LUISS

Department
of Economics and Finance

Course of Advanced Financial Economics

Comparing numerical methods for term structure fitting

Prof. Paolo Porchia
Supervisor

Prof. Paolo Santucci De Magistris
Co-supervisor

Andrei Lukianov ID 733071
Candidate

Academic Year 2020/2021

Contents

1 Abstract

Parametric yield curve models are popular among researchers and practitioners as a means of terms structure estimation. There are numerous research papers showing that the estimation procedure of such models is highly sensitive towards the choice of a numerical method. Most of the widely-used numerical methods are initiated by some starting values, but in the framework of this problem the choice of the starting values has not yet been thoroughly investigated. The choice of starting values also has a significant impact on the solution, so in this research we propose specific starting values for several popular numerical methods based on our own criteria. We also demonstrate what the cost of an improper choice of starting values can be in terms of the variation of yields. Finally, for the purpose of on less liquid, developing markets, in our research we use the data set of the government debt of Russia.

2 Introduction

Yield curve or term structure is a graphic representation of bonds' yields with maturities ranging from short-term to long-term. The yield corresponding to a certain maturity on the yield curve is the total return on a bond, if it is held to maturity. The yield curve is one of the most important concepts in economics and finance.

In macroeconomics yield curve is one of the major indicators of economic activity. For example, there is a wide array of research literature based on the predictive power of yield curve in economic recessions[1]. Yield curve control is also an important monetary policy instrument utilized by the Federal Reserve. The European bond spreads between the government bonds of the EU member states and Germany is also a major benchmark related to yield curve.

In finance the yield curve is used to price fixed income securities, it is also a major component in the pricing of derivatives such as futures. When it comes to options, government bond yields are utilized in Black-Scholes formula and in the calculation of implied volatility and greeks. In risk management yield curve is also widely used because the interest rate risk is one of the major risks in for fixed income instruments. Overall, any field in which a notion of a risk-free rate is necessary has to use yield curve as a reference.

The term structure, however, is unobservable in most cases. Countries do not usually issue the amount of zero-coupon denominated debt that covers the whole range of maturities and even then the curve would require some method of smoothing. The solution to this problem lies in the estimation of term structure based on coupon bearing bonds. Though such a method presents a number of numerical difficulties[2], it still remains one of the most widely-used methods of term structure estimation.

The approach requires an assumption about the functional form of the yield curve and one of the

[1] See Dueker 1997, Estrella and Trubin 2006

[2] See Gilli, Große, and Schumann 2010

most popular models in this field is the Nelson-Siegel (NS) model. While giving a clear advantage in specification of the curve and of the interpretation of the factors, the parameters of the model are difficult to estimate. The loss function in the problem also has multiple local minima, so when combined with the sparse data of developing markets, the vulnerabilities of particular numerical methods and the choice of starting values, the task gets even more complicated.

So far the research literature has been mostly focused on comparing numerical methods in the framework of this problem. Ahi, Akgiray, and Sener 2018 compared the performance of gradient methods BFGS and Gauss-Newton, direct search algorithms Powell and Nelder-Mead and global optimization algorithms simulated annealing (SA) and several modification of Particle Swarm Optimization (PSO). The global optimization algorithms outperformed their counterparts in terms of accuracy and robustness, while single-point methods tended to find local minima rather than global and also failed to converge in a $30-50\%$ of cases. The choice of the starting values for single-point methods was given by the coefficients of linear regression of three-factor Nelson-Siegel model on the yields generated by bootstrapping. The method was originally proposed in Diebold and Li 2006 and according to Ahi, Akgiray, and Sener 2018 produces a starting point in the general region of the global minimum. It is important in case of the single-point numerical methods because they are sensitive and they are also the default methods for many statistical packages, so they are more likely to be used for fitting the term structure with the NS model and its modification. The final conclusion referring the single point methods is that unless the starting values are carefully chosen, the results are likely to be less reliable.

Manousopoulos and Michalopoulos 2009 compares a similar set of algorithms and comes to a conclusion that direct search and global optimization algorithms are expected to achieve a smaller error than gradient based counterparts. Though being highly reliant on the choice of starting values, gradient based algorithms should not be rejected because the interpretation of the factors allows for a good initial guess. Which algorithm performs better does not have a certain answer though. However, given that the direct search and global optimization algorithms do not use any information about the gradient, authors suggest to use a gradient based method to refine the solution from other more accurate numerical methods in a two-stage optimization process.

Some researchers Gilli, Große, and Schumann 2010 argue that standards statistical methods are not appropriate for fitting a yield curve. The primary reason for that is that the optimization problem is not convex and it has multiple local extrema what makes a single-point optimization unreliable. The authors make a sample fit using numerical method Nelder-Mead with 500 randomly generated points and come to a conclusion that multiple restarts are imperative, while it might not still be accurate enough, for example, for the purpose of modelling the evolution of parameters over time. Another challenge is that in case of the NS and Nelson-Siegel-Svensson (NSS) models the problem is 'badly conditioned' due to the factors being highly correlated for certain value of the time scale parameter τ. As a result, many different sets of parameters can produce good fits, but not all of them could have economic interpretation. Therefore, it is important to constrain parameters in certain bounds to achieve identification. Finally, the authors fit the NSS model using a numerical method

called differential evolution and argue that it produces more reliable results than its gradient based counterparts.

Paper De Pooter 2007 also underlines the problem of multicollinearity of factor loadings. The author examines different modifications of NS model and comes to a conclusion that more flexible functional forms result in better in-sample fit of the term structure. Out-of-sample fit for such models improves as well. While examining the models the author also explains how the multicollinearity problem arises and that it makes the estimation procedure more complicated.

Estimation issues were also touched in Diebold and Li 2006. The paper focuses on forecasting the yield curve using a two-stage approach. The first stage is the estimation of parameters of the NS model using a linear regression. With the estimates obtained, on the second stage the authors run univariate and multivariate autoregressions on the factors of the NS model. Their models are consistent with the stylized facts and build a forecast which has a higher accuracy on long term horizon relatively to the counterparts.

3 Yield curve fitting

Yield curve fitting is a method of obtaining a yield curve from the prices of coupon bearing bonds. Firstly, the method requires an assumption about the parametric functional form of the yield curve. Given that the bonds are priced as the sum of future discounted cash flows, this parametric (i.e. depended on a number of parameters) yield curve function could be used to obtain the prices of the bonds. Ideally the parameters should be such that the sum of discounted cash flows of the bonds would be equal to the bonds' market prices. It could be measured by a loss function: the higher the errors of the price estimates are, the higher the loss. Finally, with the loss function depended only on the parameters of the yield curve, an optimization with a numerical method is run to obtain the estimates.

First we will start by looking into the model of the parametric functional form of the yield curve that we use in our research.

3.1 Nelson-Siegel model

The methods of parametric yield curve fitting can be classified into two groups: spline-based methods and function-based methods. In spline-based methods term structure is approximated by a piecewise polynomial function. In order to achieve a smooth shape, the polynomials, usually cubic ones, are linked at various points throughout the curve. As a result such a method can construct a term structure of virtually any shape. Function-based methods, on the other hand, assume that the yield curve has specific financially interpretable features. Furthermore, the shape of the yield curve is fully given by the parameters which also have interpretation. The choice between these groups of methods is in fact a trade-off between precision and interpretability.

One of the most widely used models among function based methods a is Nelson-Siegel model

Nelson and Siegel 1987. The model is used in a variety of forms. There numerous models which are modifications of the NS model and a particular example is the more flexible NSS model Svensson 1995. Different variations of NS model are used by central banks, stock exchanges, investment banks and etc. In Russia, for example, a modification of Nelson-Siegel model is utilized by the central bank and Moscow Stock Exchange.

In our research we also use the Nelson-Siegel model. The model (1) provides a functional form of the yield curve, where the yield $r(t|\Theta)$ corresponding to time to maturity t can be calculated based on the vector of parameters Θ. The vector of parameters in the model is $[\tau, \beta_0, \beta_1, \beta_2]$. The yield curve is represented as the sum of a constant β_0 and factor loadings (2) and (3) weighted by the corresponding factors β_1 and β_2. The loading on the slope component (2) corresponds to the short term level of interest rates. It is a monotonically decreasing function: its maximum value is 1 at $t = 0$ and it approaches 0 at $t \to \infty$. The loading on the hump component (3) corresponds to peaks or throughs in medium term. It monotonically increases until it reaches a peak, then it approaches 0 when $t \to \infty$. The remaining component is the level given by coefficient β_0 which reflects the long term level of interest rates. The whole yield curve approaches β_0 when $t \to \infty$.

$$r(t|\Theta) = \beta_0 + \beta_1 g(t|\tau) + \beta_2 h(t|\tau) \tag{1}$$

$$g(t|\tau) = \frac{1 - \exp(-t/\tau)}{t/\tau} \tag{2}$$

$$h(t|\tau) = \frac{1 - \exp(-t/\tau)}{t/\tau} - \exp(-t/\tau) \tag{3}$$

Figure 1 shows how the shape of the factor loadings change with τ. In case of the loading on the hump component τ determines the location of the peak: the higher its value is, the further away is the peak. In the loading of the slope component, τ determines the rate at which the component decreases. Higher value of τ makes the component decrease at a slower rate.

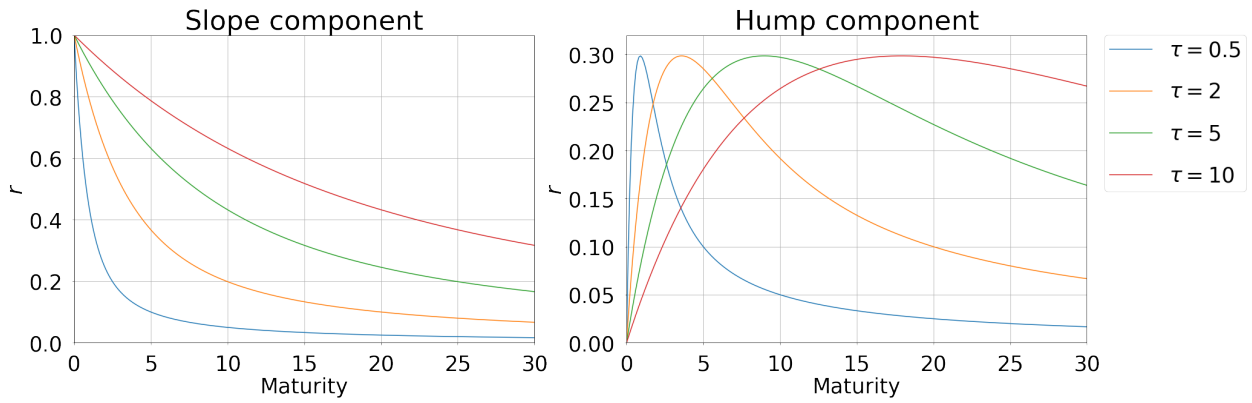

Figure 1: Nelson-Siegel model components

In the equation of the yield curve (1) the factor loadings are scaled by the corresponding factors β_1 and β_2 and are summed up with β_0. Thus, the shape of the yield curve is fully determined by the

values of β_0, β_1, β_2 and τ. These parameters of the curve could be chosen in a way to fit different data and describe a term structure of a variety of shapes.

Finally, the specification of the model also implies several restrictions on the vector of parameters Θ. The long-term level of interest rates, which means that β_0 should not be negative (4). The interest rate for $t = 0$ should not be negative either. So, knowing that the slope and the hump components are equal to 1 and 0 respectively at $t = 0$, we can derive the restriction (5). The last one is that τ should lie in the range of 0 to 30 so that the humps would also be in this range (6).

$$\beta_0 \geq 0 \tag{4}$$

$$\beta_0 + \beta_1 \geq 0 \tag{5}$$

$$0 < \tau \leq 30 \tag{6}$$

By trial and error we found out that the inclusion of restrictions does matter for several optimization methods. In particular, for some optimization methods we will also impose lower and upper bounds on betas [3].

$$0 \leq \beta_0 \leq 1 \tag{7}$$

$$-1 \leq \beta_1 \leq 1 \tag{8}$$

$$-1 \leq \beta_2 \leq 1 \tag{9}$$

3.2 Loss function

In our research we fit the NS model to prices of coupon bearing government bonds. The obtained term structure should give the estimates of the bonds' prices such that they would be as close as possible to the bonds' market quotes. In other words, we should find a vector of parameters Θ of the NS curve such that it minimizes the sum of the squares of pricing errors. In addition to that we also divide the pricing errors by 1000 in order to avoid overflow errors when fitting the yield curve (10).

$$L(\Theta) = \sum_{i=1}^{N} \left(\frac{P_i - \hat{P}_i(\Theta)}{1000} \right)^2 \tag{10}$$

We assume continuous compounding and evaluate prices of bonds as a function of the yield

[3] We express the term structure in a way that, for example, 7% yield corresponds to the number 0.07 on a curve. Therefore, the upper bound of 1 on β_0 restricts the long term level of interest rates below 100%. Though as pointed out by Manousopoulos and Michalopoulos 2009 it is a loose restriction, it does prevent numerical methods from divergence in some cases.

curve and its parameters. That being the case, the price of an individual bond is a sum of its cash flows discounted by the corresponding rates obtained from the NS curve (11).

$$P(\Theta) = \sum_{i=1}^{n} CF_i \exp(-r(t_i, \Theta)t_i) \tag{11}$$

Another loss function that can be used for this type of problem is a weighted sum of squares of bonds' pricing errors (12) Ahi, Akgiray, and Sener 2018, Hu, Pan, and Wang 2013. The underlying idea is to give lower weights to the bonds with higher Macaulay duration because otherwise such bonds contribute more to the total error while being less frequently observed. The weight w_i in that case could be defined as $\frac{1}{D_i}$ or $\frac{\frac{1}{D_i}}{\sum_{j=1}^{N}\frac{1}{D_j}}$.

$$L_D(\Theta) = \sum_{i=1}^{N} w_i\big(P_i - \hat{P}_i(\Theta)\big)^2 \tag{12}$$

We also used this function at first, but we did not see a significant improvement in accuracy whereas the computational time increased. Moreover with such a modification the function will still have a lot of local minima, so the problem of the proper choice of starting values is just as important as before.

Finally, we could state the optimization problem as the minimization of the loss function (10) under constraints (4), (5), (6) and in some cases, which we will specify also (7), (8), (9).

3.3 Numerical methods

In our research we work with widely used numerical methods that are available in most statistical packages. These methods are BFGS, Powell, Nelder-Mead, Trust-Region Constrained Algorithm. We used the implementations of these methods from library scipy in Python.

In general, there are methods which perform arguably better for our problem and which are also more robust towards the choice of starting values and to the noise in data. Amoung these methods, for example, Differential Evolution Gilli, Große, and Schumann 2010, PSO Ahi, Akgiray, and Sener 2018, Simulated Annealing Manousopoulos and Michalopoulos 2009. On the other hand, these methods are not available in such statistical packages as EViews, Stata, SPSS or in the solver tool in Excel. When more advanced software is used like Python, R or MATLAB, the numerical methods which are used by default in minimization functions are also among the ones we chose[4]. There are also numerous papers which are using the same group of methods that we use [5].

It is also important to mention that that some of the methods that we utilize support the native inclusion of restrictions like lower and upper bounds or linear restrictions, whereas some do not. Therefore, in order to implement the restriction we have to set up all the methods individually.

[4] In scipy, for example, BFGS is used by default.

[5] See Ahi, Akgiray, and Sener 2018, De Pooter 2007, Gilli, Große, and Schumann 2010, Ioannides 2003, Manousopoulos and Michalopoulos 2009

3.3.1 BFGS

Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) is a gradient-based numerical method. We are solving a problem of constrained minimization, so we chose an implementation of BFGS with restrictions on upper and lower bounds and limited computer memory usage which is called L-BFGS-B.

Given that BGFS does not support the inclusion of linear restricitons, we had to find a proper reparameterization of the loss function and of the restrictions. We define $u_0 = \beta_0$, $u_1 = u_0 + \beta_1$, $u_2 = \beta_2$, $\tau = \tau$. Then the restriction (6) remains unchanged while restrictions (4), (5) and the loss function take the following form:

$$u_0 > 0 \tag{13}$$

$$u_1 > 0 \tag{14}$$

$$r(t|\Theta) = u_0 + (u_1 - u_0)g(t|\tau) + u_2 h(t|\tau) \tag{15}$$

Furthermore, the value of vector gradient should be supplied as an input to BFGS. By default scipy uses a finite difference scheme to approximate the gradient, but it can take as an input any other function that calculates vector gradient. For instance, among possible options are automatic differentiation or a function which gives a precise value of a vector gradient as an output. The libraries in Python which perform automatic differentiation like Autograd or JAX do not support our implementation of the loss function[6]. Besides that we tried a precise vector gradient[7], but it did not increase the accuracy significantly while it took almost double the time to converge. Therefore we decided to stick to a native finite difference method.

3.3.2 Powell's method

Powell's method is a direct search algorithm. It supports bounds, but it does not support linear restriction, so in order to include the linear constraint (5) we used the same reparameterization as we used for BFGS. We also had to include restrictions on upper and lower bounds of the betas (7), (8), (9) because otherwise the method would converge out of these bounds in several cases.

3.3.3 Nelder-Mead

Like Powell's method, Nelder-Mead is also a direct search algorithm. Our optimization problem should, however, be set up differently for the Nelder-Mead because it does not support neither restrictions on the bounds, nor linear restrictions. A possible solution to that is to include restrictions

[6]Both Autograd and JAX differentiate native python and numpy code, while our loss funciton generator also has to use pandas and datetime libraries

[7]In attachments you could find the values of the partial derivatives that we used for BFGS

in the loss function, for example, like the Lagrange function does. So, let us redefine restrictions (4), (5) and (6) as penalty functions (16), (17) and (18) respectively.

$$c_0 = \max(-\beta_0, 0) \tag{16}$$

$$c_1 = \max(-(\beta_0 + \beta_1), 0) \tag{17}$$

$$c_2 = \max(\tau - 30, 0) + \max(-\tau + 0.01, 0) \tag{18}$$

If we include these penalties by just summing them with the sum of squares of errors, then the resulting loss function will not be smooth. What we could do is to use the squares of these penalty functions and to also multiply them by a constant K to increase the penalty (19). In our case we set $K = 1000$.

$$L_{NM}(\Theta) = \sum_{i=1}^{N} \left(\frac{P_i - \hat{P}_i(\Theta)}{1000} \right)^2 + K \sum_{i=0}^{2} c_i^2(\theta) \tag{19}$$

3.3.4 Trust constrained

For Trust-Region Constrained Algorithm there is a possibility of inclusion both border and linear constraints, so no modification for the loss function was necessary. However, like in the case of Powell, we also had to include lower and upper bounds on betas because otherwise it would not always converge.

3.4 Starting values

In total we looked into 5 ways of starting values generation. Some of the approaches the we used had already been employed by other researchers[8], whereas the others are relatively new.

3.4.1 Randomly generated starting values

With this approach we generated random starting values between specific lower and upper bounds using uniform distribution. For the value of τ we simply used the range $[0, 30]$. For the value of β_i we used the rule given by equation (20). In this equation we use the minimum and maximum historic values for β_i and also their historical standard deviation. The period for which we took these statistics is from January 2014 till June 2020.

$$\beta_i \sim U[\min(\beta_i) - 3\sigma_{\beta_i}, \max(\beta_i) + 3\sigma_{\beta_i}] \tag{20}$$

As a result we obtain the lower and upper bounds as in Figure 2. We consider these bounds to be loose enough to get variability, but at the same time not too wide to initiate optimization with a

[8] See Diebold and Li 2006, Manousopoulos and Michalopoulos 2009

priori wrong starting values. In total we generate 16 such starting values for each day in our dataset.

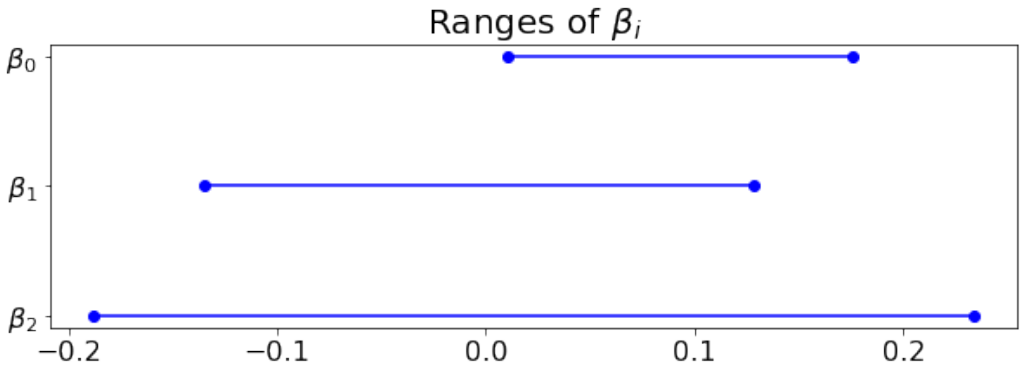

Figure 2: Upper and lower bounds of β_i

The main purpose of why we use the random starting values is because we intend to demonstrate variability in the result of the optimization problem. We also use them to obtain a better solution: after solving the problem for all 16 of them, we simply choose the result with the lowest RMSE. Such a result we call ”Best of random”.

3.4.2 Previous day starting values

Another approach that we believe should give an accurate starting point is using the solution of the previous day. Our idea comes from an observation that the term structure does not change its shape significantly from one day to another. We should add here that we initiate the optimization on the first day with the historical parameters of the NS curve from the day before. Such an approach we will call ”Recurrent”.

3.4.3 Manousopoulos' approach

The next method of starting values generation that we use was introduced by Manousopoulos and Michalopoulos 2009. The idea for this approach comes mainly from the interpretation of the parameters. For example, β_0 is taken as the average yield of M bonds with longest maturity (21). We used $M = 4$ as is recommended by the authors. The short term level of interest rates β_1 is taken as the difference between the yield of the shortest bond y_s and β_0 (22). As for the remaining parameters they are recommended to be taken as fixed (23), (24). We will denote such starting values by Θ_M. An example of a starting value generated by this method is given by picture 3.

$$\beta_0 = \frac{1}{M}\sum_{i=1}^{M} y_i \tag{21}$$

$$\beta_1 = y_s - \beta_0 \tag{22}$$

$$\beta_2 = 0 \tag{23}$$

$$\tau_1 = 1 \tag{24}$$

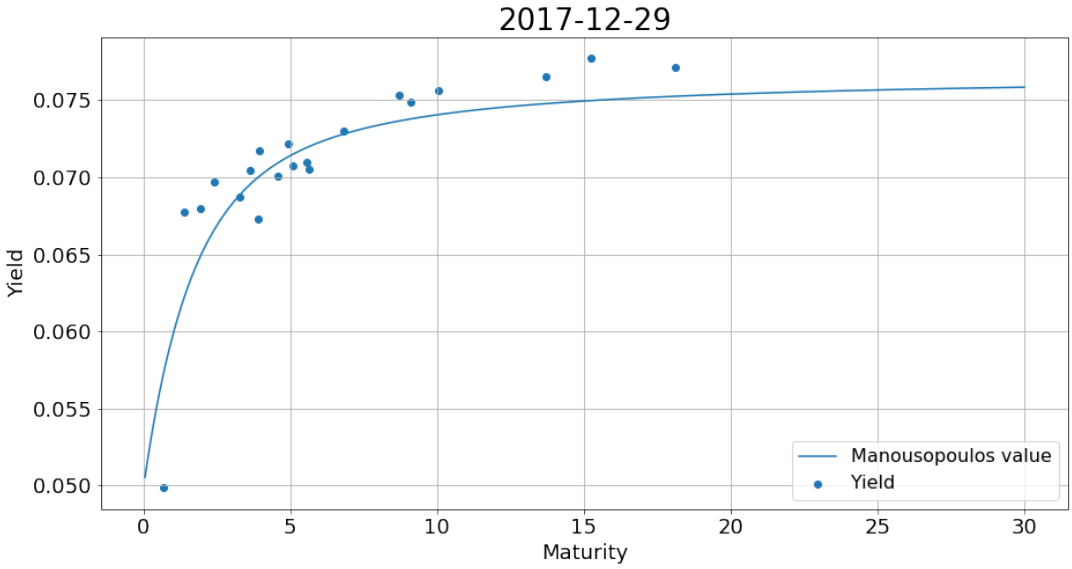

Figure 3: A starting value generated by Manousopoulos' method on one of the dates.

3.4.4 Diebold, Li approach

Finally, we would like to investigate the approach proposed by Dieblod, Li Diebold and Li 2006. The method can be divided into two stages. The first stage is obtaining the bonds' yields by the bootstrapping algorithm otherwise known as unsmoothed Fama-Bliss yields. On the second stage the yields are used as a dependent variable, while factor loadings of the NS model as features in a linear regression.

First, let us look into bootstrap Fama and Bliss 1987 in details. Let us suppose that we would like to obtain a term structure, for example, for a particular day. As inputs we have a set of bonds of different maturities, the timetable of their future cash flows and their closing prices for the day. If we calculate IRR for each of the bonds and plot them for the corresponding maturities, then such a "yield curve" will lack a time structure, because the yields of the bonds with shorter maturities will not be used to determine the yields of the bonds with longer maturities. So, what was proposed by the authors, was to do calculate the the longer yields based on the shorter yields. The first step is to calculate the yield of the shortest zero-coupon bond or, as we did, the IRR of the shortest coupon-bearing bond. In this method this yield corresponds to the maturity of this bond and before, i.e. we assume that the yield curve on this interval is flat. In the next step we calculate the yield of the second shortest bond assuming the shape of the yield curve that we already obtained from the shortest bond. Again, assuming that the yield curve is flat on the interval between the maturities of the shortest and the seconds shortest bond, we obtain another section of the yield curve. Then

we just repeat it for all the remaining bonds in the order of increasing maturity until we reach the longest bond. Once this yield is calculated, the procedure is finished. The bonds' prices are on the Russian market contain a lot of noise, so after applying the method we filter the yields by deleting the ones which are more than 4 standard deviations away from the median yield. Figure 4 gives a representation of the Fama-Bliss yields calculated on one of the dates in our data set.

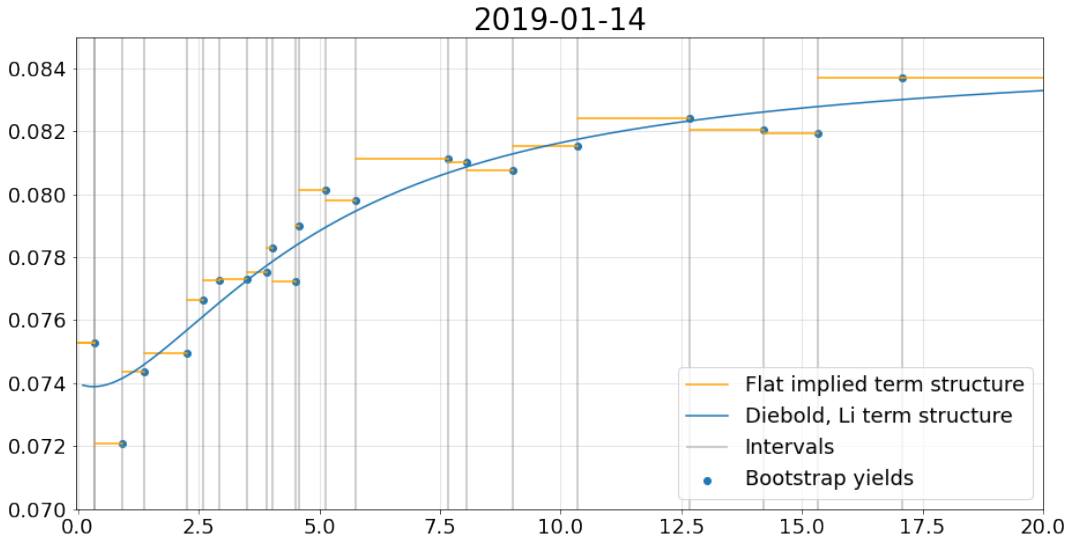

Figure 4: An example of Fama-Bliss yields, the implied flat term structure and NS curve based on Diebold, Li factors

The next step is running a linear regression based on equation (1). We follow the approach by Diebold and Li 2006 in fixing the value of τ. When picking the value of τ they refer to the loading on the hump component. As we have already noted above, τ is the parameter that determines where the loading achieves the maximum. The loading is also a medium term component, while the maturities that are often referred to as medium term are 2 to 3 years. We, just liked the authors, picked the average which is 2.5 years. With this average we calculated the value of τ at which the loading achieves maximum. Using Wolfram Alpha[9] we arrived at the value of τ equal to 1.4. Our result is consistent with other researchers. For example, Gilli, Große, and Schumann 2010 arrived at the same value of τ when working with maturity in years. They also mention that this result translates to the same value obtained by Diebold and Li 2006 who worked with maturity in months. The value, according to Gilli, Große, and Schumann 2010 is also well-chosen because it implies low correlation between the factor loadings.

Now that we have the values of τ, we could obtain the features for the regression by substituting the corresponding maturities into the factor loadings. The remaining part is just to estimate the factors by OLS regression with the bootstrap yields as the dependent variable and the loadings as features. An example of such an estimation is given by Figure 4 and the whole time series of the

[9]See the Wolfram Alpha result

factors is represented by Figure 5.

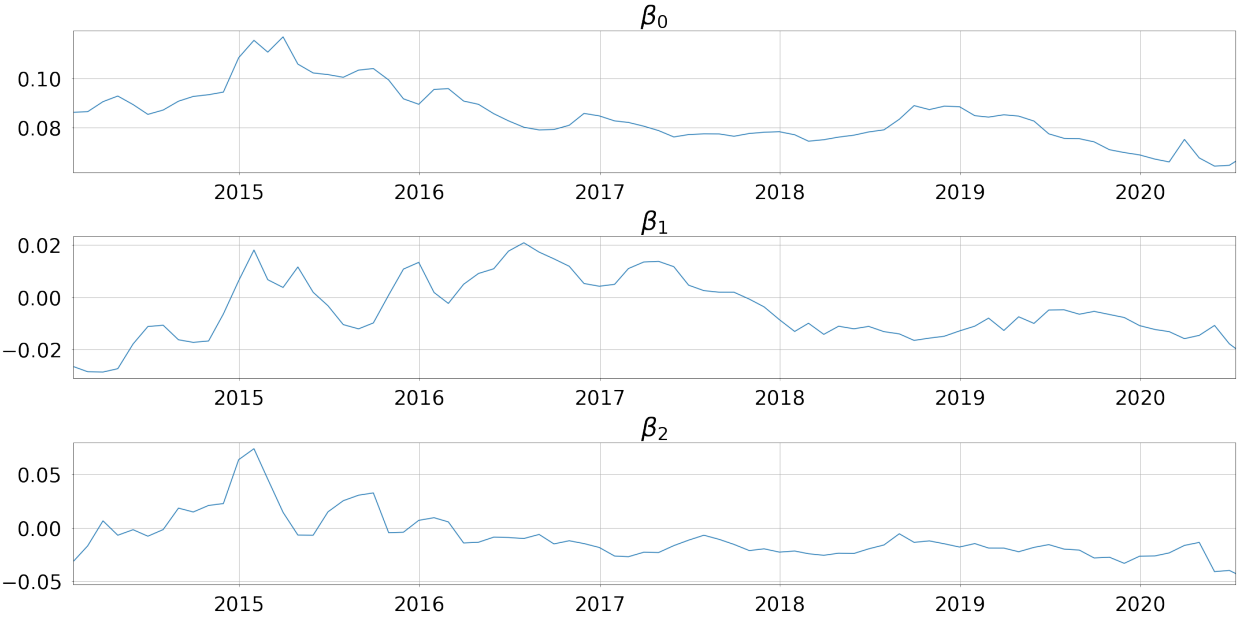

Figure 5: Time series of NSS factors obtained by the Diebold, Li method

4 Data set

Our data set comprises the data of the secondary market for the rouble denominated government debt of Russia for the period starting from year 2014 till mid 2020. The source of the data set is the data platform for investors `cbonds.ru`. From there we obtained daily prices, interest and amortisation schedules. The original source of the prices is the Moscow Stock Exchange.

Given that our numerical problem is quite sensitive towards outliers in bonds' prices, we removed all the illiquid bonds. We also removed all the bonds with the maturity of less than 1 month, because prices of such bonds tend to be distorted in such short periods, while NS model is particularly sensitive in short-term maturities.

We should note, however, that the data set required a lot of thorough inspection, because there was a substantial number of outliers in bonds' prices. There were particular dates on which some bonds experienced sudden price movements of, for example, 10% in IRR while returning to their normal price right next day. On the other hand, relative to the size of the data set, the number of such observations was minuscule, but we still removed them. In the end, we calculated the dirty prices of the bonds which we used later to sustitute into the loss function.

In total there are 44 bonds in our data set. Figure 6 gives an overview of the maturities of the bonds. The average maturity in our dataset is 6 years. On each of the days we have quotes of 24 bonds on average.

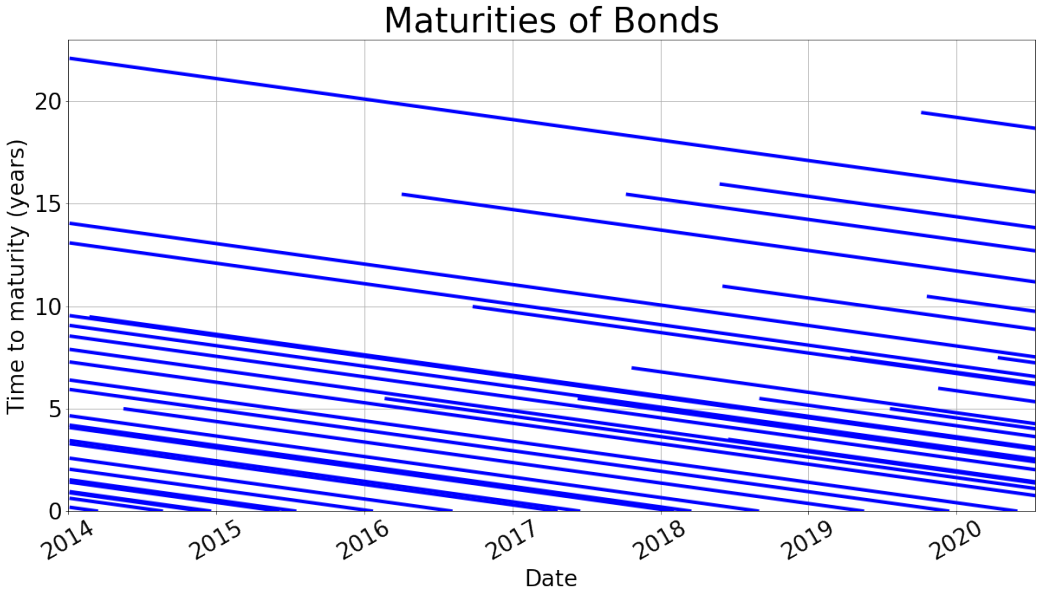

Figure 6: Bonds' maturities in the dataset

5 Criteria for the choice of starting values

We will utilise a number of criteria to compare the performance of the starting values. The first one is RMSE (25) which is an indicator of the goodness-of-fit. The lower the RMSE, the better the fit. In addition to that, RMSE will let us understand if the methods converge to different solutions under different starting values and also how significant that difference is.

$$L(\Theta) = \left(P_i - \hat{P}_i(\Theta)\right)^2 \tag{25}$$

Another important criteria is whether a numerical method successfully converged. Our idea is that if the resulting solution is not close to the market yields, then such a solution cannot be a "good" fit. We introduce our own criteria for convergence and utilise it further to separate the subset of "good" solutions.

As we noted above, the loss function in our numerical problem has multiple local minima, and as a result, numerical methods tend to stop at these local minima. What we would like to do, is to show this variation in solutions. One of the ways of doing it is by looking at the difference between the maximum and the minimum yields on a specific maturity. The maturities that we consider are of 1 year and 10 years.

The next criteria is the CPU time. There is a trade-off between the accuracy and the complexity of the numerical methods, so it is also useful to understand how significant the trade-off is.

Finally, we will also look into whether the obtained solutions can possibly have an economic interpretation.

6 Results

6.1 Minimum RMSE

First, let's have a look at which starting values showed the lowest RMSE. Figure 7 gives an overview of the results. The obvious observation is that for all the numerical methods the best out of 16 random starting values showed the lowest RMSE. The recurrent starting values, on the other hand, gave the lowest RMSE in a very small amount of cases, especially for such numerical methods as L-BFGS-B and Powell. These methods appeared to be non-robust towards the choice of starting values, so in a lot of cases the numerical methods did not converge.

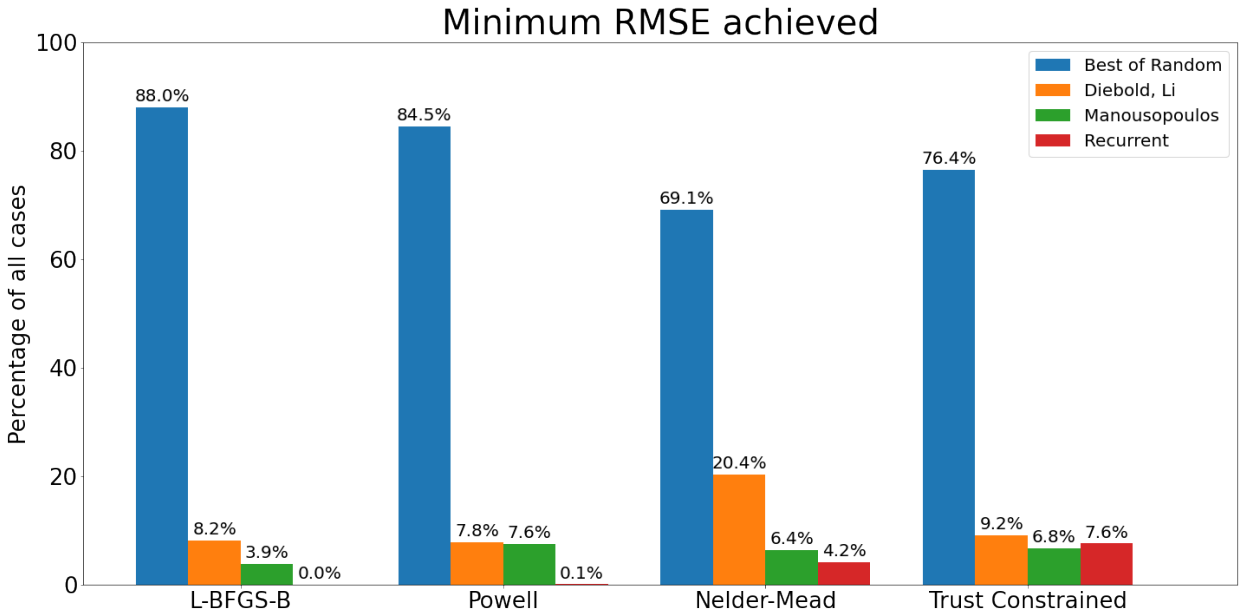

Figure 7: Shares of cases in which minimum RMSE was achieved

6.2 Convergence

Now we will make an attempt to identify those cases. Obviously, such a problem requires a convergence criteria. Our criteria is based on the following idea: if the resulting yield curve is not very different from the unsmoothed Fama-Bliss yields, then we say that the method converged. To be more precise, those cases in which 90% of the unsmoothed Fama-Bliss yields lie inside a margin of 2 standard deviations from the resulting smoothed yield curve, we will consider a convergence success. The standard deviation that we use to define the margin is the standard deviation of the unsmoothed yields on the corresponding cross-section in our dataset. A comprehensive representation with the examples of successful and unsuccessful convergences can be seen in Figures 8 and 9.

Table 1 summarizes the results after applying the criteria. The first thing to note is that the only starting values for which the amount of cases of divergence does not vary substantially are the Best

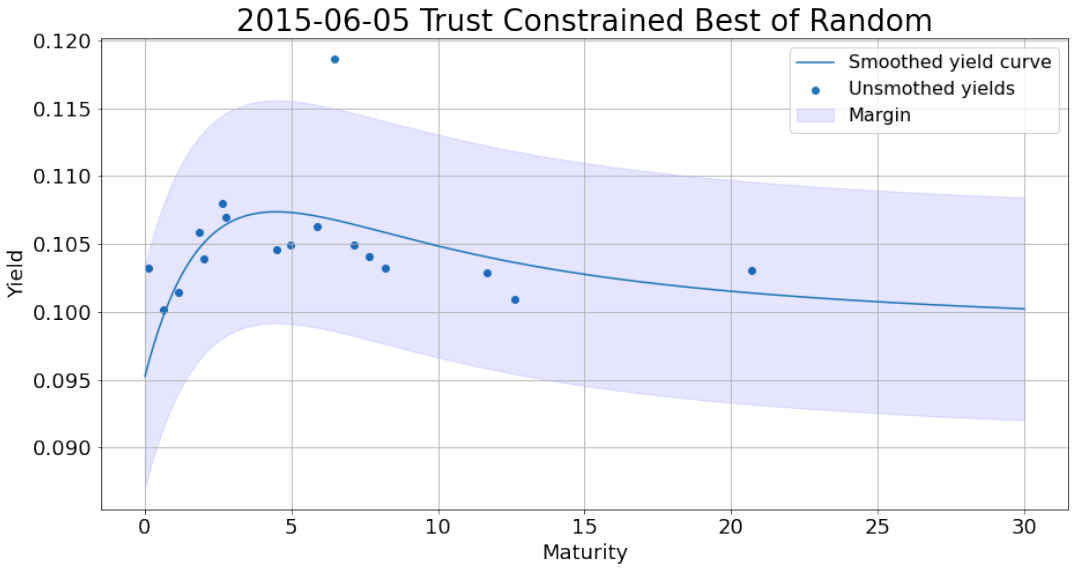

Figure 8: Successful convergence

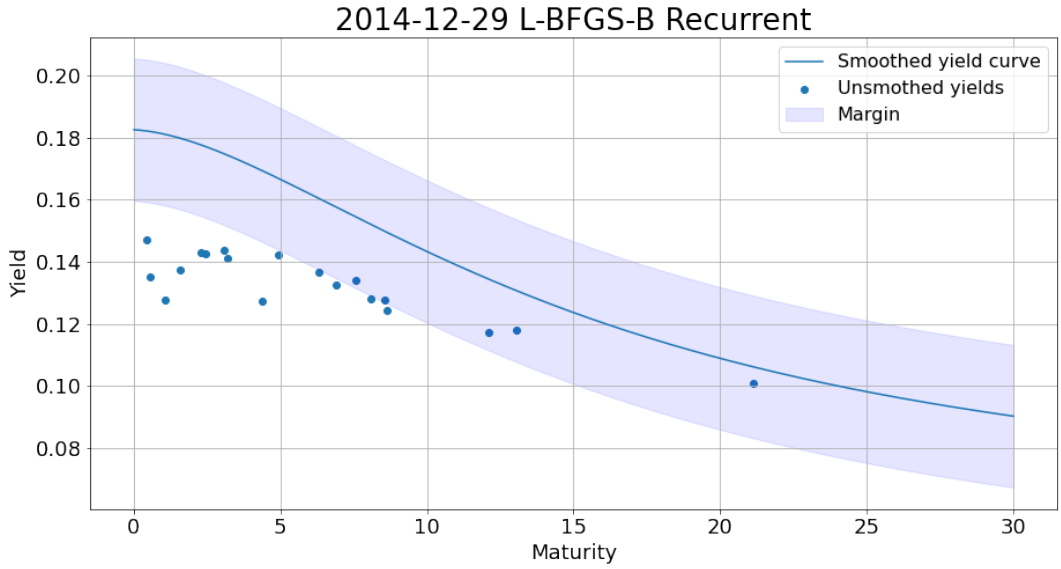

Figure 9: Unsuccessful convergence

of Random and Diebold, Li. Previous day values have a lot of cases of divergence when used with non-robust numerical methods.

The reason for that is that if in one of the days the method does not converge, then it is less likely to converge the following day because of a bad start. The less robust methods should then have a significantly higher percentage of failures for the previous day values which is true in case of L-BFGS-B, Powell and Nelder-Mead.

From this table we can also conclude that the Trust Constrained is the most robust method because the percentage of failures does not vary dramatically over different starting values.

Method	L-BFGS-B	Powell	Nelder-Mead	Trust Constrained
Best of Random	6.3	6.54	6.67	6.3
Recurrent	99.82	87.83	31.8	6.85
Manousopoulos	17.13	21.22	7.09	6.36
Diebold Li	9.6	7.58	6.24	6.36

Table 1: Convergence failures, percent

6.3 RMSE distribution

Now that we can exclude the cases of unsuccessful convergence we could also get a clear picture of the RMSE distributions. Figure 10 summarizes the results.

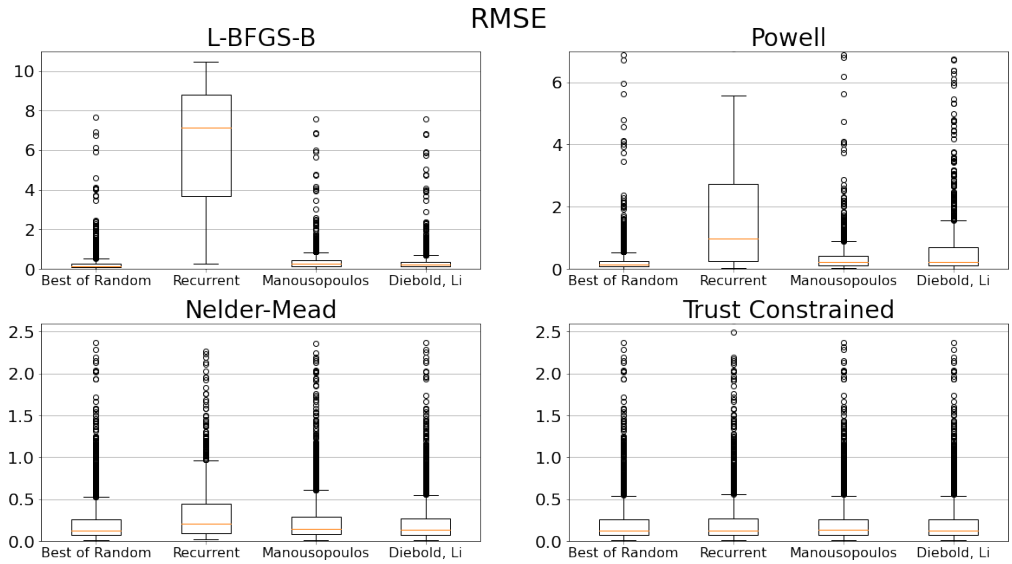

Figure 10: RMSE distributions

The first thing to comment on is the unusual distribution of the RMSE in case of the starting values of the previous day in numerical methods L-BFGS-B and Powell. As we noted above, these methods are clear outsiders when it comes to convergence. Therefore, once all the ”bad” solution are removed, there is just not that many of them left, especially in case of L-BGFS-B. Overall, given that we already know that previous day values are an unreliable choice in case of L-BFGS-B and Powell, the information that we get from the distribution of RMSE in ”good” solutions doesn't make them any more reliable.

What is more important, is that in case of some starting values the distribution of RMSE is significantly narrower, which is clearly an advantage. In numerical methods L-BFGS-B and Powell

such a starting value is quite evident: it is the best of random. Given that these methods are relatively sensitive towards the choice of starting values, running multiple optimizations from different starting points produces a much more accurate solution.

The same, however, cannot be said about Nelder-Mead and especially Trust Constrained. In case of Nelder-Mead running multiple optimization improves RMSE only by a small margin, whereas in case of Trust Constrained there is no improvement at all. Both methods are more robust, thus they tend to converge to a single solution even from a wide range of starting points.

6.4 Variation of the yields

The exclusion of the cases of unsuccessful convergence and look at the range of the yields. In particular, we are interested in 1-year and 10-year yields. The way we calculated the range is by subtracting a minimum yield from the respective maximum yield on each of the dates in our dataset. The resulting distribution is given by the boxplots in Figure 11.

There is a number of conclusions that we can draw from this result. Firstly, variation of the 1-year yield is higher than the 10-year yield. The most likely reason is that the curve doesn't fit very accurately for shorter maturities, which consequently can be explained by the possible inaccuracies in the dataset.

Secondly, all the methods produce a relatively high level of variation in solutions. We could expect the variation to be in a 2% range for the 1-year yield and in around 0.4% range for the 10-year yield. Given that the yield curve is used to price financial instruments, we could also expect the variation in their prices in a magnitude similar or higher than such of the yields.

Thirdly, Trust Constrained consistently gives a lower degree of variation in comparison with other numerical methods. As we noted above, the method proves to be robust towards the choice of starting values.

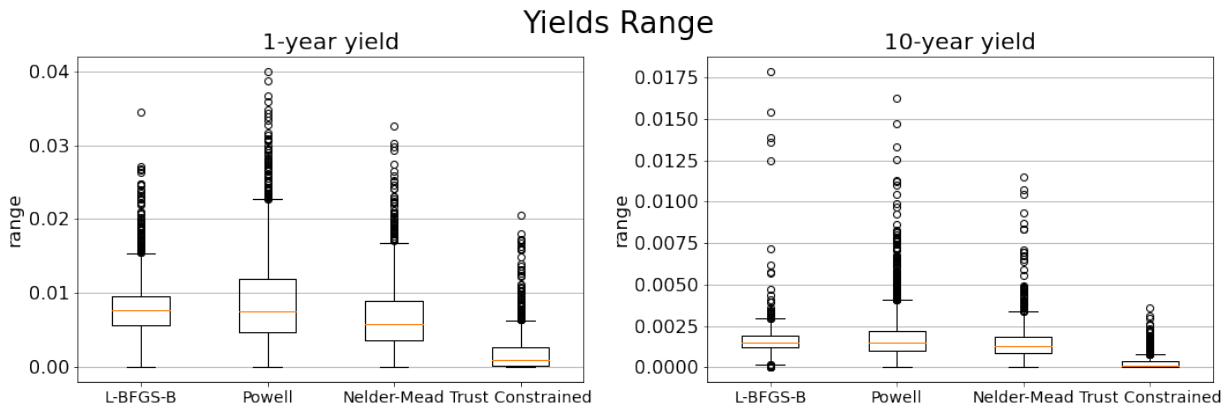

Figure 11: Yields' box plots

6.5 CPU time

Another important criteria is the CPU time. Figure 12 depicts the distributions of the time measurements grouped by numerical methods and starting values. One important remark is that the Best of Random starting value CPU time is given in terms of the average of 16 fits, whereas the other methods correspond to 1 fit only.

Not surprisingly, the most accurate starting value, which is the Best of Random, requires significantly more time in comparison with other methods. Another important observation is that even the average CPU time of Best of Random starting values is relatively high for all the numerical methods. The possible reason is that some of the random starting values give a bad starting point of optimization, so it takes more time for them to converge rather than when we use a good approximation from the beginning.

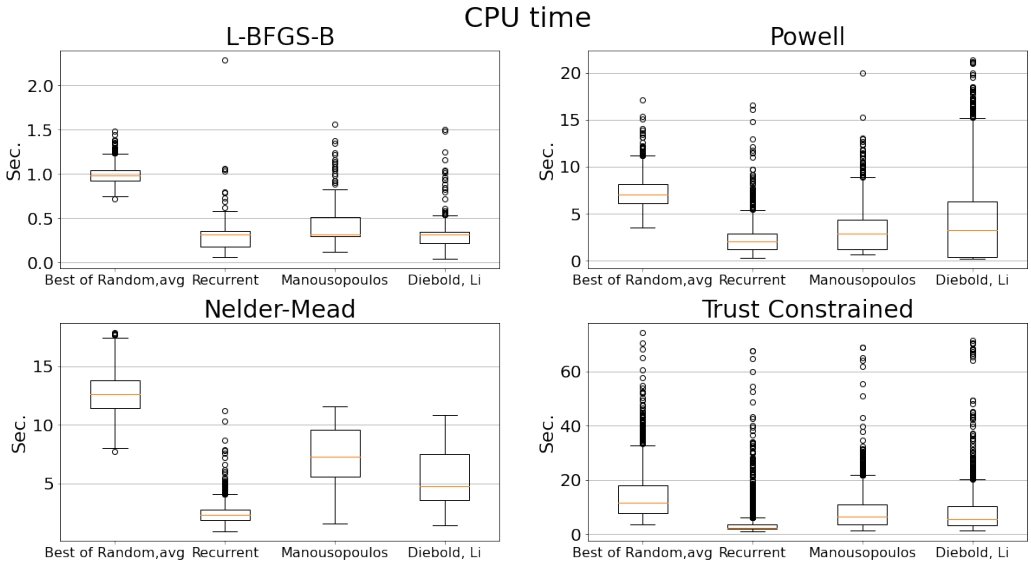

Figure 12: Box plots of CPU time. Best of Random is the average of 16 fits.

6.6 Economic Interpretation

The problem with the majority of the combinations of numerical methods and starting values that we reviewed above is that they lack economic interpretation. The term structure does not tend to change drastically from one day to another, while what we observed in most cases is the significant fluctuations in the values of the parameters of the NS model. While such solutions can score high in terms of goodness-of-fit, the parameters may not represent the time evolution of the yield curve. The only combination out of those we investigated that we believe has a time structure is the Diebold, Li starting values used with L-BFGS-B.

Diebold, Li starting value is initially a good approximation for two reasons. Firstly, it helps avoid the multicollinearity problem by fixing τ at a value with which the correlation between factor

loadings is extremely low, as we mentioned above[10]. Secondly, with such a value of τ accurate estimates of the factors could be obtained with a linear regression. This is also important, because, for example, with Manousopoulos' approach the value of τ, is chosen right, the values of β_0, β_1 are also chosen quite close to what we obtained with Diebold, Li values, but the value of β_2 is left at 0. The time series with Manousopoulos' values as starting points resembles that of the Bundesbank[11], while with Diebold, Li starting values it has the gradual evolution[12].

Another important remark is that we were unable to achieve it with a numerical method other than L-BFGS-B. We believe that the reason is that L-BFGS-B falls into a local minimum in the area of the starting point. While such a solution might not have the lowest RMSE, it keeps the factor loadings relatively uncorrelated. Other numerical methods tend to jump out of the initial area thus leaving economic interpretation behind. While the new solution might be better in terms of the goodness-of-fit, it would have highly correlated loadings which is neither good for a regression approach, neither for a numerical method.

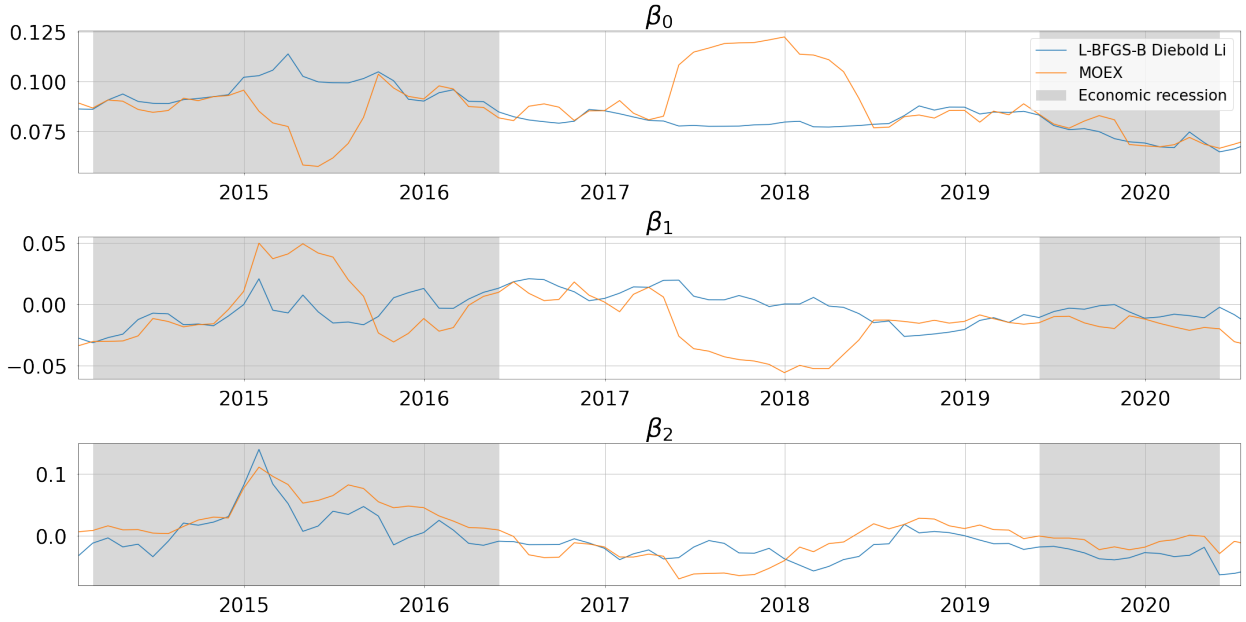

Figure 13: Factors of the NS model

Now we would like to demonstrate that the combination of L-BFGS-B with Diebold, Li starting values can produce an economically sound solution. What we want to show is that the changes in the factors (level, slope and curvature) can explain a substantial part of the variation in bonds' prices.

We already had the necessary time series of the factors and as our first step we switched to a monthly data by taking the medians of the factors in each of the months in our data set. The next

[10] See Gilli, Große, and Schumann 2010

[11] Idem

[12] See Figure 13. On the Figure we also plot the factors published by the Moscow Stock Exchange (MOEX). We should note that the time series are quite similar apart from the through in mid 2015 and the peak in end 2017 in the series of the MOEX. We believe that the long term market yield behaved in these periods closer to our estimates rather than those of the MOEX.

step was to obtain the proxies for the increments of the factors, so we fit AR(1) processes to each of the monthly time series and calculated the residuals. The resulting residuals did not indicate high pairwise correlation: -0.38 between β_0 and β_1, 0.3 between β_0 and β_2, -0.13 between β_1 and β_2. The dependent variable in our analysis is the Russian Government Bond Index (RGBI)[13] which is comprised of the most liquid bonds with maturity above 1 year. Factors in the NS model correspond to different maturities, so it is important that we use a portfolio of bonds of different maturities as well.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.542
Model:	OLS	Adj. R-squared:	0.523
Method:	Least Squares	F-statistic:	29.14
Date:	Tue, 01 Jun 2021	Prob (F-statistic):	1.51e-12
Time:	00:06:33	Log-Likelihood:	219.11
No. Observations:	78	AIC:	-430.2
Df Residuals:	74	BIC:	-420.8
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0080	0.002	4.747	0.000	0.005	0.011
x1	-2.3373	0.584	-4.004	0.000	-3.500	-1.174
x2	-0.0027	0.274	-0.010	0.992	-0.548	0.543
x3	-0.6722	0.101	-6.659	0.000	-0.873	-0.471

Omnibus:	16.184	Durbin-Watson:	2.242
Prob(Omnibus):	0.000	Jarque-Bera (JB):	51.421
Skew:	0.456	Prob(JB):	6.82e-12
Kurtosis:	6.872	Cond. No.	351.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Figure 14: Regression of RGBI on factor proxies; variables x1, x2 and x3 are the proxies of level, slope and curvature respectively

Figure 14 summarizes the results of the regression. The changes in the factors explain 54.2% of variation in the return on the bond portfolio. Quite unsurprisingly, the most contributing factor is the level with the highest negative effect. Slope is insignificant, while the curvature also has a negative effect.

7 Conclusion

Finally, we can make a recommendation on which starting values should be used for the numerical methods in our research.

[13] See `https://www.moex.com/en/index/RGBI`

In case of L-BFGS-B and Powell, we would suggest to use either the Best of Random or Diebold, Li. These numerical methods, as we noted above, are non-robust towards the choice of starting values, so an accurate initial point is very important. The most obvious approach is running multiple optimizations from randomly selected points. Even such a small number of restarts as 16, is already enough to achieve a much higher accuracy than with any other starting value. The only problem with such an approach is that it inevitably produces solutions which lack economic interpretation. In case if the interpretation is important, we would suggest to use the Diebold, Li starting values with L-BFGS-B. The value of τ that they suggest helps to resolve the problem of multicollinearity between the factor loadings thus avoiding the substitution of one factor loading for another. L-BFGS-B with such a starting value tends to converge in the proximity of the solution.

Nelder-Mead is a more robust method, so we believe that for the purpose of fitting any of the starting values could be used if they supposedly lie in the general area of the solution. The RMSE variation with Nelder-Mead is not very high while the number of diverged solutions doesn't differ dramatically except for the cases of an exceptionally "bad" start like with the values of the previous day. Running multiple optimization with Nelder-Mead does not improve RMSE significantly, whereas it takes significantly more time. When it comes to the economic interpretation of the solution, we believe that with Nelder-Mead it is difficult to obtain. The possible reason for that is that instead of converging to a local minimum in the area of a solution with a low correlation between the factor loadings like, for example, L-BFGS-B does, it finds a minimum with a lower value of RMSE, but with highly correlated loadings.

The last numerical method, Trust Constrained, is the most robust out of the four. In fact, there is not much difference in terms of accuracy among the starting values that we used, so for the purpose of fitting many initial points would do, even to a larger extent than with Nelder-Mead. The variation in RMSE is the lowest among the methods, just like the variation in the yields. Though, it is quite effective in minimizing the RMSE, such solutions lack economics interpretation for the same reasons as Nelder-Mead.

Bibliography

[AAS18] Emrah Ahi, Vedat Akgiray, and Emrah Sener. “Robust term structure estimation in developed and emerging markets”. In: *Annals of Operations Research* 260.1 (2018), pp. 23–49. DOI: 10.1007/s10479-016-2282-5. URL: https://doi.org/10.1007/s10479-016-2282-5.

[De 07] Michiel De Pooter. “Examining the Nelson-Siegel class of term structure models: In-sample fit versus out-of-sample forecasting performance”. In: *Available at SSRN 992748* (2007).

[DL06] Francis X. Diebold and Canlin Li. “Forecasting the term structure of government bond yields”. In: *Journal of Econometrics* 130.2 (2006), pp. 337–364. ISSN: 0304-4076. DOI: https://doi.org/10.1016/j.jeconom.2005.03.005. URL: http://www.sciencedirect.com/science/article/pii/S0304407605000795.

[Due97] Michael J Dueker. “Strengthening the Case for the Yield Curve as a Predictor of US Recessions”. In: *Federal Reserve Bank of St. Louis Review* 79.2 (1997), p. 41.

[ET06] Arturo Estrella and Mary Trubin. “The yield curve as a leading indicator: Some practical issues”. In: *Current issues in Economics and Finance* 12.5 (2006).

[FB87] Eugene F Fama and Robert R Bliss. “The information in long-maturity forward rates”. In: *The American Economic Review* (1987), pp. 680–692.

[GGS10] Manfred Gilli, Stefan Große, and Enrico Schumann. “Calibrating the nelson-siegel-svensson model”. In: *Available at SSRN 1676747* (2010).

[HPW13] Grace Xing Hu, Jun Pan, and Jiang Wang. “Noise as information for illiquidity”. In: *The Journal of Finance* 68.6 (2013), pp. 2341–2382.

[Ioa03] Michalis Ioannides. “A comparison of yield curve estimation techniques using UK data”. In: *Journal of Banking & Finance* 27.1 (2003), pp. 1–26.

[MM09] Polychronis Manousopoulos and Michalis Michalopoulos. “Comparison of non-linear optimization algorithms for yield curve estimation”. In: *European Journal of Operational Research* 192.2 (2009), pp. 594–602.

[NS87] Charles Nelson and Andrew F Siegel. “Parsimonious Modeling of Yield Curves”. In: *The Journal of Business* 60.4 (1987), pp. 473–89. URL: https://EconPapers.repec.org/RePEc:ucp:jnlbus:v:60:y:1987:i:4:p:473-89.

[Sve95] Lars EO Svensson. “Estimating forward interest rates with the extended Nelson & Siegel method”. In: *Sveriges Riksbank Quarterly Review* 3.1 (1995), pp. 13–26.

8 Attachments

All the calculations were implemented in Python and can be accessed at the author's repository.

The partial derivatives of the reparametarized loss function for BFGS:

$$\frac{\partial L}{\partial u_0} = 2\sum_{i=1}^{N}\left[\left(\frac{P_i - \hat{P}_i(\Theta)}{1000}\right)\frac{1}{1000}\sum_{j=1}^{K}\Big[t_j \cdot\ CF_j \cdot\ \exp(-r(t_j,u)t_j)\cdot \right.$$
$$\left.(u_0 + \frac{u_1 - u_0 + u_2}{t}\Big(1 - \exp\Big(\frac{-t}{\tau}\Big)\Big[1 + \frac{t}{\tau}\Big]\Big) + \frac{u_2}{\tau^2}\cdot\exp\Big(\frac{-t}{\tau}\Big)\Big]\right] \tag{26}$$

$$\frac{\partial L}{\partial u_0} = 2\sum_{i=1}^{N}\left[\left(\frac{P_i - \hat{P}_i(\Theta)}{1000}\right)\frac{1}{1000}\sum_{j=1}^{K}\Big[t_j \cdot\ CF_j \cdot\ (1 - g(t_j|\tau)) \cdot\ \exp(-r(t_j,u)t_j)\Big]\right] \tag{27}$$

$$\frac{\partial L}{\partial u_1} = 2\sum_{i=1}^{N}\left[\left(\frac{P_i - \hat{P}_i(\Theta)}{1000}\right)\frac{1}{1000}\sum_{j=1}^{K}\Big[t_j \cdot\ CF_j \cdot\ g(t_j|\tau) \cdot\ \exp(-r(t_j,u)t_j)\Big]\right] \tag{28}$$

$$\frac{\partial L}{\partial u_2} = 2\sum_{i=1}^{N}\left[\left(\frac{P_i - \hat{P}_i(\Theta)}{1000}\right)\frac{1}{1000}\sum_{j=1}^{K}\Big[t_j \cdot\ CF_j \cdot\ h(t_j|\tau) \cdot\ \exp(-r(t_j,u)t_j)\Big]\right] \tag{29}$$