ARTIFICIAL INTELLIGENCE COMBINED WITH NONLINEAR OPTIMIZATION TECHNIQUES AND THEIR APPLICATION FOR YIELD CURVE OPTIMIZATION

ROYA SOLTANI*

Department of Industrial Engineering, Science and Research Branch
Islamic Azad University, Tehran, Iran

SEYED JAFAR SADJADI AND MONA RAHNAMA

Department of Industrial Engineering
Iran University of Science and Technology, Tehran, Iran

(Communicated by Duan Li)

ABSTRACT. This study makes use of the artificial intelligence approaches combined with some nonlinear optimization techniques for optimization of a well-known problem in financial engineering called yield curve. Yield curve estimation plays an important role on making strategic investment decisions. In this paper, we use two well-known parsimonious estimation models, Nelson-Siegel and Extended Nelson-Siegel, for the yield curve estimation. The proposed models of this paper are formulated as continuous nonlinear optimization problems. The resulted models are then solved using some nonlinear optimization and meta-heuristic approaches. The optimization techniques include hybrid GPSO parallel trust region-dog leg, Hybrid GPSO parallel trust region-nearly exact, Hybrid GPSO parallel Levenberg-Marquardt and Hybrid genetic electromagnetism like algorithm. The proposed models of this paper are examined using some real-world data from the bank of England and the results are analyzed.

1. Introduction. One of the most important issues on bond market is to estimate the yield curve in time horizon which is used for different purposes such as investment strategies, monetary policy, etc. In the bond market, there are different kinds of bonds with different times of maturities to invest. In this paper, a bond with no coupon payment, zero-coupon bond, is considered where its yield to maturity is the spot interest rate for its maturity. Since these decisions are crucial, the yield curves and their estimation methods have been broadly considered. The estimation models of the yield curve can be classified into two popular groups: (a) Parsimonious yield curve models and (b) Spline-based models. Parsimonious models present a function for forward interest rate and have some parameters to be estimated whereas spline-based methods approximate the yield curve by some piecewise polynomial functions, which can be either quadratic or cubic [1]. In this paper, two parsimonious models, Nelson-Siegel and Extended Nelson-Siegel, are taken into

2010 Mathematics Subject Classification. Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Artificial intelligence, nonlinear programming, gradient search methods, meta-heuristic approaches, yield curve optimization, parsimonious method, financial engineering.

* Corresponding author: Roya Soltani.
consideration, which are essentially nonlinear optimization problems. Thus, nonlinear optimization techniques such as hybrid parallel versions of Trust region-dog leg, Trust region- nearly exact and Levenberg-Marquardt methods are employed for the first time to solve the resulted problem formulation. The hybridization is occurred by employing the hybrid genetic particle swarm optimization technique. In addition, the hybrid genetic electromagnetism like meta-heuristics is also employed to solve the problem. In the following, the relevant literature is reviewed in two scopes of yield curve estimation modeling and optimization. Nelson and Siegel [18] are the first who present the original parsimonious model of the yield curve by introducing a function with a few parameters for the forward interest rates. The model is considered either monotonic or with a hump. Later, Svensson [23]-[24] and Soderlind and Svensson [22] extend the model by adding some parameters which allows the curves to have a second hump. The new model is known as Extended Nelson-Siegel model or Svensson model which captures a greater variety of shapes for yield curves. Pooter [21] examined various extensions of the Nelson and Siegel model [18] with the purpose of fitting and forecasting the term structure of interest rates. Bolder and Streliski [5] investigated the behaviour of the Nelson-Siegel and the Svensson yield curve models on the data from the bank of Canada. Mcculloch [16] develops the quadratic spline-based model to estimate the yield curve. The cubic spline method proposed by Mcculloch [17] seems to have a better performance than the quadratic spline models. Bliss [4] compares five distinct methods for estimating the term structure: the Unsmoothed Fama-Bliss method, the Smoothed Fama-Bliss method, Mcculloch method (cubic spline), the Fisher-Nychka-Zervos method (cubic spline) and the extended Nelson-Siegel method. Also, there is an approach in which the nonlinear parameters are predetermined [8]. There are different optimization techniques to solve yield curve estimation models. Csajbok [7] compares different kinds of yield curve models using the Gauss-Newton algorithm. Ioannides[11] uses the BFGS algorithms for estimating the yield curve. Manousopoulos and Michalopoulos [13]-[14] study the yield curve estimation methods in Greek bond market and define the models of estimating the yield curve as a nonlinear optimization problem. Manousopoulos and Michalopoulos [15] compare different nonlinear optimization algorithms in order to estimate the yield curves. In this work, they use parsimonious modeling of the yield curve and some algorithms such as Nelder-Mead, Powel, Simulated annealing, BFGS and generalized reduced gradient are implemented to solve the yield curve optimization problem. The remaining of the paper is organized as follows. The studying issue of this paper is elaborated in section 2. Section 3 describes the solution procedure used to solve the yield curve optimization. In section 4 the computational results are presented and the proposed methods are compared. Finally in section 5 we conclude the paper along with proposing future research directions.

2. **Problem statement.** One important decision in the bond market is to estimate and optimize the yield curve. Yield curve is a curve that depicts the term structure of interest rate and describes the yield as a function of time. There are various types of yield curve. The frequently used one is spot interest rate curve or zero-coupon yield curve which considers bonds with no coupon payment. To understand about the spot interest rate for maturity $m$, consider a bond with $m$ payments. The price of a bond is determined through the present value of its future cash flows (see formula 1).
\[
\text{bondprice} = \sum_{j=1}^{m} C \times \delta(t_j) + R \times \delta(t_m)
\]  
(1)

where \( C \) is the periodic coupon payment in times \( t_j \) \( (j = 1, 2, m) \), \( R \) is the redemption payment in time \( t_m \) and is the discount function. The discount function is continuously compounded according to formula 2.

\[
\delta(t) = e^{-t \times s(t)}
\]  
(2)

where \( s(t) \) denotes the spot interest rate for maturity time \( t \). The instantaneous forward rates which guarantee the return of the investment in a single period are then defined by formula 3.

\[
fr(t) = \frac{-\delta'(t)}{\delta(t)}
\]  
(3)

Finally, the spot interest rate for maturity \( m \) is calculated as the mean forward rate over the interval \([0, m]\) (see formula 4).

\[
s(m) = \frac{1}{m} \int_0^m fr(t) dt = \frac{1}{m} \int_0^m \frac{-\delta'(t)}{\delta(t)} dt = \frac{-\ln(\delta(m))}{m}
\]  
(4)

2.1. Estimation of the zero-coupon yield curve. Formula 4 presents a general form for the spot interest rate which depends on how the instantaneous forward rate is defined. To estimate the zero-coupon yield curve, we make use of two well-known parsimonious methods, i.e. Nelson-Siegel and Extended Nelson-Siegel. First, the following notations are introduced.

**Notations**

- \( m \): maturity
- \( fr(m) \): The instantaneous forward rates for maturity \( m \)
- \( s(m) \): The spot interest rate for maturity \( m \)
- \( \beta_0 \): a level of the curve or a long-term component of the model
- \( \beta_1 \): the slope component of the model
- \( \beta_2 \): curvature component of the model(first humped shaped)
- \( \beta_3 \): curvature component of the model (second humped shaped)
- \( \tau_1 \): the speed at which slope and curvature change or decay to zero
- \( \tau_2 \): the speed at which the second curvature part of the yield curve changes or decays to zero

where \([\beta_0, \beta_1, \beta_2, \tau_1]\) and \([\beta_0, \beta_1, \beta_2, \beta_3, \tau_1, \tau_2]\) are the parameters to be estimated in Nelson-Siegel and Extended Nelson-Siegel, respectively

a. Nelson-Siegel estimation model. Nelson and Siegel [18] presented a model based on the zero-coupon bond market data in which the yield curve has a monotonic shape and can have one hump. Under these assumptions the forward interest rate is defined with formula 5.

\[
fr(m) = \beta_0 + \beta_1 \exp(-m/\tau_1) + \beta_2 \exp(m/\tau_1) \cdot \exp(-m/\tau_2)
\]  
(5)

The corresponding spot interest rate for maturity \( m \) is then defined with formula 6.

\[
s(m) = \frac{1}{m} \int_0^m fr(t) dt = \beta_0 + \beta_1 \left[ \frac{1 - \exp(-m/\tau_1)}{\tau_1} \right] + \beta_2 \left[ \frac{1 - \exp(-m/\tau_1)}{\tau_1} - \exp(-m/\tau_1) \right]
\]  
(6)
b. Extended Nelson-Siegel estimation model. To improve the flexibility of the Nelson-Siegel model and let the estimation model to have a second hump, Svensson[23]-[24] and Soderlind and Svensson[22] added two other parameters to the previous model. The function of the forward interest rate is defined according to formula 7.

\[ fr(m) = \beta_0 + \beta_1 \exp\left(-\frac{m}{\tau_1}\right) + \beta_2 \left(\frac{m}{\tau_2}\right) + \beta_3 \exp\left(-\frac{m}{\tau_2}\right) \]  

(7)

The corresponding spot interest rate for maturity \( m \) is then defined with formula 8.

\[
s(m) = \frac{1}{m} \int_0^m fr(t)dt = \beta_0 + \beta_1 \left[ 1 - \exp\left(-\frac{m}{\tau_1}\right) \right] + \beta_2 \left[ 1 - \exp\left(-\frac{m}{\tau_1}\right) \right] - \exp\left(-\frac{m}{\tau_2}\right) + \beta_3 \left[ 1 - \exp\left(-\frac{m}{\tau_2}\right) \right] - \exp\left(-\frac{m}{\tau_2}\right) \]

(8)

Where \( \tau_2 \) and \( \beta_2 \) describe the time of the second hump and the magnitude and the direction of this hump, respectively.

Through Maclaurin series (exp(\( x \)) = 1 + \( x \) + \( \frac{x^2}{2!} \) + \( \frac{x^3}{3!} \) + \( \ldots \) + \( \frac{x^n}{n!} \) + \( \ldots \)) and using the first two sentences the extended Nelson-Siegel can be approximated according to \( z(m) = \beta_0 + \beta_1 + \beta_2 \left(\frac{m}{\tau_1}\right) + \beta_3 \left(\frac{m}{\tau_2}\right) \) which is still nonlinear in \( \tau_1 \) and \( \tau_2 \).

2.2. Optimization of the zero-coupon yield curve. The optimization problem is defined as the minimization of the squared yield errors between the estimated yield curve and the market data subject to some constraints. For Extended Nelson-Siegel model that is more general than Nelson-Siegel model, the optimization problem is formulated according to model 9.

\[
\min_{\beta, \tau} \left\{ \sum_{j=1}^{m} (s_{j,m} - s(m))^2 \right\} \\
\text{s.t.} \quad 0 \leq \tau_1 \leq N \\
0 \leq \tau_2 \leq N \\
\beta_0 \geq 0 \\
\beta_0 + \beta_1 \geq 0 \\
\]

(9)

Where \( N \) is an number of maturities and \( s_{j,m} \) is market data of the \( j \)th maturity at time \( m \).

3. Solution procedures. As described in the previous section, yield curve model is nonlinear which reinforce the complexity of the problem. Nonlinear nature of the problem suggests nonlinear methods and complexity of the problem offers meta-heuristic approaches. In the followings, both approaches are presented. In all methods, the solution is represented by \( X \), for example for Extended Nelson-Siegel we have \( X = [\beta_0, \beta_1, \beta_2, \beta_3, \tau_1, \tau_2] \). In general, a new solution is obtained by \( X^{t+1} = X^t + \alpha^t d^t \), where \( d^t \) and \( \alpha^t \) are the direction and step size of the move, respectively. Each procedure has a special method for determining \( d^t \) and \( \alpha^t \).
3.1. Meta-heuristic approaches.

3.1.1. Hybrid genetic particle swarm optimization. Particle swarm optimization (PSO) is a computational approach initially proposed by [12] which is inspired from social behavior of bird flocking or fish schooling to find a rich source of food. PSO has been successfully used for continuous optimization problems where there is no need of having the gradient information of the problem. In this approach, a swarm consisting of a number of particles flies around in an $N$ dimensional search space. In optimization problems, each swarm plays the role of a solution. Fig. 1 shows the search space of the particles graphically. To move from one solution to a new one, PSO makes use of a velocity vector which simultaneously determines the direction of the fly and step of the move. The position of each particle is updated through formula 10. Following Eberhart and Shi [9] and Clerc and Kennedy [6], the modified velocity update presented in formula 11 is used which has better performance and convergence rate than the original PSO.

$$X^t_{ij} = X^t_{ij} + \nu^t_{ij}$$  \hspace{1cm} (10)

$$\nu^t_{ij} = \chi \times \nu^t_{ij} + \chi \times C_1 \times \text{rand}_1 \times (pbest^t_{ij} - X^t_{ij}) + \chi \times C_2 \times \text{rand}_2 \times (gbest^t_{ij} - X^t_{ij})$$  \hspace{1cm} (11)

where, $\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}}$, $\phi = C_1 + C_2 > 4$. $X^t_{ij}$ is the position (variable) $j$ of the $ith$ particle (solution) in the iteration $t$. $\nu^t_{ij}$ is the velocity of the $ith$ particle in the $jth$ position which simultaneously determines the direction of the fly and step of the move. A new solution in the next iteration is obtained by formula 10. $C_1$ and $C_2$ are acceleration constants which range from $[0, 4]$. $pbest^t_{ij}$ is the best position of particle $i$ in the $jth$ position at time $t$. Hence, $pbest^t_{ij} - X^t_{ij}$ is known as cognitive component. $gbest^t_{ij}$ is the $jth$ position of the global best particle. Therefore, $gbest^t_{ij} - X^t_{ij}$ is known as social component.

In the proposed hybrid genetic particle swarm optimization (HGPSO) the swarm undergo crossover and mutation operation to preserve exploration and exploitation of the search. This paper takes advantage of arithmetic crossover and non-uniform mutation in HGPSO. The main steps of this method are summarized in procedure 1.

**Procedure1 HGPSO ()**

Begin
1. Define the crossover probability ($pc$), maximum number of iterations (maxiteration);
2. Randomly generate a population of initial points of the size popsize;
3. for $t=1$ to maxiteration do
4. Evaluate the particles with respect to objective function of model 9;
5. Update personal best position of each particle (pbest);
6. Update global best position (gbest);
7. Velocity_Update();
8. Position_Update();
9. Repair the infeasible solutions;
10. Calculate the probability of each particle in the population;
11. for $i=1$:$pc \times popsize$ do
12. Apply roulette wheel selection two times to select two candidate solutions for crossover;
13. Arithmatic_crossover();
14. Repair the infeasible solutions;
15. Calculate the velocity of the two generated solutions as sum of the velocity vectors of the candidate solutions normalized to the original length of each candidate solution velocity vector;
16. Update pbest and gbest;
17. end(for)
18. for \( j=1: \text{popsize} - \text{pc} \times \text{popsize} \)
19. Randomly select a candidate solution;
20. Nonuniform\_mutation();
21. Repair the infeasible solutions;
22. update pbest and gbest;
23. end(for)
24. Update \( \chi \);
25. end(for)
End.

3.1.2. Hybrid genetic electromagnetism like algorithm. Electromagnetism like algorithm (EM) is a population based meta-heuristic which was originally designed for continuous optimization problems by [3]. The method stems from electromagnetism theory of physics in which each electrically charged particle plays the role of potential solutions in the solution space. The particles use an attraction-repulsion mechanism to move towards optimal solution. New position of each particle is determined through a random step length \( \lambda \) in the direction of the total force exerted on that particle from other particles. Fig. 2 shows the total force exerted on particle \( i \) from particles \( j \) and \( k \). Formula 12 presents the new position of the moved particle in which \( \text{RNG} \) is the allowed feasible length and the force \( F_i \) is calculated through Formulas 13 and 14.

\[
X_{i}^{t+1} = \chi_{i}^{t} + \lambda F_{i} / \| F_{i} \| (\text{RNG}), i = 1, 2, \ldots, m
\]  

\[
q_{i} = \exp(-n \cdot \frac{f(X_{i}) - f(X_{\text{best}})}{\sum_{k=1}^{m} (f(X_{k}) - f(X_{\text{best}}))})
\]

\[\text{Figure 1. PSO solution space} \quad \text{Figure 2. EM solution space}\]
The proposed hybrid genetic electromagnetism like algorithm (HGEM) uses genetic operators such as arithmetic crossover and non-uniform mutation operators in its process. The main steps of this algorithm are presented in procedure 2.

**Procedure 2 HGEM()**

1. Define the crossover probability ($pc$), maximum number of iterations ($maxiteration$) and population size ($popsize$);
2. Generate a population of $m$ particles having $n$ dimensions;
3. for $t=1$ to $maxiteration$ do
4. Calculate the probability of each particle in the population;
5. for $i=1:pc \times popsize$ do
6. Apply roulette wheel selection two times to select two candidate solutions for crossover;
7. Arithmetic_crossover();
8. Repair the infeasible solutions;
9. end(for)
10. for $j=1:(popsize - pc \times popsize)$
11. Randomly select a candidate solution;
12. Nonuniform_mutation();
13. Repair the infeasible solutions;
14. end(for)
15. Select $popsize$ number of best particles from the old and new populations;
16. Apply local search on each particle of this selected population coordinate by coordinate;
17. Calculate the amount of charge of each particle using formula 13;
18. Calculate the total force exerted on each particle using formula 14;
19. Normalize the force vector;
20. Move particles except best one in the direction of the force using formula 12;
21. end(for)

**End.**

3.2. Search methods using derivatives. Derivative search methods start with an initial solution and iteratively move to a new position in the direction $d^t$ which depends on the gradient of the problem with a specified step size $\alpha^t$. formula 15 shows the new position in the solution space [19]-[2].

$$X^{t+1} = X^t + \alpha^t d^t$$  \hspace{1cm} (15)

The two famous derivative search methods are gradient method or steepest descent and Newton method which respectively use formulas 16 and 17 as their directions in which $\nabla f(X^t)$ and $H(X^t)$ are in turn the gradient and Hessian of the objective function at point $X^t$. Note that the plus sign is used for maximization problems and the minus sign is used for minimization problems. In Newton method $\alpha^t$ is unit whilst in gradient method is not necessarily unit.

$$d^t = \pm \nabla f(X^t)$$  \hspace{1cm} (16)
\[ d^t = \pm \frac{\nabla f(X^t)}{H(X^t)} \] (17)

Hereafter denote \( \alpha^t d^t \) with \( P \). Hence, for Newton method in case of the minimization problem we have:

\[ P = -\frac{\nabla f(X^t)}{H(X^t)} \]

The drawback of these methods is that the gradient method rapidly approaches to the optimal solution but acts in a zigzag manner near the optimal solution whilst Newton method acts well only when its initial solution is near the optimal solution. To overcome these deficiencies modifications of these methods are used. In the following subsections new versions of the two modifications, Trust region and Levenberg-Marquardt, are presented.

3.2.1. Hybrid genetic PSO parallel trust region (HGPSOPTR). The original trust region starts with an initial solution and iteratively searches the solution space within a circular region known as trust region. The performance of the method highly depends on the choice of initial solution. To overcome this drawback a parallel algorithm is designed to search different regions of the solution space with their assigned trust regions in parallel. To select these regions a hybrid genetic PSO algorithm is employed. Fig. 3 shows an example search of solution space with five different trust regions. Procedure 3 summarizes the main steps of the proposed hybrid parallel trust region.

**Figure 3. Parallel trust regions**

**Procedure 3 HGPSOPTR()**

**Begin**

1. Define the number of processors (Num_pr), maximum number of iterations for the algorithm (maxiteration), maximum number of iterations for the trust region (maxit);
2. Determine maximum trust region radius(Max_radius);
3. Choose the initial $Tr_{\text{radius}}$ in the interval of $(0, \text{Max} \_ \text{radius})$;
4. Determine $\eta \in [0, \frac{1}{4})$;
5. for $itr=1$ to maxiteration do
6. for $pr=1$ to Num.pr do in parallel
7. HGPSO()
8. for $t=1$ to maxit do
9. Obtain $P$ applying either Procedure 4 or Procedure 5;
10. Evaluate $\rho^t = \frac{f(X^t) - f(X^{t+1})}{\nabla f(X^t) \times P - 0.5 \times P^T \times H \times P}$;
11. if $\rho^t < \frac{1}{4}$
12. $Tr_{\text{radius}} = \frac{1}{4} \|P\|$
13. else if $\rho^t > \frac{3}{4}$ and $\|P\| = Tr_{\text{radius}}$
14. $Tr_{\text{radius}} = \min(2 \times Tr_{\text{radius}}, \text{Max} \_ \text{radius})$
15. end(if)
16. if $\rho^t > \eta$
17. $X^{t+1} = X^t + P$
18. else
19. $X^{t+1} = X^t$
20. end(if)
21. if norm(gradient) $\leq$ specified_tolerance
22. break;
23. end(if)
24. end(for)
25. end(for)
26. The best solution obtained from all processors is gathered by a central unit and
the next iteration starts with the best solution of each processor.
27. end(for)

**End.**

Dog leg (DL). The dogleg method finds an approximate solution $P$ considering the
trust region radius constraint. It first finds the unconstrained minimizer $P^{\text{Gradient}}$
along the steepest descent or gradient direction, $-\nabla f$, defined by formula 18.

$$P^{\text{Gradient}} = \frac{\nabla f^T \times \nabla f}{\nabla f^T \times H \times \nabla f} \times (-\nabla f) \tag{18}$$

If the trust region radius is small and the solution $P^{\text{Gradient}}$ is not in the trust
region radius set $P$ according to formula 19.

$$P = \frac{Tr_{\text{radius}}}{\|\nabla f\|} \times (-\nabla f) \tag{19}$$

If $P^{\text{Gradient}}$ is in the trust region area find an unconstraint minimizer $P^{\text{Newton}}$
according to the Newton method considering formula 20.

$$P = \frac{-\nabla f(X^t)}{H(X^t)} \tag{20}$$

If it is feasible, set $P = P^{\text{Newton}}$. Otherwise, the solution $P$ follows a curved
trajectory like the one presented in Fig. 4. The approximate solution $P$ is found
by replacing the curved trajectory with a path consisting of two line segments.
The first line segment runs from the initial point to the unconstrained minimizer $P_{Gradient}$ along the steepest descent or gradient direction, $-\nabla f(X^t)$ . The second line segment runs from $P_{Gradient}$ along the Newton direction, $-\nabla f(X^t)/H(X^t)$ . The new solution $X^{t+1}$ is the intersection of the dogleg path and the boundary of the trust region. In this case $P$ is specified according to formula 21 in which $\tau$ is calculated by solving the quadratic equation 22.

$$P = P_{Gradient} + (\tau - 1)(P_{Newton} - P_{Gradient})$$  \hspace{1cm} (21)

$$\|P_{Gradient} + (\tau - 1)(P_{Newton} - P_{Gradient})\|^2 = (Tr\_radius)^2$$  \hspace{1cm} (22)

Figure 4. Dogleg trajectory

Procedure 4 summarizes the main steps of the dogleg method.

**Procedure 4 Dogleg ()**

Begin
1. if $\nabla f^T \times H \times \nabla f \leq 0$
2. set $P$ according to formula 19;
3. else
4. Calculate $P_{Gradient}$ according to equation 18;
5. if $\|P_{Gradient}\| \geq Tr\_radius$
6. set $P$ according to formula 19;
7. else
8. Calculate $P_{Newton}$ according to equation 20;
9. if
10. set $P = P_{Newton}$ ;
11. else
12. calculate $P$ according to equation 21;
13. end(if)
14. end(if)
Nearly exact (NE). As shown in formula 17 the direction of the Newton method is not defined when the Hessian matrix is zero. Therefore the Hessian matrix is modified according to formula 23.

\[ H_{\text{modified}}(X^t) = H(X^t) + \lambda I \quad (23) \]

Then \( \lambda \) is updated until a feasible point in a trust region is achieved. The procedure of this method is shown in procedure 5.

**Procedure 5 NE()**
Begin
1. Initialize \( \lambda_0 \) and trust region radius (\( Tr\_radius \));
2. for \( k=0, 1,2, \)
3. Calculate Cholesky factorization of \( H(X^t) + \lambda_k I \) and set it as \( R^T R \);
4. Solve \( R^T P_k = -\nabla f(X^t) \) and find \( P_k \);
5. Solve \( R^T q_k = P_k \);
6. set \( \lambda_{k+1} = \lambda_k + (P_k q_k)^2 \times \left( \frac{P_k - Tr\_radius}{Tr\_radius} \right) \);
7. end(for)
End.

3.2.2. **Hybrid genetic PSO parallel Levenberg-Marquardt (HGPSOPLM).** Due to ill-conditioning which might occur at point where the Hessian is singular or near singular, the modification of Hessian matrix is used according to formula 24, where \( I \) is an identity matrix.

\[ H_{\text{modified}}(X^t) = H(X^t) + \varepsilon^t I \quad (24) \]

The algorithm starts with an initial feasible solution and iteratively improves the position of the solution in the solution space according to formula 25.

\[ X^{t+1} = X^t - \frac{\nabla f(X^t)}{H_{\text{modified}}(X^t)} \quad (25) \]

At each iteration of the algorithm the value of the \( \varepsilon^t \) is modified regarding the value of \( \rho^t \) which is calculated according to formula 26.

\[ \rho^t = \frac{f(X^t) - f(X^{t+1})}{-0.5 \times \nabla f(X^t) \times P} \quad (26) \]

Where,

\[ P = -\frac{\nabla f(X^t)}{H_{\text{modified}}(X^t)} \quad (27) \]

Since the performance of the Levenberg-Marquardt algorithm depends on the choice of initial solution, a new hybrid parallel version of Levenberg-Marquardt is suggested in this paper in which a number of initial points are generated through PSO and Levenberg-Marquardt methods is implemented at each point in parallel. The main steps of the new proposed method are summarized in procedure 5.

**Procedure 5 HGPSOPLM**
Begin
1. Initialize the levenberg-Marquardt parameters: mu0 = 0.1; mulow = 0.25; muhigh = 0.75; omup = 2; omdown = 0.5;
2. Define the number of processors (Num\_pr)
3. for \(itr = 1\) to \(maxiteration\) do
4. for \(pr = 1\) to \(Num_{pr}\) do in parallel
5. HGPSO()
6. for \(t = 1:\maxit\) do
7. calculate \(P\) using formula 13;
8. \(X_{t+1}^{t} = X_{t}^{t} + P^{t}\);
9. Evaluate \(X_{t+1}^{t}^{t+1}\);
10. Calculate \(\rho^{t}\) using formula 12;
11. if \(\rho^{t} < \text{mulow}\)
12. \(\varepsilon^{t+1} = \text{omup} \times \varepsilon^{t}\)
13. else if \(\rho^{t} > \text{muhigh}\)
14. \(\varepsilon^{t+1} = \text{omdown} \times \varepsilon^{t}\)
15. end(if)
16. if \(\text{norm(gradient)} < \text{specified\_tolerance}\)
17. break;
18. end(if)
19. end(for)
20. end(for)
21. The best solution obtained from all processors is gathered by central unit and the next iteration starts with the best solution of each processor.
22. end(for)
End.

4. Computational results. A computational experiment is conducted to investigate the performance of the proposed methods. First, the methods are run using the nominal data (“zero-coupon” yields) of bank of England\(^1\) in 2011 from January 4 to January 19. Each set of data in each algorithm is run regarding Nelson-Siegel and Extended Nelson-Siegel yield models. For each model, the results are averaged upon all data and the averages and standard deviations of the errors are reported in Table 1. Fig. 5 and Fig. 6 illustrate the interval plot for Extended Nelson-Siegel and Nelson-Siegel models. Also, Fig. 7 and Fig. 8 compare the proposed methods on Extended Nelson-Siegel and Nelson-Siegel models considering 12 sets of data taken from bank of England from January 4 to January 19. As can be seen for Extended Nelson-Siegel model, HGPSOPLM results in minimum error. In case of Nelson-Siegel model, HGPSOPTR-DL has minimum error. Amongst the two proposed meta-heuristics HGEM has minimum error in both considered models. The implementation of HGPSO lonely results the maximum error but we could get better results by hybridizing it with the proposed parallel nonlinear methods. The improvement on the results is significant when Extended Nelson-Siegel model is used compared with the case of Nelson-Siegel model. The average and standard deviation of parameters are also presented in Table 2 and 3 for Extended Nelson-Siegel and Nelson-Siegel models, respectively. Table 4 and Table 5 show the eigenvalues of the Hessian matrix. For Nelson-Siegel model all eigenvalues are positive and all gradient norms are near zero. Therefore one may conclude that the Hessian matrix is positive definite and the resulted solutions are optimal. In the appendix Figs. 9a to 9f show how the obtained yield curves fit market data on January 4 which is a normal upward sloping yield curve. This shape happens when an economy is growing and investors are confident. This is the significant signs of development.

\(^1\)http://www.bankofengland.co.uk/statistics/Pages/yieldcurve/archive.aspx
Table 1. Fitness Error

| Methods     | Nelson-Siegel | Extended Nelson-Siegel |
|-------------|---------------|------------------------|
|             | Average       | Standard deviation     | Average       | Standard deviation     | Average       | Standard deviation     |
| HEM         | 0.106920      | 0.025078               | 1.07608      | 0.007397               | 0.028606582  | 2.761143               |
| HEMI        | 0.103976      | 0.01973                | 1.05098      | 0.012035               | 0.00613485   | 1.89916                |
| HEMI-NE     | 0.102208      | 0.023577               | 29.1853      | 0.004867               | 0.002573407  | 36.364                 |
| HEMI-DL     | 0.102167      | 0.023679               | 31.3681      | 0.005108               | 0.002877644  | 37.85                  |
| HEMI-PTR    | 0.102533      | 0.006306               | 27.4275      | 0.004783               | 0.002648785  | 35.74                  |

Table 2. Parameter values for Extended Nelson-Siegel model

| Methods     | β₀       | Standard deviation | β₁       | Standard deviation | β₂       | Standard deviation | τ₁       | Standard deviation |
|-------------|----------|--------------------|----------|--------------------|----------|--------------------|----------|--------------------|
| HEM         | 9.496    | 0.384              | -8.982   | 0.378              | -6.656   | 0.142              | 2.482    | 0.061              |
| HEMI        | 9.470    | 0.350              | -8.978   | 0.381              | -6.585   | 0.035              | 2.469    | 0.078              |
| HEMI-NE     | 9.796    | 1.007              | -9.324   | 1.035              | -6.688   | 0.549              | 2.501    | 0.471              |
| HEMI-DL     | 9.917    | 0.849              | -9.448   | 0.866              | -6.738   | 0.531              | 2.442    | 0.243              |
| HEMI-PTR    | 9.957    | 0.850              | -9.491   | 0.873              | -6.725   | 0.574              | 2.462    | 0.111              |

Table 3. Parameter values for Nelson-Siegel model

| Methods     | β₀       | Standard deviation | β₁       | Standard deviation | β₂       | Standard deviation | τ₁       | Standard deviation |
|-------------|----------|--------------------|----------|--------------------|----------|--------------------|----------|--------------------|
| HEM         | 5.2881   | 0.0004             | -3.8043  | 0.0053             | -5.4705  | 0.0093             | 1.5192   | 0.0013             |
| HEMI        | 5.2467   | 0.0919             | -3.5282  | 0.0732             | -5.2209  | 0.3462             | 1.6519   | 0.2967             |
| HEMI-NE     | 5.8321   | 0.0581             | -3.4160  | 0.0458             | -4.1469  | 0.0422             | 2.2462   | 0.0073             |
| HEMI-DL     | 5.0028   | 0.0806             | -4.3148  | 0.8265             | -4.0892  | 4.9073             | 2.274    | 2.6319             |
| HEMI-PTR    | 5.0832   | 0.0581             | -4.3076  | 0.0458             | -4.2517  | 0.0422             | 2.1824   | 0.0072             |

which requires effective and adequate investment [10]. In this situation investors demand an additional yield for longer maturity bonds. This is logical when there is more risk associated with long term investment. In sum, healthy economies nearly always have an upward sloping yield curve, although the interest rates that make up that curve may differ substantially from one period to another. To support our findings, the proposed algorithms were tested again on a data set from a recent year with respect to the Extended Nelson-Siegel model which has been proven to be a more accurate model than Nelson-Siegel model. We used a rather larger data set, i.e. a quarter, from 1 April till 30 June in 2015. The results have been shown in Table 6. The results support the superiority of the HEMI-PTR algorithm.

For more analysis, we estimated the yield curve for the data of a crisis period, i.e. May 2008, through the Extended Nelson-Siegel model and the HEMI-PTR algorithm which are respectively an accurate estimation model and the best algorithm in our experiments. The performance of the Levenberg-Marquardt (LM) algorithm depends on the choice of the initial solution. One time, the LM algorithm was run with a randomly generated initial solution. Another time, the LM algorithm was run with the estimation of the yesterday as an initial solution. Finally, the LM algorithm was run with HEMI results as an initial condition. The results are
Table 4. Extended Nelson-Siegel Eigenvalues and Gradient norms

| Method       | Data | Eigenvalues | Gnorm |
|--------------|------|-------------|-------|
|              |      |             |       |
| HGPSOPTR-NE  | Set1 | 219.717     | 7.245 | 2.530 | 0.041 | 0.0009 | 0.0004 |
|              | Set2 | 220.752     | 7.111 | 2.748 | 0.042 | 0.0001 | 0.011 |
|              | Set3 | 214.848     | 6.942 | 2.733 | 0.041 | 0.0009 | 0.006 |
|              | Set4 | 188.925     | 5.999 | 3.557 | 0.046 | 0.001  | 0.026 |
|              | Set5 | 198.835     | 6.254 | 3.134 | 0.072 | -0.005 | 0.003 |
|              | Set6 | 184.056     | 5.863 | 2.929 | 0.039 | 0.0002 | 0.006 |
|              | Set7 | 220.411     | 6.617 | 3.233 | 0.043 | 0    | 0.0009 |
|              | Set8 | 201.848     | 6.490 | 2.754 | 0.039 | 0    | 0.0007 |
|              | Set9 | 201.782     | 6.450 | 2.712 | 0.047 | -0.002 | 0.002 |
|              | Set10| 192.970     | 6.112 | 2.971 | 0.040 | 0.0007 | 0.002 |
|              | Set11| 192.574     | 6.087 | 3.086 | 0.043 | -0.001 | 0.001 |
|              | Set12| 206.778     | 6.637 | 2.896 | 0.042 | 0.0001 | 0.004 |
|              | Average| 203.624    | 6.484 | 2.940 | 0.045 | -0.0005 | 0.012 |

| HGPSOPTR-DL  | Set1 | 215.781     | 7.093 | 2.624 | 0.042 | 0    | 0.0009 |
|              | Set2 | 217.899     | 7.074 | 2.697 | 0.042 | 0    | 0.0009 |
|              | Set3 | 210.860     | 6.806 | 2.826 | 0.042 | 0    | 0.0009 |
|              | Set4 | 208.092     | 6.804 | 2.728 | 0.041 | 0    | 0.0008 |
|              | Set5 | 198.835     | 6.254 | 3.134 | 0.072 | -0.005 | 0.003 |
|              | Set6 | 180.292     | 5.633 | 3.236 | 0.042 | 0.0004 | 0.007 |
|              | Set7 | 220.411     | 6.617 | 3.233 | 0.043 | 0    | 0.002 |
|              | Set8 | 216.129     | 6.706 | 3.019 | 0.041 | 0.0001 | 0.002 |
|              | Set9 | 194.599     | 6.178 | 2.895 | 0.039 | 0.0001 | 0.007 |
|              | Set10| 195.845     | 6.264 | 2.811 | 0.039 | 0    | 0.0007 |
|              | Set11| 195.095     | 6.153 | 2.995 | 0.040 | 0    | 0.0007 |
|              | Set12| 207.949     | 6.682 | 2.881 | 0.042 | 0    | 0.0009 |
|              | Average| 205.149    | 6.522 | 2.940 | 0.045 | -0.0005 | 0.012 |

| HGPSOPLM     | Set1 | 225.272     | 7.332 | 2.931 | 0.042 | 0    | 0.0009 |
|              | Set2 | 225.272     | 7.332 | 2.588 | 0.042 | 0    | 0.0009 |
|              | Set3 | 212.355     | 6.863 | 2.781 | 0.042 | 0    | 0.0009 |
|              | Set4 | 210.691     | 6.901 | 2.684 | 0.042 | 0    | 0.0009 |
|              | Set5 | 198.835     | 6.254 | 3.134 | 0.072 | -0.005 | 0.003 |
|              | Set6 | 184.161     | 5.865 | 2.933 | 0.039 | 0.0002 | 0.006 |
|              | Set7 | 220.411     | 6.617 | 3.233 | 0.043 | 0    | 0.002 |
|              | Set8 | 201.916     | 6.494 | 2.751 | 0.039 | 0    | 0.0007 |
|              | Set9 | 195.245     | 6.211 | 2.868 | 0.039 | 0    | 0.0007 |
|              | Set10| 192.97      | 6.112 | 2.971 | 0.04  | 0.0001 | 0.002 |
|              | Set11| 192.575     | 6.088 | 3.086 | 0.043 | -0.001 | 0.001 |
|              | Set12| 206.734     | 6.636 | 2.895 | 0.042 | 0    | 0.0009 |
|              | Average| 204.904    | 6.536 | 2.886 | 0.044 | -0.0005 | 0.001 |

recorded in Table 7. As shown, the HGPSO results as initial solutions give better results which support the effectiveness of adding a global search component to the optimization algorithm. Figures 10a to 10e illustrate the resulted curve versus market data for selective days from May 2008. As can be seen, the shape of the yield curve is humped, i.e. first it is inverted, then grown and again inverted. In the inverted part of yield curve the economic outlook is very bleak. When the yield curve inverts, it indicates tough economic times ahead. For example, when the oil
| Method     | Data  | Eigenvalues | Gnorm  |
|-----------|-------|-------------|--------|
|           |       |       |        |
| HGPSOPTR-NE |      |       |        |
| Set 1     | 166.4771 | 6.6645 | 3.359 | 0.0499 | 0.0032 |
| Set 2     | 168.9473 | 6.8236 | 3.3593| 0.0498 | 0.0069 |
| Set 3     | 167.3736 | 6.8091 | 3.3541| 0.0489 | 0.0025 |
| Set 4     | 166.0604 | 6.5357 | 3.3699| 0.0494 | 0.0033 |
| Set 5     | 161.8373 | 5.9944 | 3.4018| 0.0482 | 0.0056 |
| Set 6     | 144.2963 | 3.0416 | 0.3247| 0.0013 | 0.0050 |
| Set 7     | 162.1993 | 5.941  | 3.4066| 0.0456 | 0.0081 |
| Set 8     | 164.1612 | 6.2443 | 3.3904| 0.0477 | 0.0105 |
| Set 9     | 162.8706 | 6.1114 | 3.3942| 0.0468 | 0.0054 |
| Set 10    | 163.7136 | 6.2145 | 3.3885| 0.0472 | 0.0090 |
| Set 11    | 167.1193 | 6.6272 | 3.3702| 0.0495 | 0.0005 |
| Set 12    | 167.1193 | 6.6272 | 3.3702| 0.0495 | 0.0005 |
| Average   | 163.1071 | 6.0838 | 3.1271| 0.0443 | 0.0045 |
|           |       |       |        |
| HGPSOPTR-DL |      |       |        |
| Set 1     | 166.4771 | 6.6645 | 3.359 | 0.0499 | 0.0032 |
| Set 2     | 168.9473 | 6.8236 | 3.3593| 0.0498 | 0.0069 |
| Set 3     | 167.3736 | 6.8091 | 3.3541| 0.0489 | 0.0025 |
| Set 4     | 166.0604 | 6.5357 | 3.3699| 0.0494 | 0.0033 |
| Set 5     | 161.8373 | 5.9944 | 3.4018| 0.0482 | 0.0056 |
| Set 6     | 145.1737 | 2.9842 | 0.0011| 0.3113 | 0.0065 |
| Set 7     | 162.1993 | 5.941  | 3.4066| 0.0456 | 0.0081 |
| Set 8     | 164.1612 | 6.2443 | 3.3904| 0.0477 | 0.0105 |
| Set 9     | 162.8706 | 6.1114 | 3.3942| 0.0468 | 0.0054 |
| Set 10    | 163.7136 | 6.2145 | 3.3885| 0.0472 | 0.0090 |
| Set 11    | 167.1193 | 6.6272 | 3.3702| 0.0495 | 0.0005 |
| Set 12    | 167.1193 | 6.6272 | 3.3702| 0.0495 | 0.0005 |
| Average   | 163.0494 | 6.0925 | 3.1002| 0.0735 | 0.0051 |
| HGPSOPLM  |       |       |        |
| Set 1     | 166.4771 | 6.6645 | 3.359 | 0.0499 | 0.0032 |
| Set 2     | 168.9473 | 6.8236 | 3.3593| 0.0498 | 0.0069 |
| Set 3     | 167.3736 | 6.8091 | 3.3541| 0.0489 | 0.0025 |
| Set 4     | 166.0604 | 6.5357 | 3.3699| 0.0494 | 0.0033 |
| Set 5     | 161.8373 | 5.9944 | 3.4018| 0.0482 | 0.0056 |
| Set 6     | 142.6002 | 3.1483 | 0.002 | 0.3515 | 0.0065 |
| Set 7     | 162.1993 | 5.941  | 3.4066| 0.0456 | 0.0081 |
| Set 8     | 164.1612 | 6.2443 | 3.3904| 0.0477 | 0.0105 |
| Set 9     | 162.8699 | 6.1117 | 3.3942| 0.0468 | 0.0035 |
| Set 10    | 162.2295 | 5.9982 | 3.4065| 0.0473 | 0.0022 |
| Set 11    | 163.7136 | 6.2145 | 3.3885| 0.0472 | 0.0009 |
| Set 12    | 167.1193 | 6.6272 | 3.3702| 0.0495 | 0.0005 |
| Average   | 163.0494 | 6.0925 | 3.1002| 0.0735 | 0.0051 |
Table 6. Fitness Error for the Extended Nelson-Siegel model (Second quarter in 2015)

| Method       | Average | Standard deviation | Average Time (in minutes) |
|--------------|---------|--------------------|--------------------------|
| HGEM         | 0.024759| 0.009674           | 3.0325                   |
| HGPSO        | 0.039497| 0.007324           | 1.1797                   |
| HGPSOPTR-NE  | 0.016541| 0.015271           | 35.3814                  |
| HGPSOPTR-DL  | 0.019875| 0.01700            | 39.1531                  |
| HGPSOPLM     | 0.015778| 0.014238           | 36.332                   |

Figure 5. Interval plot for Extended Nelson-Siegel model

Figure 6. Interval plot for Nelson-Siegel model
price decreases, we see a huge amount of budget deficit in Middle East countries [20]. Therefore, when there is serious concern about downside on economy, most investors prefer to stay in the safe side by holding significant amount of money as either cash or safe investment options such as longer term bond market, etc. This increases longer maturity bond prices, and the yields step down. Investors are also reluctant to invest on short term bonds. This lack of demand drives short-term treasury prices down and the yields up.

5. **Conclusion.** An important financial decision in a bond market is to determine the yield as a function of time. Such a function is called yield curve which could be
monotonic or involve one or more hump. In this paper we have used two famous models of the yield curve, Nelson-Siegel and Extended Nelson-Siegel, with at most one and two humps, respectively. To have an appropriate yield curve, the parameters of the yield curve models need to be optimized such that they have the least square error from market data. Our contribution is the first work in applying parallel hybrid nonlinear and meta-heuristic methods to solve the resulted nonlinear yield curve optimization problem. The proposed methods are hybrid GPSO parallel trust region-dog leg, hybrid GPSO parallel trust region-nearly exact, hybrid

**Figure 9.** Fitted models versus market data set 1
Figure 10. Fitted Extended Nelson-Siegel models resulted from HGPSOPLM versus market data

GPSO parallel Levenberg-Marquardt and hybrid genetic electromagnetism like algorithm. The methods are compared with each other and the results show that for Extended Nelson-Siegel model, HGPSOPLM method and for Nelson-Siegel model, HGPSOPTR-DL are superior to others. Since the accuracy of the yield curve estimation is very important in the bond market, developing new and efficient methods are suggested as a future research to solve the yield curve optimization problem.
Table 7. Comparison between LM results with respect to random, yesterday and PSO results as initial conditions

| Date     | Random parameters as Initial condition | PSO results as Initial condition |
|----------|----------------------------------------|---------------------------------|
| 01/05/2008 | 0.5486                                | -                               |
| 02/05/2008 | 0.0540                                | 0.0368                          |
| 06/05/2008 | 0.7640                                | 0.081                           |
| 07/05/2008 | 0.0534                                | 0.0498                          |
| 08/05/2008 | 0.0536                                | 0.084                           |
| 09/05/2008 | 0.0550                                | 0.0144                          |
| 12/05/2008 | 0.0126                                | 0.0166                          |
| 13/05/2008 | 0.0705                                | 0.0064                          |
| 14/05/2008 | 0.0825                                | 0.037                           |
| 15/05/2008 | 0.1040                                | 0.0346                          |
| 16/05/2008 | 0.2365                                | 0.120                           |
| 19/05/2008 | 0.3837                                | 0.0742                          |
| 20/05/2008 | 0.0914                                | 0.0023                          |
| 21/05/2008 | 0.2751                                | 0.1220                          |
| 22/05/2008 | 0.2099                                | 0.0838                          |
| 23/05/2008 | 0.2504                                | 0.1214                          |
| 27/05/2008 | 0.1863                                | 0.0768                          |
| 28/05/2008 | 0.2189                                | 0.0869                          |
| 29/05/2008 | 0.1662                                | 0.0880                          |
| 30/05/2008 | 0.1637                                | 0.0846                          |

Acknowledgments. We would like to thank the associate editor and the anonymous reviewers for their careful reading and valuable suggestions which improved the quality of this paper.

REFERENCES

[1] I. Baki, Yield curve estimation by spline-based model, A thesis submitted to the Graduate school of Applied Mathematics of The Middle East Technical University, 2006.
[2] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, Nonlinear Programming: Theory and Algorithms, 3rd edition, John Wiley and Sons, 2006.
[3] S. I. Birbil and S. C. Fang, An electromagnetism-like mechanism for global optimization, Journal of Global Optimization, 25 (2003), 263–282.
[4] R. R. Bliss, Testing term structure estimation methods, Working Paper WP 96-12a, Federal Reserve Bank of Atlanta, 1996.
[5] D. J. Bolder and D. Streliski, Yield Curve Modeling at the Bank of Canada, Technical Report TR 84, Bank of Canada, 1999.
[6] M. Clerc and J. Kennedy, The particle swarm explosion stability and convergence in a multi dimensional complex space, IEEE Transactions on Evolutionary Computation, 6 (2001), 58–73.
[7] A. Csajbok, Zero-coupon yield curve estimation from a central bank perspective, Working Paper WP 1998-2, National Bank of Hungary, 1998.
[8] F. X. Diebold and C. Li, Forecasting the term structure of government bond yields, Journal of Econometrics, 130 (2006), 337–364.
[9] R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, Proceedings of the Congress on Evolutionary Computation, San Diego. CA, 2000, 84–88.
[10] A. Hanjoori, M. Amiri and A. Alimi, Forecasting stock price using grey-fuzzy technique and portfolio optimization by invasive weed optimization algorithm, *Decision Science Letters*, 2 (2013), 175–184.

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

[12] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proceeding of IEEE International Conference on Neural Networks*, 1995, 1942–1948.

[13] P. Manousopoulos and M. Michalopoulos, A comparison of yield curve estimation methods: The Greek case, *Journal of Financial Decision Making*, 1 (2005), 33–46.

[14] P. Manousopoulos and M. Michalopoulos, Yield curve construction as a non-linear optimization problem, Proceedings of the 18th Hellenic Conference on Operations Research, 2006.

[15] P. Manousopoulos and M. Michalopoulos, Comparison of non-linear optimization algorithms for yield curve estimation, *European Journal of Operational Research*, 192 (2009), 594–602.

[16] J. H. McCulloch, Measuring the term structure of interest rates, *Journal of Business*, 44 (1971), 19–31.

[17] J. H. McCulloch, The tax-adjusted yield curve, *The Journal of Finance*, 30 (1975), 811–830.

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

[19] J. Nocedal and S. J. Write, *Numerical Optimization*, Springer Science and Business Media, 2006.

[20] M. Orouji, Oil price shocks and stock market returns, *Accounting*, 2 (2016), 103–108.

[21] M. Pooter, Examining the Nelson-Siegel Class of Term Structure Models, Tinbergen Institute, Discussion paper, 2007.

[22] P. Soderlind and L. E. O. Svensson, New Techniques to Extract Market Expectations from Financial Instruments, Working Paper WP 5877, National Bureau of Economic Research, 1996.

[23] L. E. O. Svensson, Estimating and Interpreting Forward Interest Rates: Sweden 1992–1994, Working Paper WP 4871, National Bureau of EconomicResearch, 1994.

[24] L. E. O. Svensson, Estimating forward interest rates with the Extended Nelson-Siegel method, *Sveriges Riksbank Quarterly Review*, 3 (1995), 13–26.

Received August 2015; 1st revision June 2016; 2nd revision October 2016.

E-mail address: roya.soltani@gmail.com and royasoltani@srbiau.ac.ir
E-mail address: sjsadjadi@iust.ac.ir
E-mail address: mona.rahnama85@gmail.com