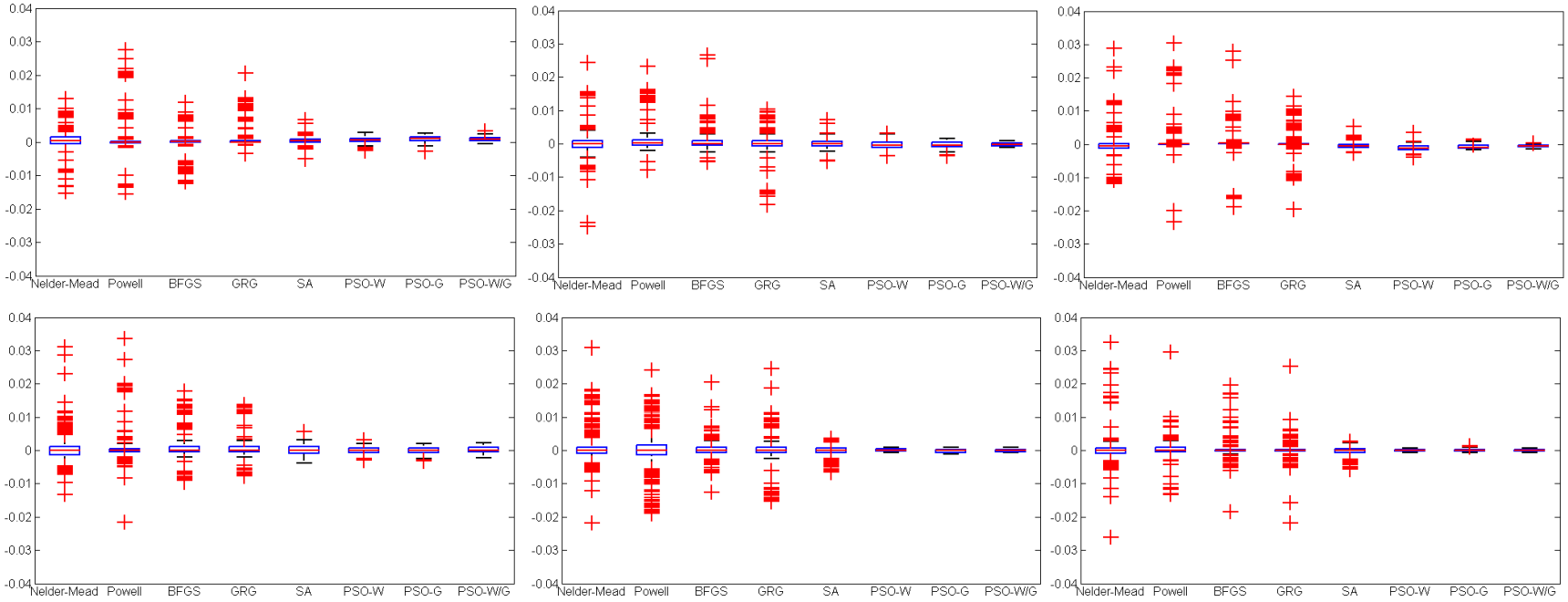


**Figure 36:** Price and starting values perturbation results of Brazilian local bonds for 3 month, 2 year and 10 year spot rate respectively.

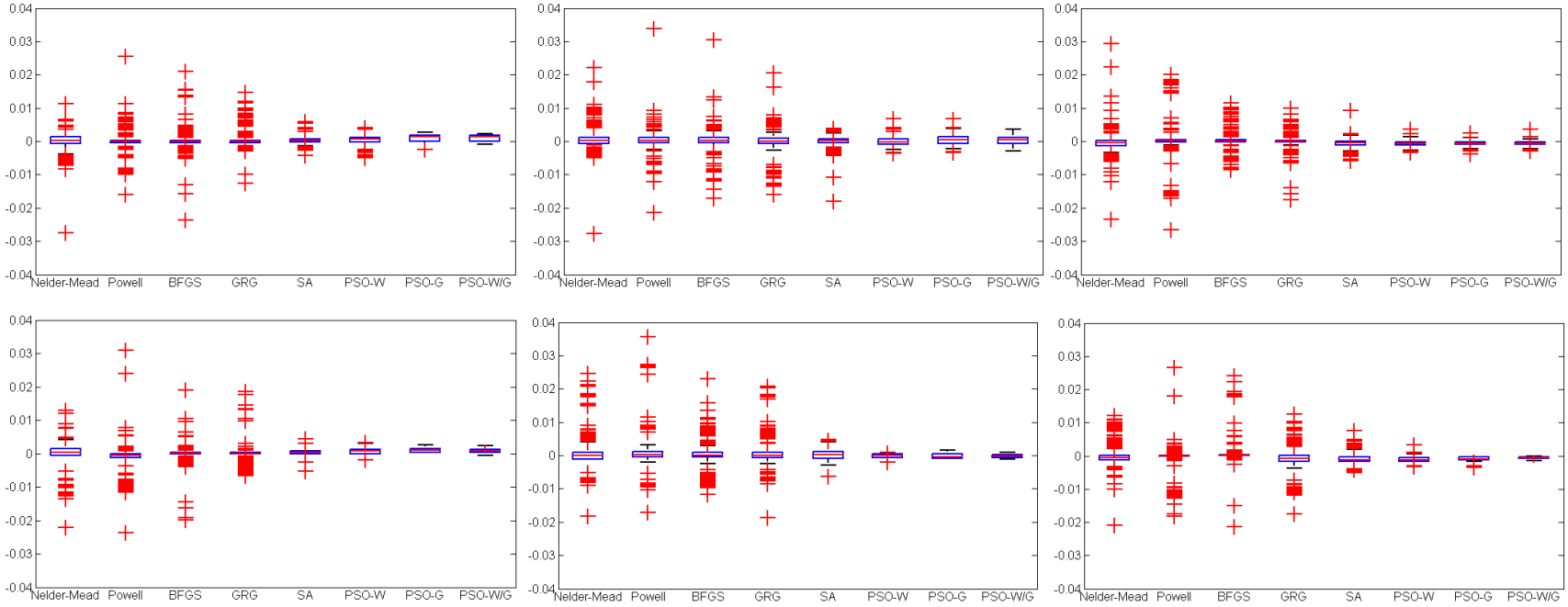


**Figure 37:** Price and starting values perturbation results of Turkish local bonds for 3 month, 2 year and 10 year spot rate respectively.

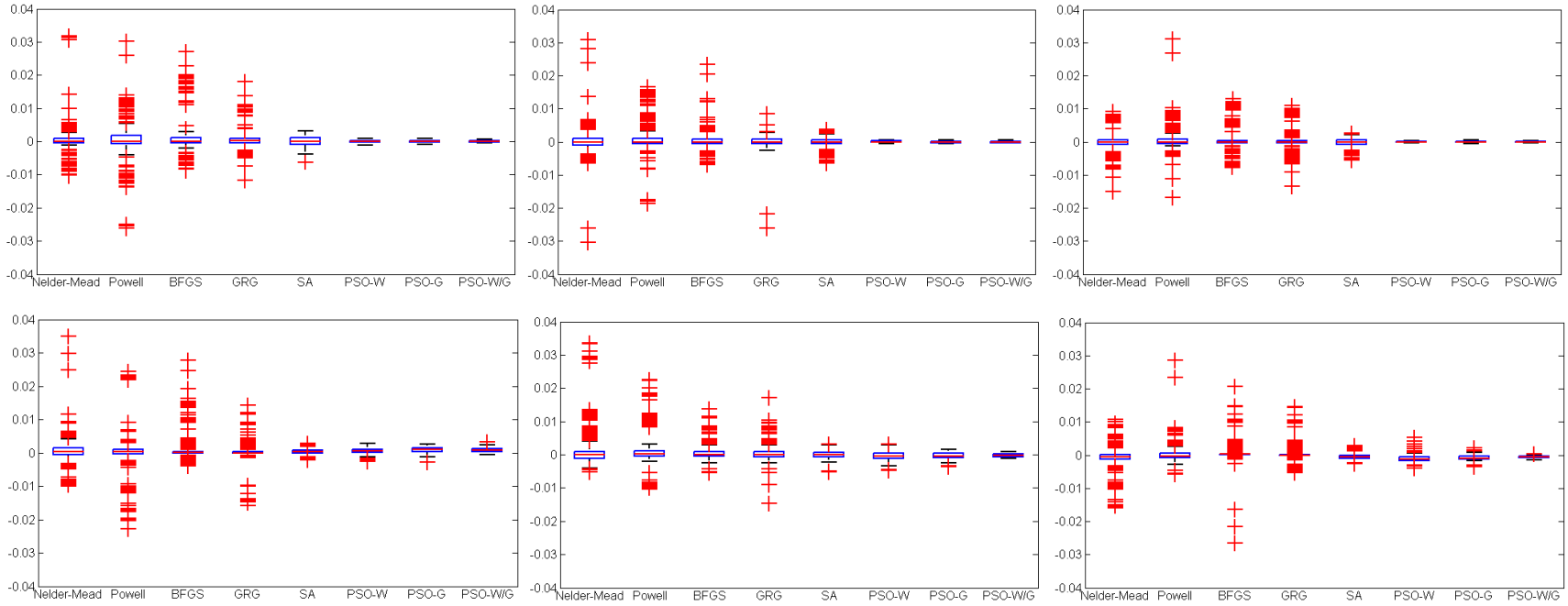


**Figure 38:** Price and starting values perturbation results of Mexican local bonds for 3 month, 2 year and 10 year spot rate respectively.

### 1.6.5 Simulated Data

In order to compare different optimization methods, they are applied to identical term structure estimation problem by setting the same starting values and same actual market data. However, in order to ensure that the above results are not an artifact of the actual datasets used, some simulation exercises (repeated a number of times K) with known data generating processes are needed. By simulating data and repeating this simulation exercise, robustness of the hybrid-PSO algorithm is checked to confirm that the superior performance is not due to original data sets used.

In this section, spot rate term structures are generated using a combination of Cox et al. (1985) and Vasicek (1977) models with assumed parameters which are chosen to generate various yield curves. For every cross-section, Vasicek model is used by changing the instantaneous rate, $r$, and the initial stochastic volatility, $V$. So that, I obtain different shapes of the term structure (upward sloping, flat or inverted). Instantaneous rate $r$ is simulated by CIR short-rate model.

The spot rates are calculated at 0.25 years and at 6-month intervals from 0.5 to 10 years. Two coupons are assumed at each maturity. The yields on the par coupon bonds with maturities of 0.25, 0.5, 1, 2, 3, 5, and 10 years are then put into the on-the-run estimation methods to generate estimated spot rates. From these estimated spot rates, I calculate estimated bond prices for all bonds. The error analysis consists of comparisons of the estimated rates and prices with the "true", or simulated, rates and prices. Afterward, I add a random error to the generated bond prices. The results obtained using the simulated data are similar to the eurobond and local bond results as it can be observed by 9 and 10.

#### *1.6.5.1 Vasicek Model*

Vasicek (1977) assumes that the term structure of interest rates has the dynamics described from the following SDE (namely Ornstein-Uhlencbeck process):

$$dr = \alpha(\gamma - r)dt + \rho dW_t \tag{26}$$

where the mean reversion rate $\alpha$, long-run mean $\gamma$ and volatility $\sigma$ are positive constants, and $W_t$ is a Brownian motion. [32] The price $P(t,T)$ of a bond is determined by using the probability distribution of the short rate path during the life of the bond. It is assumed that the market is efficient and the market price of risk due to $W_t$, denoted by $q$, is constant. According to the assumptions, Vasicek (1977) derives the following bond-pricing formula :

$$\begin{aligned} P(t,T) &= e^{A_\tau - B_\tau r} \\ \text{where } B_\tau &= \frac{1}{\alpha}(1 - e^{-\alpha\tau}) \\ \text{and } A_\tau &= (B_\tau - \tau)R(\infty) - \frac{\rho^2}{4\alpha}B_\tau^2 \\ R(\infty) &= \gamma + \frac{\rho q}{\alpha} - \frac{\rho^2}{2\alpha} \end{aligned}$$

Then, the yield curve is readily derived as :

$$R(t,\tau) = -\frac{ln(P(t,T))}{T-t} = \frac{B_\tau r - A_\tau}{\tau} \tag{27}$$

[32] For a more detailed information, please see Shimko (1992), Baxter and Rennie (1996) or Mikosch (1998)

Using the Vasicek model, the yield curve can take three category of shapes depending on the level of state variable (here it is $r$): upward-sloping (when $r$ is low), inverted (when $r$ is high) and slightly humped (when $r$ is close to $R(\infty)$). Figure 39 gives a sample of yield curves generated by Vasicek model. In this example, the long-run average short rate is 0.07, represented by the dashed line. The long-run mean is 0.5, and the volatility is 0.04. The market price of interest rate risk is selected as q = 1. The starting point of a curve represents the short rate, which in the Vasicek model is normally distributed.

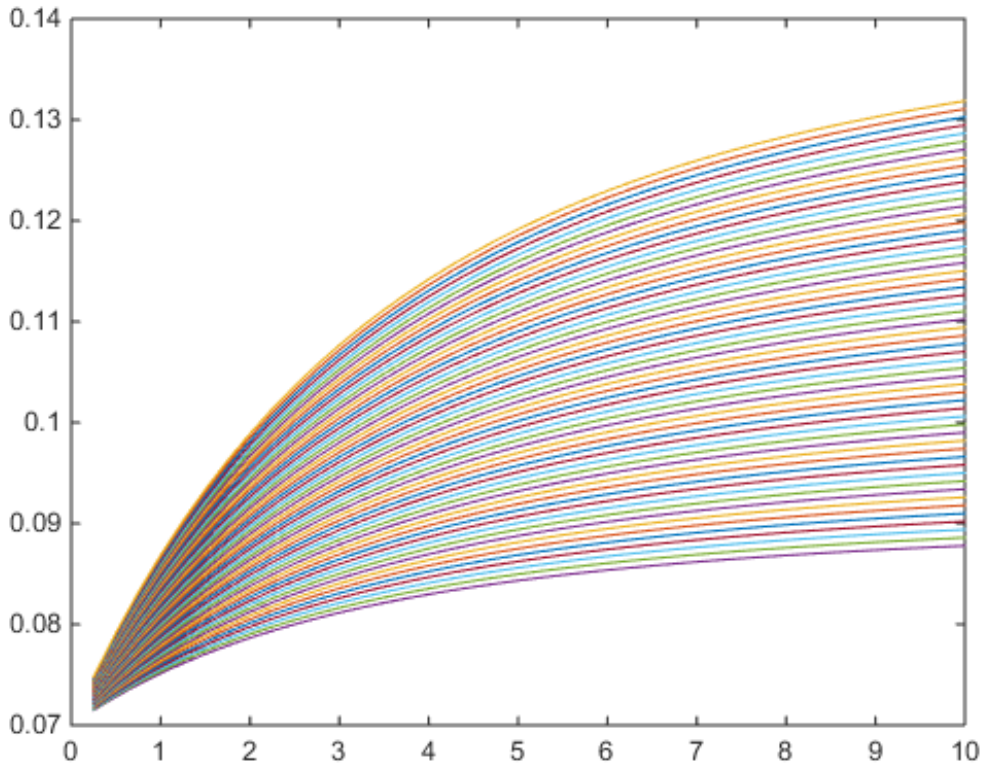


**Figure 39:** Yield curves generated by the Vasicek model at parameter values $\alpha = 0.5$, $\gamma = 0.04$, $\rho = 0.07$. for various values of r

The yield curves depicted in 39 are quite realistic since the model is subject to some observations. The first one is; volatility of long-term yields is unrealistically low, a realistic range cannot be achieved without allowing an excessively broad range for the short rate. Second one is that the rates on all maturities are perfectly correlated. This is inconsistent with what is actually observed. So in my data generation process, I randomly perturbed all the zero coupon yields in the bid-ask bound, after they are generated by the Vasicek model.

#### *1.6.5.2 CIR Model*

Cox et al. (1985) assumes the following SDE for the short rate process:

$$dr = \kappa(\theta - r)dt + \sigma\sqrt{r}dW_t \tag{28}$$

It is similar to Vasicek's, but the volatility of the short rate is proportional to the square root of its level, which implies that the short rate cannot go negative; the closer it gets to zero, the more its fluctuation abates, and the more the pull to mean dominates. The CIR model is derived from an underlying general equilibrium model of economic fundamentals, including a model of how investments in various stochastic production processes are turned into output; a model of how these production processes themselves develop over time; a model of how individual agents engage in investing, trading and consuming the production output, maximizing the expected value of an intertemporal utility function. From this equilibrium model, the authors derive a partial differential equation which all asset prices in the economy must satisfy. The solution of the equation depends on the functional forms and parameter values of the underlying economic model.

The general framework developed in Cox et al. (1985) specifies the functional forms for the underlying economic processes; from these, a dynamic term structure

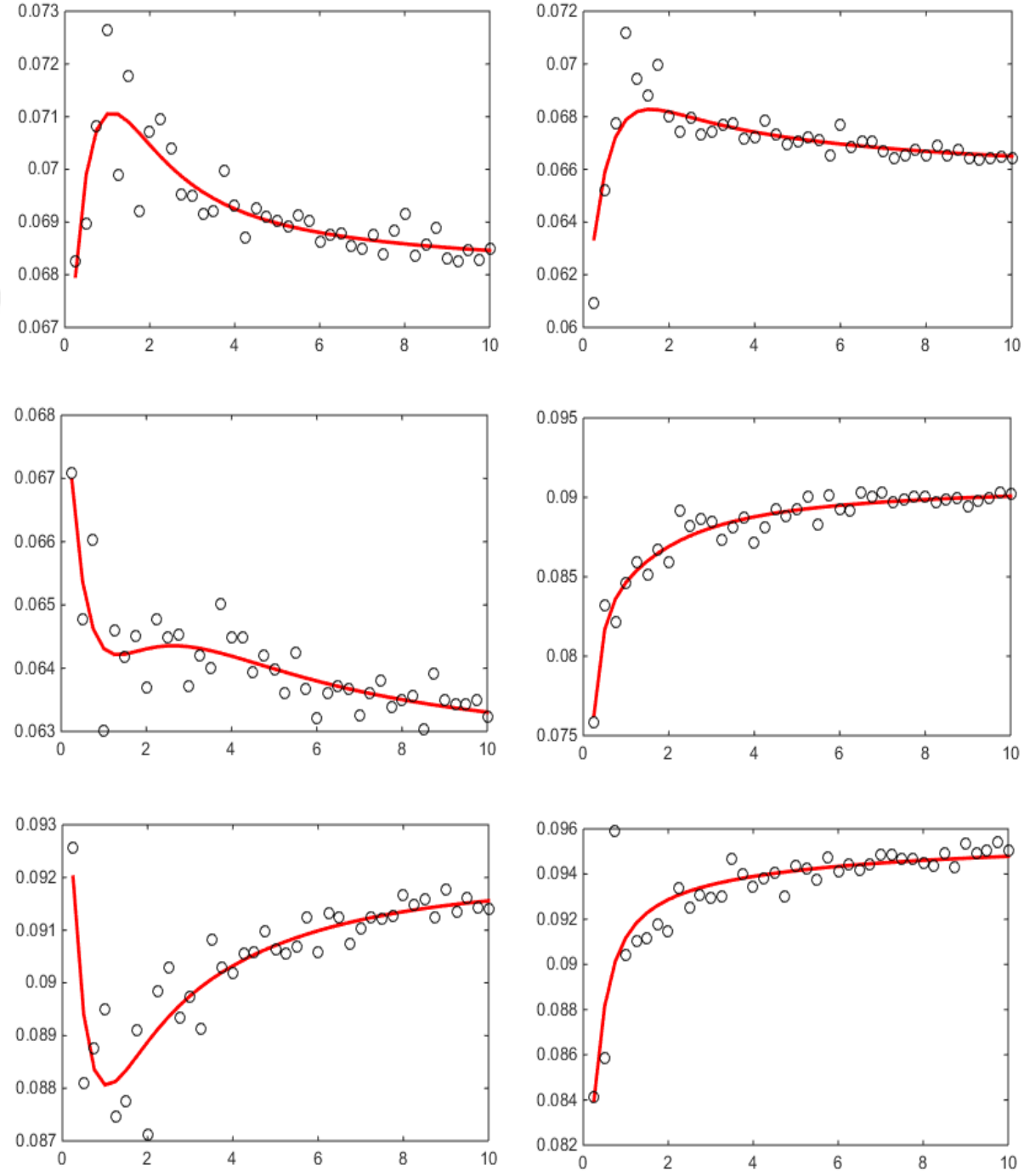


**Figure 40:** Selected fitted yield curves fitted to simulated bond yield data using the hybrid PSO optimization method. Circles indicate spot rates generated by Vasicek model.

model is derived. The resulting yield curves and their time-paths look very much like those produced by the Vasicek model, except that the distribution of rates is skewed upwards as a result of the introduction of $\sqrt{r}$ into the volatility term. Thus, the CIR model is more consistent with the facts that the conditional volatilities of yield changes are time-varying, persistent and positively correlated with the level of yields.

Figure 41 gives an illustration of various interest rate paths up to 1 year time period with parameter values $S = 0.4$, $L = 0.15$, $V = 0.05$. In my data generation process, consistently with CIR rate paths, yield curves are generated by the Vasicek model.

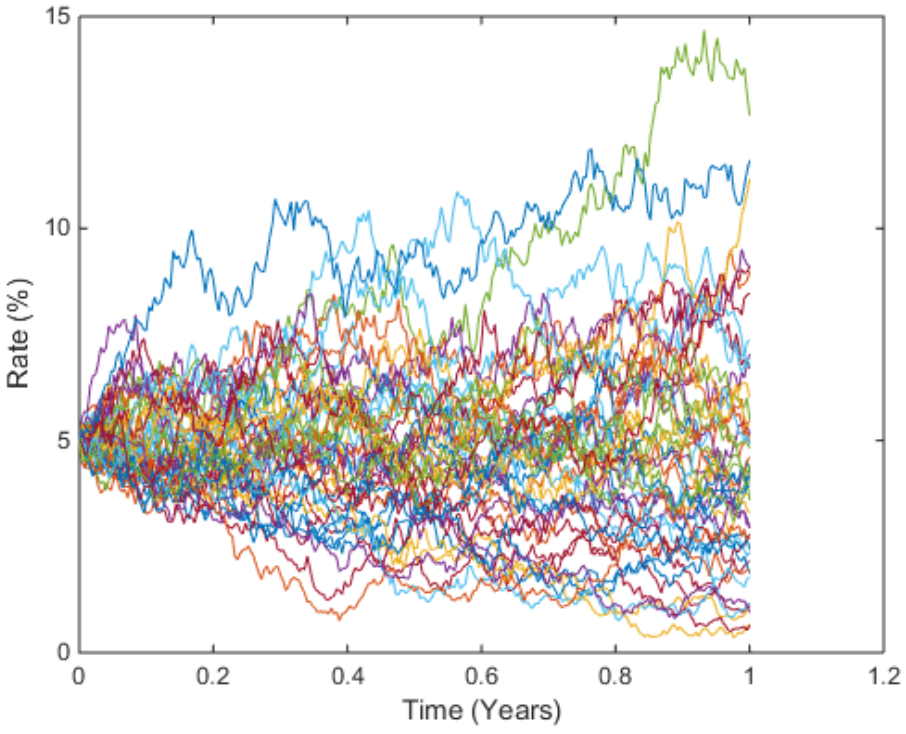


**Figure 41:** C.I.R. short rate paths at parameter values $S = 0.4$, $L = 0.15$, $V = 0.05$. Showing 100 paths, with one singled out.

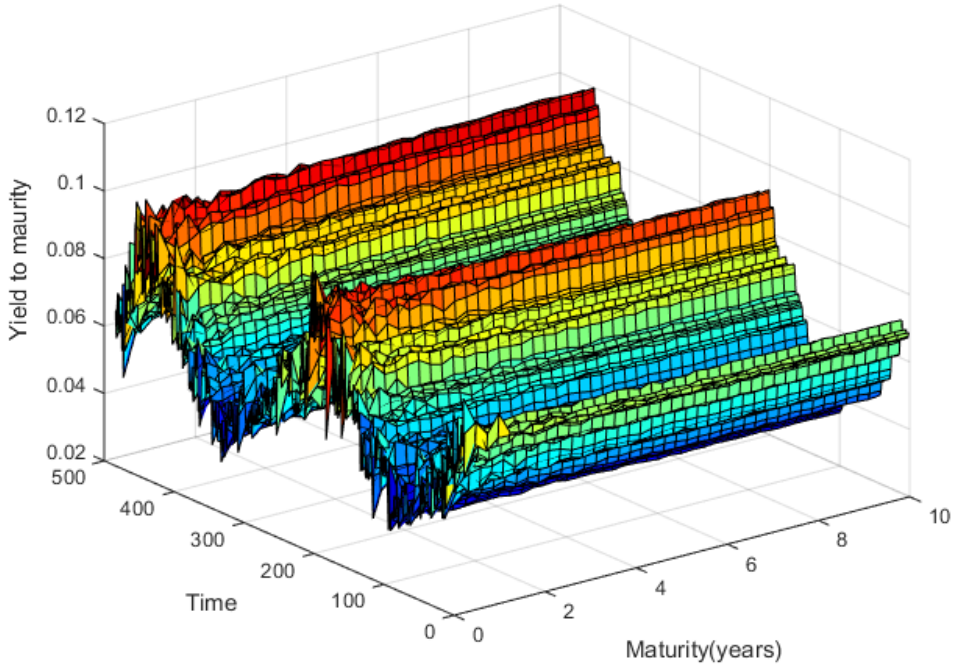


**Figure 42:** Graphical representation of yield curves generated consistently with Vasicek and CIR models

**Table 9:** Table summarizes the comparison of optimization algorithms in terms of total RMSE and MAE using simulated data

| Method | Total RMSE | Total MAE | CPU Time | Space Scanned |
|---|---|---|---|---|
| **Gradient Based Algorithms** | | | | |
| BFGS | 0,39 | 0,24 | 833,06 | 418,43 |
| GRG | 0,47 | 0,39 | 967,10 | 482,58 |
| **Direct Search Algorithms** | | | | |
| Nelder-Mead | 0,49 | 0,34 | 8,14 | 703,86 |
| Powell | 0,65 | 0,51 | 271,83 | 3429,63 |
| **Global Optimization Algorithms** | | | | |
| PSO-W | 0,37 | 0,22 | 435,04 | 1696,75 |
| PSO-G | 0,36 | 0,21 | 415,18 | 1267,29 |
| Hybrid PSO | 0,37 | 0,22 | 569,44 | 509,37 |
| SA | 0,40 | 0,24 | 715,37 | 96,97 |

**Table 10:** Deviation of three factors from their empirical proxies based on simulated data

| **Method** | $\beta_0$ | $\beta_1$ | $\beta_2$ | $\tau_1$ |
|---|---|---|---|---|
| **Gradient Based Algorithms** | | | | |
| BFGS | 0,15 | 0,23 | 5,36 | 73,60 |
| GRG | 0,21 | 0,25 | 7,80 | 74,98 |
| **Direct Search Algorithms** | | | | |
| Nelder-Mead | 0,33 | 0,56 | 6,10 | 18,49 |
| Powell | 0,09 | 205,76 | 221,18 | 45,77 |
| **Global Optimization Algorithms** | | | | |
| PSO-W | 0,03 | 0,09 | 0,31 | 2,13 |
| PSO-G | 0,06 | 0,09 | 0,36 | 2,09 |
| Hybrid PSO | 0,02 | 0,16 | 0,22 | 1,86 |
| SA | 0,05 | 0,08 | 0,05 | 10,57 |

## *1.7 Applied Application: Model Based Trading*

In an efficient market, a correctly specified term structure of interest rate estimation model is expected to exactly explain the observed bonds prices for all maturities. Thus, estimated interest rates should accurately explain the market prices of bonds in use. However, reported results show significant pricing errors e.g. Duarte et al. (2007), Jankowitsch and Nettekoven (2008), Svensson (1994), Subramanian (2001), and Jankowitsch and Pichler (2002) report significant pricing errors in their sample. Important question remained to be answered is whether the deviation of particular bond prices from estimated yield curve is caused by a model misspecification or or insufficient market efficiency.

In this section, I compare different methods of estimating the term structure of interest rates on weekly emerging bond data sample. I examined the Nelson and Siegel and Svensson functions, McCulloch's cubic spline, the linear, exponential and B-splines. Details of spline methods and estimation techniques can be found in the Appendix 4.1. I provide empirical evidence for the EM sovereign bond markets that risk-adjusted NSS based trading strategies based on bond pricing errors has a superior performance and can yield about 360 basis points p.a. abnormal return, implying that pricing errors contain economic information.

Among these term structure estimation models, linear splines have some fundamental issues. For instance, due to the fact of linear splines being continuous but not smooth, the yield curve slope may change at the knot points. The cubic spline form not only allows for a more flexible shape for the yield curve between knots,but also fixes the smoothness problem by imposing the continuity conditions on the curve as well as its first and second derivatives. In cubic spline estimation, each cubic equation is unique, so the entire curve is unique due to the constrained end points. Generally,

the yield curve is constrained to have a constant slope for the short-term, but the long end is close to flat. However, I found that cubic spline (hermite, B-splines) generate forward rates which are rather unstable which does not provide meaningful rich/cheap analysis because it would not be possible to tell if the valuations had changed due to a change in the model or a genuine change in the valuation of the bond. McCulloch (1971) solved this problem by introducing a quadratic polynomial to estimate the discount function. This approximation to the discount function is defined as a family of cubic functions which are object to being continuous and smooth around each knot. Details about the cubic functions can be found in the Appendix 4.1.3.

### 1.7.1 Bond Pricing

As explained in the first chapter, term structure estimation methods fall into two categories: function-based models and spline-based models. Among spline based models, I sticked to McCulloch and B-spline models due to their superior performance. Details and estimation of spline based models can be found in the Appendix 4.1. In this section, I give some details about theoretical background for bond pricing related to calculation of pricing errors.

The theoretical price $\hat{P}_{i,t}$ for a bond $i$ is the sum of individual cash flows. So, for a bond market with $n$ coupon bonds, each bond $i$ is characterized by its market price $P_i$ (quoted price plus accrued interest), its cash flow vector $C_i$ , and its vector of cash flow dates $T_i$ . The the market price $P_i$ is

$$P_{i,t} = \sum_{t=1}^{T_i} C_i(t) e^{-r_i(t)t} + \epsilon_i \tag{29}$$

where $\epsilon_i$ denotes the pricing error of bond $i$. Thus, the general estimation problem is to define the discount factors as a continuous function of time, which is dependent

on a specified number of free parameters, and to calibrate this function to the observed bond prices such that the pricing errors are minimized given a certain norm, e.g. sum of squared errors. In this setting, the discount factor is taken as $e^{-r_i(t)t}$

Therefore the pricing error for bond $i$ at time $t$ is :

$$\epsilon_{i,t} = P_{i,t} - \hat{P}_{i,t}$$

### 1.7.2 Pricing Errors

For comparison of pricing errors and trading strategies Svensson (1994), cubic splines model by McCulloch (1971)and B-Splines are used to estimate the term structure of interest rates. For McCulloch to avoid the effect of over parameterization, I use equidistant knots to estimate the term structure. Please see Appendix 4.1.8 for implementation details. [33] The pricing errors for each bond, are defined as the market price (clean price plus accrued interest) minus the model price (sum of discounted cash flows using the estimated term structure). Table 11 presents the statistics of the resulting pricing errors.

For NSS estimation, the mean absolute pricing error observed on each day is on average 8.3, 3.9 and 2.1 bps for Turkey, Brazil and Mexico respectively. And it is between the intervals 0-61.7, 0-20.6 and 0-35.9 bps in the whole sample. The maximum pricing error observed on each day is on average 175.8, 259.7 and 167.5 and the minimum pricing error is -209, -156 and -202 bps respectively for Turkey, Brazil and Mexico. Compared to the average bid-ask spread of around 40 bps, these numbers can be considered to be economically significant. An important question is, whether the mean absolute pricing error is only driven by some large pricing errors or most of the bonds have significant pricing errors. Therefore, the hitting rate is calculated,

[33] I also implemented more sophisticated cubic spline as described at 4.1.8. However, I observe that there is no improvement in the resulting pricing errors.

see Bliss (1997b), which represents the percentage of pricing errors within the bid-ask spread. These errors could be seen as insignificant and might not contain economic information. In my sample the hitting rate is 20 %. Thus, for 80 % of the bonds, I observe significant pricing errors. The observed pricing errors can be due to model misspecification, e.g. poor functional form, or economic reasons.

**Table 11:** Pricing error statistics for NSS

| **Turkey** | **Average** | **Std. Dev.** | **Max** | **Min** |
|---|---|---|---|---|
| **Mean absolute pricing error** | 8,3 | 9,5 | 61,7 | 0,0 |
| **Max pricing error** | 175,8 | 80,0 | 524,1 | 22,9 |
| **Min pricing error** | -209,2 | 169,9 | -35,8 | -1.397,9 |
| **Brazil** | **Average** | **Std. Dev.** | **Max** | **Min** |
| **Mean absolute pricing error** | 3,9 | 5,0 | 20,6 | 0,0 |
| **Max pricing error** | 259,7 | 315,4 | 1.401,0 | 0,6 |
| **Min pricing error** | -156,1 | 162,6 | -0,4 | -918,6 |
| **Mexico** | **Average** | **Std. Dev.** | **Max** | **Min** |
| **Mean absolute pricing error** | 2,1 | 3,3 | 35,9 | 0,0 |
| **Max pricing error** | 167,5 | 199,0 | 1.103,6 | 0,1 |
| **Min pricing error** | -202,8 | 249,5 | -0,1 | -2.079,1 |

For McCulloch model, the mean absolute pricing error observed on each day is on average 9.6, 4.0 and 2.3 bps for Turkey, Brazil and Mexico respectively. And it is between the intervals -0,1-72.2, 0-20.6 and 0-35.9 bps in the whole sample. The maximum pricing error observed on each day is on average 175.8, 259.7 and 167.5 and the minimum pricing error is -209, -156 and -202 bps respectively for Turkey, Brazil and Mexico.

### 1.7.3 Trading Strategy

If the pricing errors are at least partially caused by the deviation of individual bond prices from general market conditions, then trading strategies based on these pricing errors can yield abnormal returns, otherwise they can be just called noise. In the bond data sample that is used, the pricing errors show high auto-correlation and sometimes

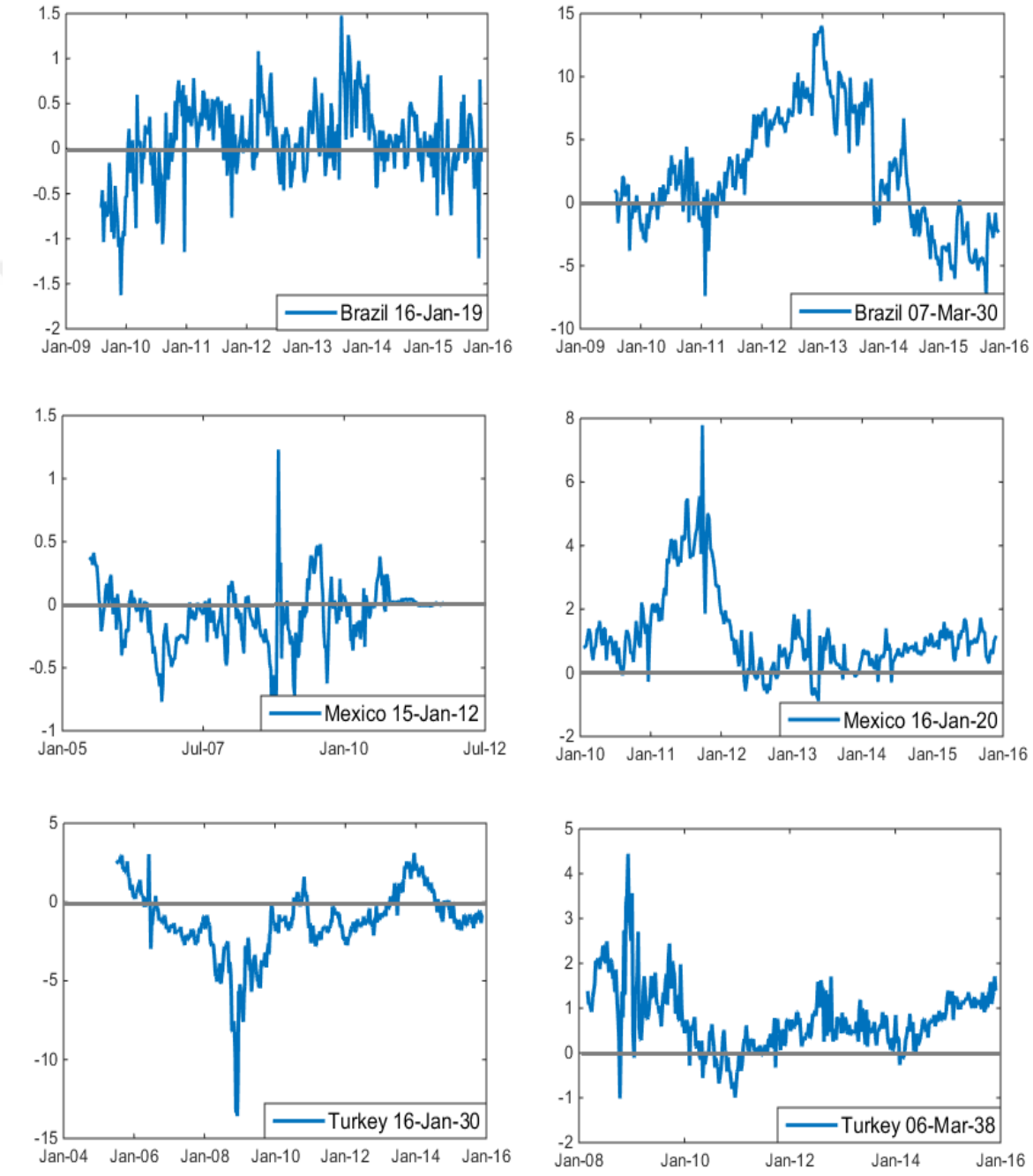


**Figure 43:** Time series of pricing errors for two representative Brazilian, Mexican and Turkish eurobonds using NSS model

**Table 12:** Pricing error statistics for McCulloch

| **Turkey** | **Average** | **Std. Dev.** | **Max** | **Min** |
|---|---|---|---|---|
| **Mean absolute pricing error** | 9,6 | 11,0 | 72,2 | -0,1 |
| **Max pricing error** | 201,4 | 91,4 | 608,0 | 27,1 |
| **Min pricing error** | -219,2 | 193,5 | -36,2 | -1.515,8 |
| **Brazil** | **Average** | **Std. Dev.** | **Max** | **Min** |
| **Mean absolute pricing error** | 4,0 | 5,3 | 20,8 | -0,2 |
| **Max pricing error** | 311,0 | 339,1 | 1.559,6 | 0,5 |
| **Min pricing error** | -156,6 | 184,0 | -0,5 | -939,9 |
| **Mexico** | **Average** | **Std. Dev.** | **Max** | **Min** |
| **Mean absolute pricing error** | 2,3 | 3,8 | 37,3 | -0,2 |
| **Max pricing error** | 179,4 | 228,4 | 1.130,0 | 0,0 |
| **Min pricing error** | -236,6 | 287,8 | -0,1 | -1.987,0 |

oscillate consistently around non-zero levels. Therefore, focusing on relative mispricing of bonds would be more sensible. To focus just on the level of the pricing error, i.e. a positive (negative) pricing error is seen as an overpricing (underpricing) like in Sercu and Wu (1997), Flavell et al. (1994) and Ioannides (2003), does not turn out to be the optimal choice, because a bond with higher or lower liquidity can produce consistent overpricing or underpricing.

The trading rule is designed as follows: Simply the bonds that are undervalued (overvalued) are bought (sold). Trading signals are focused on the mean reversion effect of the pricing errors. The trading rule is tested to detect possible sub-periods where the rule worked better than average and to avoid problems with event-time tests when there are long runs of under- or overpricing. I employ the study of Wu (1995) and Sercu and Wu (1997), who propose three alternative benchmarks of ARs (abnormal returns) to verify whether the competing functions allow any profitable trading strategies. The first benchmark measures ARs as the difference between the market holding period return, $HP_{i,t}$ and the expected holding period, $E_t[HP_{i,t}|\phi_{t-1}, \phi_t]$ for the NS function :

$$AR_{i,t} = HP_{i,t} - E_t[HP_{i,t}|\phi_{t-1}, \phi_t] \tag{30}$$

where

$$E_t[HP_{i,t}|\phi_{t-1}, \phi_t] = \frac{\hat{P}_{i,t} - \hat{P}_{i,t-1} + Coupon}{\hat{P}_{i,t}} \tag{31}$$

For every trading week, a portfolio of underpriced bonds (subscript p, for purchase), and overpriced bonds (subscript s, short for sale), is formed, weighted by the size of the mispricing $RES_{i,t-1}$). If the number of underpriced bonds on day t is $N_{p,t}$, then the mean abnormal return for day $t$ on the purchase portfolio is:

$$\bar{AR}_{p,t} = \sum_{i=1}^{N_{p,t}} \frac{RES_{i,t-1}}{\sum_{i=1}^{N_{p,t}} RES_{i,t-1}} \tag{32}$$

$$C\bar{AR}_{y,t} = \sum_{i=1}^{N_{y,t}} \bar{AR}_{i,t} \tag{33}$$

Trading strategy can be summarized as follows :

## Trading Strategy

*Step 1.* Calculate the pricing error for bond $i$ at time $t$ as $\epsilon_{i,t} = P_{i,t} - \hat{P}_{i,t}$

*Step 2.* Calculate the deviation of the actual pricing error from the moving average computed over last k trading weeks for each bond i :

$$\mu_{i,t} = \frac{1}{k} \sum_{j=t-k}^{t-1} \epsilon_{i,t}$$

$$\sigma_{i,t} = \sqrt{\frac{1}{k-1} \sum_{j=t-k}^{t-1} \epsilon_{i,t} - \mu_{i,t}}$$

*Step 3.* Trading rules :

if $\epsilon_{i,t} > \mu_{i,t} + m\sigma_{i,t}$ , then bond i is overpriced;

if $\epsilon_{i,t} < \mu_{i,t} + m\sigma_{i,t}$ , then bond i is underpriced;

Here $m$ is the multiplier. A trading signal is observed when the actual pricing error is above (below) the price plus (minus) a certain threshold. In this setup, $m$ for the threshold is taken as the width of the confidence interval. A multiplier of 1.96 would define a 95 %, one of 1.65 a 90 % confidence interval. I preferred to choose m as 1.96 which corresponds to the 95 % confidence interval. The time window k is selected as 20 weeks. (The same trading exercise is done by chancing the window from 4 weeks to 100 weeks and I have found that a window of 20-week gives the best results).

### 1.7.4 Trading Results

In this section, I present the results of the annualized abnormal returns, i.e. the excess returns of the different trading strategies compared to the returns of the benchmarks, for the time period from January 2005 to November 2015. I analyze whether the strategies can outperform the buy-and-hold market portfolio (which are simulated

using Bloomberg country specific bond indices).

Against the benchmark, strategies based on all three term-structure estimation models have performed remarkably well as can be seen in 13, 14 and 15. Hence, model-yield spread mean-reverting and betting on the persistence of the spread strategies can outperform a bond investment strategy, on a risk-adjusted basis. For various reasons, some bonds are cheap or rich for long periods in emerging markets, so betting on the existence of persistence may yield better results in terms of trading. The results of the Svensson (1994) are compared with the McCulloch (1971) by applying the same strategy to verify whether the errors are driven by the particular choice of the model. And I observed that the NSS model has a superior performance in terms of profit and loss which can be depicted from tables 13 and 14. As it can be observed from 15 abnormal returns from trading using the B-spline method uniformly underperforms the McCulloch model, therefore the NSS model.

Graphical representation for abnormal returns using NSS estimation for Brazil, Turkey and Mexico can be observed from figures 44, 48, 46 respectively. The charts show that the trading strategy has positive skewness. Cumulative abnormal return charts can be depicted from figures 45, 49 and 47 respectively. Here it can be observed that NSS model has a superior performance to McCulloch cubic spline model and McCulloch is superior to B-spline for all three countries.

**Table 13:** Profits and losses from contrarian trading strategies for each of the emerging bond markets using NSS model

| **Lag (Weeks)** | **Brazil** | | | **Mexico** | | | **Turkey** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Buy | Sell | Both | Buy | Sell | Both | Buy | Sell | Both |
| 0 | 4,769 | 4,645 | 4,311 | 5,191 | 4,440 | 3,780 | 2,721 | 3,975 | 2,953 |
| | 5,040 | 3,829 | 2,612 | 2,327 | 2,445 | 3,229 | 3,113 | 6,087 | 5,076 |
| 1 | 3,164 | 3,092 | 2,363 | 2,263 | 3,966 | 3,408 | 2,246 | 4,387 | 3,261 |
| | 4,875 | 6,292 | 5,098 | 1,057 | 1,854 | 4,455 | 5,429 | 6,485 | 9,518 |
| 4 | 4,808 | 2,678 | 2,793 | 2,772 | 5,833 | 4,204 | 2,789 | 4,132 | 2,906 |
| | 2,941 | 1,868 | 1,963 | 1,010 | 1,093 | 1,001 | 0,792 | 1,766 | 1,148 |
| 8 | 2,767 | 2,437 | 3,519 | 5,355 | 4,147 | 5,276 | 2,651 | 3,630 | 3,519 |
| | 2,753 | 10,328 | 0,124 | 0,765 | 0,499 | 0,624 | 0,973 | 1,719 | 1,203 |
| 12 | 5,795 | 5,121 | 5,508 | 3,781 | 6,120 | 4,145 | 5,907 | 3,955 | 4,727 |
| | 3,154 | 7,115 | 3,727 | 5,739 | 4,725 | 6,355 | 4,182 | 2,251 | 3,707 |
| 24 | 4,110 | 5,576 | 4,599 | 2,916 | 3,455 | 4,642 | 3,242 | 4,227 | 4,198 |
| | 5,181 | 7,288 | 5,286 | 3,664 | 4,809 | 3,606 | 4,149 | 2,586 | 3,379 |

**Table 14:** Profits and losses from contrarian trading strategies for each of the emerging bond markets using McCulloch method

| **Lag (Weeks)** | **Brazil** | | | **Mexico** | | | **Turkey** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Buy | Sell | Both | Buy | Sell | Both | Buy | Sell | Both |
| 0 | 2,965 | 3,323 | 2,722 | 3,607 | 2,859 | 2,949 | 1,761 | 2,719 | 1,636 |
| | 2,531 | 2,429 | 1,929 | 1,371 | 1,659 | 1,771 | 2,103 | 4,267 | 2,664 |
| 1 | 1,804 | 1,753 | 1,284 | 1,376 | 2,674 | 2,098 | 1,440 | 2,556 | 2,342 |
| | 3,082 | 3,954 | 2,762 | 0,731 | 1,335 | 3,308 | 3,549 | 3,633 | 5,651 |
| 4 | 3,232 | 1,630 | 1,463 | 1,738 | 3,106 | 2,697 | 1,759 | 2,895 | 1,744 |
| | 2,145 | 1,009 | 1,078 | 0,566 | 0,819 | 0,521 | 0,486 | 1,028 | 0,654 |
| 8 | 2,011 | 1,377 | 2,321 | 3,620 | 2,713 | 3,602 | 1,758 | 2,580 | 2,443 |
| | 1,690 | 5,244 | 0,089 | 0,383 | 0,266 | 0,356 | 0,529 | 0,966 | 0,795 |
| 12 | 3,809 | 3,792 | 3,417 | 2,672 | 3,307 | 2,806 | 3,514 | 2,311 | 2,683 |
| | 1,974 | 4,031 | 2,627 | 4,081 | 3,310 | 3,628 | 2,229 | 1,530 | 2,749 |
| 24 | 3,057 | 3,642 | 2,791 | 1,636 | 2,291 | 3,304 | 1,766 | 3,015 | 2,716 |
| | 3,757 | 4,045 | 3,250 | 2,023 | 2,781 | 2,140 | 2,742 | 1,579 | 2,047 |

**Table 15:** Profits and losses from contrarian trading strategies for each of the emerging bond markets using B-Splines

| **Lag (Weeks)** | **Brazil** | | | **Mexico** | | | **Turkey** | | |
|---|---|---|---|---|---|---|---|---|---|
| | Buy | Sell | Both | Buy | Sell | Both | Buy | Sell | Both |
| 0 | 2,330 | 2,892 | 2,312 | 2,647 | 1,921 | 1,968 | 1,526 | 2,637 | 1,617 |
| | 2,388 | 1,924 | 1,821 | 1,160 | 1,226 | 1,575 | 2,058 | 3,043 | 1,904 |
| 1 | 1,448 | 1,657 | 1,190 | 1,269 | 1,931 | 1,847 | 1,358 | 2,310 | 2,051 |
| | 2,347 | 3,533 | 2,684 | 0,720 | 1,194 | 2,331 | 3,208 | 2,438 | 5,184 |
| 4 | 2,455 | 1,279 | 1,268 | 1,713 | 2,832 | 2,148 | 1,579 | 2,490 | 1,541 |
| | 1,895 | 0,845 | 0,719 | 0,543 | 0,634 | 0,432 | 0,433 | 0,799 | 0,475 |
| 8 | 1,754 | 1,088 | 2,056 | 2,675 | 2,381 | 2,640 | 1,412 | 2,050 | 1,669 |
| | 1,638 | 3,678 | 0,068 | 0,347 | 0,188 | 0,300 | 0,520 | 0,801 | 0,534 |
| 12 | 3,045 | 3,088 | 3,382 | 2,348 | 2,282 | 2,058 | 3,327 | 2,144 | 2,626 |
| | 1,553 | 3,819 | 2,231 | 3,493 | 2,236 | 2,551 | 2,169 | 1,068 | 2,286 |
| 24 | 2,735 | 2,551 | 2,710 | 1,290 | 1,984 | 2,638 | 1,705 | 2,717 | 2,040 |
| | 2,661 | 3,593 | 2,815 | 1,831 | 2,683 | 2,047 | 2,025 | 1,453 | 2,001 |

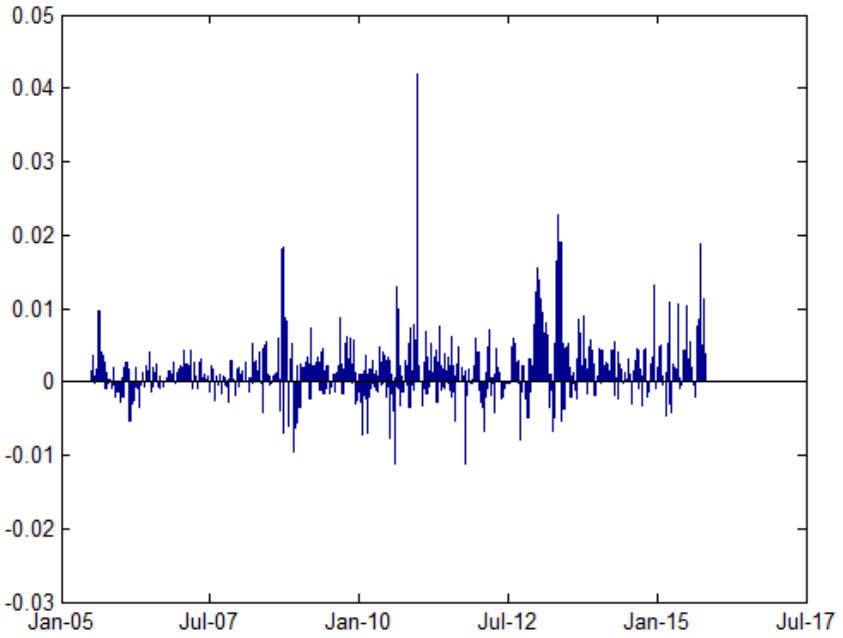


**Figure 44:** Time series of abnormal returns for Brazil government bonds using NSS model

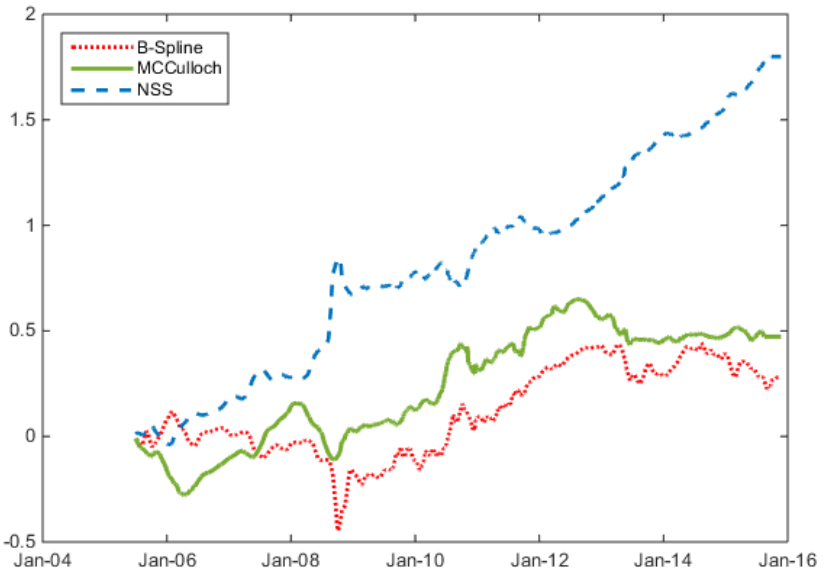


**Figure 45:** Time series of cumulative abnormal returns for Brazil government bonds

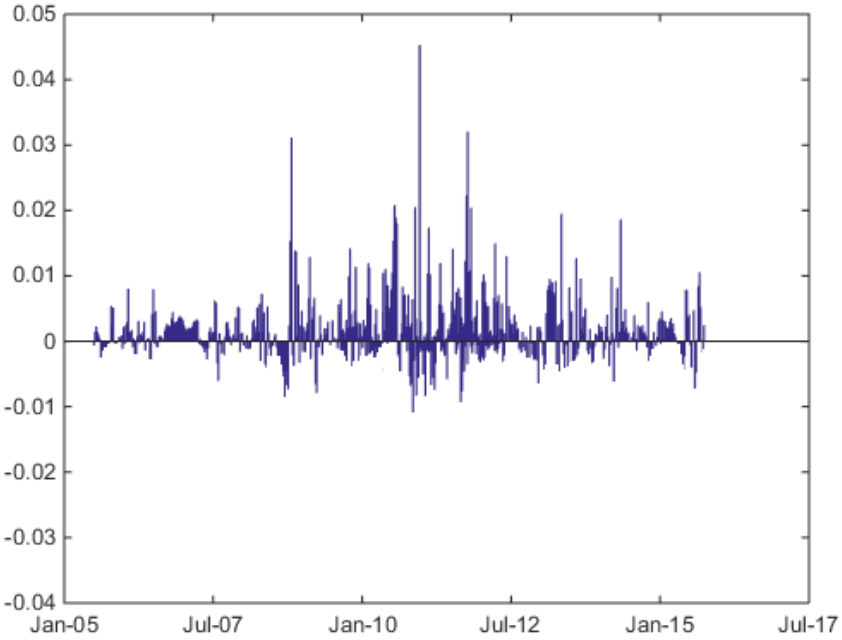


**Figure 46:** Time series of abnormal returns for Mexico government bonds using NSS model

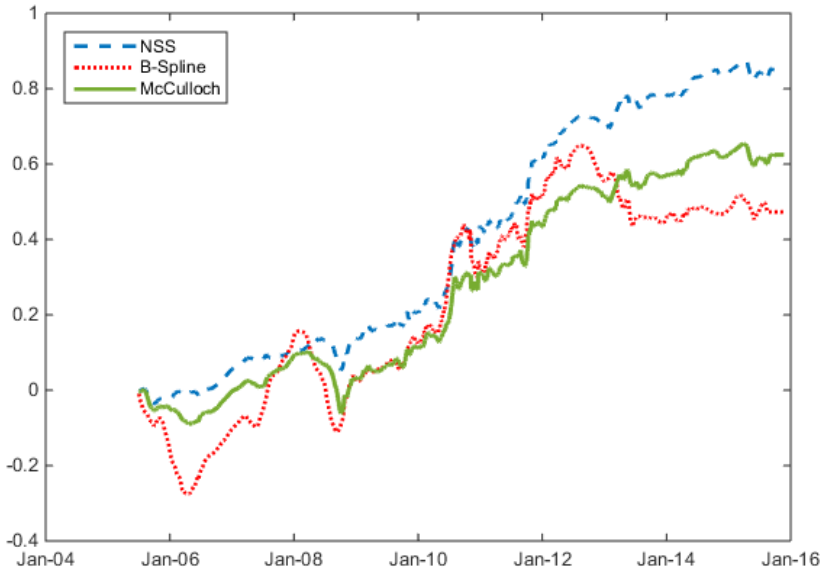


**Figure 47:** Time series of cumulative abnormal returns for Mexico government bonds

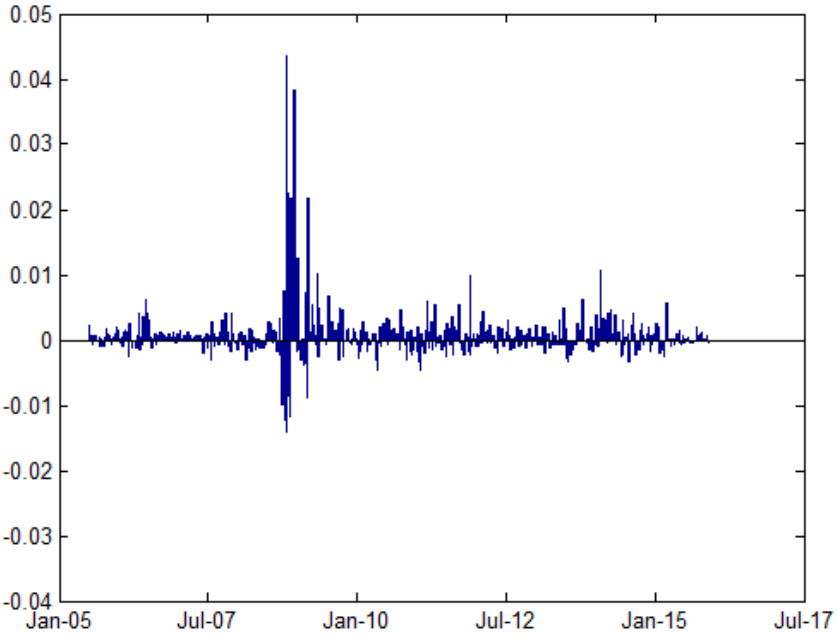


**Figure 48:** Time series of abnormal returns for Turkish government bonds using NSS model

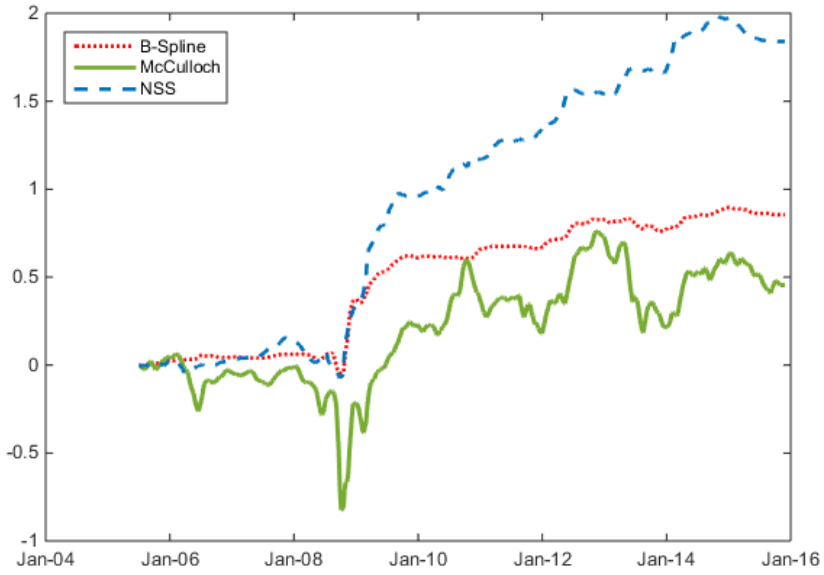


**Figure 49:** Time series of cumulative abnormal returns for Turkish government bonds

# CHAPTER II

# CROSS ASSET APPLICATION : OPTION MARKETS

## *2.1 Introduction*

There are large numbers of optimization problems in theoretical and applied finance that are difficult to estimate as they exhibit multiple local optima. Being very popular among academics, practitioners, and central bankers, NSS term structure of interest rate model sets a perfect example for multiple local optima estimation problems. In the first section of my thesis, I provide clear evidence of the noise that injudicious choice of optimization method can introduce in the estimated values, as well as suggesting and validating a method (Hybrid PSO) that works well for the NSS interest rate model. However, for researchers interested in optimization methods, it is also important to show that my findings are robust to different sets of asset classes. Thus, in this section, I extend the analysis to option markets to investigate whether Hybrid PSO algorithm is powerful enough to handle different asset classes under the same objective functions.

The foreign exchange options market is the most liquid and the largest derivative market. Therefore, economic interpretation of FX option dynamics has a critical importance. The most important factor in option pricing is implied volatility and usually forms a smile shape. On the other hand, very little is known about the construction and forecasting of the implied volatility smile regarding the literature. The volatility smile describes a function relating the implied volatility of an option to the degree of "moneyness". It is shown that the three time-varying parameters may be interpreted as factors corresponding to the level, slope and curvature.

Black and Scholes (1973) model assumes that the asset prices follow a geometric Brownian motion with constant volatility. Therefore, all options on the same asset should have the same implied volatility. In practice, in contrast to Black-Scholes model, implied volatilities tend to differ across exercise prices (or option delta) and time to expiration. For example, S&P 500 option-implied volatilities form a "smile" pattern after the 1987 market crash. Usually, in emerging market currency options, due to a devaluation expectation, a "sneer" appears-the implied volatility which increases monotonically as the exercise price rises relative to the spot level.

For describing the structure of the option prices, the main drawback of the Black-Scholes model arises from its constant volatility assumption. Dealing with non-constant volatility within an option valuation framework, however, is no easy task. In this manner, financial literature follows two different schools : the stochastic volatility models and the deterministic volatility functions. With stochastic volatility, option valuation generally requires a market price of risk parameter, which presents a difficulty of its own. On the other hand, when volatility is a deterministic function of asset price and/or time, option valuation based on the Black-Scholes partial differential equation remains possible even it is not by means of the Black-Scholes formula itself. Dumas et al. (1998) refers this case as the "deterministic volatility function" (DVF) hypothesis. Previously, Derman et al. (1996), Dupire (1994), and Rubinstein (1992) develop variations of the DVF approach. They search for a binomial or trinomial lattice that achieves an exact cross-sectional fit of reported option prices rather than positing a structural form for the volatility function.

Within the class of the function based models, I apply a version of Nelson and Siegel (1987) model which is an exponential polynomial functional form. The model

gives a good estimate of the curve and has the ability to decompose the smile function as level, slope and curvature which means that it generates an economically interpretable parameter set. I would like to exploit this aspect of the function based models, and extend its usage to the pricing of FX options, particularly for emerging markets. The main objective of this study is to model and forecast implied volatility smile, with the ultimate aim of producing relevant information for option traders. The goal of the model-building is to find a model that would reliably capture the dynamics of the implied volatility smile and forecast the future direction of IV, thus providing an economically interpretable model with valuable signals for trading and hedging.

I proceed in the following steps. First, principal component analysis is used to identify the number of factors that explains the variation of implied volatility for both developed and emerging market currency options. Second, the Nelson-Siegel model, Malz (1996), Dumas et al. (1998), Daglish et al. (2007) and methodologies to estimate the smile are applied to 10 different currency option data. Thirdly, a forecasting and stability comparison among the models is applied. Finally, the success of the forecasts are tested with their directional accuracy by simulating option trades with true market prices and calculating the profitability of the trades.

## *2.2 Literature Survey*

Implied volatility literature has two directions. One school of methods starting from Heston (1993) works with the problem under the framework of modeling the volatility of the underlying asset as a stochastic process. Another school has worked within the framework of deterministic volatility functions described in Dumas et al. (1998).

The first approach involves the calibration of a stochastic volatility model to available prices. Once you have the model parameters in such a stochastic volatility model, then you can build the whole volatility surface. A widely adopted choice is the Heston model, Heston (1993). You can use this model either with constant or time-dependent parameters, Hull and White (1987). Another choice that has been used is the SABR model that is proposed by Hagan et al. (2002). Although this is not widely used in the FX options market, it can be used to generate volatility smile curves for a given expiration date. The model is quite attractive in the sense that the implied volatilities can be expressed as explicit, but very complicated, functions of the Black-Scholes implied volatilities. The reason behind the fact that it is not generally used for the FX option data, it requires many prices available for stable calibration of all stochastic volatility models, Castagna (2010).

An exception occurs when volatility is a deterministic function of an asset price and/or time. In this case, option valuation based on the Black-Scholes partial differential equation remains possible, although not by means of the Black-Scholes formula itself. It is done via deterministic volatility functions. There is only a few data points available in the FX option market whereas there is a wider data set for the currencies of emerging markets. Due to this fact, I will not consider the stochastic volatility models in my study and focus on the functional forms which have been used to fit the volatility smile for a given expiration date directly.

Methods for estimating implied risk-neutral probability distributions can be categorized into several different groups, such as implied binomial trees, PDF approximating function methods, and volatility smoothing methods. Although these last two classes go hand-in-hand since there is a quite close connection between the implied distribution of the volatility for a given expiration date and the implied volatility

smile for the same expiration date. For a detailed explanation for this connection, please see Appendix 4.2.

The implied binomial tree method was first started in Rubinstein (1992). The tree is constructed in such a way that minimizes the deviations from a log-normal process fitting the observed option prices. Hence, the implied binomial tree is a non-parametric Bayesian technique related to stochastic process methods, focusing on modeling the evolution of the underlying asset's price. Their methodology is indeed a variation of the deterministic volatility function approach of Rubinstein (1992).

The smoothed implied volatility smile method was originally developed by Shimko (1993). The method is an approximating function applied to the volatility smile rather than to the probability distribution function. Option prices are first converted to implied volatilities using the Black-Scholes options pricing formula. A continuous approximating (smoothing) function is then fitted to the implied volatilities and the associated strike prices. This continuous implied volatility function is converted back into a continuous call price function and then equation (54) is used to obtain the PDF. The Black-Scholes model is used here simply as a transformation or mapping from one measurement space to another. The smoothed implied volatility smile method does not assume that the underlying price process is log-normal. Malz (1997) used delta, $\Delta = \frac{\partial C}{\partial S}$, rather than strike price as the input variable when fitting the implied volatility smile smoothing function. Both Shimko and Malz used low-order polynomial functional forms to fit the implied volatility smile. Campa et al. (1997) introduced the use of smoothing splines to fit the implied volatility function in their case as a function of the strike price.

Several other methods in the literature have been proposed towards producing

a volatility smile, considering the availability of a limited number of option prices. Some of them are really simply very general interpolation and smoothing schemes capable of fitting almost perfectly all available data, provided that no-arbitrage condition of the resulting surface is held. For a detailed description of the no-arbitrage condition in the volatility surface, one can look at Fengler (2006) and the references therein. These tools produce good results assuming that there are many prices that you can work with.

A simple interpolation method that gained a lot of popularity in the academic literature is the second order polynomial function (in deltas) proposed by Malz(1996). This study has been a basic model to explain stylized movement in the volatility smile for a lot of academic studies that has come afterward, e.g. Dumas et al. (1998), Daglish et al. (2007).

There is also Vanna-Volga method that is known as an empirical model that can be used to infer an implied volatility smile from the three available quotes for a given maturity. It is based on the construction of locally replicating portfolios whose related hedging costs are added to the corresponding Black-Scholes prices so that one can have smile consistent values. The methodology has been wonderfully explained in Wystup (2003). Although this methodology work well empirically, the model parameters lack of the financial interpretations as other interpolation techniques.

However, implied volatility models depend on a wide variety of possible state variables. One example is modeling the implied volatility via the proportional moneyness $(K/S_t - 1)$. This was used in Skiadopoulos et al. (2000) in a principal components analysis. They were able to identify two factors which explain about 60% of the variance in S&P 500 data. Their results were consistent across years. But to implement

a smile-consistent no-arbitrage stochastic volatility model for the pricing and hedging of futures options, they need three factors. One was required for the underlying asset and the other two for the implied volatility.

Rubinstein (1992) and Jackwerth and Rubinstein (1996), among others, show that the implied volatilities of stock and stock index options exhibit a pronounced "skew" (that is, the implied volatility is a decreasing function of strike price). For foreign currencies, this skew becomes a "smile" (that is, the implied volatility is a U-shaped function of strike price).

The first attempts to model the volatility smile were by Rubinstein (1992), Derman et al. (1996) and Dupire (1994). These authors show how a one-factor model for an asset price, known as the implied volatility function (IVF) model, can be developed so that it is exactly consistent with the current volatility surface.

## *2.3 Data Description*

In this section, data sets are described in detail. The dataset contains the weekly currency option prices traded in the over-the-counter (OTC) market over the period September 2005 to April 2013, retrieved from Bloomberg. The options are European-style and they are written on both emerging markets currencies (Brazil, Mexico, Turkey, India, and South Africa) and developed markets currencies (Europe, Switzerland, U.K., Japan, and Canada) for four constant times to maturity (1 month, 3 months, 6 month and 1 years). The constant time-to-maturity feature of the OTC currency option market is useful as it allows us to interpolate the volatility of the options with adjacent maturities. Note that liquidity of options is quite important to have a robust analysis. In the analysis, five emerging countries which have very liquid currency option markets, due to their increasing popularity among investors,

are used. Characteristics of the options are collected from Reuters 3000 Extra. In order to assure that a homogenous and reliable sample is used in my analysis all the options on certain deltas are excluded which are illiquid. 10 delta call, 25 delta call, ATM, 10 delta put and 25 delta put options written on the given currencies are used in order to meet the criteria discussed above.

The OTC currency option market has very specific trading conventions. Unlike bond or equity options that are typically traded in terms of option premiums at different strike prices, currency options trade in terms of implied volatilities. More specifically, implied volatility quotes are most commonly available in the form of three types of option combinations: the *delta-neutral* straddle, the risk reversal, and the strangle. A straddle $\sigma_{STR}$ is a portfolio of a call option and a put option with the same strike price and maturity.[1] The risk reversal (RR) measures the difference in implied volatility between an out-of-the-money call option and an out-of-the-money put option. The moneyness level is defined in terms of the Black-Scholes delta of the option and is conventionally set at 25-delta. The strangle (STR) corresponds more precisely to a butterfly spread and measures the difference between the average volatility of the two 25-delta options and the delta-neutral straddle implied volatility. In summary:

$$\sigma_{ATM} = \sigma_{50\Delta Call} \tag{34}$$

$$\sigma_{RR} = \sigma_{25\Delta Call} - \sigma_{25\Delta Put} \tag{35}$$

$$\sigma_{STR} = \frac{\sigma_{25\Delta Call} + \sigma_{25\Delta Put}}{2} - \sigma_{ATM} \tag{36}$$

[1]For the straddle to be delta-neutral, the strike price needs to be sufficiently close to the forward price to make the implied volatility quote of the straddle an at-the-money implied volatility (ATM).

Equations 34 show that the straddle is a measure of the level of the implied volatility, the risk reversal is a measure of the slope of the implied volatility smile, and the strangle is a measure of the curvature of the implied volatility smile. From the three quotes, I can derive the level of implied volatilities at the three levels of delta:

$$\sigma_{25C} = \sigma_{ATM} + \frac{1}{2}\sigma_{25-RR} + \sigma_{STR} \tag{37}$$

$$\sigma_{25P} = \sigma_{ATM} + \sigma_{STR} - \frac{1}{2}\sigma_{RR} \tag{38}$$

**FX Implied Volatility Data:** Table 16 contains summary statistics for the sample of currency options related to the three option combinations defined by equations 34. The sample averages for at-the-money implied volatilities show that EM currencies are the most volatile currency pair in my sample, followed by developed ones. The average term structure of at-the-money implied volatility is upward-sloping for all the underlying assets.

I have used weekly quoted at-the-money, risk reversal and strangle implied volatility data for USD/TRY, USD/BRL, USD/MXN, USD/INR, USD/ZAR, EUR/USD, GBP/USD, USD/CHF, USD/JPY and USD/CAD currencies in the time period from September 2005 to April 2011, to derive call and put volatilities. See Appendix 4.2 for the conventions for market quotations. Summary statistics for the data sets are reported in 17. It can be observed from the table that emerging markets have more risk premia and standard deviation of implied volatilities in emerging markets is higher than that of developed markets.

**Table 16:** Sample statistics for implied vol. of emerging market currencies

| | Delta | Mean | Median | Max | Min | Std. Dev. | Skewness | Kurtosis |
|---|---|---|---|---|---|---|---|---|
| **USDTRY** | 10 | 12.78 | 11.84 | 43.87 | 5.65 | 4.59 | 2.23 | 11.19 |
| | 25 | 12.87 | 11.77 | 45.90 | 5.63 | 4.89 | 2.28 | 11.48 |
| | 50 | 13.96 | 12.64 | 49.77 | 6.20 | 5.41 | 2.23 | 10.97 |
| | 75 | 16.32 | 14.86 | 55.90 | 7.61 | 6.29 | 2.09 | 9.71 |
| | 90 | 18.88 | 17.14 | 62.87 | 8.64 | 7.22 | 2.06 | 9.14 |
| **USDBRL** | 10 | 12.78 | 11.84 | 43.87 | 5.65 | 4.59 | 2.23 | 11.19 |
| | 25 | 12.87 | 11.77 | 45.90 | 5.63 | 4.89 | 2.28 | 11.48 |
| | 50 | 13.96 | 12.64 | 49.77 | 6.20 | 5.41 | 2.23 | 10.97 |
| | 75 | 16.32 | 14.86 | 55.90 | 7.61 | 6.29 | 2.09 | 9.71 |
| | 90 | 18.88 | 17.14 | 62.87 | 8.64 | 7.22 | 2.06 | 9.14 |
| **USDMXN** | 10 | 11.41 | 9.59 | 67.67 | 3.88 | 6.73 | 3.06 | 20.07 |
| | 25 | 11.57 | 9.65 | 66.53 | 4.73 | 7.26 | 3.15 | 17.97 |
| | 50 | 12.45 | 10.33 | 71.43 | 4.90 | 8.26 | 3.16 | 16.99 |
| | 75 | 14.31 | 11.83 | 77.03 | 5.53 | 9.75 | 2.89 | 14.20 |
| | 90 | 16.71 | 13.80 | 86.57 | 6.18 | 11.18 | 2.69 | 12.50 |
| **USDINR** | 10 | 8.90 | 8.63 | 22.78 | 4.06 | 3.18 | 1.02 | 4.89 |
| | 25 | 8.77 | 8.34 | 26.22 | 3.89 | 3.58 | 1.52 | 6.99 |
| | 50 | 9.08 | 8.65 | 30.00 | 3.80 | 4.05 | 1.94 | 9.25 |
| | 75 | 10.20 | 9.58 | 35.78 | 4.15 | 4.87 | 2.18 | 10.38 |
| | 90 | 11.41 | 10.76 | 33.16 | 4.81 | 5.05 | 1.47 | 6.08 |
| **USDZAR** | 10 | 16.56 | 15.15 | 57.69 | 10.16 | 5.49 | 3.63 | 23.43 |
| | 25 | 16.86 | 15.28 | 61.40 | 10.93 | 5.76 | 3.65 | 22.86 |
| | 50 | 17.93 | 16.28 | 65.70 | 11.25 | 6.36 | 3.39 | 20.34 |
| | 75 | 20.44 | 18.43 | 71.40 | 12.21 | 7.31 | 2.96 | 16.19 |
| | 90 | 23.24 | 20.97 | 75.69 | 13.23 | 8.14 | 2.67 | 13.68 |

**Table 17:** Sample statistics for implied vol. of developed market currencies

| | Delta | Mean | Median | Max | Min | Std. Dev. | Skewness | Kurtosis |
|---|---|---|---|---|---|---|---|---|
| **EURUSD** | 10 | 12.32 | 11.76 | 30.75 | 5.13 | 4.76 | 1.00 | 4.28 |
| | 25 | 11.45 | 10.91 | 28.64 | 4.84 | 4.24 | 1.07 | 4.63 |
| | 50 | 10.83 | 10.32 | 27.00 | 4.68 | 3.88 | 1.24 | 5.28 |
| | 75 | 10.75 | 10.16 | 26.75 | 4.81 | 3.83 | 1.49 | 6.02 |
| | 90 | 11.03 | 10.37 | 28.77 | 4.95 | 4.05 | 1.72 | 6.74 |
| **USDGBP** | 10 | 11.63 | 10.59 | 34.98 | 5.14 | 5.10 | 1.89 | 7.26 |
| | 25 | 10.72 | 9.73 | 31.15 | 4.81 | 4.50 | 1.87 | 7.11 |
| | 50 | 10.02 | 8.98 | 28.50 | 4.55 | 4.07 | 1.93 | 7.28 |
| | 75 | 9.85 | 8.71 | 27.45 | 4.64 | 3.91 | 1.99 | 7.33 |
| | 90 | 10.07 | 8.85 | 28.50 | 5.03 | 4.02 | 2.11 | 7.77 |
| **CHFUSD** | 10 | 11.92 | 11.36 | 28.89 | 5.61 | 3.73 | 1.29 | 5.65 |
| | 25 | 11.32 | 10.86 | 26.27 | 5.45 | 3.40 | 1.22 | 5.30 |
| | 50 | 10.99 | 10.53 | 24.55 | 5.25 | 3.25 | 1.13 | 4.95 |
| | 75 | 11.19 | 10.79 | 24.29 | 5.35 | 3.37 | 1.08 | 4.56 |
| | 90 | 11.71 | 11.34 | 25.29 | 5.49 | 3.68 | 1.04 | 4.31 |
| **USDJPY** | 10 | 13.68 | 12.48 | 46.08 | 6.62 | 5.93 | 2.16 | 9.47 |
| | 25 | 12.19 | 11.11 | 40.59 | 6.39 | 4.83 | 2.14 | 9.94 |
| | 50 | 11.09 | 10.29 | 35.53 | 6.13 | 4.01 | 2.08 | 10.20 |
| | 75 | 10.58 | 10.08 | 30.84 | 5.88 | 3.32 | 1.84 | 9.51 |
| | 90 | 10.81 | 10.56 | 29.02 | 5.83 | 3.12 | 1.56 | 8.14 |
| **USDCAD** | 10 | 10.76 | 9.67 | 28.07 | 6.00 | 4.29 | 1.74 | 6.42 |
| | 25 | 10.54 | 9.51 | 27.44 | 5.88 | 4.12 | 1.68 | 6.30 |
| | 50 | 10.62 | 9.74 | 28.25 | 5.75 | 4.12 | 1.64 | 6.32 |
| | 75 | 11.20 | 10.21 | 30.44 | 5.83 | 4.34 | 1.60 | 6.36 |
| | 90 | 11.92 | 10.90 | 33.17 | 5.99 | 4.70 | 1.60 | 6.42 |

## *2.4 Theoretical Framework*

In this section, first general overview of econometric and factor models are presented. Then, theoretical background of volatility smile is explored. Finally, the performance metrics discussed and the models tested are described.

### 2.4.1 Principal Component Analysis

The advantage of using factor based models is that they can deal with high number of parameters reducing them to a few number. Secondly, factor models can eliminate idiosyncratic movements which possibly include measurement error and local shocks. Hence, the factor models yield a more reliable signal for policy makers and prevents them from reacting to idiosyncratic movements. Finally, factor modelers can remain agnostic about the structure of the economy and do not need to rely on overly tight assumptions as it is sometimes the case in structural models.

In order to determine the existence and the number of common factors affecting the movements in implied volatilities across different currencies, it is first assumed that the variance of changes in the volatility smile of any given currency can be decomposed into common variance and unique variance. Common variance is shared by movements of all volatility term structures included in the system whereas unique variance is specific to a particular currency and includes also an error component. In this section, I explore the interactions of the IVS dynamics between different currency pairs. We are interested in whether changes in the IVS of all currency pairs follow some systematic factors that need not (necessarily) be orthogonal.

### 2.4.2 Volatility Smile

Options traded in financial markets have a rather flat volatility structure before the 1987 market crash, but rules of the game have changed one afterward. After the 1987 crash, people realized that extreme events were more likely than the log normal distribution suggests (See Appendix 4.2). The implied volatilities of the options often have a skewed structure, commonly called 'the volatility smile'. One of the long-standing problems in options pricing has been how to reconcile this structure with the Black-Scholes model usually used by options traders. In this study, the Black-Scholes model is extended to make it consistent with the smile.

The Black-Scholes model takes the volatility constant across different strikes and the index distribution at any option's expiration is lognormal, and all options on the index have the same implied volatility. But, ever since the '87 crash, the market's implied Black-Scholes volatilities for FX options have shown a negative relationship with strike prices while out-of-the-money puts trade at higher implied volatilities than out-of-the-money calls.

Now, I will define the volatility smile and examine how it typically appears in the market. A brief summary is given in Appendix 4.2.

*Definition:***Volatility Smile:** For an expiry $T$, the volatility smile maps the implied volatilities to a given function $P = f(K)$ of the strike price. Hence, the smile indicates the volatility parameter to plug into the Black-Scholes formula struck at the level $K = f^{-1}(P)$.

Description of the data used will be explained in detail in the data section. For the moment, I only mention that all data used is obtained from Bloomberg, where

the implied volatility quotes are in terms of forward deltas.

### 2.4.3 Performance Metrics

Performance measures of the volatility smile models are analyzed in three dimensions: goodness-of-fit, forecasting capability, and stability of parameters. First performance measure is goodness-of-fit, for which the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are used. Clearly, the fitting quality is higher for the model that generates smaller errors. Secondly, MAE and RMSE of models for out-of-sample data are compared. Finally, parameter stability of the models is compared by evaluating the deviation in parameter sets and graphical illustrations for the evolution of model parameters over the time are presented to observe their stability.

***Goodness-of-fit*** The performance statistics MAE, and RMSE can be calculated as:

$$\text{MAE} = \sum_{i=1}^{N} \frac{|\hat{\sigma}_i - \sigma_i|}{N},$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N} \frac{(\hat{\sigma}_i - \sigma_i)^2}{N}}.$$

where $N$ represents the number of selected deltas or moneyness levels. RMSE places a greater weight upon larger errors and therefore better indicates how well the models fit the data at each particular observation. A low value for the mean is assumed to indicate that the model is flexible and, on average, is able to fit the volatility smile fairly accurately. MAE is the average distance between theoretical and observed implied volatilities in absolute value terms. This measure is not as easily influenced by extreme observations as RMSE. Therefore, these two measures are complementary.

***Forecasting*** A good approximation to volatility smile dynamics should not only fit well in-sample, but also should forecast well out-of-sample. The prediction errors

are calculated as the difference of forecasted implied volatility and observed implied volatility in terms of RMSE and MAE. Namely, implied volatility values for each delta at a certain time point (weekly, daily or monthly) are forecasted using previous implied volatilities on a selected time window. The time window is iterated for each time point in the data sample. My model's one-period-ahead forecasting results are reported in Table 23 and 24.

***Stability*** Past research have mainly focused on the performance of the volatility smile fit and have paid scant attention to parameter stability. Knowing that NS parameters are estimated on a daily basis and each of these parameters have a specific financial interpretation, it is expected to generate smooth parameters over the time. For instance, consider, the first parameter of NS model ($\beta_0$), which is interpreted as the long-run at-the-money level of implied volatility. From one day to the next, jumps of several percentage points for the estimates of this parameter will be totally unacceptable, even if the smile curve fit is quite good. To this end, I depicted the parameters obtained by the NS model and other compared models together with the day-to-day values of the parameters.

### 2.4.4 Models Tested

Castagna (2010) points out three criteria for the representation of the volatility surface: parsimony, consistency and intuitiveness :

- Parsimony: The representation contains the smallest amount of information needed to retrieve the entire volatility smile.
- Consistency: The information contained in the representation is along expiries and strikes so as to make the integration of missing easily possible, either by interpolation or extrapolation.

- Intuitiveness: The information provides the user a clear picture of the volatility surface. Volatilities on different deltas move dependently. From principal component analysis exercises, one may reasonably assume that the degrees of freedom are only three: **level, slope, and curvature**.

Four different methods were used to extract the implied volatility smile curve from the same option price data of different currencies. Three estimation methods, Malz(1997), Dumas(1998) and Hull(2007) were taken from the literature. The remaining Nelson-Siegel methodology were developed for this research. These four estimation methods represent a broad variety of volatility smile fitting and estimation methods. The use of broadly differing estimation methods is useful for distinguishing method-specific results from those due to the common pricing equations and data sets. The details of the models can be found in Appendix 4.2.

**Models Tested**

1. **Malz :** $\sigma(\Delta) = b_0 atm_t + b_1 rr_t(\Delta - 0.5) + b_2 str_t(\Delta - 0.50)^2$
2. **Dumas et. al. :** $\sigma(\Delta) = \max(0.01, a_0 + a_1\Delta + a_2\Delta^2)$
3. **Daglish et. al. :** $\sigma(\Delta) = c_{1,t} log(\Delta) + c_{2,t} log(\Delta)^2 + c_{3,t} log(\Delta)^3 + c_{4,t} log(\Delta)^4$
4. **Nelson Siegel :** $\sigma(\Delta) = \beta_{0,t} + \beta_{1,t} \frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t} \left( \frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)} \right)$

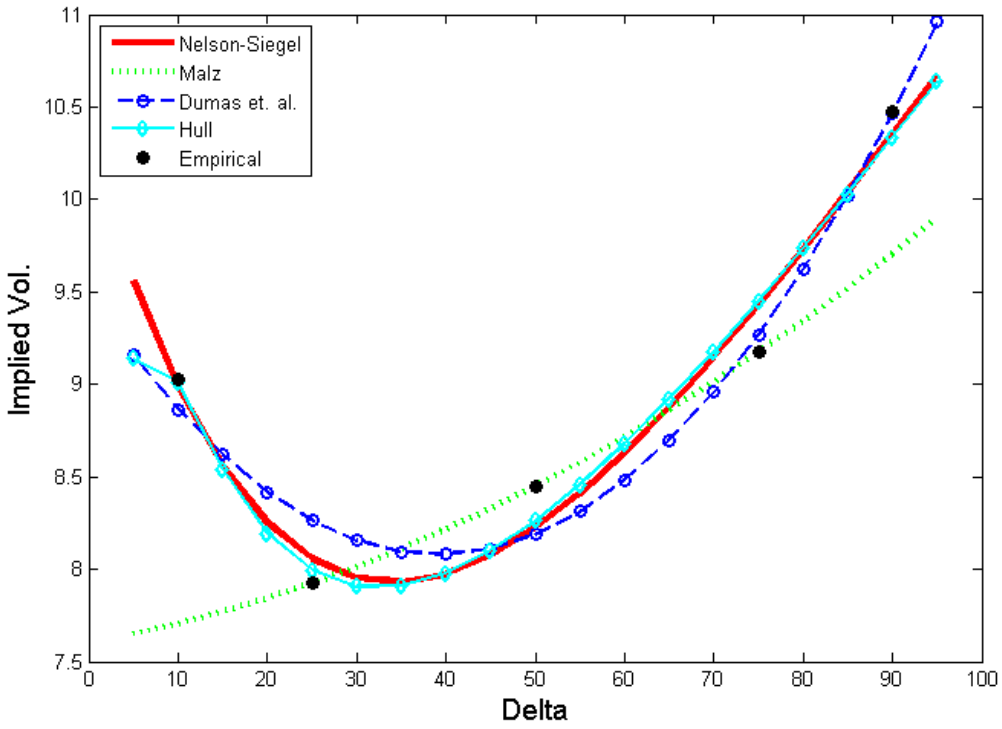


**Figure 50:** Actual and implied volatility curves fitted according to the suggested version of Nelson-Siegel methodology and two other benchmark methods. The data used is the BRL/USD option data with maturity 1 month for the date 03 January 2006

#### *2.4.4.1 Malz*

A continuous volatility smile is constructed by interpolating a particular functional form through the observed market prices of options. Malz assumes the specification

$$\sigma(\Delta) = b_0 atm_t + b_1 rr_t(\Delta - 0.5) + b_2 str_t(\Delta - 0.50)^2$$

In this specification, the volatility smile has three components: a linear function of the at-the-money volatility, a linear function of the risk reversal price and the deviation of delta from 0.5, and a quadratic function of the strangle price and the deviation of delta from 0.5.

This functional form is the simplest one that captures the basic information about the smile which the three option prices express. The at-the-money or straddle volatility gives the general level of implied volatility; it is a measure of location for the volatility smile. The risk reversal price indicates the skew in the volatility smile. The strangle price indicates the degree of curvature of the volatility smile, that is the degree to which the volatilities of out-of-the-money options exceed the at-the-money volatility.

#### *2.4.4.2 Dumas*

Dumas estimates the volatility function, $\sigma(X)$, by fitting the implied volatility levels to reported option prices at time $t$. Because $\sigma(X)$ is an arbitrary function, they post a number of different structural forms including:

$$\sigma(\Delta) = \max(0.01, a_0 + a_1\Delta + a_2\Delta^2)$$

The quadratic functional form given in this equation may seem questionable for two related reasons. First, the use of the parabolic form, for which there is no basis in fact and which are purely extrapolative in nature, may influence the results. Of course, the probability weights received by values of the underlying asset far from the current value become extremely small very quickly, so they may or may not play a negligible role in the analysis.

Second, it is questionable, mathematically speaking, to let the volatility grow quadratically with the state variable, because such a volatility function violates the assumptions for existence of the solution of a stochastic differential equation, so-called slow-growth and Lipschitz conditions.

#### *2.4.4.3 Daglish et. al.*

Daglish(2007) discusses the way in which the traders use the volatility smile for hedging. Derman(1999) also discusses alternative approaches to hedging against asset price movements. The 'volatility-by-strike' or 'sticky strike' rule assumes that the implied volatility for an option with a given strike price and maturity will be unaffected by changes in the underlying asset price. Another popular approach is the 'volatility-by-moneyness' or 'sticky delta' rule. This assumes that the volatility for a particular maturity depends only on the moneyness.

Daglish(2007) tests a series of functional forms for the volatility surface from a perspective of sticky strike or sticky delta forms, and finally decides on a version like the following

$$\sigma(\Delta) = c_{1,t}log(\Delta) + c_{2,t}log(\Delta)^2 + c_{3,t}log(\Delta)^3 + c_{4,t}log(\Delta)^4$$

The paper also considers more restricted forms of the model, where progressively $c_4$ and $c_3$ are set to zero, to render a more parsimonious model for the volatility smile, and looking at their out of sample performances, I decide on this model. In fact, they also consider the case where the coefficients are time dependent. However, since I are focused on the smile rather than the surface, I will treat them as constants when I implement the model.

#### *2.4.4.4 Nelson-Siegel*

The Nelson-Siegel curve model can be viewed as a constant plus a Laguerre function, which is a polynomial times an exponential decay term and is a popular mathematical approximating function as it can be seen in Courant and Hilbert (1953) The

corresponding functional form is

$$\sigma(\Delta) = \beta_{0,t} + \beta_{1,t}\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t}\left(\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)}\right) \tag{39}$$

where $\Theta$ is the set of the model parameters $[\beta_0, \beta_1, \beta_2, \tau]$ and $\Delta$ is option delta.

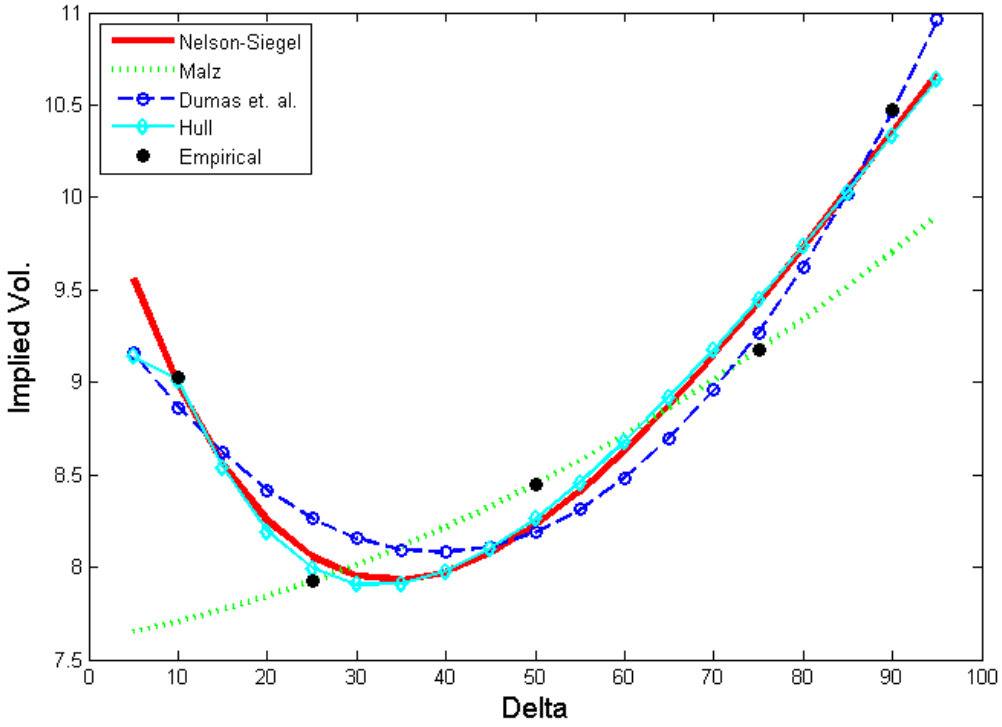


**Figure 51:** Actual and implied volatility curves fitted according to the suggested version of Nelson-Siegel methodology and two other benchmark methods. The data used is the BRL/USD option data with 1-month maturity for the date 03 January 2006

I model and forecast the Nelson-Siegel factors as univariate AR(1) processes. The volatility forecasts based on underlying univariate AR(1) factor specifications are :

$$\sigma_{t+h}(\Delta, \Theta) = \beta_{0,t+h} + \beta_{1,t+h}\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t+h}\left(\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)}\right) \tag{40}$$

where

$$\beta_{i,t+h} = c_i + \gamma_i \beta_{i,t},\ \text{i=1,2,3}$$

## *2.5 Empirical Analysis*

The empirical analysis is divided into three parts. In the first part, a number of factors explaining the variation in implied volatilities are extracted. In the second and third parts, in-sample and out-of sample performances of the models are compared. In the fourth part, coefficient stability of the models are provided.

**Principal Components:**

Investors or portfolio managers can minimize their currency exposure while benefiting international diversification by using out-of-the-money options in each currency that they invest in. However, this approach brings the cost that is the sum of various option premiums for the investor. To explain the variation in these currency options, it is critical to distinguish the systemic risk factors that have general impact on these options. I use the standard methodology for extracting the most important uncorrelated sources of variation in the volatility smile across ten currency option markets. The empirical findings demonstrate that three common factors can explain a vast proportion of the variation in implied volatilities across currencies. Thus, an international portfolio can be hedged by using these three factors against currency risk.

I applied dynamic principal component analysis to determine the nature and commonalities on the variation of implied volatilities for five emerging and five developed markets. To achieve this, I measure the weekly differences of implied volatilities across different levels of moneyness and different ranges of days to expiry (expiry buckets).

For each sub-sample, I extract the principal components from the correlation matrix that has been sustained by the residuals from the regressions. The corresponding analysis reveals that there are three significant components across currencies for both emerging and developed markets.

**Table 18:** Principal Component Analysis
Principal component analysis for implied vol. of developed and emerging market currencies in the period of 2005-2011

| **Year** | **Factors** | **Implied Volatility Variations** | | |
|---|---|---|---|---|
| | | *All Markets* | *Developed* | *Emerging* |
| **Entire Period** *(6 April 2005 - 30 April 2011)* | **Factor 1** | 48,88% | 76,06% | 54,23% |
| | **Factor 2** | 70,84% | 87,25% | 78,86% |
| | **Factor 3** | 80,05% | 94,38% | 88,06% |
| **Pre-Crisis Period** *(6 April 2005 - 8 August 2007)* | **Factor 1** | 36,65% | 64,43% | 45,05% |
| | **Factor 2** | 55,66% | 81,67% | 64,61% |
| | **Factor 3** | 65,73% | 93,29% | 81,01% |
| **Crisis Period** *(9 August 2007 - 29 August 2008)* | **Factor 1** | 51,59% | 77,25% | 57,80% |
| | **Factor 2** | 74,64% | 88,56% | 83,03% |
| | **Factor 3** | 83,31% | 94,96% | 90,84% |
| **Post-Crisis Period** *(1 April 2009 - 30 April 2011)* | **Factor 1** | 41,85% | 71,74% | 43,39% |
| | **Factor 2** | 60,08% | 84,96% | 67,05% |
| | **Factor 3** | 71,59% | 94,29% | 81,70% |

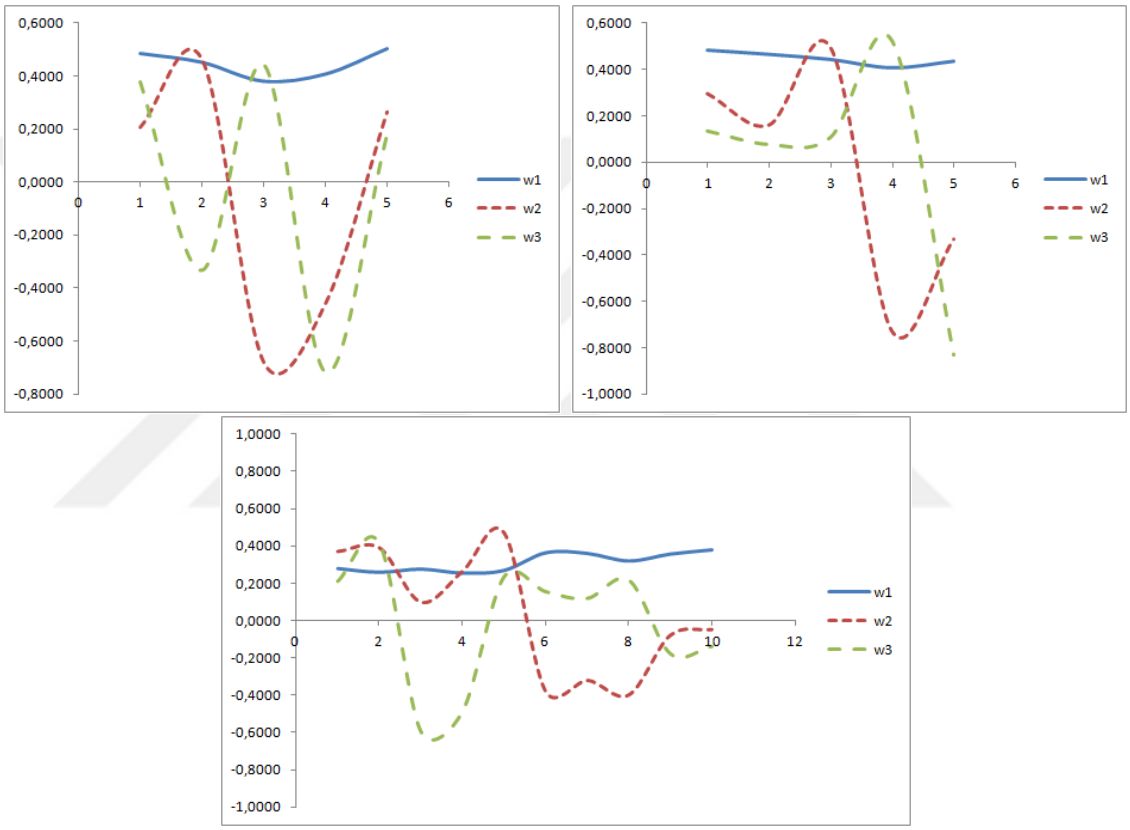


**Figure 52:** The upper left panel shows the first three principal components for emerging market and the upper right shows for developed market currency options with 1 month expiry. The lower panel shows the first three principal components for all market currency options with 1 month expiry

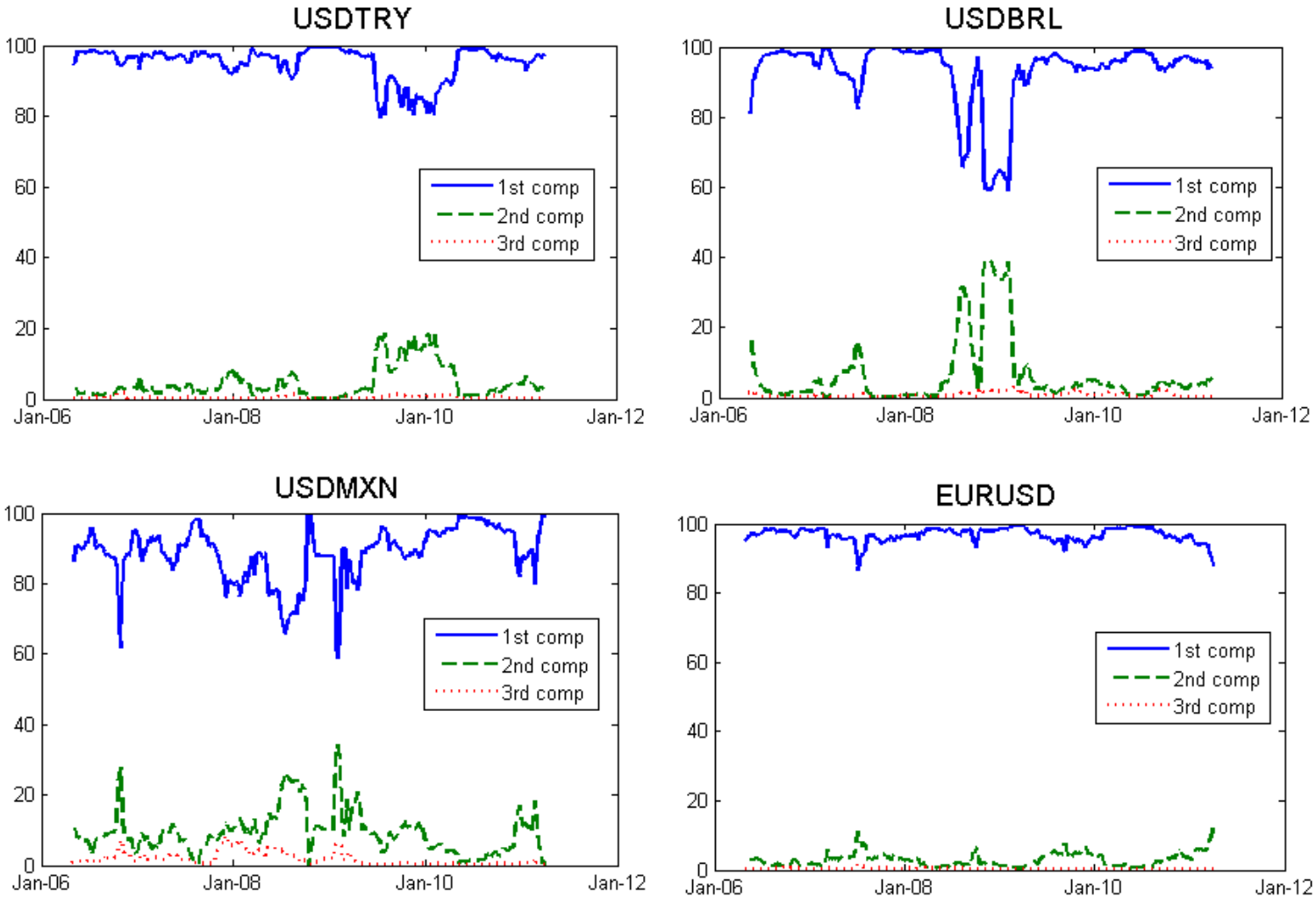


**Figure 53:** The figure shows the first three principal components for all market currency options with 1 month expiry

**Table 19:** Principal Component Analysis for Different Maturity Options
Principal component analysis for implied vol. of developed and emerging market currencies in the period of 2005-2011 for options with maturity range from 1 month to 1 year.
The columns indicate the cumulative coverage for each component.

| **EM** | **w1** | **w2** | **w3** | **w4** | **w5** |
|---|---|---|---|---|---|
| **1 Month** | 53,42% | 70,98% | 83,63% | 92,45% | 100,00% |
| **3 Month** | 55,21% | 75,01% | 84,82% | 93,38% | 100,00% |
| **6 Month** | 51,18% | 70,64% | 84,23% | 93,32% | 100,00% |
| **1 Year** | 53,37% | 70,27% | 85,47% | 93,84% | 100,00% |

| **Developed** | **w1** | **w2** | **w3** | **w4** | **w5** |
|---|---|---|---|---|---|
| **1 Month** | 74,10% | 84,98% | 91,86% | 97,73% | 100,00% |
| **3 Month** | 72,76% | 85,06% | 92,24% | 97,73% | 100,00% |
| **6 Month** | 72,66% | 84,72% | 92,13% | 97,84% | 100,00% |
| **1 Year** | 72,67% | 84,09% | 92,18% | 97,95% | 100,00% |

| **All** | **w1** | **w2** | **w3** | **w4** | **w5** |
|---|---|---|---|---|---|
| **1 Month** | 50,40% | 65,16% | 74,41% | 80,82% | 85,51% |
| **3 Month** | 51,30% | 66,54% | 75,96% | 81,16% | 85,76% |
| **6 Month** | 50,61% | 64,37% | 73,81% | 80,93% | 86,16% |
| **1 Year** | 50,13% | 64,66% | 73,14% | 80,73% | 85,80% |

**In-Sample Results :**

Using the FX implied volatility data for five emerging market and five developed markets, I now estimate the volatility functions of Malz (1997), Dumas et al. (1998), Daglish et al. (2007) and Nelson-Siegel three factor method. Here, I fixed the time to maturity of the options as one month. Results for different maturities ranging from one week to 2 years are available upon request. The results are consistent with the one-month results. So, I depicted one-month maturity options for each currency.

Summary statistics on the goodness-of-fit are provided in table 20 and 21. To assess the quality of the fitted models, I compare the root mean squared errors (RMSE) and mean absolute errors (MAE) of four for five emerging market currencies and five developed market currencies with different deltas. The estimations are run on a weekly data. Results for daily and monthly data have the similar pattern with weekly results and are available upon request.

I structure the conclusions derived from the in-sample results into two categories. The first part of the conclusions would be about each model across different moneyness levels (deltas). The second part of the conclusions would compare different models with each other at various moneyness levels and for different market currencies.

**Across Moneyness Levels**

It is obvious from the theoretical formulation of Malz method, it has perfect goodness-of-fit for the levels of 25, 50 and 75 deltas. However, this leads to a much worse accuracy for 10 and 90 deltas. This observation is consistent across all ten currencies explored. Namely, Malz method overfit the 25, 50, 75 deltas and as a result significantly underfits 10 and 90 deltas. On the contrary, the accuracy structure is reversed for Dumas' methodology, i.e. accuracy level for 25, 50, 75 deltas is

worse than 10 and 90 deltas. This pattern is observed through out all ten currencies explored. What Malz and Dumas' method have in common is that they both use quadratic polynomials to approximate the smile. This yields a trade off between the goodness-of-fit at or around the money and away from the money.

On the other hand, the story for Daglish and Nelson Siegel models is a little different. For the Daglish methodology, the quality of the goodness-of-fit degrades from one end to the other end of the smile. In other words, it performs much better at 10 or 25 deltas compared to 75 and 90 deltas. Amongst the investigated four methods, Nelson Siegel has the most consistent accuracy structure across different deltas and across different currencies. As a typical example, 50 shows a comparison of the fitting performances of the discussed three methods for USDTRY currency.

**Across Market Currencies**

When one compares table 20 and 21, it is clear that all the models perform better in developed market currencies than in emerging market currencies. This can be explained in two different ways. Financially speaking, the shape of the volatility smiles of developed market currencies are indeed smiles whereas the shape of the volatility smiles of emerging market currencies are rather smirks. Mathematically speaking, skewness of the implied volatilies for emerging market currencies are much higher than the skewness of the developed market currencies, see tables 16 and 17. For emerging market currencies, estimation results are heterogeneous among the models. When all models are compared, as table 20 suggests, the RMSE's are almost the same for the methods Nelson-Siegel, Dumas et al. (1998) and Daglish et al. (2007). In case of developed markets, Nelson-Siegel three factor model and Daglish et al. (2007) have slightly better fitting performance when compared to other models.

In summary, I have a pretty clear idea about when Malz method is more accurate or less accurate than the others. But it's hard to argue one way or the other about the accuracies of other methodologies as they have different accuracy patterns. I can only argue that the Nelson Siegel method has the most consistent pattern across different deltas and across different currencies.

**Table 20:** In-Sample Fit Results for Emerging Markets
The table reports the in-sample fitting results for five emerging markets as root mean square errors (RMSE) and mean absolute errors (MAE)

| Model | Delta | USDTRY | | USDBRL | | USDMXN | | USDINR | | USDZAR | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| **Malz** | **10** | 53,17 | 43,38 | 209,51 | 1772,49 | 163,87 | 1178,51 | 132,03 | 738,80 | 148,55 | 454,02 |
| | **25** | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | **50** | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | **75** | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | **90** | 171,00 | 170,27 | 356,04 | 1880,20 | 307,54 | 1202,97 | 159,31 | 529,47 | 236,84 | 490,07 |
| **Dumas et. al.** | **10** | 19,43 | 3,08 | 40,15 | 19,83 | 34,55 | 19,70 | 14,38 | 7,63 | 41,40 | 11,77 |
| | **25** | 26,18 | 7,08 | 67,30 | 92,89 | 54,30 | 76,45 | 30,06 | 39,16 | 65,40 | 38,47 |
| | **50** | 22,60 | 3,41 | 59,11 | 107,21 | 47,79 | 64,45 | 35,13 | 39,01 | 24,76 | 21,84 |
| | **75** | 56,04 | 18,84 | 105,62 | 102,86 | 94,32 | 78,71 | 37,92 | 19,77 | 83,74 | 41,81 |
| | **90** | 31,07 | 5,95 | 55,48 | 22,27 | 50,30 | 20,25 | 17,10 | 2,89 | 48,35 | 12,59 |
| **Hull et. al.** | **10** | 3,45 | 0,06 | 6,03 | 0,44 | 5,03 | 0,28 | 2,87 | 0,12 | 4,30 | 0,13 |
| | **25** | 18,03 | 1,61 | 31,52 | 12,00 | 26,29 | 7,77 | 14,98 | 3,34 | 22,49 | 3,51 |
| | **50** | 49,87 | 12,29 | 87,18 | 91,79 | 72,72 | 59,42 | 41,43 | 25,56 | 62,22 | 26,86 |
| | **75** | 73,41 | 26,64 | 128,34 | 198,94 | 107,05 | 128,77 | 60,99 | 55,39 | 91,60 | 58,20 |
| | **90** | 38,21 | 7,22 | 66,81 | 53,91 | 55,72 | 34,89 | 31,75 | 15,01 | 47,68 | 15,77 |
| **Nelson-Siegel** | **10** | 25,03 | 3,86 | 34,23 | 7,56 | 31,85 | 8,17 | 9,00 | 0,84 | 42,22 | 9,05 |
| | **25** | 38,16 | 10,01 | 57,71 | 28,56 | 50,29 | 25,13 | 16,49 | 5,50 | 70,45 | 27,36 |
| | **50** | 19,45 | 2,75 | 49,59 | 69,15 | 41,59 | 44,17 | 29,19 | 22,83 | 23,07 | 14,80 |
| | **75** | 70,35 | 26,82 | 128,77 | 203,60 | 111,93 | 135,81 | 54,67 | 54,34 | 99,47 | 66,61 |
| | **90** | 43,00 | 9,99 | 75,71 | 69,89 | 65,41 | 41,88 | 30,81 | 16,71 | 62,92 | 24,75 |

**Table 21:** In-Sample Fit Results for Developed Markets
The table reports the in-sample fitting results for five developed markets as root mean square errors (RMSE) and mean absolute errors (MAE)

| Model | Delta | EURUSD | | USDGBP | | CHFUSD | | USDJPY | | USDCAD | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| **Malz** | **10** | 64,96 | 28,80 | 67,15 | 38,49 | 62,11 | 74,54 | 188,88 | 266,53 | 31,40 | 7,53 |
| | **25** | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | **50** | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | **75** | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| | **90** | 45,54 | 16,30 | 31,88 | 11,66 | 53,88 | 70,07 | 79,00 | 68,33 | 43,99 | 13,83 |
| **Dumas et. al.** | **10** | 8,94 | 0,57 | 10,92 | 0,92 | 7,58 | 0,66 | 29,43 | 6,60 | 5,81 | 0,25 |
| | **25** | 17,57 | 2,15 | 20,37 | 3,31 | 15,34 | 3,60 | 56,06 | 23,49 | 10,04 | 0,72 |
| | **50** | 12,79 | 1,18 | 10,94 | 1,19 | 14,03 | 4,44 | 30,47 | 7,72 | 7,61 | 0,45 |
| | **75** | 12,44 | 1,00 | 9,98 | 0,82 | 12,08 | 3,18 | 18,85 | 5,13 | 13,81 | 1,30 |
| | **90** | 6,77 | 0,29 | 6,41 | 0,31 | 5,69 | 0,56 | 12,77 | 2,11 | 7,57 | 0,39 |
| **Hull et. al.** | **10** | 1,07 | 0,01 | 0,89 | 0,01 | 1,16 | 0,02 | 1,64 | 0,02 | 0,99 | 0,01 |
| | **25** | 5,58 | 0,21 | 4,64 | 0,16 | 6,08 | 0,51 | 8,59 | 0,56 | 5,17 | 0,16 |
| | **50** | 15,45 | 1,59 | 12,83 | 1,21 | 16,83 | 3,89 | 23,76 | 4,32 | 14,31 | 1,24 |
| | **75** | 22,74 | 3,45 | 18,88 | 2,62 | 24,77 | 8,43 | 34,98 | 9,37 | 21,06 | 2,69 |
| | **90** | 11,84 | 0,93 | 9,83 | 0,71 | 12,89 | 2,28 | 18,21 | 2,54 | 10,96 | 0,73 |
| **Nelson-Siegel** | **10** | 2,85 | 0,06 | 3,05 | 0,14 | 7,15 | 14,69 | 7,74 | 1,26 | 5,08 | 0,19 |
| | **25** | 6,68 | 0,33 | 7,05 | 0,66 | 10,22 | 7,70 | 21,12 | 5,96 | 8,48 | 0,49 |
| | **50** | 11,17 | 0,99 | 9,42 | 0,99 | 13,71 | 5,90 | 27,29 | 6,51 | 6,77 | 0,38 |
| | **75** | 16,38 | 1,82 | 12,93 | 1,43 | 18,69 | 9,19 | 29,66 | 9,08 | 17,29 | 1,94 |
| | **90** | 8,80 | 0,53 | 6,81 | 0,39 | 12,64 | 15,25 | 15,52 | 3,03 | 10,40 | 0,70 |

**Out-Of-Sample Results :**

In this section, I compare the forecasting capabilities of each methodology on different market currencies across options with different maturities and different forecast horizons. I use exactly the same time period for the data, i.e. ranging from January 2006 to April 2011. Only this time, I have options with various maturities ranging from 1 week to 1 year. Similarly, I have different choices for the forecast horizons again ranging from 1 week to 1 year as opposed to one month maturity data. I have explored all of these combinations that make sense. Clearly, one would never forecast 1 year ahead implied volatility smile of options with one week maturities. The following table indicates the combinations on which I have done the forecast analysis.

**Table 22:** Executed forecast analysis

| | | **Forecast Horizon** | | | |
|---|---|---|---|---|---|
| | | 1 Week | 1 Month | 3 Month | 6 Month |
| **Option Maturity** | 1 Month | X | X | | |
| | 3 Month | X | X | X | |
| | 6 Month | X | X | X | X |
| | 1 Year | X | X | X | X |

The performance metrics of the methodologies do not change significantly by neither the maturity of the data set nor the duration of the forecast horizon. Hence, for the results of table 23 and 24 I used 3 month options in my data set, and 1 month horizon for the forecast as a representative. The results hold robust regardless of the choices of forecast horizon and option maturities.

**Techniques of Forecasting**

If the estimation method of the implied volatility smile is economically more reasonable, simple models of the dynamics of its parameters should be able to perform better in out-of-sample forecasting. In this section, I deal with such a forecasting

exercise. Starting from January 2006, I fit all the models on the first 52 observations in order to estimate implied volatilities for each market currency. Here, the size of the rolling training window, 52 weeks, is not a random selection. [2] I have tried different sizes of training window ranging from 1 week to 104 weeks. Any window shorter than 20 weeks yields dramatically poor results for any of the methods. At the other extreme, for any window size larger than 70 weeks, the contribution that is supposed to come from AR(1) extension disappears due to smoothing out effects. Practically, all of the AR(1) forecasts reduces to AR(0) forecasts.

Main econometric tools used to forecast the future implied volatility smile are random walk and AR(1). For example, I model and forecast the Nelson-Siegel factors as random walk and univariate AR(1) processes. The volatility forecasts based on underlying random walk specifications are :

$$\sigma_{t+h}(\Delta) = \beta_{0,t} + \beta_{1,t}\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t}\left(\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)}\right) \tag{41}$$

The volatility forecasts based on underlying univariate AR(1) factor specifications are :

$$\sigma_{t+h}(\Delta,\Theta) = \beta_{0,t+h} + \beta_{1,t+h}\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t+h}\left(\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)}\right) \tag{42}$$

where

$$\beta_{i,t+h} = c_i + \gamma_i\beta_{i,t},\ \text{i=1,2,3}$$

Parameter specifications for other methods are summarized in 2.5.

[2]The forecasts were calculated from rolling samples, keeping the sample size constant each week. In other words, after calculating each forecast, the furthest observations are dropped, the observations for the most recent week are added to the sample, and the model is re-estimated

Although, one can use many other econometric models such as AFRI(1,d), VAR(1), BVAR(1), VECM, etc as in Chalamandaris (2011) and Diebold et. al. (2005), I have not observed significant improvements coming from more complex econometric models. This is partly due to availability of few data points for each volatility smile. As an example, I presented the results in tables 23 and 24 also for the VAR(1) case, but clearly the results are almost the same with AR(1). In fact, this is true for many other more complex econometric models. For a detailed explanation of random walk, AR(1) and VAR(1) methodologies, I refer to Brooks(2002).

Here, I do not include the Malz methodology for the forecasting results as the nature of the methodology does not allow us to change its parameters from one period to another. Therefore, it is impossible to apply any of the econometric models that I have explored.

**Forecast Performance**

I compare the forecasting capabilities of all the models provided in 2.5. Following the results of table 23 and 24, one striking observation is that all estimation and forecasting procedures perform much better for the developed market currencies than the emerging market counterparts. As 16 and 17 shows, the range and the variance of implied volatility smiles for emerging market currencies are significantly larger than the ones for developed market. This makes the forecasting analysis harder for the emerging market volatility smiles. Hence, all the models generate larger MAE and RMSE for forecasting emerging market implied volatilities.

**Forecasting Models**

1. **Random Walk :** $\sigma_{t+h}(\Delta, \Theta) = \sigma_t(\Delta, \Theta)$

2. **AR(1) on volatility levels :** $\sigma_{t+h}(\Delta, \Theta) = c(\delta) + \gamma\sigma_t(\Delta, \Theta)$

3. **VAR(1) on volatility levels :** $\sigma_{t+h}(\Delta, \Theta) = c + \Gamma\sigma_t(\Delta, \Theta)$ where $\sigma_t = [\sigma_t(10), \sigma_t(25), \sigma_t(50), \sigma_t(75), \sigma_t(90)]^T$.

4. **Dumas et al. (1998) model:** $\sigma_{t+h}(\Delta) = a_{0,t} + a_{1,t}\Delta + a_{2,t}\Delta^2$

5. **Dumas' model with AR(1):** $\sigma_{t+h}(\Delta) = a_{0,t+h} + a_{1,t+h}\Delta + a_{2,t+h}\Delta^2$

$$a_{i,t+h} = c_i + \gamma_i a_{i,t}, \text{ i=1,2,3}$$

6. **Daglish et al. (2007) model :** $\sigma_{t+h}(\Delta) = c_{1,t}log(\Delta) + c_{2,t}log(\Delta)^2 + c_{3,t}log(\Delta)^3 + c_{4,t}log(\Delta)^4$

7. **Daglish with AR(1) :** $\sigma_{t+h}(\Delta) = c_{1,t+h}log(\Delta) + c_{2,t+h}log(\Delta)^2 + c_{3,t+h}log(\Delta)^3 + c_{4,t+h}log(\Delta)^4$

$$c_{i,t+h} = d_i + \gamma_i c_{i,t}, \text{ i=1,2,3}$$

8. **NS :** $\sigma_{t+h}(\Delta) = \beta_{0,t} + \beta_{1,t}\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t}\left(\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)}\right)$

9. **NS with AR(1) :** $\sigma_{t+h}(\Delta) = \beta_{0,t+h} + \beta_{1,t+h}\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} + \beta_{2,t+h}\left(\frac{1-e^{(-\Delta/\tau)}}{\Delta/\tau} - e^{(-\Delta/\tau)}\right)$

$$\beta_{i,t+h} = c_i + \gamma_i \beta_{i,t}, \text{ i=1,2,3}$$

The difficulty of forecasting the smile for the emerging market currencies only gets

worse as the moneyness level (delta) increases, because almost all the smiles have a tendency to go up towards the right end. Consequently, the range and the variance of in the money implied volatilities are much higher than out of the money ones. So, all the models generate larger MAE and RMSE for forecasting in the money implied volatilities. This phenomenon does not happen for the developed market currencies since they are rather flat towards the edges of the smile compared to emerging market currencies.

As far as forecasting performances are concerned, it is hard to distinguish Dumas from Daglish in both random walk and AR(1) cases. But in both cases, Nelson Siegel is the winner for the emerging markets and the developed markets. Note that all of these methods have a better forecasting capability when I use the random walk model. Hence, one should not compare the Dumas methodology with random walk to Nelson Siegel with AR(1).

One possible explanation for the superior performance of the Nelson Siegel methodology is the fact that "one may reasonably assume that the degrees of freedom are only three : (i)level, (ii) slope, and (iii)convexity. In fact, as a principal component analysis can show, most shape variations can be explained either by a parallel shift of the smile or by a tilt to the right or left or by a relative change of the wings with respect to the central strike." (Castagna, 2010). The connection between level, slope and curvature of the smile and the $\beta_0$, $\beta_1$, $\beta_2$ parameters are going to be further explained in section 3.3. This connection ,which the other models do not have, yields a superior forecasting capability as well as a superior stability which will be shown in the next section.

**Stability Results**

**Table 23:** Out-Of-Sample Fit Results for Emerging Markets
The table reports the out-of-sample fitting results for five developed markets as root mean square errors (RMSE) and mean absolute errors (MAE)

| Model | Delta | USDTRY | | USDBRL | | USDMXN | | USDINR | | USDZAR | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| **Random Walk** | **10** | 1,50 | 2,42 | 2,07 | 3,95 | 1,98 | 4,94 | 1,40 | 2,22 | 1,55 | 3,15 |
| | **25** | 1,62 | 2,64 | 1,82 | 3,27 | 1,69 | 4,47 | 1,40 | 2,24 | 1,56 | 3,01 |
| | **50** | 1,81 | 2,95 | 2,13 | 3,86 | 1,97 | 5,22 | 1,63 | 2,71 | 1,75 | 3,34 |
| | **75** | 2,09 | 3,41 | 2,62 | 4,78 | 2,40 | 6,14 | 2,06 | 3,37 | 1,99 | 3,80 |
| | **90** | 2,37 | 3,88 | 3,18 | 5,91 | 3,04 | 7,60 | 2,41 | 3,58 | 2,20 | 4,29 |
| **AR(1) on Vol** | **10** | 1,49 | 2,52 | 2,01 | 3,61 | 2,41 | 8,80 | 1,44 | 2,20 | 1,65 | 3,52 |
| | **25** | 1,63 | 2,74 | 1,96 | 3,89 | 2,52 | 13,65 | 1,43 | 2,34 | 1,67 | 3,22 |
| | **50** | 1,93 | 3,26 | 2,11 | 4,24 | 2,74 | 14,95 | 1,51 | 2,73 | 1,90 | 3,57 |
| | **75** | 2,11 | 3,53 | 2,79 | 5,29 | 3,35 | 16,04 | 2,12 | 3,50 | 2,11 | 3,98 |
| | **90** | 2,39 | 4,04 | 3,32 | 6,67 | 3,94 | 17,24 | 2,43 | 3,57 | 2,23 | 4,25 |
| **VAR on Vol** | **10** | 1,56 | 2,57 | 2,06 | 3,95 | 2,73 | 14,36 | 1,49 | 2,40 | 6,22 | 60,42 |
| | **25** | 1,65 | 2,80 | 2,05 | 4,17 | 3,07 | 13,90 | 1,69 | 3,42 | 3,13 | 19,82 |
| | **50** | 1,84 | 3,13 | 2,41 | 5,32 | 3,50 | 15,24 | 1,98 | 4,20 | 4,40 | 33,13 |
| | **75** | 2,12 | 3,59 | 2,90 | 6,65 | 4,06 | 16,66 | 2,45 | 4,96 | 13,06 | 156,36 |
| | **90** | 2,36 | 4,07 | 3,23 | 6,10 | 4,41 | 18,59 | 2,65 | 4,49 | 11,61 | 132,92 |
| **Dumas et. Al.** | **10** | 1,51 | 2,42 | 2,04 | 3,86 | 1,95 | 4,83 | 1,38 | 2,13 | 1,56 | 3,11 |
| | **25** | 1,58 | 2,60 | 1,81 | 3,30 | 1,71 | 4,47 | 1,36 | 2,22 | 1,54 | 3,01 |
| | **50** | 1,78 | 2,93 | 2,09 | 3,83 | 1,92 | 5,16 | 1,62 | 2,74 | 1,72 | 3,30 |
| | **75** | 2,16 | 3,43 | 2,71 | 4,84 | 2,55 | 6,23 | 2,11 | 3,40 | 2,06 | 3,83 |
| | **90** | 2,32 | 3,83 | 3,05 | 5,79 | 2,93 | 7,46 | 2,35 | 3,52 | 2,15 | 4,24 |
| **Daglish et. Al.** | **10** | 1,54 | 2,43 | 2,11 | 3,98 | 2,02 | 4,94 | 1,42 | 2,21 | 1,61 | 3,16 |
| | **25** | 1,55 | 2,63 | 1,75 | 3,22 | 1,62 | 4,41 | 1,33 | 2,21 | 1,52 | 2,98 |
| | **50** | 1,89 | 2,98 | 2,22 | 3,89 | 2,04 | 5,23 | 1,70 | 2,78 | 1,82 | 3,36 |
| | **75** | 2,30 | 3,51 | 2,90 | 5,00 | 2,75 | 6,36 | 2,21 | 3,50 | 2,22 | 3,93 |
| | **90** | 2,28 | 3,83 | 2,95 | 5,66 | 2,84 | 7,34 | 2,26 | 3,42 | 2,10 | 4,20 |
| **NS** | **10** | 1,39 | 2,20 | 1,90 | 3,49 | 1,75 | 4,41 | 1,31 | 2,00 | 1,47 | 2,81 |
| | **25** | 1,47 | 2,37 | 1,73 | 3,04 | 1,58 | 4,09 | 1,31 | 2,02 | 1,42 | 2,77 |
| | **50** | 1,62 | 2,65 | 1,97 | 3,51 | 1,74 | 4,66 | 1,49 | 2,57 | 1,60 | 3,01 |
| | **75** | 1,99 | 3,12 | 2,51 | 4,39 | 2,36 | 5,68 | 1,90 | 3,09 | 1,93 | 3,52 |
| | **90** | 2,15 | 3,55 | 2,78 | 5,25 | 2,71 | 6,78 | 2,20 | 3,23 | 2,01 | 3,86 |
| **AR(1) NS** | **10** | 1,42 | 2,38 | 2,09 | 4,20 | 2,42 | 7,68 | 1,40 | 2,04 | 1,66 | 3,51 |
| | **25** | 1,47 | 2,49 | 2,33 | 6,33 | 2,41 | 7,19 | 1,32 | 2,22 | 1,80 | 4,02 |
| | **50** | 1,66 | 2,79 | 2,85 | 9,06 | 3,03 | 8,20 | 1,64 | 3,16 | 2,30 | 5,90 |
| | **75** | 1,99 | 3,19 | 3,40 | 9,49 | 4,14 | 11,70 | 2,17 | 4,10 | 3,09 | 6,41 |
| | **90** | 2,17 | 3,55 | 3,49 | 8,80 | 4,96 | 15,68 | 2,33 | 4,50 | 3,43 | 6,61 |
| **AR(1) Daglish** | **10** | 1,56 | 2,61 | 2,43 | 5,14 | 3,03 | 10,27 | 1,47 | 2,26 | 1,89 | 4,13 |
| | **25** | 1,57 | 2,81 | 2,75 | 8,07 | 3,21 | 11,35 | 1,42 | 2,41 | 1,93 | 4,50 |
| | **50** | 1,96 | 3,19 | 3,62 | 10,79 | 4,35 | 13,47 | 1,79 | 2,91 | 2,58 | 5,33 |
| | **75** | 2,39 | 3,76 | 4,55 | 12,49 | 5,45 | 15,57 | 2,32 | 3,66 | 3,12 | 6,33 |
| | **90** | 2,33 | 4,06 | 4,44 | 13,57 | 5,55 | 17,53 | 2,40 | 3,83 | 2,91 | 6,85 |
| **AR(1) Dumas** | **10** | 1,52 | 2,55 | 2,28 | 4,61 | 2,66 | 8,45 | 1,45 | 2,21 | 1,74 | 3,86 |
| | **25** | 1,57 | 2,71 | 2,54 | 6,97 | 2,64 | 7,94 | 1,43 | 2,42 | 1,92 | 4,37 |
| | **50** | 1,77 | 2,99 | 3,09 | 10,05 | 3,27 | 9,06 | 1,75 | 3,48 | 2,48 | 6,50 |
| | **75** | 2,16 | 3,46 | 3,74 | 10,48 | 4,53 | 12,99 | 2,30 | 4,45 | 3,35 | 9,41 |
| | **90** | 2,30 | 3,84 | 3,82 | 9,78 | 5,49 | 17,36 | 2,53 | 5,00 | 3,78 | 11,36 |

**Table 24:** Out-Of-Sample Results for Developed Markets
For the volatility smiles implied by options written on the exchange rates in my developed markets sample, the table reports root mean squared errors (RMSE) and mean abolute errors (MAE) of one-month-ahead forecasts for 3 month options

| Model | Delta | EURUSD | | USDGBP | | CHFUSD | | USDJPY | | USDCAD | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| **Random Walk** | **10** | 1,41 | 2,14 | 1,46 | 2,47 | 1,21 | 1,80 | 2,24 | 3,35 | 1,34 | 1,87 |
| | **25** | 1,23 | 1,84 | 1,26 | 2,09 | 1,11 | 1,65 | 1,71 | 2,57 | 1,25 | 1,80 |
| | **50** | 1,13 | 1,69 | 1,14 | 1,87 | 1,04 | 1,53 | 1,35 | 2,05 | 1,24 | 1,82 |
| | **75** | 1,13 | 1,69 | 1,08 | 1,76 | 1,06 | 1,54 | 1,10 | 1,64 | 1,32 | 2,00 |
| | **90** | 1,22 | 1,84 | 1,14 | 1,86 | 1,17 | 1,69 | 1,03 | 1,46 | 1,47 | 2,28 |
| **AR(1) on Vol** | **10** | 1,42 | 2,16 | 1,50 | 2,50 | 1,23 | 1,82 | 2,28 | 3,40 | 1,38 | 1,91 |
| | **25** | 1,23 | 1,86 | 1,30 | 2,13 | 1,12 | 1,66 | 1,73 | 2,60 | 1,29 | 1,85 |
| | **50** | 1,38 | 1,93 | 1,69 | 2,34 | 1,25 | 1,70 | 2,51 | 3,07 | 1,30 | 1,87 |
| | **75** | 1,15 | 1,71 | 1,11 | 1,80 | 1,04 | 1,52 | 1,09 | 1,60 | 1,37 | 2,03 |
| | **90** | 1,24 | 1,86 | 1,18 | 1,91 | 1,15 | 1,67 | 1,00 | 1,42 | 1,52 | 2,29 |
| **VAR on Vol** | **10** | 1,45 | 2,20 | 1,51 | 2,53 | 1,30 | 1,88 | 2,23 | 3,34 | 1,32 | 1,86 |
| | **25** | 1,27 | 1,91 | 1,31 | 2,16 | 1,18 | 1,73 | 1,73 | 2,61 | 1,28 | 1,82 |
| | **50** | 1,18 | 1,76 | 1,19 | 1,94 | 1,10 | 1,60 | 1,38 | 2,08 | 1,30 | 1,89 |
| | **75** | 1,19 | 1,76 | 1,13 | 1,82 | 1,12 | 1,61 | 1,11 | 1,66 | 1,44 | 2,12 |
| | **90** | 1,27 | 1,88 | 1,17 | 1,93 | 1,23 | 1,76 | 1,03 | 1,46 | 1,60 | 2,39 |
| **Dumas et. Al.** | **10** | 1,38 | 2,12 | 1,42 | 2,44 | 1,19 | 1,79 | 2,17 | 3,31 | 1,32 | 1,85 |
| | **25** | 1,23 | 1,84 | 1,30 | 2,11 | 1,10 | 1,65 | 1,83 | 2,64 | 1,25 | 1,80 |
| | **50** | 1,11 | 1,67 | 1,10 | 1,84 | 1,02 | 1,52 | 1,30 | 2,04 | 1,21 | 1,80 |
| | **75** | 1,13 | 1,69 | 1,08 | 1,75 | 1,06 | 1,54 | 1,08 | 1,62 | 1,34 | 2,00 |
| | **90** | 1,21 | 1,83 | 1,13 | 1,85 | 1,15 | 1,67 | 1,03 | 1,46 | 1,45 | 2,26 |
| **Daglish et. Al.** | **10** | 1,40 | 2,13 | 1,45 | 2,46 | 1,20 | 1,79 | 2,21 | 3,32 | 1,34 | 1,86 |
| | **25** | 1,20 | 1,82 | 1,24 | 2,08 | 1,08 | 1,64 | 1,76 | 2,59 | 1,21 | 1,78 |
| | **50** | 1,12 | 1,68 | 1,11 | 1,85 | 1,03 | 1,52 | 1,29 | 2,04 | 1,23 | 1,81 |
| | **75** | 1,16 | 1,72 | 1,12 | 1,77 | 1,10 | 1,57 | 1,11 | 1,62 | 1,39 | 2,03 |
| | **90** | 1,19 | 1,82 | 1,10 | 1,84 | 1,13 | 1,66 | 1,01 | 1,45 | 1,41 | 2,25 |
| **NS** | **10** | 1,25 | 1,99 | 1,31 | 2,23 | 1,08 | 1,64 | 2,03 | 3,02 | 1,23 | 1,70 |
| | **25** | 1,15 | 1,69 | 1,27 | 1,97 | 1,08 | 1,53 | 1,73 | 2,43 | 1,14 | 1,65 |
| | **50** | 1,08 | 1,60 | 1,06 | 1,71 | 0,95 | 1,45 | 1,24 | 1,86 | 1,12 | 1,67 |
| | **75** | 1,05 | 1,58 | 0,99 | 1,59 | 1,00 | 1,44 | 0,99 | 1,50 | 1,26 | 1,86 |
| | **90** | 1,11 | 1,72 | 1,08 | 1,69 | 1,11 | 1,60 | 0,97 | 1,40 | 1,31 | 2,06 |
| **AR(1) NS** | **10** | 1,35 | 1,99 | 1,35 | 2,26 | 1,11 | 1,66 | 2,01 | 3,13 | 1,28 | 1,79 |
| | **25** | 1,21 | 1,74 | 1,31 | 2,02 | 1,11 | 1,53 | 1,75 | 2,47 | 1,25 | 1,70 |
| | **50** | 1,11 | 1,62 | 1,02 | 1,77 | 0,99 | 1,42 | 1,30 | 2,00 | 1,19 | 1,72 |
| | **75** | 1,07 | 1,67 | 1,07 | 1,62 | 1,08 | 1,42 | 1,08 | 1,60 | 1,33 | 1,93 |
| | **90** | 1,19 | 1,70 | 1,06 | 1,73 | 1,14 | 1,59 | 0,98 | 1,39 | 1,42 | 2,17 |
| **AR(1) Daglish** | **10** | 1,42 | 2,17 | 1,51 | 2,51 | 1,24 | 1,83 | 2,27 | 3,40 | 1,40 | 1,93 |
| | **25** | 1,24 | 1,86 | 1,29 | 2,12 | 1,11 | 1,67 | 1,83 | 2,67 | 1,26 | 1,83 |
| | **50** | 1,15 | 1,73 | 1,14 | 1,89 | 1,05 | 1,54 | 1,37 | 2,13 | 1,27 | 1,85 |
| | **75** | 1,21 | 1,78 | 1,15 | 1,83 | 1,12 | 1,58 | 1,19 | 1,71 | 1,43 | 2,05 |
| | **90** | 1,23 | 1,86 | 1,13 | 1,88 | 1,15 | 1,66 | 1,09 | 1,53 | 1,43 | 2,24 |
| **AR(1) Dumas** | **10** | 1,41 | 2,16 | 1,47 | 2,49 | 1,23 | 1,82 | 2,23 | 3,39 | 1,38 | 1,91 |
| | **25** | 1,27 | 1,90 | 1,36 | 2,17 | 1,14 | 1,67 | 1,90 | 2,71 | 1,30 | 1,86 |
| | **50** | 1,14 | 1,72 | 1,13 | 1,88 | 1,03 | 1,53 | 1,35 | 2,11 | 1,26 | 1,86 |
| | **75** | 1,17 | 1,75 | 1,10 | 1,79 | 1,09 | 1,55 | 1,14 | 1,68 | 1,42 | 2,08 |
| | **90** | 1,24 | 1,86 | 1,15 | 1,89 | 1,18 | 1,68 | 1,09 | 1,51 | 1,54 | 2,33 |

In order to assess the validity of parameter stability for the methodologies investigated, I report the coefficient of variations (the ratio of the standard deviation to the mean) of the model's parameter estimates across the 358 weeks in table 8. It is natural to look at the standard deviation of the parameter estimates to find out if there is any considerable variation in the coefficient estimates from week to week, implying perhaps that the volatility function is not stable through time. However, the functional forms used by each methodology to fit the smile is significantly different from each other. Hence, comparing their stability comparisons using their standard deviations would be misleading. I have to resort to the coefficient of variation to make this comparison properly. Also, I are not able to include the Malz methodology in this section as its parameters are perfectly stable since they are constant all the time.

**Table 25:** Parameter Stability

The table reports the stability comparison of Nelson-Siegel, Dumas et. al. and Daglish et. al. method parameters for emerging and developed markets from Jan 2006 to Apr 2011.

| Stdev/Mean | Dumas et. al. | | | Nelson-Siegel | | | | Hull et. al. | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | a0 | a1 | a2 | Beta0 | Beta1 | Beta2 | Tau | c0 | c1 | c2 | c3 |
| **Emerging Market Currencies** | | | | | | | | | | | |
| **USDTRY** | 0,37 | 1,31 | 0,37 | 0,39 | 0,42 | 0,38 | 0,01 | 2,55 | 0,64 | 0,54 | 0,50 |
| **USDBRL** | 0,43 | 1,99 | 17,11 | 0,54 | 0,61 | 0,55 | 0,02 | 1,75 | 0,70 | 0,61 | 0,58 |
| **USDMXN** | 0,59 | 2,66 | 0,76 | 0,78 | 0,89 | 0,83 | 0,06 | 3,10 | 1,13 | 0,96 | 0,90 |
| **USDINR** | 0,37 | 3,48 | 0,62 | 0,63 | 0,83 | 0,69 | 0,03 | 7,63 | 1,12 | 0,88 | 0,81 |
| **USDZAR** | 0,33 | 1,11 | 0,55 | 0,39 | 0,44 | 0,47 | 0,03 | 5,19 | 0,89 | 0,66 | 0,58 |
| **Developed Market Currencies** | | | | | | | | | | | |
| **EURUSD** | 0,39 | 0,73 | 0,57 | 0,53 | 0,26 | 0,51 | 0,09 | 0,46 | 0,87 | 3,22 | 7,19 |
| **USDGBP** | 0,44 | 0,69 | 0,58 | 0,42 | 0,28 | 0,74 | 0,10 | 0,47 | 0,66 | 1,70 | 14,18 |
| **CHFUSD** | 0,31 | 0,83 | 0,55 | 0,43 | 0,31 | 0,68 | 0,07 | 0,38 | 1,34 | 4,11 | 1,48 |
| **USDJPY** | 0,45 | 0,78 | 0,46 | 0,58 | 0,22 | 0,41 | 0,12 | 0,71 | 1,46 | 5,38 | 11,10 |
| **USDCAD** | 0,40 | 2,34 | 0,45 | 0,44 | 0,74 | 0,52 | 0,06 | 0,52 | 2,26 | 1,65 | 0,91 |

From the forecasting results, it was clear that the accuracy and forecasting capabilities of all the methodologies are significantly better in developed market currencies than the emerging market currencies. Surprisingly, I do not observe this feature for the parameter stability. The coefficient of variation comparion is inconclusive for this case, it is more or less the same for both sets of currencies. One striking observation is that the coefficient $a_1$ and $a_2$ in Dumas methodology and $c_0$ and $c_3$ in Daglish's methodology show signs of instability for some specific currencies. See USDBRL and USDCAD currencies for $a_1$ and $a_2$ and see USDINR, USDGBP and USDJPY currencies for $c_0$ and $c_3$. Notice that these bursts of unstability never happen for Nelson-Siegel. It has a fairly small coefficient of variations for each of its parameters across all the currencies.

In order to understand the reason behind this, I further investigate the time series behavior of the parameters for each method. To examine explicitly the issue of coefficient stability, 2.5 has three panels containing plots of the time-series estimates of the implied volatilities for the models Nelson-Siegel, Dumas and Daglish. "This evidence indicates that the in-sample estimates for the Deterministic Volatility Function model seem to be unstable. This inference implies that changes in the coefficient estimates may not be entirely due to economic factors, but may be the result of overfitting." (Dumas et al. (1998)) Indeed Dumas arrives at the same conclusion in his paper and a similar argument is true for Daglish's methodology. The three parameters ($\beta_0$, $\beta_1$, $\beta_2$) of Nelson-Siegel methodology have economic interpretations, therefore immune to this deficiency. To more clearly express the connection of NS estimation parameters ($\beta_0$, $\beta_1$, $\beta_2$) with the empirical proxies (ATM, risk-reversal and butterly) , I put the time evolution of these proxies on top of the parameter estimates. As a last remark about NS, the parameter $\tau$ is harder to interpret in financial terms, but it is by far the most stable one among all the parameters. Hence, it has no significance as far as

the stability is concerned.

In order to explicitly examine the issue of parameter stability, graphical representations of at-the-money, risk-reversal and strangle factors of Nelson-Siegel method are given in 2.5. As it can be observed from the figure, method is capable of generating stable and realistic factors. Also, when I compare second and third parameters of Nelson-Seigel method and the method of Dumas et. al., it can be observed from 25 that deviation in NS factors is much smaller.

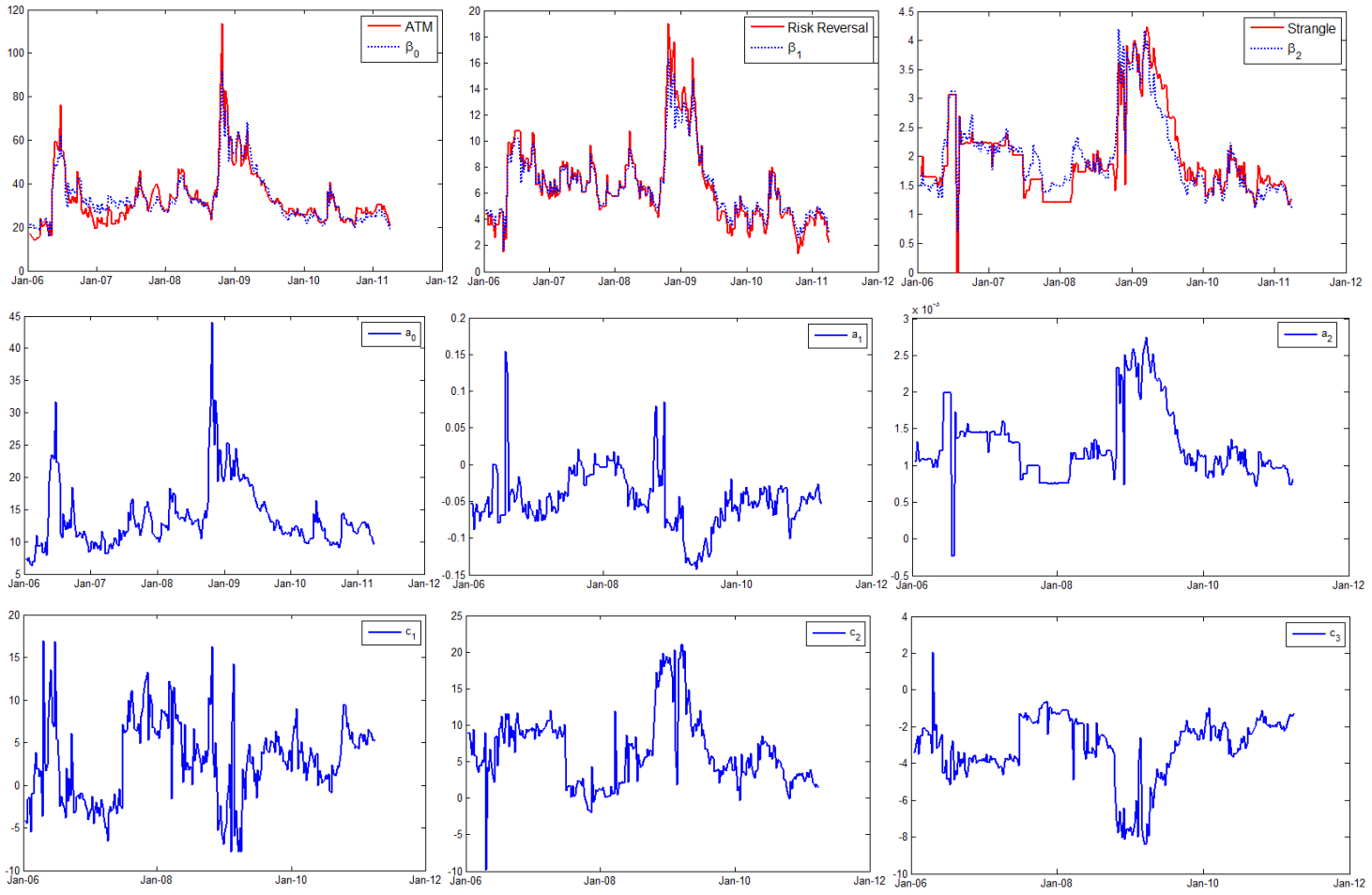


**Figure 54:** Stability of NS,Dumas et. al. and Daglish et. al. parameters for USD/TRY option date for the period Jan 2006-Apr 2011. The parameters are scaled to match the corresponding empirical quantity.

# CHAPTER III

# CONCLUSION

I began by showing why the NS and NSS functional forms were attractive and widely used for estimating the term structure of interest rates from bond prices. I then noted that the NS and in particular the NSS functions could lead to potential degeneracy that would make parameter estimation challenging. However, not all optimization methods are equally susceptible to these problems. The types of degeneracy that the NSS function is prone to creating greater problems for gradient and direct search methods then for methods based on global optimization methods. The relative immunity of global optimization algorithms to problems inherent in the NS and NSS functions was confirmed repeatedly in my in-sample stability and out-of-sample, and robustness tests. The global optimization methods, and in particular Hybrid PSO algorithm, also generally showed the best in- and out-of-sample goodness-of-fits. Only in the case computational speed was another method, the Nelder-Mead, clearly superior to the alternatives.

I believe that the method I use to arrive at starting values, derived from Diebold and Li (2006), starts all the single-point optimizations at a point that may be expected to be in the general region of the globe minimum. Researchers who do not use as carefully chosen starting values and are using gradient-based or other methods that rely on a single starting value, are likely to find their results to be less reliable than shown in this study, if they perform similar robustness checks.

My hypothesis is that the superior performance of the PSO methods lies in using

multiple starting values, employing a randomized search process, and keeping track of past best solutions (i.e., having memory). For ill-conditioned problems such as NS and NSS can give rise to, BFGS is subject to numerical problems. Variables that are not uniquely identified can cause oscillation between extreme solutions pairs when trying to find an improvement in the objective function using gradients that are near zero. Methods that do not rely on gradients are much less susceptible to problems when optimizing such ill-conditioned problems.

Single-point methods are susceptible to finding local, rather than global, minima. They are, therefore, critically dependent on the topography of the objective function surface and the initial starting value. As I have noted, estimating term structures from coupon bond prices presents the optimization algorithm with a very complex surface over which to search for a global minimum. Randomizing within a single-point search algorithm, as Simulated Annealing does, only partially mitigates the problem of converging on the first local minimum found in the search. The multiple starting point algorithms, Nelder-Mead, PSO, and Genetic algorithms, overcome the problem hanging up on local minimum, which single-point methods are prone to, by beginning with a wide range of starting points and then converging these to a single solution. In the process, these methods "sweep" a large area of the objective function surface, rather than following a single line. Finally, the memory of past individual particle and global best solutions inherent in the PSO class of optimization algorithms adds a level of information and flexibility that the other methods lack. The hybrid PSO improves on previous PSO methods by focusing exclusively on global best in moving particle forward after there has been a degree of convergence and individual particle best information is no longer as valuable.

For academics and practitioners estimating term structures, this study provides

clear evidence of the noise that injudicious choice of optimization method can introduce in the estimated values, as well as suggesting and validating a method that works well for the NSS model. For researchers interested in optimization methods, this study presents an example of a problem that is not suited to number of optimizations methods, including the defaults in most statistical packages, but can be handled by the newer global optimization methods and suggests why this is the case.

Unfortunately, gradient-based methods such as BFGS and GRG are the default methods used by statistical packages, and therefore are likely to be the ones being used when estimating term structures using the NS and NSS functions. My conclusion, therefore, is that empirical researchers and practitioners needing reliable estimates of the term structure should give careful consideration to the choice of functional form and the optimization method used to estimate its parameters.

I compared the optimization algorithms using some performance metrics and I have made several robustness checks such as application of the model and the optimization algorithms to the local currency bonds. Afterward, I checked the trading performance of the model in terms cheap and rich analysis. As a final step, I applied the hybrid PSO algorithm in a different asset class, which is foreign exchange options in emerging markets. I have seen that using a NSS type of deterministic implied volatility function for the FX options in emerging markets yields more stable results and has better predictive power and trading results compared to its peers.

# Bibliography


Afonso, Antonio, Manuel M. F. Martins. 2012. Level, slope, curvature of the sovereign yield curve, and fiscal behaviour. *Journal of Banking and Finance* **36**(6) 1789–1807.

Annaert, Jan, Anouk G.P. Claes, Marc J.K. deCeuster, Hairui Zhang. 2010. Estimating the yield curve using the Nelson-Siegel model.

Annaert, Jan, Anouk G.P. Claes, Marc J.K. deCeuster, Hairui Zhang. 2012. The estimation of Svensson model term structures and their volatilities.

Baxter, Martin, Andrew Rennie. 1996. *Financial calculus: an introduction to derivative pricing*. Cambridge university press.

BIS. 2005. Zero-coupon yield curves: Technical documentation. Tech. Rep. 25, Bank of International Settlements.

Black, F., M. Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy* **81**(3) 637–654.

Bliss, R. 1997a. Movements in the term structure of interest rates. *Economic Review* **82**(4) 16–33.

Bliss, R. 1997b. Testing term structure estimation methods. *Advances in Futures and Options Research* **9** 197–231.

Bloomberg. 2010. Bloomberg. Http://www.bloomberg.com.

Bolder, D., S. Gusba. 2002. Exponentials, polynomials, and fourier series: More yield curve modelling at the bank of canada. Tech. Rep. 29, Bank of Canada.

Bolder, D., D. Streliski. 1999. Yield curve modeling at the Bank of Canada. Tech. Rep. 84, Bank of Canada.

Brown, S., P. Dybvig. 1986. The empirical implications of the Cox, Ingersoll, Ross theory of the term structure of interest rates. *Journal of Finance* **41**(3) 617–630.

Campa, M., P. Chang, Reider L., W. Buiter, B. Eichengreen. 1997. Erm bandwidths for emu and after: Evidence from foreign exchange options. *Economic Policy* **12**(24) 53–89.

Castagna, A. 2010. *FX Options and smile risk*. John Wiley and Sons, New York.

Clerc, M., J Kennedy. 2002. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 58–73.

Courant, E., D. Hilbert. 1953. *Methods of Mathematical Physics*. John Wiley and Sons, Newyork.

Cox, J., S. Ross, M. Rubinstein. 1976. Option pricing: A simplified approach. *Journal of Financial Economics* **7**(3) 229–263.

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

Csajbók, Attila. 1999. Zero-coupon yield curve estimation from a central bank perspective.

Daglish, T., J. Hull, W. Suo. 2007. Implied volatility functions: Empirical tests. *Quantitative Finance* **7**(5) 507–524.

De Pooter, M. 2007. Examining the Nelson-Siegel class of term structure models: In-sample fit versus out-of-sample forecasting performance.

Derman, E., I. Kani, N. Chriss. 1996. Implied trinomial trees of the volatility smile. *Wilmott Magazine* .

Diebold, Francis X., Monika Piazzesi, Glenn D. Rudebusch. 2005a. Modeling bond yields in finance and macroeconomics. *American Economic Review* **95**(2) 415–420.

Diebold, Francis X., Monika Piazzesi, Glenn D. Rudebusch. 2005b. Modeling bond yields in finance and macroeconomics. PIER Working Paper Archive 05-008, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania. URL `http://ideas.repec.org/p/pen/papers/05-008.html`.

Diebold, F.X., C. Li. 2006. Forecasting the term structure of government bond yields. *Journal of Econometrics* **130** 337–364.

Dierckx, P. 1993. *Curve and Surface Fitting with Splines*. Claredon Press.

Duarte, Jefferson, Francis A Longstaff, Fan Yu. 2007. Risk and return in fixed-income arbitrage: Nickels in front of a steamroller? *Review of Financial Studies* **20**(3) 769–811.

Dumas, B., J. Fleming, R. Whaley. 1998. Implied volatility functions: Empirical tests. *Journal of Finance* **53**(6) 2059–2106.

Dupire, A. 1994. Pricing with a smile. *Risk* **7** 18–20.

Fama, E., R. Bliss. 1987. The information in long-maturity forward rates. *American Economic Review* **77**(4) 16–33.

Fengler, Matthias R. 2006. *Semiparametric modeling of implied volatility*. Springer Science & Business Media.

Fisher, M., D. Nychka, D. Zervos. 1994. Fitting the term structure of interest rates with smoothing splines. Tech. rep., U.S. Federal Reserve Board.

Flavell, Richard, Nigel Meade, Gerry Salkin. 1994. The gilt market: Models and model-based trading. *Journal of the Operational Research Society* 392–408.

Fylstra, I.C. 1998. Design and use of the Microsoft Excel Solver. *Interfaces* 29–55.

Gilli, Manfred, Stefan Grose, Enrico Schumann. 2010. Calibrating the Nelson-Siegel-Svensson model.

Gimeno, Ricardo, Juan M. Nave. 2006. Genetic algorithm estimation of interest rate term structure.

Gurkaynak, R., B. Sack, J. Wright. 2007. The U.S. Treasury yield curve: 1961 to the present. *Journal of Monetary Economics* 2291–2304.

Hagan, Patrick S, Deep Kumar, Andrew S Lesniewski, Diana E Woodward. 2002. Managing smile risk. *The Best of Wilmott* 249.

Heston, S.L. 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of financial studies* **6**(2) 327–343.

Hull, J.C., A. White. 1987. The pricing of options on assets with stochastic volatilities. *Journal of Finance* (42) 281–300.

Ioannides, M. 2003. A comparison of yield curve estimation techniques using UK data. *Journal of Banking and Finance* **27** 1–26.

Jackwerth, Jens Carsten, Mark Rubinstein. 1996. Recovering probability distributions from option prices. *The Journal of Finance* **51**(5) 1611–1631.

Jankowitsch, Rainer, Michaela Nettekoven. 2008. Trading strategies based on term structure model residuals. *The European Journal of Finance* **14**(4) 281–298.

Jankowitsch, Rainer, Stefan Pichler. 2002. Parsimonious estimation of credit spreads. *Available at SSRN 306779* .

Judd, K.L. 1998. *Numerical Methods in Economics*. MIT Press, Cambridge, MA.

Kirkpatrick, S., C. Gelatt, M. Vecchi. 1983. Optimization by simulated annealing. *Science* **220** 671–680.

Krippner, Leo. 2012. A theoretical foundation for the Nelson and Siegel class of yield curve models.

Kruger, CJC. 2003. Constrained cubic spline interpolation. *Chemical Engineering Applications* **1**(1).

Lagarias, J., J. Reeds, M. Wright, P. Wright. 1965. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal on Optimization* **9** 112–147.

Lancester, P., K. Salkauskas. 1986. *Curve Fitting and Surface Fitting: An Introduction*. Academic Press.

Lasdon, L.S., A.D. Waren, A. Jain, M. Ratner. 1978. Design and testing of a generalized reduced gradient code for nonlinear programming. *ACM Transactions on Mathematical Software* **4** 34–50.

Lekkos, I. 2001. Factor models and the correlation structure of interest rates: Some evidence for USD, GBP, DEM and JPY. *Journal of Banking and Finance* **8**(25) 1427–1445.

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

Litterman, R., J. Scheinkman. 1994. Explorations into factors explaining money market returns. *Journal of Finance* **49** 1861–1882.

Malz, A.M. 1996. Using option prices to estimate realignment probabilities in the european monetary system: the case of sterling-mark. *Journal of International Money and Finance* 717–748.

Malz, A.M. 1997. Estimating the probability distribution of the future exchange rate from options prices. *Journal of Derivatives* **81**(3) 18–36.

Manousopoulos, P., M. Michalopoulosa. 2007. Comparison of non-linear optimization algorithms for yield curve estimation. *European Journal of Operational Research* **192** 594–602.

Martellini, L., P. Priaulet, Priaulet S. 2003. *Fixed Income Securities: Valuation, Risk Management and Portfolio Strategies*. John Wiley and Sons, Newyork.

McCulloch, J. H. 1971. Measuring the term structure of interest rates. *The Journal of Business* **44** 19–31.

McCulloch, J. H. 1975. The tax-adjusted yield curve. *The Journal of Finance* **30** 881–830.

Mikosch, Thomas. 1998. *Elementary stochastic calculus with finance in view*. World scientific.

Nelder, J.A., R. Mead. 1965. A simplex method for function minimization. *Computer Journal* **7** 308–313.

Nelson, C.R., A.F. Siegel. 1987. Parsimonious modeling of yield curves. *The Journal of Business* **60**(4) 473–489.

Pedersen, M.E.H., Chipperfield A.J. 2010. Simplifying particle swarm optimization. *The Journal of Applied Soft Computing* 618–628.

Powell, M.J.D. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal* **7** 155–162.

Rubinstein, M. 1992. Implied binomial trees. *Journal of Finance* **49**(4) 771–818.

Seppala, J., P. Viertio. 1996. The term structure of interest rates: Estimation and interpretation. Tech. Rep. 19, Bank of Finland.

Sercu, Piet, Xueping Wu. 1997. The information content in bond model residuals: An empirical study on the belgian bond market. *Journal of Banking & Finance* **21**(5) 685–720.

Shea, G. S. 1984. Pitfalls in smoothing interest rate term structure data: Equilibrium models and spline approximations. *Journal of Financial and Quantitive Studies* **19**(3) 253–269.

Shi, Y., R. Eberhart. 1998. A modified particle swarm optimizer. *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence* 69–73.

Shimko, D. 1993. Bounds of probability. *Risk* **6** 33–37.

Shimko, David C. 1992. *Finance in continuous time: a primer*. Kolb Publ.

Skiadopoulos, George, Stewart Hodges, Les Clewlow. 2000. The dynamics of the s&p 500 implied volatility surface. *Review of Derivatives Research* **3**(3) 263–282.

Subramanian, KV. 2001. Term structure estimation in illiquid markets. *The Journal of Fixed Income* **11**(1) 77–86.

Svensson, Lars EO. 1994. Estimating and interpreting forward interest rates: Sweden 1992-1994. Tech. rep., National Bureau of Economic Research.

Trelea, I.C. 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters 85* 317–325.

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

Vasicek, O. A., H. G. Fong. 1981. Term structure modelling using exponential splines. *The Journal of Finance* **37** 339–348.

Wu, Xueping. 1995. Term structures in the belgian market: model estimation and pricing error analysis. Ph.D. thesis, Katholieke Universiteit, Faculteit der Economische en Toegepaste Economische Wetenschappen.

Wystup, Uwe. 2003. The market price of one-touch options in foreign exchange markets. *Derivatives week* **12**(13) 8–9.

Yallup, Peter J. 2012. Models of the yield curve and the curvature of the implied forward rate function. *Journal of Banking and Finance* **36**(1) 121–135.

# CHAPTER IV

# APPENDIX

## *4.1 Review of Splines*

In this section I will take a look to the basic concepts about *splines*. I tried to explain the concepts in formal and informal ways to make it easier to understand. First I will give a formal definition of *splines* and introduce the notation then, starting with the *linear spline* derivation, I will discuss about the conceptual and computational nature of *splines*. At last I will examine the functions offered in McCulloch (1971) and find the logic behind them.

*Splines* are *piecewise polynomials* defined between *knot points*. The word *spline* comes from the tools (flat splines) used to draw the curves that cannot be drawn with French curves. Flat splines are a piece of thin flat wood or some other flexible material. It's fixed at some points and passes through the data points.

### 4.1.1 Spline Definition

$(n+1)$th order ($n$th degree) *spline* defined as follows:

$$S : [a, b] \to \mathbb{R}$$

*Splines* are defined between the first and last point of the *knot sequence*. The *knot sequence* $t_i$ is strictly increasing.

$$a = t_0 < t_1 < ... < t_{k-1} < t_k = b, i = 0, ..., k$$

I define *piecewise polynomials* that form our *spline* between two adjacent points.

$$P_i : [t_i, t_{i+1}] \to \mathbb{R}$$

$$P_i = a_{i0} + a_{i1}x + a_{i1}x^2 + ... + a_{in}x^n$$

$$S(t) = P_0(t), t_0 \leq t < t_1,$$

$$S(t) = P_1(t), t_1 \leq t < t_2,$$

$$\vdots$$

$$S(t) = P_{k-1}(t), t_{k-1} \leq t < t_k,$$

At last I impose the *continuity* constraints.

$$P_i(x_i) = y_i = P_{i-1}(x_i), i = 1, ..., n-1$$

$$P_i'(x_i) = P_{i-1}'(x_i), i = 1, ..., n-1$$

$$P_i''(x_i) = P_{i-1}''(x_i), i = 1, ..., n-1$$

$$\vdots$$

$$P_i^{(n-1)}(x_i) = P_{i-1}^{(n-1)}(x_i), i = 1, ..., n-1$$

I denote the *continuity* of *splines* as following. This expression means that our function has a continuous $(n-1)$th derivative.

$$S(t) \in C^{(n-1)}$$

*Continuity* is an important parameter of splines. In statistical applications like our subject, yield curve estimation, *cubic splines* satisfy our expectations. The more *continuity* is the more fluctuation of curve.

### 4.1.2 Linear Splines

The simplest functions that I can fit on data is *linear splines*. I can explain the *linear splines* as the lines drawn with a ruler between the adjacent data points. They are continuous but their derivatives are not. It's useful and easy to implement for plotting data but they are problematic for analytical purposes. Some examples of linear splines are data plots in MATLAB and Excel and the yield curves on Bloomberg terminal.

Here I explain two derivation methods for *linear splines*. Both of them gives the same result but they differ in the computational aspects.

#### *4.1.2.1 Linear Spline Derivation - I*

All linear splines can be defined as

$$S(x) = a_0|t - t_0| + a_1|t - t_1| + ... + a_k|t - t_k|$$

In order to determine the coefficients $a_i$ I should solve he linear system that

$$S(t_i) = y_i, i = 0, 1, ..., k$$

I can express this linear system in matrix form as

$$\begin{bmatrix} 0 & |t_0 - t_1| & \dots & |t_0 - t_n| \\ |t_1 - t_0| & 0 & \dots & |t_1 - t_n| \\ \vdots & \vdots & \ddots & \vdots \\ |t_n - t_0| & |t_n - t_1| & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Even though these expressions do not look like the formal *spline* definition, just because the *absolute value* function is a *piecewise linear function* the linear combination of the *absolute values* coincide with the definition given above (Bolder and Gusba, 2002).

#### *4.1.2.2 Linear Spline Derivation - II*

Alternatively I can define the *linear splines* by using the *basis functions*. The *basis function* that I use satisfies the constraint that $B_i(t_j) = \delta_{ij}$. $\delta_{ij}$ is the *Kronecker delta* that is equal to 1 if $i = j$ and 0 if $i \neq j$.

$$B_0(t) = \left\{ \begin{array}{cc} \frac{t_1 - 1}{t_1 - t_0} & : t_0 \leq t < t_1 \\ 0 & : t_1 \leq t \end{array} \right\}$$

$$B_i(t) = \left\{ \begin{array}{cc} 0 & : t < t_{i-1} \\ \frac{t - t_{i-1}}{t_i - t_{i-1}} & : t_{i-1} \leq t < t_i \\ \frac{t_{i+1} - t}{t_{i+1} - t_i} & : t_i \leq t < t_{i+1} \\ 0 & : t_{i+1} \leq t \end{array} \right\}$$

$$B_k(t) = \left\{ \begin{array}{ll} 0 & : t < t_{k-1} \\ \frac{t-t_{k-1}}{t_k - t_{k-1}} & : t_{i-1} \leq t < t_i \end{array} \right\}$$

The *basis functions* also called *pyramide* or *tent* functions due to their shapes. The logic of these basis functions introduces us the further concepts about higher order splines and *B-splines*. See the figure below for the *tent* functions.

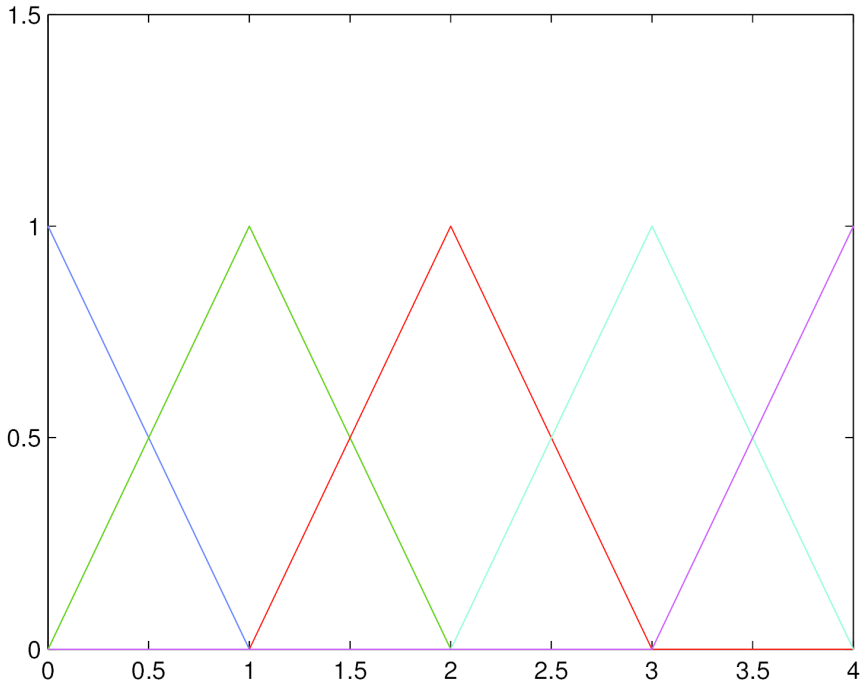


**Figure 55:** *Pyramide* or *tent* functions

The coefficients for the *basis functions* are the ordinate values of data points itself (remember that $B_i(t_j) = \delta_{ij}$.).

$$S(t) = \sum_{i=0}^{k} y_i \cdot B_i(t)$$

The spline described above can also be expressed as follows:

$$S(t) = y_{i-1}\frac{|t - t_i|}{t_i - t_{i-1}} + y_i\frac{|t - t_{i-1}|}{t_i - t_{i-1}} : t_{i-1} \leq t < t_i$$

(Lancester and Salkauskas, 1986)

### 4.1.3 Cubic Spline Derivation

*Cubic splines* are frequently used in *curve fitting* applications. They have a continuous second derivative in most situations. The higher order splines don't have too much benefit but they increase the computational burden. Using the definition above, I know that a *cubic spline* defined as follows:

$$S(t) = \left\{ \begin{array}{l} a_{0,0} + a_{0,1}(t) + a_{0,2}(t)^2 + a_{0,3}(t)^3 : t_0 \leq t < t_1 \\ a_{1,0} + a_{1,1}(t) + a_{1,2}(t)^2 + a_{1,3}(t)^3 : t_1 \leq t < t_2 \\ \vdots \\ a_{(k-1),0} + a_{(k-1),1}(t) + a_{(k-1),2}(t)^2 + a_{(k-1),3}(t)^3 : t_{k-1} \leq t < t_k \end{array} \right\}$$

I could try to find the parameters by using this definition but this makes the matrix I would try to solve too large. Moreover the matrix would have computational difficulties. Instead I suggest an alternative derivation. I assume that the second derivative of the *spline* takes the values $z_i, i = 0, 1, ..., k$ at *knot points* that I find by constructing an easy-to-solve linear system. From the formal definition of *splines* I know that *cubic splines* should have continuous second derivative at *knot points*. I define the second derivative of *cubic spline* as a *linear spline*.

$$z_i = S''(t_i)$$

$$h_i = t_{i+1} - t_i$$

I used the second *linear spline* method to define the second derivative and then take the integral of the expression in order to find the *spline*.

$$S''(t) = z_{i-1}\frac{|t-t_i|}{t_i - t_{i-1}} + z_i\frac{|t-t_{i-1}|}{t_i - t_{i-1}}$$

$$S'(t) = \int \left( z_{i-1}\frac{t_i - t}{t_i - t_{i-1}} + z_i\frac{|t-t_{i-1}|}{t_i - t_{i-1}} \right) dt$$

$$S'(t) = -\frac{z_{i-1}}{2h_i}(t_i - t)^2 + \frac{z_i}{2h_i}(t - t_{i-1})^2 + C_i,$$

$$S(t) = \int \left( -\frac{z_{i-1}}{2h_i}(t_i - t)^2 + \frac{z_i}{2h_i}(t - t_{i-1})^2 + C_i \right) dt$$

$$S(t) = \frac{z_{i-1}}{6h_i}(t_i - t)^3 + \frac{z_i}{6h_i}(t - t_{i-1})^3 + C_i x + D_i,$$

Here I used a math trick. I changed the integral constants $C_i$ and $D_i$ in order to make change of variables possible.

$$C_i = -c_i + d_i$$

$$D_i = c_i t_i - d_i t_{i-1}$$

$$S(t) = \frac{z_{i-1}}{6h_i}(t_i - t)^3 + \frac{z_i}{6h_i}(t - t_{i-1})^3 + c_i(t_i - t) + d_i(t - t_{i-1}).$$

By solving $S(t_{i-1}) = y_{i-1}$ I get that

$$c_i = \frac{1}{h_i}\left( y_{i-1} - \frac{z_{i-1}h_i^2}{6} \right)$$

In a similar way I solve $S(t_i) = y_i$ and I get that

$$d_i = \frac{1}{h_i}\left( y_i - \frac{z_i h_i^2}{6} \right)$$

Since the second derivative of the spline is continuous, the first derivative should also be continuous. Using this fact I obtain following equations.

$$S'(t) = \frac{\partial S(t)}{\partial t} = -\frac{z_{i-1}}{2h_i}(t_i - t)^2 + \frac{z_i}{2h_i}(t - t_{i-1})^2 - \frac{1}{h_i}\left(y_{i-1} - \frac{z_{i-1}h_i^2}{6}\right) + \frac{1}{h_i}\left(y_i - \frac{z_i h_i^2}{6}\right)$$

$$\lim_{t \to t_i^-} S'(t) = S'(t_i^-) = \frac{z_i h_i}{3} + \frac{m_{i-1}h_i}{6} + \frac{y_i - y_{i-1}}{h_i}$$

$$\lim_{t \to t_i^+} S'(t) = S'(t_i^+) = -\frac{z_i h_{i+1}}{3} - \frac{m_{i-1}h_i}{6} + \frac{y_{i+1} - y_i}{h_{i+1}}$$

If I solve $S'(t_i^-) = S'(t_i^+)$ I get

$$\Delta_i = \frac{h_i}{h_i + h_{i+1}} m_{i-1} + 2m_i + \frac{h_{i+1}}{h_i + h_{i+1}} m_{i+1} = \frac{6\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right)}{h_i + h_{i+1}}, i = 1, 2, ..., k-1$$

As you can see I can define this equation only for the *intermediate knot points*. This means that I have $k-1$ equation where I have $k+1$ coefficients. The first and last knot points don't have any constraints so I can freely determine the values that second derivative of *spline* takes there. I chose to make them zero and I get the system follows.

$$\begin{bmatrix} 2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1-\lambda_1 & 2 & \lambda_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1-\lambda_2 & 2 & \lambda_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1-\lambda_{k-1} & 2 & \lambda_{k-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_{k-1} \\ z_k \end{bmatrix} = \begin{bmatrix} 0 \\ \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{k-1} \\ 0 \end{bmatrix}$$

where $\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}$.

(Bolder and Gusba, 2002)

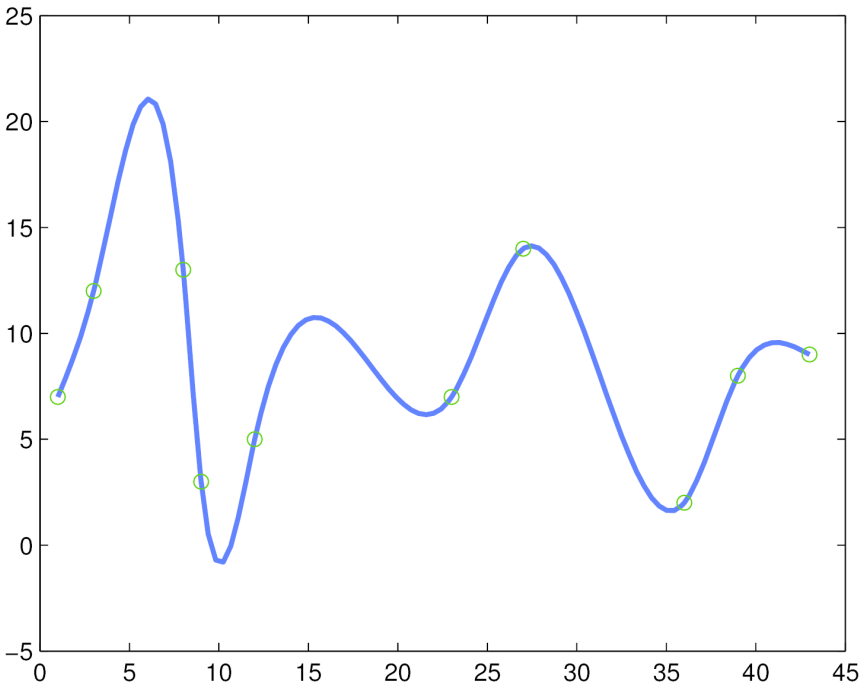


**Figure 56:** Cubic spline

### 4.1.4 B-Splines

In our final *spline* derivation, I use *basis functions* called *B-splines*. *B-splines* are *piecewise functions* whose linear combinations includes every *spline* of the same order. Using *B-splines* make the *spline fitting* easier and more flexible.

I define the *B-spline* of order 1 (degree 0) as follows.

$$B_{1,i}(t) = \left\{ \begin{array}{ll} 0 & : t < t_i \\ 1 & : t_i \leq t < t_{i+1} \\ 0 & : t_{i+1} \leq t \end{array} \right\}$$

For the higher degree I can use the recurrence relation.

$$B_{n,i}(t) = \frac{t - t_i}{t_{i+j-1} - t_i} B_{n-1,i}(t) + \frac{t_{i+j} - t}{t_{i+j} - t_{i+1}} B_{n-1,i+1}(t)$$

See the figure following

*B-splines* never take negative values and they have local support.

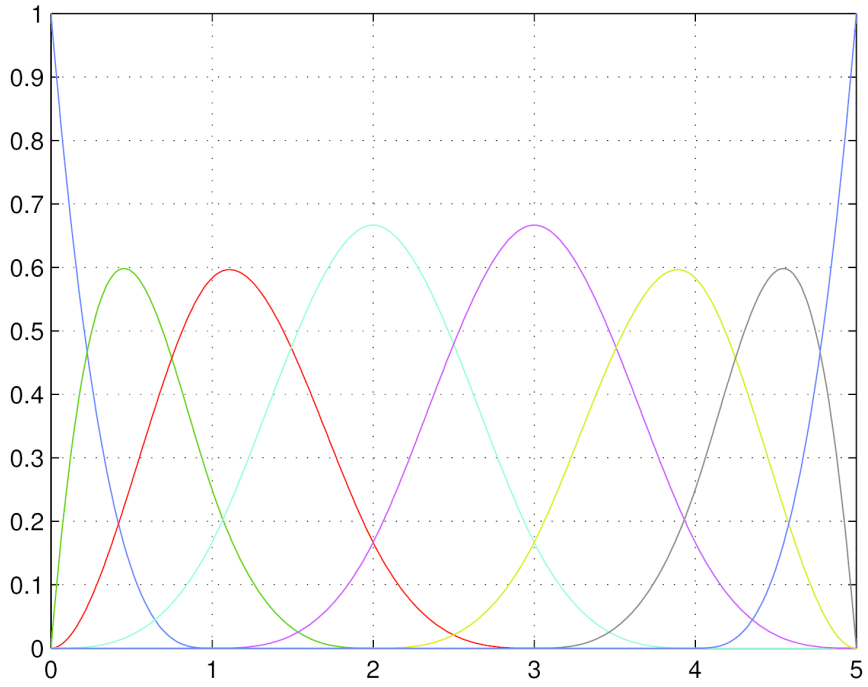


**Figure 57:** B-Spline Basis Functions for knot sequence $0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5$

$$B_{n,i}(t) \geq 0, t \in [t_i, t_i + n)$$

$$B_{n,i}(t) = 0, t \notin [t_i, t_{i+n})$$

As I defined *B-splines* by the recurrence relation, I can also define their derivatives

recursively.

$$B'_{n,i}(t) = (n-1)\left[\frac{B_{n-1,i}(t)}{t_{i+n-1}-t_i} - \frac{B_{n-1,i+1}(t)}{t_{i+n}-t_{i+1}}\right]$$

Integral of B-splines seem to be more complex.

$$\int_{t_i}^{t} B_{n,i}(u)du = \left\{ \begin{array}{cc} 0 & t \le t_i \\ \frac{t_{i+n}-t_i}{n} & t \ge t_{i+n} \\ \frac{t_{i+n}-t_i}{n}\sum_{j=0}^{n-1}\frac{t-t_{i+j}}{t_{i+n}-t_{i+j}}B_{n-j,i+j}(t) & t_i < t < t_{i+n} \end{array} \right\}$$

Integral of B-Splines are crucial when I impose a penalty for roughness.

After having the knowledge of these important properties I can dig into the *spline fitting* with *cubic B-splines*.

As proven in advanced texts all splines of order $n$ with knot sequence $t_i, i = 0, 1, ..., k$ can be obtained with the linear combination of the B-splines of same order. I denote the coefficients of the *B-splines* with the sequence $a_i$ and the column vector $\tilde{a}$ and the values of *B-spline* functions at $t$ with row vector $\tilde{B}(t)$.

In our sample the *data points* I try to fit and the *knot points* coincide. I have $k$ points to fit. At these points $y_i = \tilde{B}(t_i) \cdot \tilde{a}$. Now I have $k+1$ equations. For the end-point conditions (in this paper I use *natural splines* whose $(n-2)$th derivatives at end points are equal to zero.) I get 2 additional equations so I need $k+3$ *B-splines* to find a fit. This implies that I need $k+7$ knot points where I have $k+1$ of them. Now I should create an *augmented knot sequence* with additional *knot points* $t_{-3}, t_{-2}, t_{-1}, t_{k+1}, t_{k+2}, t_{k+3}$ which satisfy $t_{-3} = t_{-2} = t_{-1} = t_0 < ... < t_k = t_{k+1} = t_{k+2} = t_{k+3}$. The new knot point sequence requires us to define B-splines with coincident knots.

B-splines are defined for the knots $t_i = t_{i+1} = ... = t_{i+n-1} < t_{i+n}$ as follows.

$$B_{n,i}(t) = \left\{ \begin{array}{cc} \left(\frac{t_{i+n}-t}{t_{i+n}-t_i}\right)^{n-1} & t \in [t_i, t_{i+n}] \\ 0 & t \notin [t_i, t_{i+n}] \end{array} \right\}$$

In a similar way, of the knots $t_i < t_{i+1} = ... = t_{i+n-1} = t_{i+n}$:

$$B_{n,i}(t) = \left\{ \begin{array}{cc} \left(\frac{t-t_i}{t_{i+n}-t_i}\right)^{n-1} & t \in [t_i, t_{i+n}] \\ 0 & t \notin [t_i, t_{i+n}] \end{array} \right\}$$

At last I construct our system as follows.

$$\begin{bmatrix} \tilde{B}''(t_0) \\ \tilde{B}(t_0) \\ \tilde{B}(t_1) \\ \vdots \\ \tilde{B}(t_k) \\ \tilde{B}''(t_k) \end{bmatrix} \cdot \begin{bmatrix} a_{-3} \\ a_{-2} \\ a_{-1} \\ \vdots \\ a_{k-2} \\ a_{k-1} \end{bmatrix} = \begin{bmatrix} 0 \\ y_0 \\ y_1 \\ \vdots \\ y_k \\ 0 \end{bmatrix}$$

Actually the matrix of B-spline values whose rows are $\tilde{B}''(t_0)$, $\tilde{B}(t_i)$ and $\tilde{B}''(t_k)$ is almost tridiagonal so the calculations become too easy.

$$\begin{bmatrix} B''_{4,-3}(t_0) & B''_{4,-2}(t_0) & B''_{4,-1}(t_0) & 0 & \dots & 0 & 0 & 0 \\ B_{4,-3}(t_0) & B_{4,-2}(t_0) & B_{4,-1}(t_0) & 0 & \dots & 0 & 0 & 0 \\ 0 & B_{4,-2}(t_1) & B_{4,-1}(t_1) & B_{4,0}(t_1) & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & B_{4,k-3}(t_k) & B_{4,k-2}(t_k) & B_{4,k-1}(t_k) \\ 0 & 0 & 0 & 0 & \dots & B''_{4,k-3}(t_k) & B''_{4,k-2}(t_k) & B''_{4,k-1}(t_k) \end{bmatrix}$$

(Dierckx, 1993)

### 4.1.5 Cubic Hermite Splines

Another spline worths to investigate is cubic Hermite spline. Similar to the other examples discussed in this paper, Hermite splines are also cubic but there is a difference in the continuity equations. Instead of the second-order continuity Hermite spline fix the first-order derivative to a predetermined number. This provides a chance to arrange the slope at knot points thus modify the curve in order to increase smoothness. But it also causes a discontinuous second derivative or so-called 'knuckles' in the first derivative curves.

To construct the Hermite spline, one can use the four basis functions which have certain properties. The coefficients of these basis functions are desired function and derivative values for the both end of the interval.

$$B_{0,0}(t) = 2(\frac{t-t_i}{t_{i+1}-t_i})^3 - 3(\frac{t-t_i}{t_{i+1}-t_i})^2 + 1$$

$$B_{1,0}(t) = (\frac{t-t_i}{t_{i+1}-t_i})^3 - 2(\frac{t-t_i}{t_{i+1}-t_i})^2 + \frac{t-t_i}{t_{i+1}-t_i}$$

$$B_{0,1}(t) = -2(\frac{t-t_i}{t_{i+1}-t_i})^3 + 3(\frac{t-t_i}{t_{i+1}-t_i})^2$$

$$B_{1,1}(t) = (\frac{t-t_i}{t_{i+1}-t_i})^3 - (\frac{t-t_i}{t_{i+1}-t_i})^2$$

The plots of these basis functions are following:

The spline between $t_i$ and $t_{i+1}$ is defined as

$$y_i B_{0,0} + s_i B_{1,0} + y_{i+1} B_{0,1} + s_{i+1} B_{1,1}$$

The second issue with the Hermite splines is the choice of the derivatives. One of them is the Bessel Method which chooses the derivatives at the knot point as

$$s_0 = \frac{1}{t_2 - t_0}\left[\frac{(t_2 + t_1 - 2t_0)(y_1 - y_0)}{t_1 - t_0} - \frac{(t_1 - t_0)(y_2 - y_1)}{t_2 - t_1}\right]$$

$$s_i = \frac{1}{t_i + 1 - t_i - 1}\left[\frac{(t_{i+1} - t_i)(y_i - y_{i-1})}{t_i - t_{i-1}} - \frac{(t_i - t_{i-1})(y_{i+1} - y_i)}{t_{i+1} - t_i}\right]$$

$$s_k = \frac{1}{t_k - t_{k+2}}\left[\frac{(t_k - t_{k-1})(y_{k-1} - y_{k-2})}{t_{k-1} - t_{k-2}} - \frac{(2t_k - t_{k-1} - t_{k-2})(y_k - y_{k-1})}{t_k - t_{k-1}}\right]$$

Another method is the constrained splines suggested by Kruger (2003). This time the derivatives at the knot points are:

$$s_i = \left\{ \begin{array}{ll} 0 & : \text{if slope changes sign} \\ \frac{2}{\frac{t_{i+1}-t_i}{y_{i+1}-y_i} + \frac{t_i - t_{i-1}}{y_i - y_{i-1}}} & : \text{otherwise} \end{array} \right\}$$

for the intermediate points. For the end-points:

$$s_0 = \frac{3}{2}\frac{y_1 - y_0}{t_1 - t_0} - \frac{s_1}{2}$$

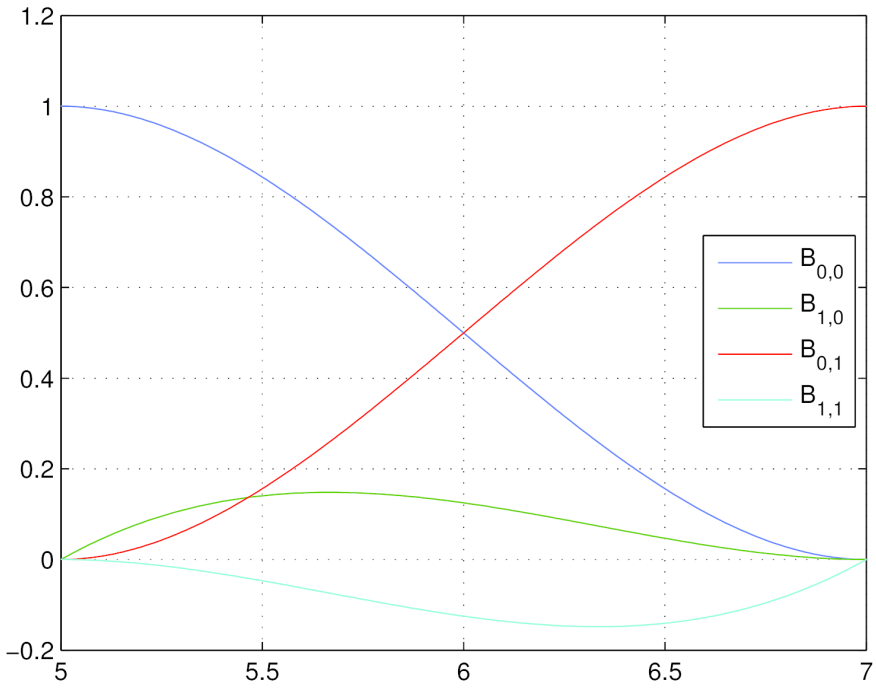


**Figure 58:** Hermite spline basis functions

$$s_k = \frac{3}{2}\frac{y_k - y_{k-1}}{t_k - t_{k-1}} - \frac{s_{k-1}}{2}$$

### 4.1.6 Least Squares Criterion

In some situations I cannot constructs such a linear system to find spline coefficients. This can be caused by that I do not observe the data points that I try to fit or I just do not want a perfect fit for some reason. In this situations I should find the spline with minimal deviation from data points. In order to do this, I try to minimize the sum of the residuals.

$$\epsilon_{total} = \sum |S(t_i) - y_i|$$

As you can realize the measure of errors given above is not differentiable so it is harder to find the minimum of it. Instead I use a differentiable measure of errors, simply the sum of squares of residuals.

$$l^2 = \sum [S(t_i) - y_i]^2$$

If I rewrite the equation in matrix form, I obtain that:

$$l^2 = \epsilon^T \epsilon$$

where

$$\epsilon = \begin{bmatrix} \tilde{B}''(t_0) \\ \tilde{B}(t_0) \\ \tilde{B}(t_1) \\ \vdots \\ \tilde{B}(t_k) \\ \tilde{B}''(t_k) \end{bmatrix} \cdot \begin{bmatrix} a_{-3} \\ a_{-2} \\ a_{-1} \\ \vdots \\ a_{k-2} \\ a_{k-1} \end{bmatrix} - \begin{bmatrix} 0 \\ y_0 \\ y_1 \\ \vdots \\ y_k \\ 0 \end{bmatrix}$$

Solving the equation $\frac{\partial l^2}{\partial \vec{a}} = 0$ I find:

$$\vec{a} = (B^T B)^{-1} B^T \vec{y}$$

### 4.1.7 Smoothing Criterion

In some cases, a perfect fit is undesirable since it cause dramatic moves in curve. To keep balance between smoothness and goodness of fit, I employ a penalty for excess roughness. Roughness at a point implied by curvature of curve at that point. The sharper the corner, the smaller the curvature. Since I wanted our penalty function to increase when moves of the function sharper I employ integral of the square of the second derivative.

$$\int_{t_0}^{t_k} (S''(t))^2 dt = \int_{t_0}^{t_k} (a^T \cdot (\tilde{B}''(t))(\tilde{B}''(t))^T \cdot a dt = a^T \cdot \int_{t_0}^{t_k} ((\tilde{B}''(t))(\tilde{B}''(t))^T dt \cdot a$$

Since the matrix $H = \int_{t_0}^{t_k} ((\tilde{B}''(t))(\tilde{B}''(t))^T dt$ is constant for a certain knot point sequence, such a formulation makes the calculations easier. It's also worth to note that $H$ is a band-diagonal matrix since the $B''_{k,i} B''_{k,i}$ is equal to 0 where $|i - j| < 2(k - 1)$.

### 4.1.8 McCulloch Estimation Method

The essence of McCulloch method is the use of linear relation between discount factor and bond prices. This method does not directly estimate spot rate, it estimates discount factor which can be transformed into spot rate function.

In McCulloch (1971), the price of the coupon bond calculated as using continuous coupon payment. See the following for McCulloch's simplified bond price formulation:

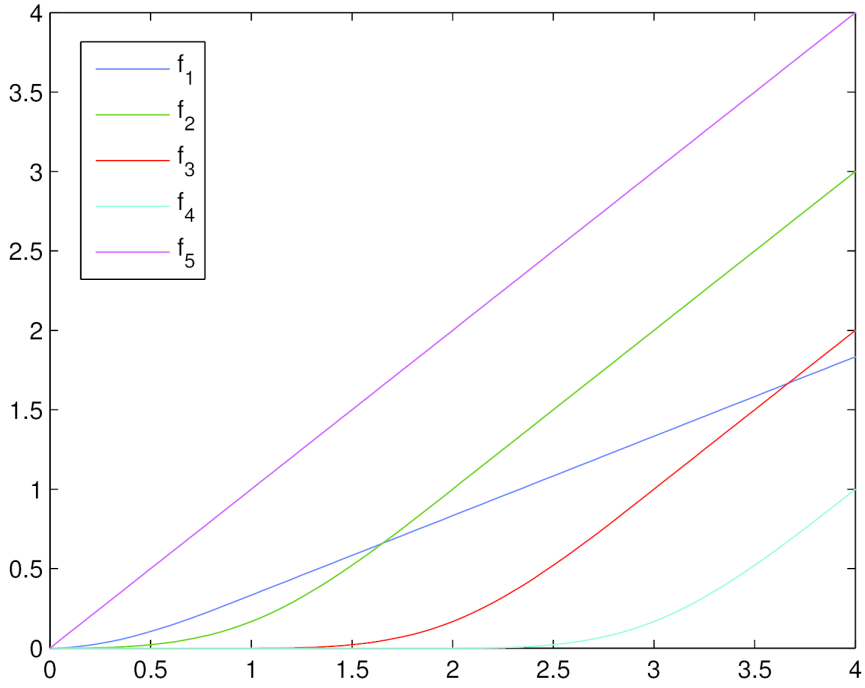


**Figure 59:** McCulloch's Basis Functions

$$p(t,T) = c_T \cdot d(t,T) + c \int_t^T d(t,m) \cdot \partial m$$

In our research I used discrete coupon stream that I described above. It results a better estimation and less error.

McCulloch postulates a linear combination of *basis functions*:

$$d(t,m) = a_0 + \sum_{j=1}^{k} a_j f_j(m-t)$$

As the nature of discount factor dictates, the discount factor for zero maturity $d(t,t)$ should be equal to one. This requires $a_0 = 1$ and $f_j(0) = 0$ for $j = 1, 2, ..., k$.

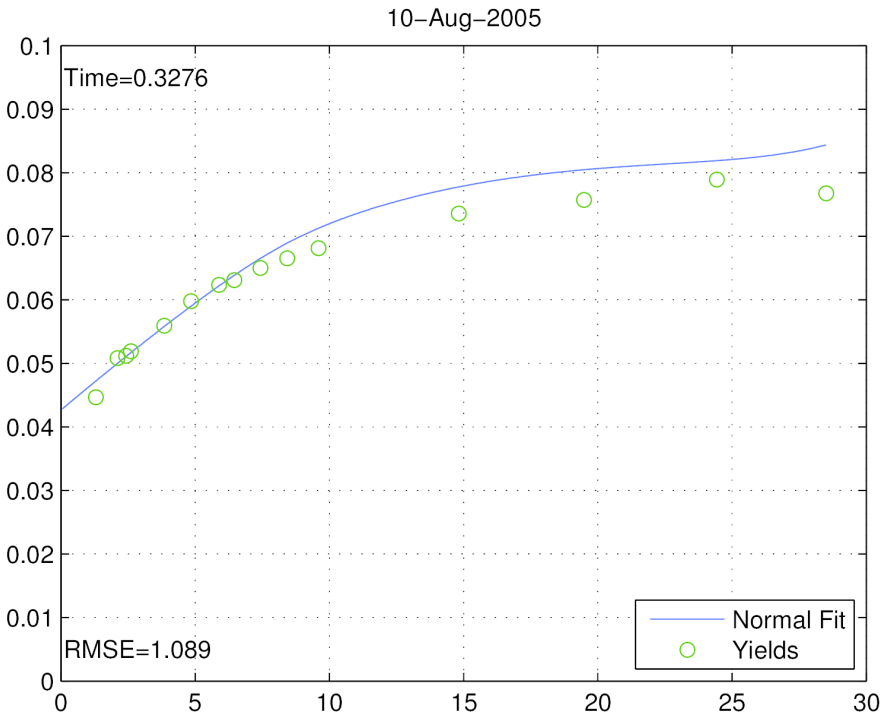


**Figure 60:** Example of a McCulloch yield curve

Here I assume that $c_i$ and $T_i$ represents the cash flow amount and cash flow date of a coupon bond. The linear system that I use for estimating the parameters calculated as following:

$$p(t, T_m) = \sum_{i=0}^{m} c_i \cdot \delta(t, T_i)$$

$$p(t, T_m) = \sum_{i=0}^{m} c_i \cdot [1 + \sum_{j=1}^{k} a_j f_j(T_i - t)]$$

$$p(t, T_m) - \sum_{i=0}^{m} c_i = \sum_{i=0}^{m} c_i \cdot \sum_{j=1}^{k} a_j f_j(T_i - t)$$

$$p(t, T_m) - \sum_{i=0}^{m} c_i = \sum_{j=1}^{k} a_j \cdot \sum_{i=0}^{m} f_j(T_i - t) \cdot c_i$$

If I express it in matrix form I get:

$$y = \begin{bmatrix} p_1 - \sum_{i=0}^{m} c_1 i \\ p_2 - \sum_{i=0}^{m} c_2 i \\ \vdots \\ p_n - \sum_{i=0}^{m} c_n i \end{bmatrix}$$

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}$$

$$X = \begin{bmatrix} \sum_{i=0}^{m} f_1(T_{1i} - t) \cdot c_{1i} & \sum_{i=0}^{m} f_2(T_{1i} - t) \cdot c_{1i} & \dots & \sum_{i=1}^{m} f_k(t_{1i}) \cdot C_{1i} \\ \sum_{i=0}^{m} f_1(T_{2i} - t) \cdot c_{2i} & \sum_{i=0}^{m} f_2(t_{2i}) \cdot C_{2i} & \dots & \sum_{i=1}^{m} f_k(t_{2i}) \cdot C_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{m} f_1(T_{ni} - t) \cdot C_{ni} & \sum_{i=1}^{m} f_2(t_{ni}) \cdot C_{ni} & \dots & \sum_{i=1}^{m} f_k(t_{ni}) \cdot C_{ni} \end{bmatrix}$$

Our objective is to minimize the residual between y and $X \cdot a$. As I introduce in Section 3.5 the usual solution for $a$ is as follows.

$$a = (X^T X)(X^T y)$$

#### *4.1.8.1 Knot points*

In McCullochs method, you can control the number of variables by determining the number of knot points. If one choose the number equal to the number of the bonds in data set, he will have almost an exact fit but it will cause unwanted oscillations. McCulloch suggests that using $\sqrt{n}$ will be enough for a good fit. Moreover it prevents the oscillations caused by mispriced bonds. For choosing these $\sqrt{n}$ knot points,

McCulloch offers following formula.

$$d_j = t_h + \theta(t_{h+1} - t_h)$$

$$h = \text{greatest integer in } \frac{(j-1)n}{k-1}$$

$$\theta = \frac{(j-1)n}{k-1} - h$$

where $k$ is the number of knot points and $n$ is the number of bond in data set. The formula always results in that $d_1 = 0$ and $d_k = t_n$. In need, you can also determine the $k$ manually. Note that when you increase $k$, therefore number of parameters, you will acquire a better fit, though smoothness of curve decreases.

**McCulloch Procedure**

1. For each bond $i$
   (a) Find the cash flow amounts and dates.
   (b) $y_i$ = (Price of bond) − (Sum of cash flow amounts)
   (c) For each basis function $f_j$
      i. $X_{i,j}$ = Sum of the values of $f_j$ at cash flow dates × corresponding cash flow amounts
2. Find coefficients $a = (X^T X)(X^T y)$

## *4.2 Implied Volatility Conventions*

### 4.2.1 Market Quotations

The simplified formula for 0.25 delta volatilities is

$$\sigma_{25C} = \sigma_{ATM} + \frac{1}{2}\sigma_{25-RR} + \sigma_{25-STR} \tag{43}$$

$$\sigma_{25P} = \sigma_{ATM} - \frac{1}{2}\sigma_{25-RR} + \sigma_{25-STR} \tag{44}$$

where $\sigma_{25C}$ is the call volatility, $\sigma_{25P}$ is the put volatility, $\sigma_{25-RR}$ is the risk reversal quotation and $\sigma_{25-STR}$ is the quoted strangle volatility. Note that

$$\sigma_{25-RR} = \sigma_{25C} - \sigma_{25P} \tag{45}$$

The market strangle volatility is defined as

$$\sigma_{25-STR} = \sigma_{25C} + \sigma_{25P} - 2\sigma_{ATM} \tag{46}$$

### 4.2.2 Basics of Hedging by Options

Primary feature of the Black and Scholes (1973) model is that the values of the contingent claims do not depend on investors' risk preferences. That means that an option can be hedged with stock to create an instantaneously riskless portfolio. Second assumption is that the stock prices evolve lognormally with constant volatility at any time and market level. The stock price evolution over time is described by

$$\frac{dS}{S} = \mu dt + \sigma dW \tag{47}$$

where $S$ is the stock price, $\mu$ is the expected return and $dW$ is a Brownian motion with a mean of zero and a variance equal to $dt$. The Black-Scholes formula for a call option follows from applying the general method of risk-neutral valuation to an asset

whose evaluation is specifically assumed to follow (47).

The value of a vanilla option contract is computed with Black and Scholes (1973) formula

$$v(S_t, K, \sigma, \phi) = v(S_t, r_d, r_f, K, \sigma, t, T, \phi) \tag{48}$$

$$= \phi[S_t e^{-r_f \tau} N(\phi d_+) - K e^{-r_d \tau} N(\phi d_-)] \tag{49}$$

$$= \phi e^{-r_d \tau}[f(t,T) N(\phi d_+) - K N(\phi d_-)] \tag{50}$$

where $d_\pm = \frac{ln(\frac{f(t,T)}{K}) \pm \frac{1}{2}\sigma^2 \tau}{\sigma\sqrt{\tau}}$, $K$ is the strike price, $\sigma$ is the volatility, $\phi = +1$ for call, $\phi = -1$ for put and $N(x)$ is the cumulative normal distribution function.
The sensitivity of the vanilla option with respect to the underlying could be interpreted in several different ways. In equity markets, one would use the spot delta (i.e. sensitivity with respect to the spot price). However, in the FX markets, the default is the forward delta (i.e. sensitivity with respect to the forward price). One can also use the premium adjusted delta. The details of these conventions are described in Wystup(2010).

The number of forward contracts one would buy to hedge a short position is slightly different from the number of units of the underlying required when using spot delta. It is given by the following formula.

$$\Delta_f(K, \sigma, \phi) \triangleq \frac{\partial v}{\partial S} = \phi N(\phi d_+) \tag{51}$$

Put-call parity yields $\Delta_f(K, \sigma, +1) - \Delta_f(K, \sigma, -1) = 1$.

One would enter $\Delta_f \times N$ forward contracts to forward-hedge a short vanilla option position. In FX options smile tables the forward delta is often preferred. The main reason for this is the fact that the sum deltas of corresponding call and put options is 1,i.e. the -delta call must have the same volatility as the 75-delta put. This symmetry only works for forward deltas, and it is due to the put-call parity equation above.

In terms of the relationship between the implied volatility as a function of delta and the implied volatility as a function of strike price different markets have different quoting mechanisms. While it is common in equity markets to quote strike-volatility or strike-price pairs, this is usually not the case in FX markets. Many customers on the buy-side receive implied volatility-delta pairs from their market data provider. This data is usually the result of a suitable calibration and transformation output. For given forward delta $\Delta_f$ and the corresponding volatility $\sigma$, the strike can be obtained as

$$K = f e^{-\phi N^{-1}(\phi \Delta_f)\sigma\sqrt{\tau}+\frac{1}{2}\sigma^2\tau}. \tag{52}$$

### 4.2.3 Generating Implied Distribution from Option Prices

Approximating function methods begin with the option-pricing relation in Cox et al. (1976), who show that the price of an option is the discounted risk-neutral expected value of the payoffs:

$$\begin{aligned} C(t,K) &= e^{-r(T-t)} \int_0^\infty (S_T - K)^+ f(S_T) dS_T \\ P(t,K) &= e^{-r(T-t)} \int_0^\infty (K - S_T)^+ f(S_T) dS_T \end{aligned} \tag{53}$$

where $C(t,K)$ and $P(t,K)$ are the prices of European calls and puts observed at time $t$ having expiries at $T$ and strike prices of $K$; $r$ is the riskless rate of interest, and

$f(S_T)$ is the risk-neutral probability density function for the value of the underlying asset $S$ at time $T$. Parametric approximating function methods assume that $f(S_T)$ has a particular functional form, chosen to allow for a variety of possible shapes. Parameter values are found by minimizing some function of the fitted price errors. For a given expiration $T$ and current stock price $S_t$, the collection $\{e^{r(T-t)}C(t,K)\}$ of undiscounted option prices of different strikes yields the risk-neutral density function $f$ of the final spot $S_T$ thrrough the relationship

$$e^{r(T-t)}C(t,K) = \int_0^\infty (S_T - K)^+ f(S_T) dS_T \tag{54}$$

Differentiate this twice with respect to $K$ to obtain

$$f(K) = \frac{\partial^2 C}{\partial K^2} \tag{55}$$

One can look at the work of Dupire (1994) for further details.