Journal of Statistical Theory and Practice (2026) 20:30
https://doi.org/10.1007/s42519-026-00545-8

ORIGINAL ARTICLE

Spline Quantile Regression

Ta-Hsin Li[1] · **Nimrod Megiddo**[2]

Received: 30 June 2025 / Accepted: 9 January 2026

Abstract
Quantile regression is a powerful tool capable of offering a richer view of the data as compared to least-squares regression. Quantile regression is typically performed individually on a few quantiles or a grid of quantiles without considering the similarity of the underlying regression coefficients at nearby quantiles. When needed, an ad hoc post-processing procedure such as kernel smoothing is employed to smooth the individually estimated coefficients across quantiles and thereby improve the performance of these estimates. This paper introduces a new method, called spline quantile regression (SQR), that unifies quantile regression with quantile smoothing and jointly estimates the regression coefficients across quantiles as spline functions. We discuss the computation of the SQR solution as a linear program (LP) using an interior-point algorithm. We also experiment with some gradient algorithms that require less memory than the LP algorithm. The performance of the SQR method and these algorithms is evaluated using simulated and real-world data.

Keywords function estimation · gradient descent · interior point · linear program · quantile periodogram · quantile regression · smoothing · spline

1 Introduction

Quantile regression (QR) is a powerful statistical tool that complements the conventional least-squares regression [14, 15]. Quantile regression postulates the conditional quantile, rather than the conditional mean, of a dependent variable as a function of the explanatory variables. One can explore this relationship by varying the quantile level to get a richer view of the data than that offered by least-squares regression. Recent years have witnessed further development of the quantile regression method. Examples include, just to name a few, the fast algorithms to compute quantile regression

✉ Ta-Hsin Li
thl024@outlook.com

Nimrod Megiddo
megiddo@us.ibm.com

[1] Formerly affiliated with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

[2] IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Published online: 11 February 2026

when the number of regressors is very large [11], the statistical analyses of quantile processes [4, 9], and the techniques to overcome quantile crossings for applications that require the quantiles to be monotone at any point in the domain of the regressor [1, 6, 10, 31, 33].

In this article, we are interested in situations where the underlying quantile regression coefficients vary smoothly across quantiles. In typical applications, the smoothness is either ignored entirely by performing quantile regression independently at different quantiles, or handled separately by employing an ad hoc post-processing procedure such as kernel smoothing to smooth the raw estimates across quantiles [14, pp. 158–159].

We offer an alternative solution, called spline quantile regression (SQR). The SQR method extends the original QR problem at an individual quantile into a function estimation problem across all quantiles. This functional view of quantile regression has two advantages over the conventional view of parameter estimation at a fixed quantile: First, it provides a global perspective of the effect of explanatory variables on all quantiles. Secondly, it has the capacity of offering more accurate assessment of the effect at a given quantile by leveraging the information from neighboring quantiles. In the SQR problem, the regression coefficients are treated as smooth functions of the quantile level in a space of splines, and the resulting model is fitted to the data jointly on a grid of quantiles. A penalty term is added to the QR cost function to regularize the smoothness of the functional regression coefficients. The SQR method unifies quantile regression with quantile smoothing and provides an estimator of regression coefficients as spline functions. In this article, we focus on the computation of the SQR solution. We evaluate and demonstrate the SQR solution with simulated and real-data examples.

Splines have been used in the context of quantile regression to represent nonparametric regression functions [2, 11, 17, 23]. In these methods, the smoothness of the regression function is regularized in a way similar to the spline smoothing problem under the least-squares framework [32]. The SQR problem, on the other hand, represents the regression coefficients as spline functions of the quantile level. The resulting model is a linear function of the regressors for fixed quantile level, and a nonlinear smooth function of the quantile level for fixed regressors. Schnabel and Eilers [30] consider a situation of univariate nonparametric quantile regression where the conditional quantile is a bivariate function of the covariate and the quantile level represented by spline basis, and the penalty is imposed on the differences of the spline coefficients rather than the derivatives of the regression function. The resulting estimates are computed by iteratively reweighted least-squares (IRLS). Das and Ghosal [7] deal with a similar situation by taking a Bayesian approach using an MCMC algorithm. Park and He [24] employ a model similar to ours for function estimation with no smoothness regularization, resulting in a composite quantile regression problem rather than a penalized quantile regression problem. Yoshida [34] extends this work by introducing a group lasso-type regularization on the spline coefficients to serve the purpose of variable selection, which differs from a smoothness regularization on the functional regression coefficients. The resulting estimates are computed using IRLS.

To retain the numerical characteristics of quantile regression, we employ the integral of the ℓ_1-norm of the second derivatives as the penalty to regularize the smoothness

of the functional regression coefficients. A similar measure was employed in Koenker et al. [17] for nonparametric quantile regression. We show that the resulting SQR problem can be reformulated as a primal-dual pair of linear program (LP) and solved using the interior-point algorithm developed originally for the ordinary QR problem by Portnoy and Koenker [26].

In this article, we also experiment with some gradient algorithms as computationally more efficient alternatives to approximate the LP solution. This investigation is largely motivated by the success of gradient algorithms in training neural network models with non-smooth objective functions [8, 29]. The success stories have generated renewed interest in trying to better understand the behavior of gradient algorithms in non-smooth situations (e.g., [3, 18]). As in machine learning applications, our aim is not to use these algorithms as a replacement of LP to produce the exact solution. Instead, we are interested in algorithms that provide sufficiently good approximations to the LP solution but with reduced computational burden, especially the demand on computer memory.

The remainder of this article is organized as follows. Section 2 describes the SQR problem. Section 3 discusses the LP reformulation and proves the suitability of the interior-point algorithm for solving the SQR problem. Section 4 presents numerical examples with simulated and real-word data to demonstrate the SQR method. Section 5 describes the gradient-based BFGS, ADAM, and GRAD algorithms. Section 6 contains the experimental results of the gradient algorithms. Concluding remarks are given in Section 7. The R functions that implement the SQR method in the package 'qfa' are described in Appendix.

2 Spline Quantile Regression

Let $\{y_t : t = 1, \ldots, n\}$ be a set of n observations of a dependent variable and $\{\mathbf{x}_t : t = 1, \ldots, n\}$ be the corresponding values of a p-dimensional regressor. Under the quantile regression framework [14], it is assumed that the conditional quantile of y_t at a quantile level $\tau \in (0, 1)$ given $\mathbf{x}_t$ can be expressed as

$$Q_{y_t}(\tau \mid \mathbf{x}_t) = \mathbf{x}_t^T \boldsymbol{\beta}(\tau). \tag{1}$$

Given an increasing sequence of quantile levels $\{\tau_\ell : \ell = 1, \ldots, L\} \subset (0, 1)$, the standard QR method produces an estimate for each $\boldsymbol{\beta}_\ell := \boldsymbol{\beta}(\tau_\ell)$ independently by solving

$$\hat{\boldsymbol{\beta}}_\ell := \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{t=1}^{n} \rho_{\tau_\ell}(y_t - \mathbf{x}_t^T \boldsymbol{\beta}) \quad (\ell = 1, \ldots, L), \tag{2}$$

where $\rho_\tau(y) := y(\tau - I(y < 0))$ denotes the objective function of quantile regression at quantile level τ, with $I(\cdot)$ being the indicator function. In typical situations, the quantile levels in $\{\tau_\ell\}$ are specified by the user based on the need of the application. In some occasions, one may want to consider the quantiles that correspond to all distinct

quantile regression solutions [14, 25] similar to the quantiles associated with the order statistics.

In this paper, we are interested in estimating $\boldsymbol{\beta}(\tau)$ in (2) as a function of τ in a closed subinterval $[a, b] \subset (0, 1)$. We propose to obtain an estimate of this function by solving the spline quantile regression (SQR) problem

$$\hat{\boldsymbol{\beta}}(\cdot) := \underset{\boldsymbol{\beta}(\cdot)\in\mathcal{F}^p}{\operatorname{argmin}} \left\{ n^{-1} \sum_{\ell=1}^{L} \sum_{t=1}^{n} \rho_{\tau_\ell}(y_t - \mathbf{x}_t^T \boldsymbol{\beta}(\tau_\ell)) + c \sum_{\ell=1}^{L} w_\ell \, \|\ddot{\boldsymbol{\beta}}(\tau_\ell)\|_1 \right\}, \tag{3}$$

where $\mathcal{F}$ is the space spanned by cubic spline basis functions on $[a, b]$, $c \geq 0$ is a smoothing or penalty parameter, $\ddot{\boldsymbol{\beta}}(\tau)$ denotes the second derivative of $\boldsymbol{\beta}(\tau)$, and $\{w_\ell\}$ is a user-specified sequence of nonnegative constants that allow different contributions from different quantiles to the penalty term.

We employ the ℓ_1-norm of the second derivatives in (3) as the roughness measure of the functional coefficient $\boldsymbol{\beta}(\cdot)$ in order to retain the LP characteristics of the original QR problem [14]. This is the same reason offered in Koenker et al. [17] to justify the ℓ_1-norm penalty for the spline-based nonparametric regressor in a nonparametric quantile regression problem. An alternative to the ℓ_1-norm is the squared ℓ_2-norm, which is commonly used for spline smoothing under the least-squares framework [32]. This option would result in a quadratic program (QP) instead of an LP. We choose to deal with this problem elsewhere.

Let $\{\phi_k(\tau) : k = 1, \ldots, K\}$ denote a set of spline basis functions of $\tau \in [a, b]$ (e.g., B-splines with $K = L + 2$). Then, any function $\beta_j(\tau)$ in $\mathcal{F}$ can be expressed as

$$\beta_j(\tau) = \sum_{k=1}^{K} \phi_k(\tau)\theta_{jk} = \boldsymbol{\phi}^T(\tau)\boldsymbol{\theta}_j \quad (j = 1, \ldots, p),$$

where

$$\boldsymbol{\phi}(\tau) := [\phi_1(\tau), \ldots, \phi_K(\tau)]^T \in \mathbb{R}^K, \quad \boldsymbol{\theta}_j := [\theta_{j1}, \ldots, \theta_{jK}]^T \in \mathbb{R}^K.$$

Therefore, for any $\boldsymbol{\beta}(\tau) := [\beta_1(\tau), \ldots, \beta_p(\tau)]^T \in \mathcal{F}^p$, we can write

$$\boldsymbol{\beta}(\tau) = \boldsymbol{\Phi}(\tau)\boldsymbol{\theta},$$

where

$$\boldsymbol{\Phi}(\tau) := \mathbf{I}_p \otimes \boldsymbol{\phi}^T(\tau) \in \mathbb{R}^{p\times pK}, \quad \boldsymbol{\theta} := [\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_p^T]^T \in \mathbb{R}^{pK}.$$

With this notation and $c_\ell := ncw_\ell$, the SQR problem (3) can be restated as

$$\hat{\boldsymbol{\theta}} := \underset{\boldsymbol{\theta}\in\mathbb{R}^{pK}}{\operatorname{argmin}} \left\{ \sum_{\ell=1}^{L} \sum_{t=1}^{n} \rho_{\tau_\ell}(y_t - \mathbf{x}_t^T \boldsymbol{\Phi}(\tau_\ell)\boldsymbol{\theta}) + \sum_{\ell=1}^{L} c_\ell \, \|\ddot{\boldsymbol{\Phi}}(\tau_\ell)\boldsymbol{\theta}\|_1 \right\} \tag{4}$$

and

$$\hat{\boldsymbol{\beta}}(\tau) := \boldsymbol{\Phi}(\tau)\hat{\boldsymbol{\theta}}, \tag{5}$$

where $\ddot{\boldsymbol{\Phi}}(\tau)$ denotes the second derivative of $\boldsymbol{\Phi}(\tau)$. In other words, the SQR problem (3) can be solved by searching for the vector $\hat{\boldsymbol{\theta}}$ in $\mathbb{R}^{pK}$ according to (4) and then converting it to the desired function $\hat{\boldsymbol{\beta}}(\cdot)$ according to (5).

3 Linear Program Reformulation

Like the ordinary QR problem [14, p. 7], the SQR problem in (4) can be reformulated as a linear program (LP) with nonnegative decision variables:

$$\begin{aligned}
&(\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\delta}}, \hat{\mathbf{u}}_1, \hat{\mathbf{v}}_1, \hat{\mathbf{r}}_1, \hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{u}}_L, \hat{\mathbf{v}}_L, \hat{\mathbf{r}}_L, \hat{\mathbf{s}}_L) := \\
&\quad \underset{(\boldsymbol{\gamma},\boldsymbol{\delta},\mathbf{u}_1,\mathbf{v}_1,\mathbf{r}_1,\mathbf{s}_1,\ldots,\mathbf{u}_L,\mathbf{v}_L,\mathbf{r}_L,\mathbf{s}_L)\in\mathbb{R}^d_+}{\operatorname{argmin}} \sum_{\ell=1}^{L}\{\tau_\ell \mathbf{1}_n^T\mathbf{u}_\ell + (1-\tau_\ell)\mathbf{1}_n^T\mathbf{v}_\ell + \mathbf{1}_p^T\mathbf{r}_\ell + \mathbf{1}_p^T\mathbf{s}_\ell\} \\
&\quad \text{s.t.} \quad \begin{cases} \mathbf{X}\boldsymbol{\Phi}(\tau_\ell)(\boldsymbol{\gamma}-\boldsymbol{\delta}) + \mathbf{u}_\ell - \mathbf{v}_\ell = \mathbf{y} \quad (\ell = 1,\ldots,L), \\ c_\ell \ddot{\boldsymbol{\Phi}}(\tau_\ell)(\boldsymbol{\gamma}-\boldsymbol{\delta}) - (\mathbf{r}_\ell - \mathbf{s}_\ell) = \mathbf{0} \quad (\ell = 1,\ldots,L), \end{cases}
\end{aligned} \tag{6}$$

where $\mathbf{1}_n$ and $\mathbf{1}_p$ are the n-dimensional and p-dimensional vectors of 1's, $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ is the regression design matrix, and $d := 2pK + 2nL + 2pL$ is the total number of decision variables to be optimized. Among the decision variables in (6), $\boldsymbol{\gamma} \in \mathbb{R}_+^{pK}$ and $\boldsymbol{\delta} \in \mathbb{R}_+^{pK}$ are primary variables which determine the solution $\hat{\boldsymbol{\theta}}$ in (4) such that

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\gamma}} - \hat{\boldsymbol{\delta}}. \tag{7}$$

The remaining variables $\mathbf{u}_\ell \in \mathbb{R}_+^n$, $\mathbf{v}_\ell \in \mathbb{R}_+^n$, $\mathbf{r}_\ell \in \mathbb{R}_+^p$, and $\mathbf{s}_\ell \in \mathbb{R}_+^p$ are auxiliary variables introduced just for the purpose of linearizing the objective function in (4).

In the canonical form, the LP problem (6) can be expressed as

$$\min\{\mathbf{c}^T\boldsymbol{\xi} \,|\, \mathbf{A}\boldsymbol{\xi} = \mathbf{b}; \boldsymbol{\xi} \in \mathbb{R}_+^{2pK+2nL+2pL}\}, \tag{8}$$

where

$$\mathbf{c} := [\mathbf{0}_p^T, \mathbf{0}_p^T, \tau_1\mathbf{1}_n^T, (1-\tau_1)\mathbf{1}_n^T, \mathbf{1}_p^T, \mathbf{1}_p^T, \ldots, \tau_L\mathbf{1}_n^T, (1-\tau_L)\mathbf{1}_n^T, \mathbf{1}_p^T, \mathbf{1}_p^T]^T,$$

$$\boldsymbol{\xi} := [\boldsymbol{\gamma}^T, \boldsymbol{\delta}^T, \mathbf{u}_1^T, \mathbf{v}_1^T, \mathbf{r}_1^T, \mathbf{s}_1^T, \ldots, \mathbf{u}_L^T, \mathbf{v}_L^T, \mathbf{r}_L^T, \mathbf{s}_L^T]^T,$$

$$\mathbf{A} := \begin{bmatrix} \mathbf{X}\,\boldsymbol{\Phi}(\tau_1) & -\mathbf{X}\,\boldsymbol{\Phi}(\tau_1) & \mathbf{I}_n & -\mathbf{I}_n & \mathbf{0} & \mathbf{0} & & & \\ \vdots & \vdots & & & & & \ddots & & \\ \mathbf{X}\,\boldsymbol{\Phi}(\tau_L) & -\mathbf{X}\,\boldsymbol{\Phi}(\tau_L) & & & & & \mathbf{I}_n & -\mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ c_1\,\ddot{\boldsymbol{\Phi}}(\tau_1) & -c_1\,\ddot{\boldsymbol{\Phi}}(\tau_1) & \mathbf{0} & \mathbf{0} & \mathbf{I}_p & \mathbf{I}_p & & & \\ \vdots & \vdots & & & & & \ddots & & \\ c_L\,\ddot{\boldsymbol{\Phi}}(\tau_L) & -c_L\,\ddot{\boldsymbol{\Phi}}(\tau_L) & & & & & \mathbf{0} & \mathbf{0} & \mathbf{I}_p & -\mathbf{I}_p \end{bmatrix},$$

and

$$\mathbf{b} := [\underbrace{\mathbf{y}^T, \ldots, \mathbf{y}^T}_{L \text{ times}}, \underbrace{\mathbf{0}_p^T, \ldots, \mathbf{0}_p^T}_{L \text{ times}}]^T \in \mathbb{R}^{nL+pL}.$$

Given the canonical form (8), one can compute the SQR solution by any general-purpose LP solvers in open-source or commercial software packages. For example, the `lp` function in the R package 'lpSolve' [5] provides an interface to the open-source software `lp_solve` that solves LP problems by a simplex method. However, in using these solvers, the high dimensionality of decision variables and constraints in (8) can be a challenge to both computer memory and computer time.

In the following, we present a special-purpose algorithm based on the primal-dual interior-point method of Portnoy and Koenker [26]. This method was originally employed to solve the ordinary QR problem [14, pp. 199–202]. As it turns out, the SQR problem (4) can also be solved by this method with suitable modifications.

First, let us consider the dual LP problem associated with the primal problem (8):

$$\max\{\mathbf{b}^T\boldsymbol{\lambda} | \mathbf{A}^T\boldsymbol{\lambda} \leq \mathbf{c}; \boldsymbol{\lambda} \in \mathbb{R}^{nL+pL}\}. \tag{9}$$

The quantity $\boldsymbol{\lambda}$ may be interpreted as the Lagrange multiplier for the equality constraints in (8).

Theorem 1 *The dual LP problem (9) can be rewritten as*

$$\max\{\mathbf{b}^T\boldsymbol{\zeta} | \mathbf{D}^T\boldsymbol{\zeta} = \mathbf{a}; \boldsymbol{\zeta} \in [0, 1]^{nL+pL}\}, \tag{10}$$

where

$$\mathbf{D} := [\boldsymbol{\Phi}^T(\tau_1)\,\mathbf{X}^T, \ldots, \boldsymbol{\Phi}^T(\tau_L)\,\mathbf{X}^T, 2c_1\,\ddot{\boldsymbol{\Phi}}^T(\tau_1), \ldots, 2c_L\,\ddot{\boldsymbol{\Phi}}^T(\tau_L)]^T,$$
$$\mathbf{a} := \sum_{\ell=1}^{L}\{(1-\tau_\ell)\,\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\mathbf{1}_n + c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\mathbf{1}_p\}.$$

Proof By partitioning $\boldsymbol{\lambda}$ according to the structure of $\mathbf{b}$ such that

$$\boldsymbol{\lambda} := [\boldsymbol{\lambda}_1^T, \ldots, \boldsymbol{\lambda}_L^T, \boldsymbol{\lambda}_{L+1}^T, \ldots, \boldsymbol{\lambda}_{2L}^T]^T,$$

the inequalities $\mathbf{A}^T\boldsymbol{\lambda} \le \mathbf{c}$ in (9) can be written more elaborately as

$$\begin{aligned}
\sum_{\ell=1}^{L}\{\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\boldsymbol{\lambda}_\ell + c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\boldsymbol{\lambda}_{L+\ell}\} &\le \mathbf{0}_p,\\
-\sum_{\ell=1}^{L}\{\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\boldsymbol{\lambda}_\ell + c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\boldsymbol{\lambda}_{L+\ell}\} &\le \mathbf{0}_p,\\
\boldsymbol{\lambda}_\ell \le \tau_\ell\mathbf{1}_n, \quad -\boldsymbol{\lambda}_\ell \le (1-\tau_\ell)\mathbf{1}_n \quad (\ell = 1,\ldots,L),\\
-\boldsymbol{\lambda}_{L+\ell} \le \mathbf{1}_p, \quad \boldsymbol{\lambda}_{L+\ell} \le \mathbf{1}_p \quad (\ell = 1,\ldots,L).
\end{aligned}$$

These inequalities are equivalent to

$$\begin{aligned}
\sum_{\ell=1}^{L}\{\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\boldsymbol{\lambda}_\ell + c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\boldsymbol{\lambda}_{L+\ell}\} &= \mathbf{0}_p,\\
\boldsymbol{\lambda}_\ell \in [\tau_\ell - 1, \tau_\ell]^n \quad (\ell = 1,\ldots,L),\\
\boldsymbol{\lambda}_{L+\ell} \in [-1,1]^p \quad (\ell = 1,\ldots,L).
\end{aligned}$$

By a change of variables,

$$\boldsymbol{\zeta}_\ell := \boldsymbol{\lambda}_\ell + (1-\tau_\ell)\mathbf{1}_n, \quad \boldsymbol{\zeta}_{L+\ell} := \frac{1}{2}(\boldsymbol{\lambda}_{L+\ell} + \mathbf{1}_p),$$

we can write

$$\mathbf{b}^T\boldsymbol{\lambda} = \sum_{\ell=1}^{L}\mathbf{y}^T\boldsymbol{\lambda}_\ell = \sum_{\ell=1}^{L}\mathbf{y}^T\boldsymbol{\zeta}_\ell - \sum_{\ell=1}^{L}(1-\tau_\ell)\mathbf{y}^T\mathbf{1}_n = \mathbf{b}^T\boldsymbol{\zeta} + \text{constant},$$

$$\begin{aligned}
&\sum_{\ell=1}^{L}\{\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\boldsymbol{\lambda}_\ell + c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\boldsymbol{\lambda}_{L+\ell}\}\\
&= \sum_{\ell=1}^{L}\{\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\boldsymbol{\zeta}_\ell + 2c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\boldsymbol{\zeta}_{L+\ell}\}\\
&\quad - \sum_{\ell=1}^{L}\{(1-\tau_\ell)\,\boldsymbol{\Phi}^T(\tau_\ell)\,\mathbf{X}^T\mathbf{1}_n + c_\ell\,\ddot{\boldsymbol{\Phi}}^T(\tau_\ell)\mathbf{1}_p\},
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\lambda}_\ell \in [\tau_\ell - 1, \tau_\ell]^n &\leftrightarrow \boldsymbol{\zeta}_\ell \in [0,1]^n,\\
\boldsymbol{\lambda}_{L+\ell} \in [-1,1]^p &\leftrightarrow \boldsymbol{\zeta}_{L+\ell} \in [0,1]^p.
\end{aligned}$$

Combining these expressions proves the assertion. □

To fully justify the use of the interior-point method of Portnoy and Koenker [26], we need to verify that the primal problem (8) can also be put into the required form. This is accomplished by the following theorem.

Theorem 2 *The primal problem (8) can be rewritten as*

$$\min\{\mathbf{a}^T\boldsymbol{\theta}|\mathbf{D}\boldsymbol{\theta}+\mathbf{z}-\mathbf{w}=\mathbf{b};\boldsymbol{\theta}\in\mathbb{R}^{pK};\mathbf{z},\mathbf{w}\in\mathbb{R}_+^{nL+pL}\},\tag{11}$$

where $\boldsymbol{\theta} := \boldsymbol{\gamma}-\boldsymbol{\delta}$, $\mathbf{z} := [\mathbf{u}_1^T,\ldots,\mathbf{u}_L^T,\ 2\mathbf{s}_1^T,\ldots,2\mathbf{s}_L^T]^T$, *and* $\mathbf{w} := [\mathbf{v}_1^T,\ldots,\mathbf{v}_L^T,\ 2\mathbf{r}_1^T,\ldots,2\mathbf{r}_L^T]^T$.

Proof Observe that the equality constraints in (8) can be written as

$$\mathbf{D}\boldsymbol{\theta}+\mathbf{z}-\mathbf{w}=\mathbf{b}.\tag{12}$$

Under these constraints, we have $\mathbf{v}_\ell=\mathbf{X}\boldsymbol{\Phi}(\tau_\ell)\boldsymbol{\theta}+\mathbf{u}_\ell-\mathbf{y}$ and $\mathbf{r}_\ell=c_\ell\ddot{\boldsymbol{\Phi}}(\tau_\ell)\boldsymbol{\theta}+\mathbf{s}_\ell$. Substituting these expressions in (8) yields

$$\begin{aligned}
\mathbf{c}^T\boldsymbol{\xi} &= \sum_{\ell=1}^{L}\{\tau_\ell\mathbf{1}_n^T\mathbf{u}_\ell+(1-\tau_\ell)\mathbf{1}_n^T\mathbf{v}_\ell+\mathbf{1}_p^T\mathbf{r}_\ell+\mathbf{1}_p^T\mathbf{s}_\ell\}\\
&= \sum_{\ell=1}^{L}\{\tau_\ell\mathbf{1}_n^T\mathbf{u}_\ell+(1-\tau_\ell)\mathbf{1}_n^T(\mathbf{X}\boldsymbol{\Phi}(\tau_\ell)\boldsymbol{\theta}+\mathbf{u}_\ell-\mathbf{y})+\mathbf{1}_p^T(c_\ell\ddot{\boldsymbol{\Phi}}(\tau_\ell)\boldsymbol{\theta}+\mathbf{s}_\ell)+\mathbf{1}_p^T\mathbf{s}_\ell\}\\
&= \sum_{\ell=1}^{L}\{(1-\tau_\ell)\mathbf{1}_n^T\mathbf{X}\boldsymbol{\Phi}(\tau_\ell)\boldsymbol{\theta}+c_\ell\mathbf{1}_p^T\ddot{\boldsymbol{\Phi}}(\tau_\ell)\boldsymbol{\theta}+\mathbf{1}_n^T\mathbf{u}_\ell+2\mathbf{1}_p^T\mathbf{s}_\ell-(1-\tau_\ell)\mathbf{1}_n^T\mathbf{y}\}\\
&= \mathbf{a}^T\boldsymbol{\theta}+\|\mathbf{z}\|_1+\text{constant}.
\end{aligned}$$

The assertion follows upon noting that $\mathbf{z}$ is not a free parameter under the condition (12) because it is equivalent to $\mathbf{D}\boldsymbol{\theta}-\mathbf{w}\le\mathbf{b}$ and $\mathbf{z}:=\mathbf{b}-(\mathbf{D}\boldsymbol{\theta}-\mathbf{w})$.

The primal-dual pair given by (10) and (11) are in the canonical form required by the interior-point algorithm of Portnoy and Koenker [26]. This algorithm solves the primal-dual pair jointly by using Newton's method in which positivity constraints are enforced by log barriers [16]. An implementation of this algorithm as a FORTRAN code, `rqfnb.f`, is available in the 'quantreg' package. The R function `rq.fit.fnb` in this package invokes the FORTRAN code to compute the solution to the ordinary QR problem. For the SQR problem, it suffices to modify this function using proper input variables.

The R function `rq.fit.fnb` was developed to take input variables from the dual formulation,

$$\max\{\mathbf{y}^T\boldsymbol{\zeta}|\mathbf{X}^T\boldsymbol{\zeta}=(1-\tau)\mathbf{X}^T\mathbf{1}_n;\boldsymbol{\zeta}\in[0,1]^n\},$$

of the ordinary QR problem $\min_{\boldsymbol{\beta}}\sum_{t=1}^{n}\rho_\tau(y_t-\mathbf{x}_t^T\boldsymbol{\beta})$. Because the dual problem for SQR is given by (10), it suffices to replace the response vector $\mathbf{y}$ by $\mathbf{b}$, the design matrix

$\mathbf{X}$ by $\mathbf{D}$, and the right-hand-side vector $(1-\tau)\mathbf{X}^T\mathbf{1}_n$ in the equality constraint by $\mathbf{a}$. In addition, we set the initial value of $\boldsymbol{\zeta}$ to $[(1-\tau_1)\mathbf{1}_n^T, \ldots, (1-\tau_L)\mathbf{1}_n^T, 0.5\mathbf{1}_{pL}^T]^T$. The modified R function, `rq.fit.fnb2`, turns out to be much more efficient in terms of computer memory and time than standard LP solvers, such as the `lp` function, for solving the SQR problem.

To develop a data-driven method for selecting the smoothing parameter c, we adopt a technique which was used by Koenker et al. [17] for nonparametric quantile regression at a fixed quantile (see also [14, p. 234]). In this technique, the mean objective function of quantile regression at the fitted values is treated as the fidelity measure, analogous to the standard error of the residuals in least-squares regression, and the number of points interpolated (or closely approximated) by the fitted values is treated as the complexity measure, similar to the number of parameters in least-squares regression. We extend the fidelity and complexity measures to the SQR problem. Let $\sigma_c(\tau_\ell) := n^{-1}\sum_{t=1}^n \rho_{\tau_\ell}(y_t - \mathbf{x}_t^T\hat{\boldsymbol{\beta}}(\tau_\ell))$ be the fidelity measure at τ_ℓ and $m_c(\tau_\ell) := \sum_{t=1}^n I(|y_t - \mathbf{x}_t^T\hat{\boldsymbol{\beta}}(\tau_\ell)| < \epsilon)$ be the complexity measure at τ_ℓ. Then, the resulting Bayesian information criterion (BIC), also known as Schwarz information criterion (SIC), takes the form

$$\mathrm{BIC}(c) := 2n\log\left(L^{-1}\sum_{\ell=1}^{L}\sigma_c(\tau_\ell)\right) + (\log n)\left(L^{-1}\sum_{\ell=1}^{L}m_c(\tau_\ell)\right). \tag{13}$$

The corresponding AIC criterion is obtained by replacing $\log n$ in (13) with 2. As usual, BIC imposes a heavier penalty on the complexity than AIC when $\log n > 2$. As a result, BIC tends to produce a smoother estimate than AIC.

In the R implementation, we reparameterize c by `spar` in a way similar to the smoothing parameter in the R function `smooth.spline` [28], i.e., let $c := r \times 1000^{\mathtt{spar}-1}$ with $r := n^{-1}\sum_{\ell=1}^L \|\mathbf{X}\boldsymbol{\Phi}(\tau_\ell)\|_1 / \sum_{\ell=1}^L w_\ell \|\ddot{\boldsymbol{\Phi}}(\tau_\ell)\|_1$. In addition, we use the `smooth.spline` function to generate the knots from a given set of quantile levels, and use the `splineDesign` function in the R package 'splines' [28] to generate the corresponding cubic spline basis functions and their second derivatives.

4 Examples

To demonstrate the capability of SQR in providing more accurate estimates, we employ a simulated data set of time series generated from a quantile autoregressive (QAR) model

$$y_t = a_0(U_t) + a_1(U_t)\, y_{t-1}. \tag{14}$$

In this model, $\{U_t\}$ is an i.i.d. sequence of $U(0,1)$ random variables, $a_0(u)$ takes the form of the quantile function of $N(0, 0.4^2)$, $a_1(u)$ is a piecewise-linear function $0.85 + 0.1u + 0.25(u-0.5)I(u > 0.5)$. The model is a slight modification of the example in Koenker [14, p. 262] where $a_0(u)$ is the quantile function of $N(0,1)$ and $a_1(u)$ is a linear function $0.85 + 0.25u$. For the QAR model (14), the conditional

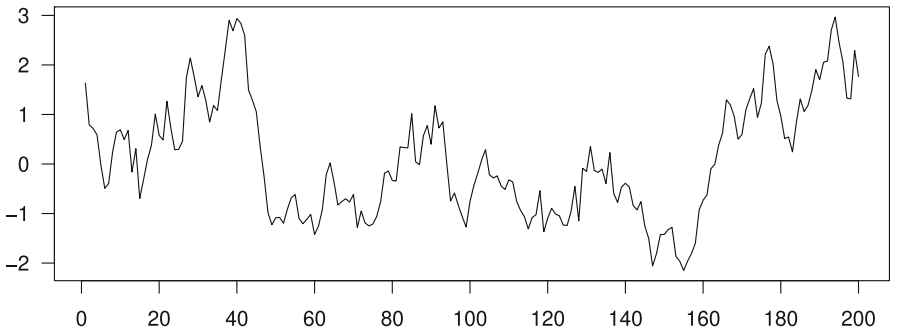

Fig. 1 A simulated time series according to the QAR model (14) ($n = 200$)

quantile function of y_t given y_{t-1} can be written as $Q_{y_t}(\tau \mid y_{t-1}) = a_0(\tau) + a_1(\tau)\, y_{t-1}$ $(0 < \tau < 1)$.

Figure 1 shows a series simulated from (14). Figure 2 depicts the SQR estimates and the QR estimates of $a_0(\cdot)$ and $a_1(\cdot)$ from this series on the quantiles $\{0.05, 0.06, \ldots, 0.95\}$. The smoothing parameter of SQR is selected by AIC and BIC, respectively. To measure the accuracy of these estimates, we employ the absolute errors, $L^{-1} \sum_{\ell=1}^{L} |\hat{a}_j(\tau_\ell) - a_j(\tau_\ell)|$ $(j = 0, 1)$. For the SQR estimates with AIC, the errors are both 0.009; for the SQR estimates with BIC, the errors are 0.009 and 0.008, respectively. The corresponding errors of the QR estimates are 0.015 and 0.013. Visual inspection of Figures 2(b) and (c) shows that the SQR estimate with BIC follows the shape of $a_1(\cdot)$ more faithfully than the SQR estimate with AIC.

A more comprehensive assessment of the estimation accuracy is presented in Figure 3. Based on 1000 Monte Carlo runs, Figure 3 shows the total mean absolute error $\mathrm{MAE}_0 + \mathrm{MAE}_1$ against the value of smoothing parameter `spar`, where

$$\mathrm{MAE}_j := E\left\{ L^{-1} \sum_{\ell=1}^{L} |\hat{a}_j(\tau_\ell) - a_j(\tau_\ell)| \right\} \quad (j = 0, 1).$$

As can be seen, the SQR estimates outperform the QR estimates for a range of smoothing parameter values, with the best choice being `spar` = 0.9. Table 1 further demonstrates that the data-driven criteria AIC and BIC are both able to select a smoothing parameter for SQR to outperform QR for both sample sizes. The improved accuracy of SQR over QR comes largely from estimating $a_1(\cdot)$.

For comparison, Figure 3 also contains the results (labeled QR-S) from the post-smoothing method which applies the `smooth.spline` function to the QR estimates for each coefficient independently with the smoothing parameter selected by GCV (default). This method produces better estimates than the QR method but remains inferior to the SQR method in terms of total MAE.

Having validated SQR with simulated data as a more accurate estimator than QR for estimating the conditional quantile function, we present some real-data examples in the remainder of this section to demonstrate its application.

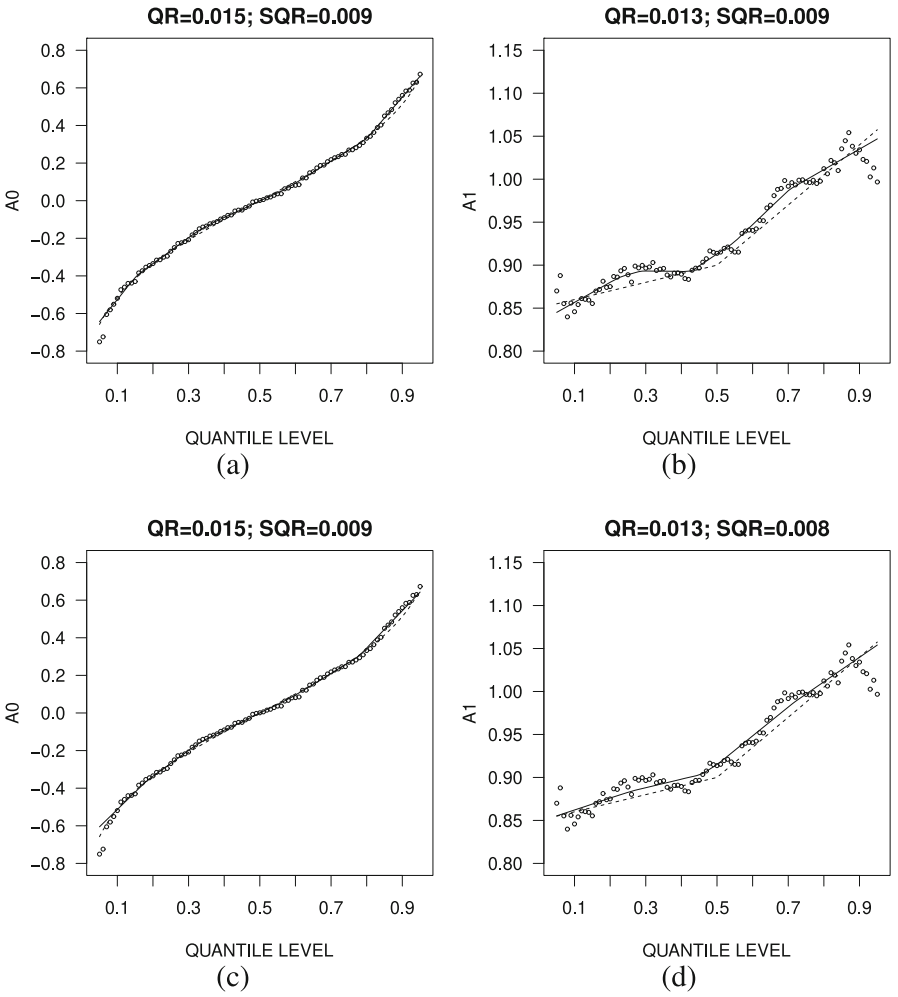

Fig. 2 SQR estimates of $a_0(\cdot)$ and $a_1(\cdot)$ in the QAR model (14) from the time series shown in Figure 1. The smoothing parameter is chosen by (a)-(b) AIC and (c)-(d) BIC. Open circles depict the QR estimates. Dashed lines depict the true values

The first real-data example is the Engel food expenditure data that comes with the 'quantreg' package [14, pp. 300–302]. It is assumed that the household expenditure on food, y, obeys a quantile regression model

$$Q_y(\tau \mid x) = \beta_1(\tau) + \beta_2(\tau)\,(x - \mu),$$

where x is the household income ($\times 1000$) with mean μ. The data set contains $n = 235$ records.

By following Koenker [14], we estimate the coefficients $\beta_1(\cdot)$ and $\beta_2(\cdot)$ by QR on the quantile grid $\{0.02, 0.03, \ldots, 0.98\}$ with μ replaced by the sample mean of income.

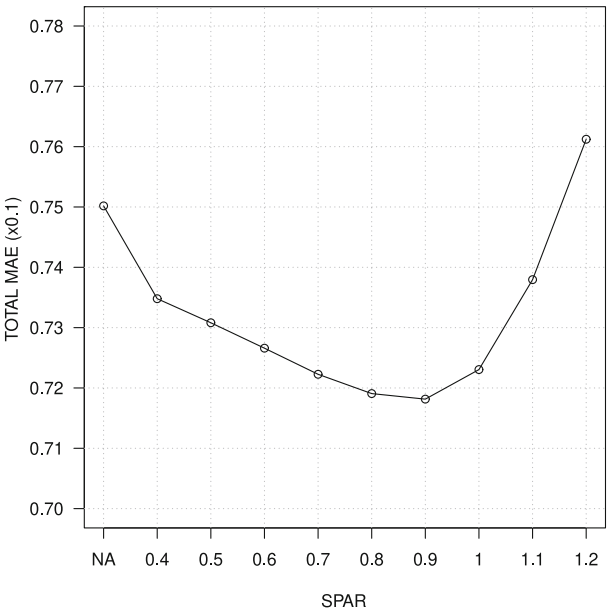

Fig. 3 Total mean absolute error for estimating the functional coefficients in the QAR model (14) by SQR with different values of smoothing parameter `spar`. NA stands for the ordinary QR. Results are based on 1000 Monte Carlo runs with $n = 200$

Table 1 Mean Absolute Error of SQR Estimates with AIC and BIC for the QAR Model (14)

n	MAE_0 ($\times 0.01$)				MAE_1 ($\times 0.01$)				Total MAE ($\times 0.01$)			
	QR	QR-S	SQR^a	SQR^b	QR	QR-S	SQR^a	SQR^b	QR	QR-S	SQR^a	SQR^b
200	3.77	3.73	3.66	3.69	3.74	3.71	3.56	3.51	7.51	7.44	7.22	7.20
500	2.21	2.19	2.17	2.21	2.08	2.06	1.98	1.94	4.29	4.25	4.15	4.15

a, smoothing parameter selected by AIC; *b*, smoothing parameter selected by BIC. Results are based on 1000 Monte Carlo run.

These estimates are compared with the SQR estimates obtained with automatically selected smoothing parameter by AIC and BIC. The results are shown in Figure 4 as functions of τ. The confidence band for the SQR estimates in this figure is borrowed from the asymptotic normality of the ordinary quantile regression estimates under the non-iid settings [14, p. 34], as implemented by the `summary.rq` function in the R package 'quantreg' with the option `se='nid'`. Additional research is needed to develop a similar theory for the SQR estimates.

In comparison with the QR estimates (open circles in Figure 4), the SQR estimates are less noisy, more visually appealing, and potentially more accurate for assessing the impact of household income on the food expenditure. As expected, BIC produces a "smoother" estimate than AIC due to a heavier penalty on the model complexity. However, without ground true, one should not favor one over the other.

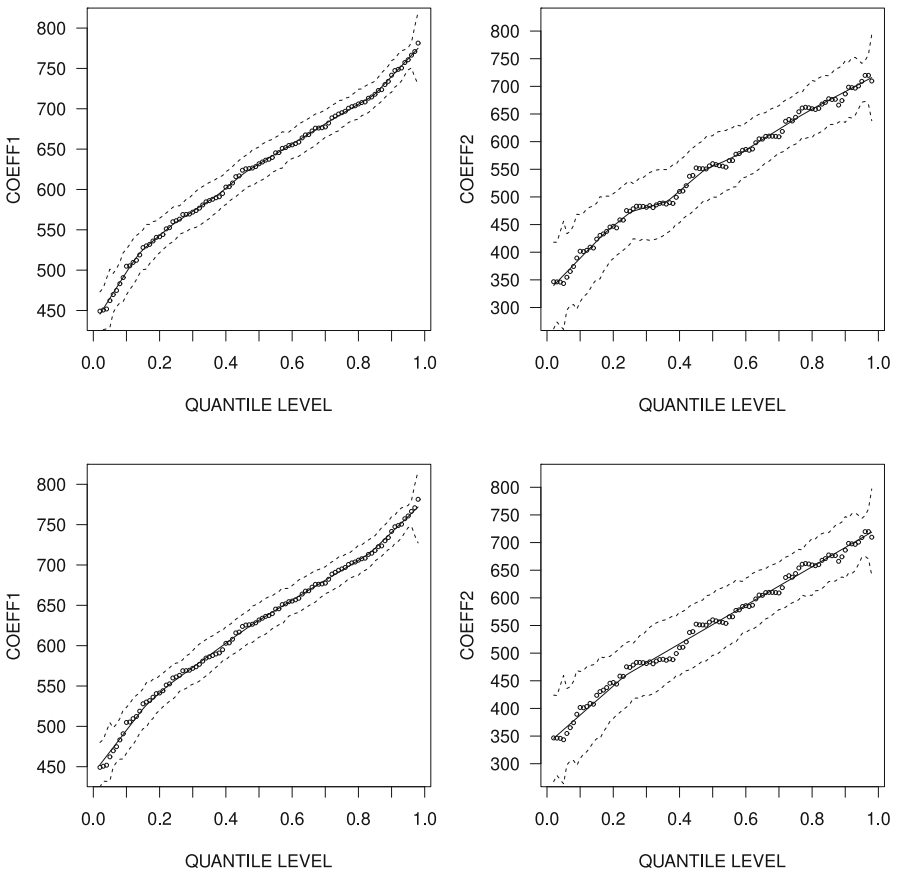

Fig. 4 SQR estimates of the functional coefficients $\beta_1(\cdot)$ (left) and $\beta_2(\cdot)$ (right) for the Engel food expenditure data. Smoothing parameters are selected by AIC (top) and BIC (bottom). Dashed lines depict a 95% pointwise confidence band. Open circles depict the QR estimates

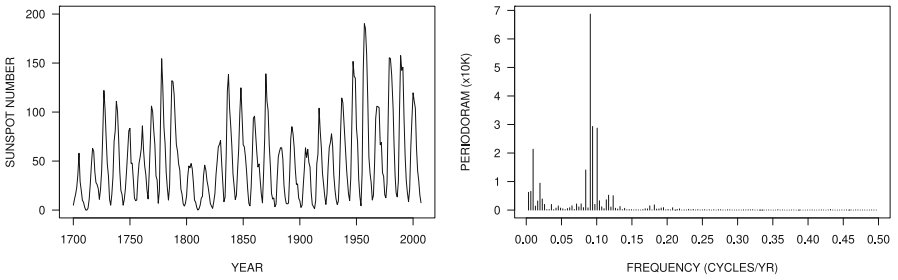

Fig. 5 Time series of yearly sunspot numbers from year 1700 to 2007 ($n = 308$) and its conventional periodogram

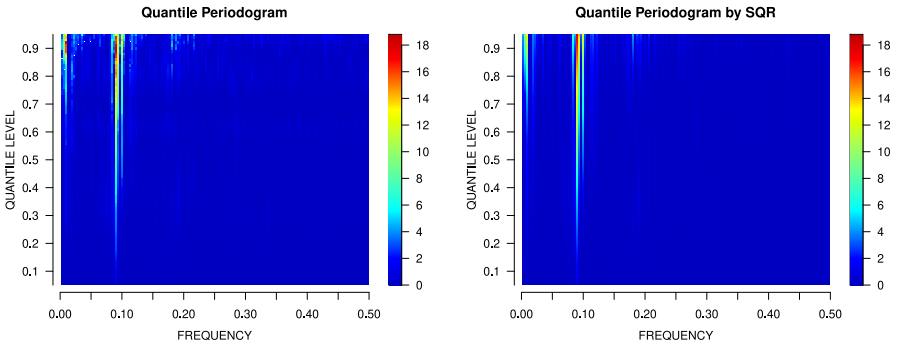

Fig. 6 Quantile periodogram of yearly sunspot numbers constructed by trigonometric quantile regression using QR (left) and SQR with BIC (right)

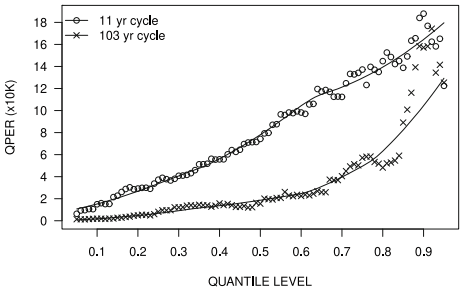

Fig. 7 Cross-section of quantile periodogram, constructed by SQR with BIC, of yearly sunspot numbers at 11 year and 103 year frequencies. Open circles depict the result from QR

While the SQR estimates in the previous example do not differ significantly from the QR estimates, bigger differences are observed in the second real-data example. This example is the time series of yearly sunspot numbers (Figure 5). This series was examined in Li [19] and Li [20] for its periodicity through the so-called quantile periodogram shown in the left panel of Figure 6. The quantile periodogram is an alternative to the conventional periodogram for spectral analysis of time series. It is derived from trigonometric quantile regression instead of Fourier transform. Formally, with f taking values in the set of normalized Fourier frequencies in the interval $(0, 0.5)$ and τ taking values in the interval $(0, 1)$, the quantile periodogram is defined as

$$(n/4)[\hat{\beta}_2^2(f,\tau) + \hat{\beta}_3^2(f,\tau)],$$

where $\hat{\beta}_2(f,\tau)$ and $\hat{\beta}_3(f,\tau)$ are given by QR using the trigonometric or harmonic regressor $\mathbf{x}_t(f) := [1, \cos(2\pi ft), \sin(2\pi ft)]^T$. As a bivariate function of f and τ, the quantile periodogram in the left panel of Figure 6 suggests that the sunspot numbers not only have the well-known cycle of 11 years ($f = 28/308 \approx 0.091$), but the magnitude of this cycle exhibits an increasing trend with the increase of quantile level, i.e., the cycle tends to be stronger at higher quantiles and weaker at lower quantiles.

Instead of QR, the trigonometric quantile regression can be performed by SQR. The resulting quantile periodogram is shown in the right panel of Figure 6. To construct this quantile periodogram, a single smoothing parameter is employed in the trigonometric

SQR problems for all frequencies. It is chosen by minimizing the average BIC across the frequencies. Needless to say, one may also choose to employ different smoothing parameters for different frequencies to achieve greater flexibility but with higher computational burden and statistical variability.

Compared to the QR-based quantile periodogram, the SQR-based quantile periodogram appears smoother across quantiles, not only around the peak frequencies but also in the background. The difference between these periodograms can be further appreciated by inspecting the cross-section plot in Figure 7, where the quantile periodograms are shown as functions of τ at two peak frequencies corresponding to the 11 year cycle ($f = 28/308$) and the 103 year cycle ($f = 3/308$). Compared to the QR estimates, the SQR estimates depict a better-defined less-noisy trend which grows steadily with τ. The effect of smoothing is most notable at higher quantiles, especially in the curve of the 103 year cycle.

The final real-data example is the US infant birth weight data from the National Bureau of Economic Research (NBER)[1] for year 2022. A similar data set from year 1997 was used in Koenker [14, p. 20] to demonstrate the usefulness of the quantile regression coefficients at multiple quantiles. By following this study, we extract the data records for black or white mothers between ages 18 and 45. The study in Koenker [14, p. 20] included 16 explanatory variables for the birth weight (in grams). We simplify the study by considering only 8 explanatory variables: infant's sex (0 for Girl, 1 for Boy), mother's race (0 for White, 1 for Black), age (18–45), and marital status (0 for Unmarried, 1 for Married), and prenatal medical care named `Precare` (0 if the first visit in the first trimester of the pregnancy, 1 if no prenatal visit, 2 if the first visit in the second trimester, 3 if the first visit in the last trimester). For simplicity, mother's education level, smoking status, and weight gain during pregnancy are not used in the model. As in Koenker [14, p. 22], we use a quadratic function to represent the effect of age.

By following Koenker [14, p. 21], we compute the QR estimates on the quantile grid $\{0.05, 0.06, \ldots, 0.95\}$. These estimates are obtained from a random sample of $n = 50000$ records. The corresponding SQR estimates are computed with `spar` $= -1$. Figure 8 shows the QR and SQR estimates as functions of τ. The 90% pointwise confidence band (shaded gray areas) for the QR estimates are constructed using the estimated standard errors of the ordinary quantile regression estimates under the iid assumption (`se='iid'` in `summary.rq`). The same errors are borrowed as approximation in constructing the confidence band for the SQR estimates. As in Koenker [14, p. 21], the dashed and dotted horizontal lines depict the ordinary least-squares estimate of the mean effect and a 90% confidence interval of the estimate.

The QR estimates in Figure 8(a) are very similar to those in Koenker [14, p. 21]. The SQR estimates in Figure 8(b) present the quantile-dependent effect of the explanatory variables with better-defined and less-noisy patterns which are potentially more accurate in estimating the underlying conditional quantile function.

[1] https://www.nber.org/research/data/vital-statistics-natality-birth-data

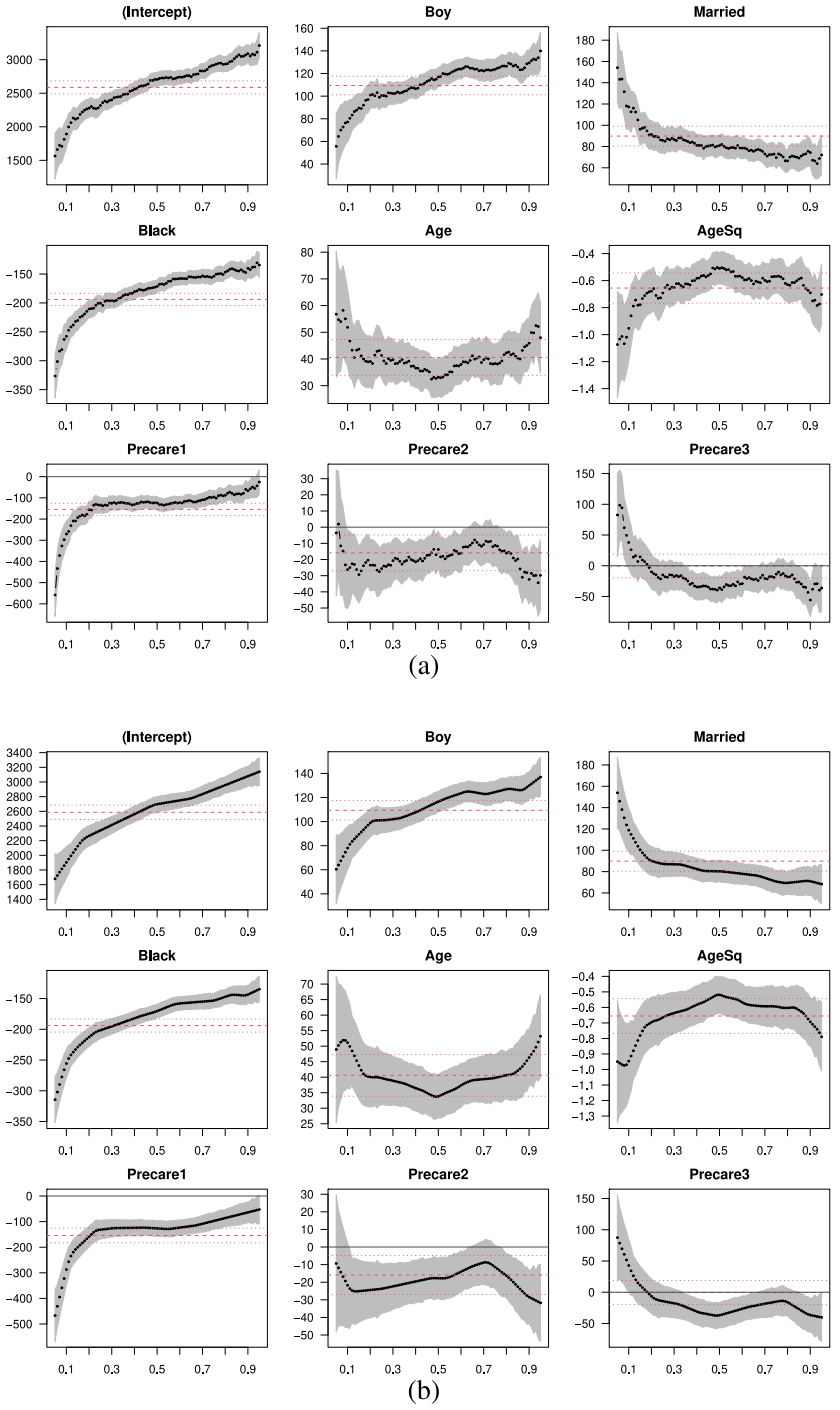

Fig. 8 Quantile regression for birth weight. (a) QR estimates. (b) SQR estimates. Shaded gray area depicts a 90% pointwise confidence band

5 Gradient Algorithms

The LP method provides the exact solution to the SQR problem. However, the artificially inflated number of decision variables can be a challenge to the computer memory when $n + p$ is large. Indeed, the total number of decision variables in the primal-dual pair (10) and (11) equals $2(n + p)L + pK$, which is far greater than the pK variables in the original regression coefficients. This challenge calls for alternative algorithms that consume less memory but still provide reasonably good solutions that may not be exact. Gradient algorithms, which directly optimize the objective function in (4) with respect to $\boldsymbol{\theta}$ in $\mathbb{R}^{pK}$, are obvious candidates. The remaining issue is whether these algorithms work with the non-smooth objective function of quantile regression.

Despite the existence of counterexamples [3], gradient algorithms have been successfully used in practice to solve optimization problems with non-smooth objective functions (e.g., not everywhere differentiable or continuously differentiable). This is evidenced, for example, by the effectiveness of such algorithms in training neural network models involving non-smooth activation functions [8, 29]. We consider three such algorithms for the SQR problem in (4), where the objective function is convex, continuous, and piecewise linear, but not everywhere differentiable.

The first algorithm is the well-known Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [22]. To supply this algorithm with a gradient function, we set the derivatives of $\rho_\tau(y)$ and $|y|$ to zero when $y = 0$. This operation should not have a significant effect on the computed optimization solution when the overall gradient is determined jointly by many contributors, most of which having properly defined derivatives, as is likely the case for the SQR problem. The BFGS algorithm has been found effective for non-smooth problems in practice, but a general theory of convergence remains lacking. A mathematical analysis of its behavior for certain non-smooth functions including $|y|$ can be found in Lewis and Overton [18].

The BFGS algorithm is a quasi-Newton method that uses an approximate Hessian matrix to capture the local curvature of the objective function together with a line search to find the optimal step size for each iteration. The `optim` function in the R package 'stats' [28] has an option for the BFGS algorithm. This implementation employs a backtracking line search strategy and works remarkably well in our trigonometric SQR experiment. Another R implementation, `bfgs` (https://rdrr.io/rforge/rHanso/man/bfgs.html), takes a more aggressive bisection approach in line search, guided additionally by the so-called curvature condition [22]. Unfortunately, in our trigonometric SQR experiment, this implementation tends to terminate prematurely due to failures in line search.

The Hessian matrix update and the line search in BFGS are computationally expensive. For more economical alternatives, we turn to a basic gradient algorithm without the help of Hessian matrix and line search. Due to their computational simplicity, such algorithms have become increasingly popular for training neural network models [8, 29]. A successful example is known as ADAM [13]. In this algorithm, the usual gradient in a gradient descent iteration is replaced by an exponentially weighted average of all past gradients and normalized by the square root of an exponentially weighted average of squared gradients. The ADAM algorithm may oscillate around

Table 2 Limited Line Search in the GRAD Algorithm

Limited line search, performed periodically in GRAD after a warm-up phase. The criterion for accepting a trial step size and the discount factor $b := 0.2$ are the same as in `optim` for BFGS. (i) (ii) $s_0 :=$ default step size, (iii) (iv) $s_0 :=$ current step size; $\kappa_0 :=$ number of trials.
$s \leftarrow \min\{1, s_0 \times b^{-\lfloor \kappa_0/2 \rfloor}\}$ (initialize trial step size); $\kappa \leftarrow 0$ (initialize trial count)
`While` s not accepted and $\kappa < \kappa_0$ `Do`
$\quad s \leftarrow s \times b; \kappa \leftarrow \kappa + 1$
`End While`
`Return` (i) (iv) s if accepted s_0 otherwise, (ii) (iii) s if accepted $s_0 \times b$ otherwise

the minimizer indefinitely because the step size does not go to zero with the iteration. Therefore, it needs to be terminated after a predefined number of iterations.

The third algorithm, which we call GRAD, is an enhanced version of ADAM. In this algorithm, we modify ADAM by using a limited line search to adjust the step size during iteration rather than holding the step size constant. The line search, shown in Table 2, follows the same backtracking strategy as in `optim`. However, it is performed not in every iteration but with a much lower frequency and only after a warm-up period; it is also performed with a small number of trials including both increase and decrease in step size. Furthermore, there are four options with different choices for the starting step size and the returning step size when none of the trial step sizes are acceptable. In Table 2, options (i) and (ii) start with the default step size used in the warm-up period, whereas options (iii) and (iv) start with the current step size. When the line search fails to produce an acceptable step size, options (i) and (iv) fall back to the default step size, whereas options (ii) and (iii) fall back to the discounted current step size. Note that only option (iii) offers gradually decreased step sizes when increases are unacceptable as suggested by the convergence theory of subgradient methods for non-smooth problems [27]. None of these options guarantees a monotone reduction in the objective function. Option (i) deviates from ADAM the least, as the step size remains at the default value except when a different trial step size becomes acceptable. Option (iv) can produce the same result as option (i) when a step size different from the default value is never found acceptable.

6 Experimentation with Gradient Algorithms

In this section, we conduct two experiments to evaluate the accuracy of the gradient algorithms discussed in the previous section as approximations to the LP solution. We do not intend to demonstrate their computational advantages in handling large-scale problems.

The first experiment uses the Engel food expenditure data discussed earlier. Table 3 contains the total mean absolute errors of the gradient algorithms for approximating the two functional coefficients obtained by LP on the quantiles $\{0.02, 0.03, \ldots, 0.98\}$. The

Table 3 Approximation Error for Engel's Food Expenditure Data

Algorithm	Number of Iterations								
	0	50	100	150	200	300	400	500	1000
BFGS	6.9064	6.7616	6.6951	1.0965	0.2112	0.1316			
ADAM	6.9064	6.7194	6.6910	6.6903	6.6904	6.6904	6.6904	6.6904	6.6904
GRAD	6.9064	6.7194	6.6835	6.6771	6.6774	6.6027	6.6034	6.5307	6.4270

QR is used as initial value in iteration 0. For ADAM and GRAD, warm-up phase = 70, frequency of step size update = 20, initial step size $s_0 = 0.4$, and discount factor $b = 0.2$. GRAD uses option (i) for line search with $\kappa_0 = 5$.

best performance is achieved by BFGS, which approximates the LP solution closely with 300 iterations (further iteration makes no significant gains). ADAM and GRAD are unable to attain such accuracy despite a large number of additional iterations. GRAD is an improvement over ADAM in reducing the approximation error. Trading memory requirement with computer time is a typical property of gradient algorithms when they work. This is the case for BFGS. An informal test shows that BFGS takes 50 seconds to complete the 300 iterations versus 12 seconds by LP.

The second experiment employs a simulated time series of length $n = 512$. Motivated by the quantile periodogram [19], we use the gradient algorithms to compute the SQR coefficients $\hat{\beta}_1(f,\tau)$, $\hat{\beta}_2(f,\tau)$, and $\hat{\beta}_3(f,\tau)$ of the regressor $\mathbf{x}_t(f) := [1, \cos(2\pi ft), \sin(2\pi ft)]^T$ with $\tau \in \{0.10, 0.11, \ldots, 0.90\}$ at 255 Fourier frequencies $f \in \{v/n : v = 1, \ldots, \lfloor (n-1)/2 \rfloor\}$. We compare the resulting $\hat{\beta}_2(f,\tau)$ and $\hat{\beta}_3(f,\tau)$, which define the quantile periodogram, with the corresponding LP solutions. The average sum of squared errors across the quantiles and frequencies is used to measure the accuracy of the gradient algorithm.

The time series $\{y_t\}$ is generated by a nonlinear mixture model in Li [21]. For brevity, the exact form of this model is omitted here because it is not important for our discussion. It suffices to say that the 255 LP solutions have a variety of patterns based on which we evaluate the approximations produced by the gradient algorithms. One of the solutions is shown in Figure 9 together with the approximations produced by BFGS and ADAM and the corresponding QR solution. Both BFGS and ADAM are terminated after 100 iterations. As can be seen, the estimates from BFGS and ADAM appear remarkably similar to the estimates from LP.

Figure 10 shows the the boxplot of approximation errors of BFGS and ADAM for all 255 frequencies against the number of iterations. This result confirms that both BFGS and ADAM, with sufficiently large number of iterations, provide reasonably good approximations to the LP solution. The BFGS algorithm is able to offer a higher accuracy of approximation than ADAM after an initial warm-up period. The ADAM algorithm has a more successful start in early stages of the iteration, but stops improving beyond a certain number of iterations.

Figure 11 compares GRAD with ADAM and BFGS based on the average approximation error across all 255 frequencies. Line search in GRAD is performed once every 10 iterations with 5 trial step sizes after 70 warm-up iterations. Thanks to these modifications, GRAD is able to outperform ADAM in terms of the approximation

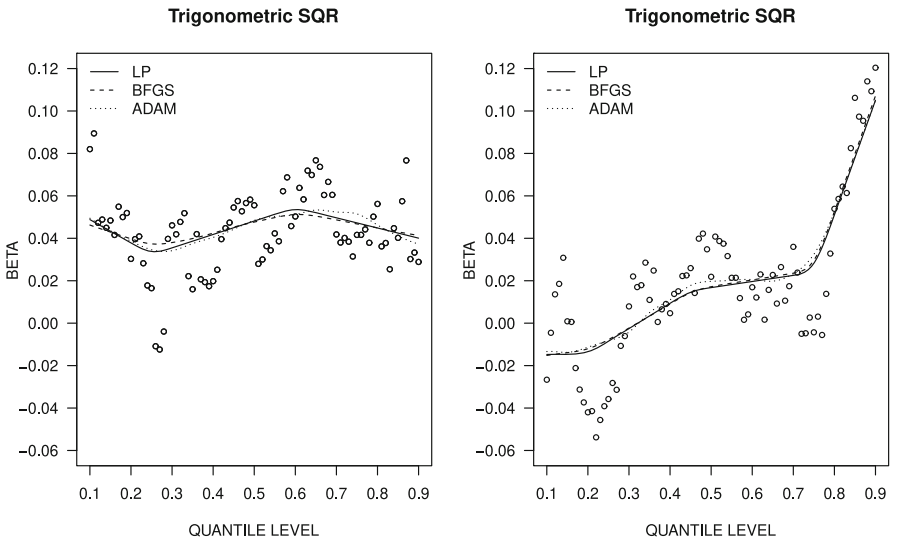

Fig. 9 Coefficients of trigonometric SQR at frequency $f = 102/512$ computed by LP (solid line), BFGS (dashed line), and ADAM (dotted line). Open circles depict the QR solution

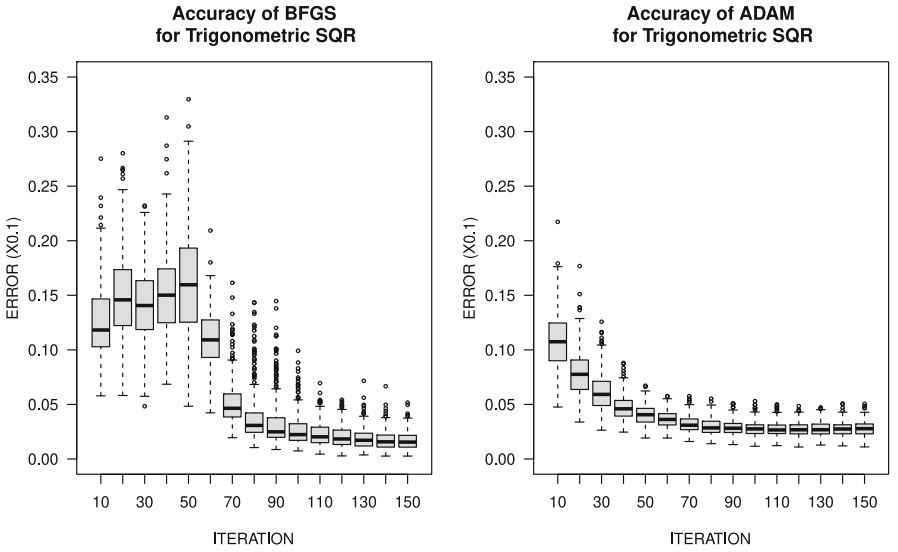

Fig. 10 Boxplot of approximation errors of trigonometric SQR at 255 frequencies versus the number of iterations computed by BFGS (left) and ADAM (right)

error and the objective function after the warm-up iterations. Among the four options in GRAD, option (iii) is most effective in reducing the objective function, whereas option (i) is least effective because it does not shrink the step size when the limited line search fails. In terms of reducing the approximation error, option (ii) turns out to be most effective.

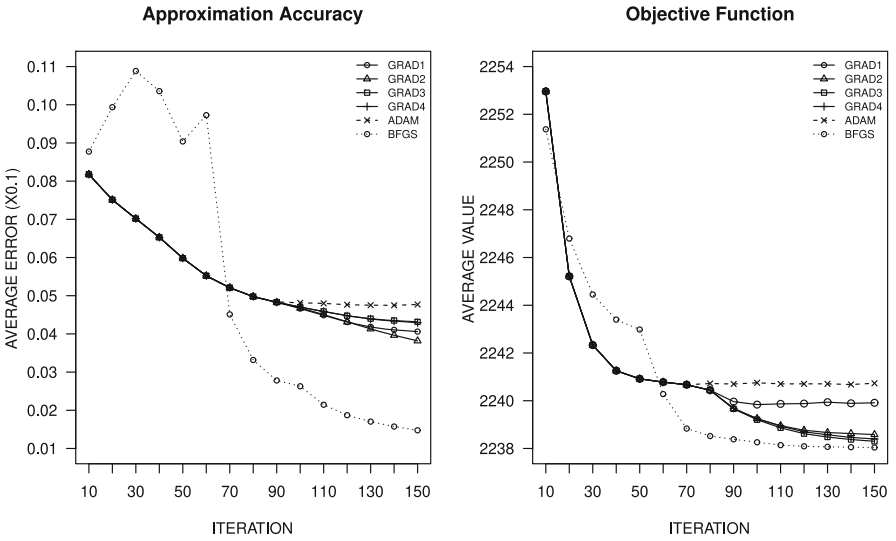

Fig. 11 Average approximation error (left) and average objective function (right) of trigonometric SQR across 255 frequencies versus the number of iterations computed by BFGS, ADAM, and GRAD with 4 options

It is interesting to observe that after 150 iterations the average approximation error of GRAD remains much larger than that of BFGS, although the average objective function of GRAD becomes near that of BFGS. Such discrepancies are often observed in problems where the objective function has a shallow minimum in some variables. With BFGS, possible differences in curvature among variables are equalized by the Hessian matrix. This enables BFGS to move quickly toward the solution with the same step size for all variables. Being a first-order algorithm, neither GRAD nor ADAM possesses this capability. Slow improvement (if any) is expected for these algorithms due to the requirement of small step sizes.

7 Concluding Remarks

In this article, we consider the problem of fitting linear models by quantile regression at multiple quantile levels where the coefficients of regressors are represented by spline functions of the quantile level and penalized to ensure smoothness across quantiles. Using the ℓ_1-norm of the second derivatives as penalty, the resulting spline quantile regression or SQR problem can be reformulated as a linear program (LP) and solved by an interior-point algorithm. The SQR solution complements the ordinary quantile regression solution obtained independently for each quantile level. In additional to a more appealing visualization, an improved estimation accuracy can be achieved by SQR when the underlying functional coefficients are suitably smooth.

The SQR method employs a single parameter to control the smoothness for all coefficients across the entire interval of quantile level. This global treatment can be

improved by incorporating different smoothing parameters on different coefficients and by applying the method locally to different regions of the quantile level. Asymptotic theory of the SQR method should also be developed in future research to facilitate statistical inference. Like any smoothing method, SQR requires the underlying functions to be sufficiently smooth. In our case, they should be well approximated by the splines. However, it remains possible that certain weak local features be confused with noise. Alternative methods such as wavelets are better equipped to handle functions with variable smoothness characteristics. Moreover, the SQR method does not enforce the monotonicity of the conditional quntile function to eliminate the rare possibility of quantile crossing [16, p. 55]. It is conceivable that an inequaity constraint of the form $\mathbf{x}_t^T \boldsymbol{\beta}(\tau_{\ell+1}) \geq \mathbf{x}_t^T \boldsymbol{\beta}(\tau_\ell)$ $(\ell = 1, \ldots, L-1; t = 1, \ldots, n)$ could be introduced to mitigate the problem. This constrained SQR solution would retain the LP nature but increase the computational burden. How to enforce the monotonicity effectively on the entire domain of the regressor for all quantiles remains an interesting topic for future research.

In this article, we also consider three gradient algorithms, BFGS, ADAM, and GRAD, as possible candidates to provide approximations to the LP solution with less computer memory. Our experiments show that it is possible for these gradient algorithms, especially BFGS, to produce reasonably good approximations to the LP solution, but a large number of iterations may be required, especially for ADAM and GRAD. The variable step size in GRAD is more effective than the fixed step size in ADAM. How to improve ADAM and GRAD to achieve a similar performance to BFGS without a significant increase of computational burden remains an interesting problem for future research.

Appendix: R Functions

The following functions are implemented in the R package 'qfa' (version $\geq$ 4.1) available at https://cran.r-project.org and https://github.com/thl2019/QFA.

- `sqr`: a function that computes the SQR solution on a set of quantile levels by the interior-point algorithm of Koenker et al. (1994) with or without user-supplied smoothing parameter `spar`.
- `sqdft`: a function that computes the quantile discrete Fourier transform (QDFT) of time series data based on trigonometric SQR solutions on a set of quantile levels with or without user-supplied smoothing parameter `spar`.
- `qdft2qper`: a function that converts the QDFT produced by `sqdft` into a quantile periodogram (QPER).
- `qfa.plot`: a function that produces a quantile-frequency image plot for a quantile spectrum.
- `tsqr.fit`: a low-level function that computes the trigonometric SQR solution on a set of quantile levels for a given frequency with a given smoothing parameter `spar`.

- `sqr.fit.optim`: a function that computes the SQR solution on a set of quantile levels with a given smoothing parameter `spar` by BFGS, ADAM, or GRAD algorithm.

Acknowledgements The authors would like to thank Dr. R. Koenker for advice on the FORTRAN code of the interior-point algorithm.

References

1. Ando T, Li K-C (2025) Simplex quantile regression without crossing. Ann Stat 53:144–169
2. Andriyana Y, Gijbels I, Verhasselt A (2014) P-splines quantile regression estimation in varying coefficient models. TEST 23:153–194
3. Asl A, Overton M (2020) Analysis of the gradient method with an Armijo-Wolfe line search on a class of non-smooth convex functions. Optimization Methods and Software 35:223–242
4. Belloni A, Chernozhukov V, Chetverikov D, Fernández-Val I (2019) Conditional quantile processes based on series or many regressors. J Econ 213:4–29
5. Berkelaar M (2024) Package 'lpSolve'. https://cran.r-project.org/web/packages/lpSolve/lpSolve.pdf
6. Bondell H, Reich B, Wang H (2010) Noncrossing quantile regression curve estimation. Biometrika 97:825–838
7. Das P, Ghosal S (2018) Bayesian non-parametric simultaneous quantile regression for complete and grid data. Comput Stat Data Anal 127:172–186
8. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, Cambridge, MA
9. Hao M, Lin Y, Shen G, Su W (2023) Nonparametric inference on smoothed quantile regression process. Comput Stat Data Anal 179:107645
10. He X (1997) Quantile curves without crossing. Am Stat 51:186–192
11. He X, Pan X, Tan K, Zhou W (2023) Smoothed quantile regression with large-scale inference. J Econ 232:367–388
12. Kim M-O (2007) Quantile regression with varying coefficients. Ann Stat 35:92–108
13. Kingma D, Ba J (2015) Adam: a method for stochastic optimization. arXiv:1412.6980
14. Koenker R (2005) Quantile Regression. Cambridge University Press, Cambridge
15. Koenker R, Bassett G (1978) Regression quantiles. Econometrica 46:33–50
16. Koenker R, Ng P (2005) A Frisch-Newton algorithm for sparse quantile regression. Acta Math Appl Sin 21:225–236
17. Koenker R, Ng P, Portnoy S (1994) Quantile smoothing splines. Biometrika 81:673–680
18. Lewis A, Overton M (2013) Non-smooth optimization via quasi-Newton methods. Math Program 141:135–163
19. Li T-H (2012) Quantile periodograms. J Am Stat Assoc 107:765–776
20. Li T-H (2014) Time Series with Mixed Spectra. CRC Press, Boca Raton, FL
21. Li T-H (2020) From zero crossings to quantile-frequency analysis of time series with an application to nondestructive evaluation. Applied Stochastic Models for Business and Industry 36:1111–1130
22. Nocedal J, Wright S (2006) Numerical Optimization, 2nd edn. Springer, New York
23. Oh H-S, Lee T, Nychka D (2011) Fast nonparametric quantile regression with arbitrary smoothing methods. J Comput Graph Stat 20:510–526
24. Park S, He X (2017) Hypothesis testing for regional quantiles. J Stat Planning Inference 191:13–24
25. Portnoy S (1991) Asymptotic behavior of the number of regression quantile breakpoints. SIAM J Sci Stat Comput 12:867–883
26. Portnoy S, Koenker R (1997) The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators. Stat Sci 12:279–300
27. Polyak B (1987) Introduction to Optimization, Chapter. 5. New York: Optimization Software, Inc
28. R Core Team (2024) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
29. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747
30. Schnabel S, Eilers P (2013) Simultaneous estimation of quantile curves using quantile sheets. AStA Adv Stat Anal 97:77–87

31. Szendrei T, Bhattacharjee A, Schaffer M (2025) Fused LASSO as non-crossing quantile regression. arXiv:2403.14036v3
32. Wahba G (1975) Smoothing noisy data with spline functions. Numer Math 24:383–393
33. Wu Y, Liu Y (2009) Stepwise multiple quantile regression estimation using non-crossing constraints. Statistics and Its Interface 2:299–310
34. Yoshida T (2021) Quantile function regression and variable selection for sparse models. Canadian J Stat 49:1196–1221