Zero–coupon yields estimated by zero–degree splines

Mojmír Simerský*

April 2018

Abstract

The paper addresses the problem of zero–coupon yield curve (YC) estimation from a portfolio of coupon–bearing instruments, primarily coupon bonds. A fast and stable iterative procedure is proposed and implemented. The optimization problem is formulated in a matrix form, the principal cashflow matrix having the dimension given by the number of instruments (bonds) and the number of knots on the time axis. The number of instruments is arbitrary, as well as the number of time knots.

In our concept of "equivalent cashflows", each future cashflow at a time t is replaced by two cashflows, one at the left, the other at the right knot respective to the time t. We solve then a simplified problem of estimating the YC from a portfolio of instruments whose future cashflows occur only at predefined times. The method allows for further additional constraints, e.g., an ultimate forward rate fixing or predefined discount factor at some time. We touch also on the asymptotic case of a very dense partitioning of the time axis.

The optimization is carried out in the space of forward rate functions of the simplest form – zero–degree splines, i.e., piecewise constant functions. Our approach is thus a generalization of the bootstrap method with no requirements on the bond maturity ladder and, at the same time, with optional smoothing.

This work relates to our previous article [4], where the idea of equivalent cashflows was introduced. Czech bond yields estimated by the proposed method can be found in [5].

Keywords: yield curve estimation, nonparametric regression, penalized splines, bootstrap, Czech bond market
JEL: E43

1 Introduction

In our view, there does not exist a substantiated theory answering the question of the zero–coupon yield curve functional form. Nonparametric approaches are therefore free to choose from a class of functions and observe two criteria: estimation errors and YC "smoothness". The errors, seen either in terms of bond prices, i.e., as differences between the market gross

*Contact: simersky@gmail.com, Location: Prague, Czech Republic

price and the price implied by the YC, or, in yield terms, as parallel shifts of the YC such that the sum of discounted flows of a bond equals its gross price, are easy to define. The notion of YC smoothness (or roughness) is far from unambiguous and the metrics differ case by case.

The sum of squared price/yield errors and some proxy of the spline roughness is thus to be minimized. When modelling the forward rate, we opted for the simplest of splines – a step function. This coarse approximation is mitigated by a substantial refinement of the time partition – in the maturity interval up to, say, 30 years, tens or even hundreds of time knots are feasible. The problem of YC "waviness", encountered with cubic splines and even with linear ones [4], is alleviated. Governing matrices are of a very simple form. The minimization is solved iteratively by the Newton–Raphson (N–R) method with an explicit Hessian. Thanks to the Hessian simple structure, the N–R runs fast and stable.

The algorithm has been used for estimating the Czech Government YCs, particularly in the period 2014–2017, when the Czech National Bank was weakening the CZK exchange rate by long–term currency interventions. In this paper, only technical issues will be treated; an account of Czech bond yields can be found in the accompanying paper [5].

2 Penalty function, equivalent cashflows

Let's consider a set of M debt instruments with known future cash flows occurring only at discrete times $t_1 < t_2 < \ldots < t_{N-1}$, where $t_1 \geq 0$.

The yield curve generates a set of discount factors at the knots. We denote by $f(t)$ the instantaneous forward rate; in continuous compounding, the discount factor d at a time t is

$$d(t) = \exp\left(- \int_0^t f(\tau)\,\mathrm{d}\tau \right) \tag{1}$$

and the spot curve

$$s(t) = \frac{\int_0^t f(\tau)\,\mathrm{d}\tau}{t}\,, \qquad \text{for } t > 0 \quad \text{and} \quad s(0) = f(0)\,.$$

The discount factors at the knots form a $(N \times 1)$ vector, denoted by $\mathbf{d}$. (Hereafter, vectors and matrices will be set in bold.) We arrange the nominal cashflows into a $(M \times N)$ matrix $\mathbf{T}$. The matrix $\mathbf{F}$ of discounted cashflows is a transformation of $\mathbf{T}$, where each column has been multiplied by the corresponding discount factor

$$\mathbf{F} = \mathbf{T}\,\mathrm{diag}(\mathbf{d})\,.$$

All cashflows are supposed to be positive, which is the standard case, i.e., all elements of $\mathbf{T}$ and $\mathbf{F}$ are nonnegative.

The price errors $\boldsymbol{\epsilon}$ of the instruments with respect to the YC are simply $\boldsymbol{\epsilon} = \mathbf{F}\mathbf{1} - \mathbf{p}$, where $\mathbf{1}$ stands for the unit vector and $\mathbf{p}$ is the vector of gross market prices.

The piecewise constant forward rate at the knots $\mathbf{t}$ can be expressed in terms of its initial value $f_0 = f(0)$ and the "jumps" $j_i = f_{i+1} - f_i$,

$$f_k = f_0 + \sum_{i=1}^{k} j_i \, .$$

The integration of f in the exponent of (1) leads to

$$\int_0^{t_i} f(t)\,\mathrm{d}t = f_0 t_i + (\mathbf{A}\,\mathbf{j})_i \, ,$$

where the elements A_{ij} of the $(N \times N)$ matrix $\mathbf{A}$ follow a simple relation $A_{ij} = \max(t_i - t_{j-1}, 0)$, $t_0 = 0$. The discount vector $\mathbf{d}$ as a function of $\mathbf{j}$ and f_0 is now

$$\mathbf{d} = \exp(-(\mathbf{A}\,\mathbf{j} + f_0\,\mathbf{t}))\, ,$$

this shorthand notation should be understood component-wise.

The penalty function L (loss function) to be minimized reads $2\,L(\mathbf{j}) = \mathrm{e}^{\lambda}\,\mathbf{j}^\top \mathbf{j} + \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}$, where a smoothing parameter λ has been introduced.

Now we would try to make L less sensitive to the number of knots and instruments, so that the level of smoothing is governed primarily by λ. The number of instruments M may be accounted for via the mean squared error (MSE), i.e., as the denominator of $\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}$. Further, we observe that the smoothness term $\mathbf{j}^\top \mathbf{j}$ depends on the number of time knots. The sum of jumps should be, for a given portfolio of instruments, approximately constant and independent of λ. If we double the number of knots, the number of terms is double, and each term is approximately a quarter of the one divided. This suggests that the term $\mathbf{j}^\top \mathbf{j}$ should be normalized by multiplying it by N.

The normalized penalty function can be now recast as

$$2\,L(\mathbf{j}) = \mathbf{j}^\top \mathbf{j} + \varphi\,\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}\, , \quad \text{with} \quad \varphi = \frac{\mathrm{e}^{-\lambda}}{MN}\, . \tag{2}$$

The gradient $\mathbf{g}$ of L with respect to $\mathbf{j}$ is a row vector of length N

$$\mathbf{g} = \frac{\partial L}{\partial \mathbf{j}} = \mathbf{j}^\top + \varphi\,\boldsymbol{\epsilon}^\top \left(\frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{j}}\right) .$$

The derivative at the right–hand side is a $(M \times N)$ matrix and equals $-\mathbf{F}\,\mathbf{A}$ (see [4]), so that the gradient is finally

$$\mathbf{g}^\top = \mathbf{j} - \varphi\,\mathbf{A}^\top \mathbf{F}^\top \boldsymbol{\epsilon}\, . \tag{3}$$

The next task is to arrange real cashflows, which occur at arbitrary times, into the matrix $\mathbf{F}$ of discounted cashflows allowed only at the nominal times $\mathbf{t}$.

Let there be a nominal cashflow T_t at a time t, where $t_k < t < t_{k+1}$. The discount factor at t, given by the estimated yield curve, is d_t. We would like to split $F_t = T_t d_t$ into two flows F_k and F_{k+1} in a way that the first two moments, the sum and the duration, remain unchanged. The resulting pair of cashflows is, in this sense, **equivalent** to the original cashflow

$$F_k + F_{k+1} = F_t, \qquad F_k\,t_k + F_{k+1}\,t_{k+1} = F_t\,t\, .$$

The discount factor d_t is easily found from t, d_k, and f_k. Clearly, the equivalent cashflows conserve pricing errors and durations of all instruments.

However, we have simplified the real problem. Even if $\boldsymbol{\epsilon}$ remains unchanged, the vector $\mathbf{F}^\top\boldsymbol{\epsilon}$ does not, so the gradient would be different. To treat this complication analytically could be perhaps possible, but it would be impractical for numerical computation. Intuitively, the splitting may be detrimental to the convergence and/or stability of the iteration process. On the other hand, the denser the time partition, the lesser the influence of this approximation on the gradient value. Also, if the estimation errors are small, $\mathbf{F}^\top\boldsymbol{\epsilon}$ will end up close to zero irrespective of the approximation due to the cashflow splitting.

We must make a similar compromise when transforming price errors to yield errors. Estimating yield errors rather than price errors is preferable [1] and instrumental in freeing λ from the dependence on the price and flows units. The transformation matrix $\mathbf{D}$ $(M \times M)$ reads $\mathbf{D} = [\mathrm{diag}(\mathbf{F}\,\mathbf{t})]^{-1}$. Indeed, in the linear approximation, a parallel shift of YC is equal to the derivative by f_0, which can be found as

$$\frac{\partial(\mathbf{F}\,\mathbf{1})}{\partial f_0} = -\mathbf{F}\,\mathbf{t}\,.$$

As the nominator $\mathbf{F}\,\mathbf{1}$ represents the vector of gross prices, the transformation of price errors to yield errors is indeed given by $\mathbf{D}$.

For the sake of not complicating the notation, we will not introduce different symbols for $\mathbf{p}$, $\mathbf{F}$ and $\boldsymbol{\epsilon}$ after this transformation. Henceforth, it is to be understood that the elements in the i–th row have already been divided by the $\sum_{i=1}^{N} T_{ij}\, d_j^p t_j$, where $\mathbf{d}^p$ is the discount vector computed in the preceding iteration, i.e., it does not enter into the derivative operations as a variable. The convergence of $\mathbf{d}$ during the iteration procedure would justify this shortcut.

3 The N–R method

3.1 Iteration formula

The Hessian $\mathbf{H}$ for the N–R method can be obtained, after some algebra, as

$$\mathbf{H} = \frac{\partial \mathbf{g}^\top}{\partial \mathbf{j}} = \mathbf{I} + \varphi\,\mathbf{A}^\top\mathbf{F}^\top\mathbf{F}\,\mathbf{A} + \varphi\,\mathbf{A}^\top \mathrm{diag}(\mathbf{F}^\top\boldsymbol{\epsilon})\,\mathbf{A}\,. \tag{4}$$

In the last term, the vector $\mathbf{F}^\top\boldsymbol{\epsilon}$ should be small for good YC estimates and we omit it in the N–R procedure. (Empirically, it was found that this omission even improves the convergence of the numerical iterations at the start, when this term is not yet negligible.)

The N–R iteration is, in principle, straightforward

$$\mathbf{j} \leftarrow \mathbf{j} + \mathbf{H}^{-1}\mathbf{g}^\top\,.$$

For numerical processing, this update can be set in a more convenient form, so that the repeated matrix multiplication by $\mathbf{A}$ and $\mathbf{A}^\top$ in the Hessian is eliminated.

Let's denote $\mathbf{B} = (\mathbf{A}^\top\mathbf{A})^{-1}$ and introduce a new variable to be iterated $\mathbf{y} = \mathbf{A}\,\mathbf{j}$. Then the N–R update of $\mathbf{y}$ can be rearranged as

$$\mathbf{y} \leftarrow \mathbf{y} + \left(\mathbf{B} + \varphi\,\mathbf{F}^\top\mathbf{F}\right)^{-1} \left(\mathbf{B}\,\mathbf{y} - \varphi\,\mathbf{F}^\top\boldsymbol{\epsilon}\right) . \tag{5}$$

The only problematic part of the update is the inversion. We maintain that the matrix $\mathbf{B} + \varphi\,\mathbf{F}^\top\mathbf{F}$ is always positive definite and therefore invertible.

$\mathbf{A}^\top\mathbf{A}$ is, by definition, symmetric and positive definite, so that its inverse is positive definite, too. We note that $\mathbf{A}^{-1}$ is lower triangular, with only the main diagonal and two sub-diagonals containing nonzero elements. It follows that $\mathbf{B}$ is a five-diagonal symmetric matrix.

If the number of instruments is lower than the number of time knots, which could be quite often, the matrix $\mathbf{F}^\top\mathbf{F}$ is singular. Even if this is not the case and the number of instruments is greater, some rows of $\mathbf{F}^\top\mathbf{F}$ may be zero, because the equivalent cashflows may not map into every knot of $\mathbf{t}$. Nevertheless, all elements of F are nonnegative; therefore, $\mathbf{F}^\top\mathbf{F}$ is positive semidefinite and $\mathbf{B} + \varphi\,\mathbf{F}^\top\mathbf{F}$ is positive definite.

A suitable method of numerical inversion is therefore the $LDL^\top$ decomposition, with the subsequent solving two sets of triangular linear systems. As a stopping rule, some measure of the difference between the consecutive iterations of $\mathbf{y}$ could be the simplest choice.

3.2 Computing issues

The input data to the estimation procedure is

- future nominal cashflows, with the cashflow times in years according to the adopted day count convention
- spot gross prices, i.e., clean prices with accrued interest by the same day count convention
- short–term treasury or money market rate (e.g., overnight)
- value of the smoothing parameter λ.

The most important program parameters are

- maximum cashflow time (30 years)
- number of time knots (40)
- maximum number of iterations (100)
- value of $\max(|\Delta y_i|)$ set as a stopping rule (10^{-5}).

Due to their number, the placement of the time knots would be impractical if user-defined. A functional form imbedded in the program is used instead. Because the linear time division is clearly disadvantageous, we opted for the quadratic function, as the next one in simplicity. The i-th time knot is $t_i = a + b\,i^2$, where a and b are the constants chosen so that $t_1 = \frac{1}{12}$ (the shortest time knot, one month), $t_N = 30$ (the last knot, 30 years).

The program is written in Excel VBA, but does not rely on Application firmware, so that it can be rewritten in any procedural programming language. It is simple and straightforward; when computing the new update of $\mathbf{y}$, small "tricks of the trade" are implemented, namely an adaptive Newton step and a matrix inversion strategy that skips the $LDL^\top$ decomposition if the rate of convergence is deemed satisfactory. As a result, the number of inversions may be smaller than the number of iterations.

We will discuss the Czech YCs elsewhere [5]; as for the performance, we mention only some averages over a set of about 700 Czech Government bond YCs in the years 2014 – 2017: smoothing parameter $\lambda = 0$, computing time 30 ms, number of iterations 13, number of inversions 2, number of instruments 18, maximum bond maturity 20 years. The computing time is understood to be the runtime of the N–R minimization subroutine alone and refers to the processing on a standard notebook (Lenovo IdeaPad Yoga 14 3).

4 Boundary conditions

At the short end, a value $f(0) = s(0)$ has to be supplied as an input parameter. Its influence on the estimation is rather limited. The time division at the start is dense and a possible misfit between the money market and the bond market short–term levels vanishes very quickly.

At the long end, the estimated forward curve remains constant for the times exceeding the last cashflow (or its split part). Because the spot and forward curves converge to each other for large times, the ultimate value of the spot curve, i.e., its value at infinity, equals the value of the forward curve at the last cashflow.

The presented method, however, permits a simple modification of these boundary conditions. The idea is to augment the penalty function with an additional term to impose one of the following constraints:

- $s(t_N) = f(t_N)$, these two rates intersect at the last knot time rather than at infinity; the final time may be specified out of the range of estimation, e.g., at 100 years,
- $f(t_N) = UFR$, the ultimate forward rate can be incorporated as a constraint,
- $s(t_N) = s$, s is a requested value reflecting, e.g., a desired discount factor at t_N.

We will elaborate only on the first case; the remaining two can be treated similarly.

The condition $s(t_N) = f(t_N)$ is equivalent to

$$\frac{(\mathbf{A}\,\mathbf{j})_N}{t_N} = \mathbf{1}^\top \mathbf{j}\,, \text{ or } \mathbf{q}^\top \mathbf{j} = 0\,, \text{ where } \quad \mathbf{q} = \mathbf{A}^\top \mathbf{e}_N - t_N \mathbf{1}\,,$$

$\mathbf{e}_N$ is the unit basis vector. If we add a term $k\,\mathbf{j}^\top \mathbf{q}\,\mathbf{q}^\top \mathbf{j}$ with a large constant $k > 0$ to the penalty function (2), the optimization procedure will see to it that $\mathbf{q}^\top \mathbf{j}$ converges to a sufficiently small value.

Consequently, the gradient (3) will be augmented with $k\,\mathbf{q}\,\mathbf{q}^\top \mathbf{j}$ and the Hessian (4) with $k\,\mathbf{q}\,\mathbf{q}^\top$. When applying the transformation from $\mathbf{j}$ to $\mathbf{y}$, the matrix $\mathbf{B} = (\mathbf{A}^\top \mathbf{A})^{-1}$ in (5) becomes $\left(\mathbf{A}^\top\mathbf{A}\right)^{-1} + k\left(\mathbf{A}^\top\right)^{\mathbf{-1}}\mathbf{q}\,\mathbf{q}^\top\mathbf{A}^{\mathbf{-1}}$. This modification of $\mathbf{B}$ is the sole impact the $s(t_N) = f(t_N)$ constraint has on the optimization procedure.

The matrix $\left(\mathbf{A}^\top\right)^{\mathbf{-1}}\mathbf{q}\,\mathbf{q}^\top\mathbf{A}^{\mathbf{-1}}$ has a very simple form. We observe that $\mathbf{A}^{-1}$ can be decomposed as $\mathbf{A}^{\mathbf{-1}} = \mathbf{J}\,\mathbf{R}\,\mathbf{J}$, where $\mathbf{J}$ is lower triangular. The only nonzero elements are the values of 1 on the main diagonal and -1 on the first subdiagonal. $\mathbf{R}$ is diagonal and it holds $R_{ii} = (t_i - t_{i-1})^{-1}$, $t_0 = 0$. Applying this decomposition, we arrive at

$$\left(\mathbf{A}^\top\right)^{\mathbf{-1}}\mathbf{q}\,\mathbf{q}^\top\mathbf{A}^{\mathbf{-1}} = \frac{(t_N\mathbf{e}_{N-1} - t_{N-1}\mathbf{e}_N)(t_N\mathbf{e}_{N-1} - t_{N-1}\mathbf{e}_N)^T}{(t_N - t_{N-1})^2}\,,$$

so that only four elements at the lower right corner of $\mathbf{B}$ are to be modified. In a similar vein, the condition $f(t_N) = UFR$ requests that $(\mathbf{1}^\top\mathbf{j} - UFR) \to 0$. It can be shown that again only four elements of $\mathbf{B}$ are affected. But, unlike the first case, the last two elements of the gradient term $\mathbf{B}\,\mathbf{y}$ in (5) must also be modified. For $s(t_N) = s$, $\mathbf{B}$ needs to be changed at the lower right corner only, together with the last element of the gradient term.

5 Continuous approximation

Even though our approach can be rightly classified as nonparametric, its "parametric" features are worth exploring. Contrary to the parametric models, where the number of instruments cannot be less than the number of free parameters, here we can estimate a YC even with a single instrument with a single cashflow. An interesting question arises what the functional form of such YC is and how it depends on the estimation set–up.

The condition $\mathbf{g}^\top = \mathbf{0}$ in (3) can be rewritten as

$$\left(\mathbf{A}^\top\right)^{-1}\mathbf{j} = \varphi\,\mathbf{F}^\top\boldsymbol{\epsilon} \tag{6}$$

With some insight into the structure of $\mathbf{A}^{-1}$ we arrive at the explicit formula

$$\left(\mathbf{A}^\top\right)^{-1}_{mn} = \delta_{mn}\,r_m - \delta_{m,n+1}(r_m + r_{m-1}) + \delta_{m,n+2}\,r_{m-1}, \tag{7}$$

where δ_{mn} is the Kronecker delta and $r_m = 1/(t_m - t_{m-1}), r_0 = 0$. The right-hand side of (7) can be recognized as a formula for numerical differentiation of $\mathbf{j}$ with respect to the time index m. Indeed, denoting $x_m = ((\mathbf{A}^\top)^{-1}\mathbf{j})_m$

$$x_{m-1} = r_{m-1}\,j_{m-1} - (r_m + r_{m-1})j_m + r_m\,j_{m+1}\,. \tag{8}$$

For the equidistant division of the time axis, i.e., for $r =$const, (8) is equivalent to a numerical approximation of the second derivative. An analysis in the limit of infinite number of knots N offers itself. We denote $\Delta = t_N/N$ and expand

$$r_{m+1} = r + r'\Delta + r''\frac{\Delta^2}{2}, \quad j_{m+1} = j + j'\Delta + j''\frac{\Delta^2}{2}$$

and similarly, for $m-1$. Substituting into (8), we find that there remain only terms with Δ^2 and that the continuous functions $x(\mu), j(\mu), r(\mu)$, $\mu \in \langle 0, t_N\rangle$ obey a simple relation

$x = (rj')'$. Further, let's suppose that the knot times follow a functional form $t(\mu)$. Then, by a similar reasoning, we can write $r(\mu) = 1/t'(\mu)$ and the left-hand side of (6) is finally $x = (j'/t')'$.

In the continuous approximation, the right-hand side of (6) consists of a series of Dirac's delta functions, each multiplied by a constant. We conclude that the integral of the right-hand side is a piecewise constant function with the discontinuities (jumps) at the cashflow times. This integral should therefore be equal to $x = (j'/t')'$, i.e., between the cashflows it holds $j' = k_1 t'$, or $j = k_0 + k_1 t$, where k_0, k_1 are constants. It is worth noting that $j(t)$ is a linear spline with knots at the cashflow times regardless of the time division $t(\mu)$ of the time axis.

To test this somehow careless reasoning, we run the YC estimation from a portfolio of only three zero–coupon bonds, but with a division of 300 knots in the time interval $\langle 0, 30\,\mathrm{y}\rangle$. In this setup, the Hessian was a five-diagonal matrix with dimensions 300×300. The cashflows were situated at 5, 15, and 25 years and priced at 92, 60, and 52 percent of the nominal.

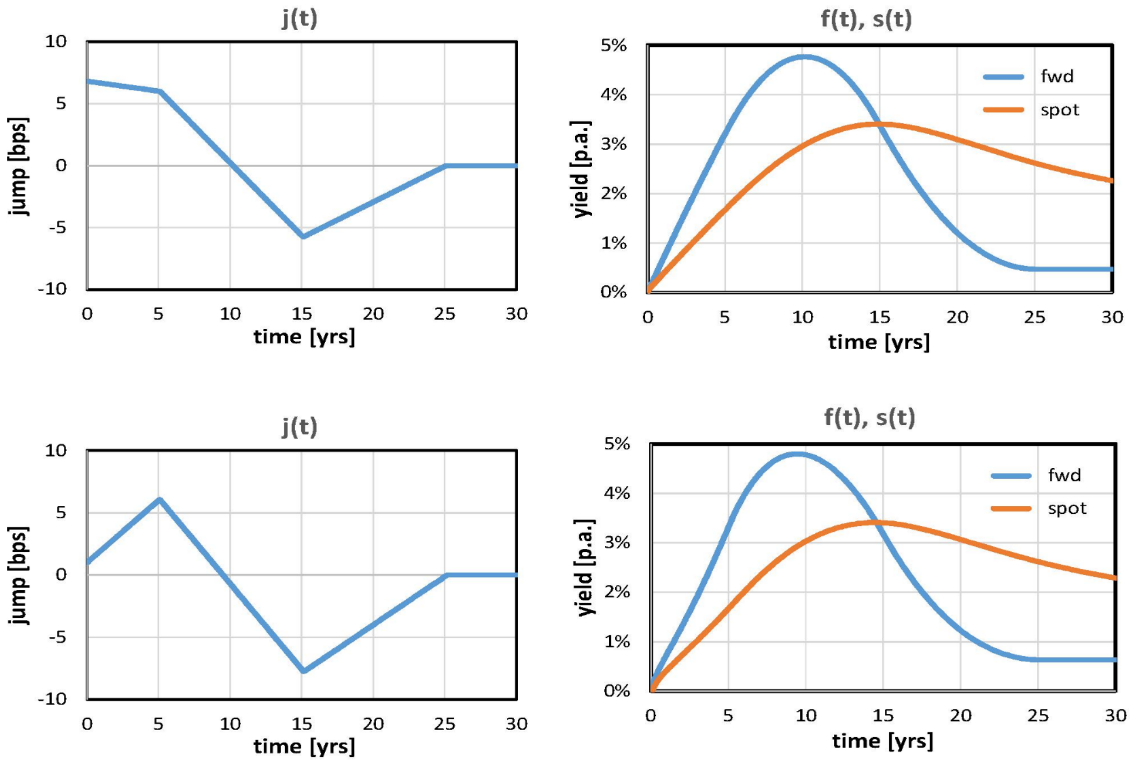

Fig.1. Spot curves $s(t)$, forward rates $f(t)$, and forward rate jumps $j(t)$ estimated in the limit of very dense partition of the time axis. Upper part: linear time division, lower part: quadratic time division. The portfolio consists of 3 zero-coupon bonds with maturities 5, 15, and 25 years, priced at 92, 60, and 52, respectively.

The computed forward rates $f(t)$ and the jumps $j(t)$ calculated thereof are plotted in Fig.1 (upper part: linear time division, lower part: quadratic time division). We note that $j(t)$

is a linear spline in both cases. Even if $j(t)$ are quite different, the spot curves are close (the differences - not shown - lie in the range of units of basis points).

In the general case of many cashflows, we deduce that for the linear time division, $t = \mu$, the forward rate as a function of t , $f(t) = \int_0^t j(\mu)\, \mathrm{d}\mu$, is a quadratic spline. The quadratic time division $t = \mu^2/t_N$ leads to a more complicated forward curve: a cubic spline in a new variable $\sqrt{t_N t}$.

6 Conclusion

We have proposed and implemented a simple, fast and stable nonparametric procedure for the estimation of zero–coupon yield curves from coupon instruments. This is our second attempt to procure us with a tool that does not suffer from shortcomings (in our view) like variable number of time knots as a function of the portfolio and/or goodness of fit, *ad hoc* roughness penalty, a portfolio–dependent degree of smoothing, etc. [6], [7]. The recent overview [2] seems to support the parametric approach, which is not justifiable by any coherent theory, just because it does not suffer from the above–mentioned shortcomings. We assume that the problem encountered in various smoothing spline approaches – the "waviness" of resulting YCs – is due to the degree of splines in use, often the cubic splines. In our first approach described in [4], we experimented with a linear spline as a model of the forward rate. The outcome was a viable computer program, but some stabilization measures had to be devised even for rather small Hessians (20×20). The current approach with the zero–degree splines marks an undisputed improvement. The coarseness of approximating functions is compensated for by a substantial increase in the number of permanently fixed time knots.

The choice of the smoothing parameter λ is, loosely speaking, market–dependent. The level of quoted pricing spreads and/or user preferences on the smoothness of the YC should determine its value. In [4] we spoke in favor of monitoring a degree of "fairness" rather than smoothness of the YC. Fairness targeting would not only complicate the processing by adding another iteration loop, but, most importantly, would damage the stability of the YC estimation. After extensive numerical experiments, we came to a conclusion that the parameter that comes closest to satisfy a subjective notion of the YC fairness is the number of inflexion points on the spot curve. This parameter, being discrete, may exhibit extreme sensitivity to bond prices and estimation setting. An application of the N–R method to non–smooth optimization could complicate the matter further. After pondering pros and cons, we prefer the simpler approach presented here over that we brought in [4], where our fairness metrics was proposed. We touch upon the question of smoothing in [5], when analyzing data sets from the price fixing [3] of the Czech bond market.

The Czech Government bond YCs are available on our site www.zyc.cz. There we also offer an Excel sample file, which can be copied and supplied with user data. The library module lib.xlam is functionally equivalent to that we use in our own processing and its basic parameters are those described in the Computing issues section.

References

[1] BIS Papers, *Zero–coupon yield curves: technical documentation*, Monetary and Economic Department, No 25. (2005)

[2] ECB Statistics Papers, No.27: *Yield curve modelling*, Per Nymand-Andersen (2018)

[3] https://www.mtsdata.com/content/data/public/cze/fixing/index.php

[4] Simersky, M., *A nonparametric zero-coupon yield curve* (2016), www.zyc.cz

[5] Simersky, M., *Czech Government Bond yields under FX pressure* (to appear)

[6] Tanggaard, C., *Nonparametric Smoothing of Yield Curves*, Review of Quantitative Finance and Accounting, No. 9 (1997), pp. 251–267.

[7] Waggoner D.F., *Spline Methods for Extracting Interest Rate Curves from Coupon Bond Prices*, "Working Paper", No. 97-10, Federal Reserve Bank of Atlanta, Atlanta, (1997)