Numer Algor (2016) 71:59–75
DOI 10.1007/s11075-015-9984-7

ORIGINAL PAPER

An efficient generalized least squares algorithm for periodic trended regression with autoregressive errors

Jaechoul Lee · Anthony Dini · William Negri

Received: 2 September 2014 / Accepted: 4 March 2015 / Published online: 14 April 2015

Abstract Time series data with periodic trends like daily temperatures or sales of seasonal products can be seen in periods fluctuating between highs and lows throughout the year. Generalized least squares estimators are often computed for such time series data as these estimators have minimum variance among all linear unbiased estimators. However, the generalized least squares solution can require extremely demanding computation when the data is large. This paper studies an efficient algorithm for generalized least squares estimation in periodic trended regression with autoregressive errors. We develop an algorithm that can substantially simplify generalized least squares computation by manipulating large sets of data into smaller sets. This is accomplished by coining a structured matrix for dimension reduction. Simulations show that the new computation methods using our algorithm can drastically reduce computing time. Our algorithm can be easily adapted to big data that show periodic trends often pertinent to economics, environmental studies, and engineering practices.

Keywords Big data · Dimension reduction · GLS estimation · Model with autocorrelated errors · Periodic trended time series regression

J. Lee (✉)
Department of Mathematics, Boise State University, Boise, ID 83725, USA
e-mail: jaechlee@math.boisestate.edu

A. Dini · W. Negri
Department of Civil Engineering, Boise State University, Boise, ID 83725, USA

1 Introduction

Suppose that we analyze time series data by fitting a linear regression model. The model can be expressed as

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \tag{1}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ is a vector of n responses, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is a $n \times k$ matrix with jth covariate $\mathbf{x}_j = (x_{j1}, \ldots, x_{jn})^{\mathrm{T}}$ for $j \in \{1, \ldots, k\}$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)^{\mathrm{T}}$ is a k-dimensional vector of unknown regression parameters, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^{\mathrm{T}}$ is a n-dimensional vector of zero-mean autocorrelated errors with variance-covariance matrix $\boldsymbol{\Gamma}$. The regression parameter associated with a covariate is often estimated by the generalized least squares (GLS) method. Computing GLS solution is based on a general linear model form. Specifically, the GLS estimator for $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}}$, is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\mathbf{y} \tag{2}$$

with variance

$$\mathrm{var}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\mathbf{X})^{-1}. \tag{3}$$

The GLS estimator $\hat{\boldsymbol{\beta}}$ is the best linear unbiased estimator (BLUE) for $\boldsymbol{\beta}$ if the regression model errors $\{\varepsilon_t\}$ have a second-moment stationary autocorrelation function. Ordinary least squares (OLS) estimators are also often calculated since OLS estimators can be asymptotically BLUE under some mild conditions [4]. OLS estimators do not need $\boldsymbol{\Gamma}$ for their values but require it for their variances. However, most statistical software computes OLS estimator variances by treating the data as being independent, producing biased OLS estimator variances. Also, GLS methods can be more efficient when there is substantial autocorrelation [16]. We do not particularly favor one method over the other but focus in this paper on an efficient GLS algorithm.

In practice, the variance-covariance matrix $\boldsymbol{\Gamma}$ is unknown, and we must estimate $\boldsymbol{\Gamma}$ to find GLS. The resulting estimator with estimated $\boldsymbol{\Gamma}$ is called the feasible generalized least squares (FGLS) estimator. The FGLS estimate is obtained by iterating GLS computation until all the model parameter estimates converge. This GLS method can be computationally demanding when the sample size n is large, mostly because GLS requires the inverse of $n \times n$ matrix $\boldsymbol{\Gamma}$. Some numerical algorithms have been developed to bypass the direct inverse matrix calculation for more efficient GLS. For instance, the Durbin-Levinson algorithm and the Innovations algorithm sequentially compute one-step ahead predictions and their mean square prediction errors in lieu of $\boldsymbol{\Gamma}^{-1}$ [2, 13]. Some GLS algorithms do not require the full-sized $n \times n$ matrix inverse. Lee [7] develops a fast algorithm for GLS estimators and their variances using an end-points matching approach. This algorithm simplifies the GLS computation by inverting a dimension-reduced $\frac{n}{2} \times \frac{n}{2}$ matrix, instead of the $n \times n$ matrix $\boldsymbol{\Gamma}$. These algorithms work fast enough for small or relatively large sets of data but take longer to compute as the data sets get larger with a computation complexity of order at least $O(n^2)$. This could be a computational challenge when GLS is iterated for FGLS and/or when GLS is applied to big data. This could be even more troublesome when the GLS method is applied to unknown multiple changepoint problems where all possible segmentations are considered for any possible changepoint times.

We consider periodic trended regression with first-order autoregressive errors. This regression model is applied in many econometric, environmental, and engineering practices. Our objective is to develop an efficient GLS algorithm for such periodic trended regression models. For this, we propose using a dimension reduction matrix that significantly reduces the dimension of the regression model by compressing replicative periodic patterns in the data. This matrix is made by matching those responses with the identical periodic patterns. Appealingly, our algorithm requires an inverse of $m \times m$ dimensional matrix with $m \ll n$ and m not increasing with n. As a result, unlike other algorithms, our algorithm has a constant computation complexity, not proportional to the sample size n, in computing the inverse matrix.

The rest of this paper proceeds as follows: Section 2 illustrates our algorithm in some fundamental periodic trended regression models in which the matrix inverse dimension for GLS computation is greatly reduced to a small constant. In Section 3, the GLS methods using our algorithm are compared with other existing GLS methods. Section 4 concludes with some remarks.

2 An efficient GLS algorithm

Although some GLS algorithms would reduce a significant amount of computing time in many practices, they still might need very demanding computation if n is big and/or if GLS is iterated to improve accuracy. However, if the errors $\{\varepsilon_t\}$ in a periodic trended regression model are a first-order autoregressive, AR(1), process, we find that the GLS computation complexity can be greatly further reduced. To elaborate, when we fit a time series regression model to a periodic trend plus AR(1) noise data, the model can be expressed as a sum of sinusoidal waves:

$$y_t = \beta_0 + \sum_{i=1}^{k} \alpha_{i1} \cos(2\pi t/T_i + \alpha_{i2}) + \varepsilon_t,$$

where β_0 is the overall mean of $\{y_t\}$, α_{i1} is the amplitude associated with period T_i, and α_{i2} is a phase shift which is not necessarily zero in actual data. We assume that T_i's are known. Use of a trigonometric identity reexpresses this model as a linear model in terms of all regression parameters:

$$y_t = \beta_0 + \sum_{i=1}^{k} \{\beta_{i1} \cos(2\pi t/T_i) + \beta_{i2} \sin(2\pi t/T_i)\} + \varepsilon_t,$$

where $\beta_{i1} = \alpha_{i1} \cos(\alpha_{i2})$ and $\beta_{i2} = -\alpha_{i1} \sin(\alpha_{i2})$. We illustrate our GLS algorithm in the following four periodic trended regression models.

Model 1 We first consider simple periodic trended regression with AR(1) errors:

$$y_t = \beta_0 + \beta_1 \cos(2\pi t/T) + \varepsilon_t \tag{4}$$

for $t \in \{1, \ldots, n\}$ with $n = CT - 1$. Here y_t is the observed value at time t with period T, $\{\varepsilon_t\}$ are a stationary AR(1) error series in the sense that $\varepsilon_t = \phi\varepsilon_{t-1} + Z_t$,

$|\phi| < 1$, with $\{Z_t\}$ being a zero-mean white noise series with variance σ^2, and C is the number of cycles in $\{y_t\}$.

We illustrate our algorithm when the data has a period of $T = 12$ (equivalent to monthly data). For data with a different period, the same approach can be used. The GLS estimator for $\boldsymbol{\beta} = (\beta_0, \beta_1)^{\mathrm{T}}$ in (4) can be written as:

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^{\mathrm{T}} = \left(\sum_{t=1}^{n} w_{0t} y_t, \sum_{t=1}^{n} w_{1t} y_t \right)^{\mathrm{T}}. \tag{5}$$

The weights $\{w_{0t}\}$ and $\{w_{1t}\}$ for the GLS estimator $\hat{\boldsymbol{\beta}}$ on $\{y_t\}$ are dependent on the AR parameter ϕ and the covariate $\cos(2\pi t/T)$. An explicit expression for these GLS weights in (5) is presented in Appendix. Figure 1 displays these weights over the corresponding time t when $C = 3$ for simplicity. Interestingly, as the weights $\{w_{0t}\}$ and $\{w_{1t}\}$ periodically change, those responses with the same weights can be merged, subsequently producing equivalent GLS results. For $C = 3$, matching such periodic patterns in $\{w_{0t}\}$ and $\{w_{1t}\}$ gives us the following $8 \times n$ dimensional matrix:

$$\mathbf{D} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}.$$

We call $\mathbf{D}$ a matching matrix as the column element in $\mathbf{D}$ signifies whether the weight w_{0t} (or w_{1t}) is identical (coded as 1) or not (coded as 0) for time t.

Extension of $\mathbf{D}$ to larger cycles is straightforward. To specify $\mathbf{D}$ in a general case with $C \in \{3, 4, \ldots\}$, we first define the following two sub-pattern matrices:

$$\mathbf{D}^* = (\mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_6, \mathbf{e}_5, \mathbf{e}_4, \mathbf{e}_3, \mathbf{e}_2), \quad \mathbf{D}_* = (\mathbf{e}_7, \mathbf{e}_8, \mathbf{e}_7),$$

where $\mathbf{e}_i$ is an 8-dimensional column vector having one in the ith element and zeros in the other elements for $i \in \{1, \ldots, 8\}$. For a general C cycle, we then simply repeat $\mathbf{D}^*$ and $\mathbf{D}_*$ by $C - 1$ times to obtain the following $8 \times n$ matching matrix:

$$\mathbf{D} = (\mathbf{e}_1, \mathbf{D}^*, \mathbf{D}_*, \mathbf{D}^*, \cdots, \mathbf{D}_*, \mathbf{D}^*, \mathbf{e}_1).$$

Next, we multiply both sides of the regression model (1) by $\mathbf{D}$, resulting in a dimension-reduced regression model:

$$\mathbf{y}_{\mathrm{r}} = \mathbf{X}_{\mathrm{r}} \boldsymbol{\beta} + \boldsymbol{\varepsilon}_{\mathrm{r}}. \tag{6}$$

Note that $\mathbf{y}_{\mathrm{r}} = \mathbf{D}\mathbf{y}$ is 8×1, $\mathbf{X}_{\mathrm{r}} = \mathbf{D}\mathbf{X}$ is 8×2, and $\boldsymbol{\varepsilon}_{\mathrm{r}} = \mathbf{D}\boldsymbol{\varepsilon}$ is 8×1. Interestingly, this dimension-reduced model (6) has a similar expression to the linear model (1) but consists of *only* 8-dimensional, *not* n-dimensional, column vectors. The dimension-reduced response vector $\mathbf{y}_{\mathrm{r}}$ has then the compressed 8×8 variance-covariance matrix:

$$\boldsymbol{\Gamma}_{\mathrm{r}} = \mathrm{var}(\mathbf{y}_{\mathrm{r}}) = \mathbf{D}\boldsymbol{\Gamma}\mathbf{D}^{\mathrm{T}}.$$

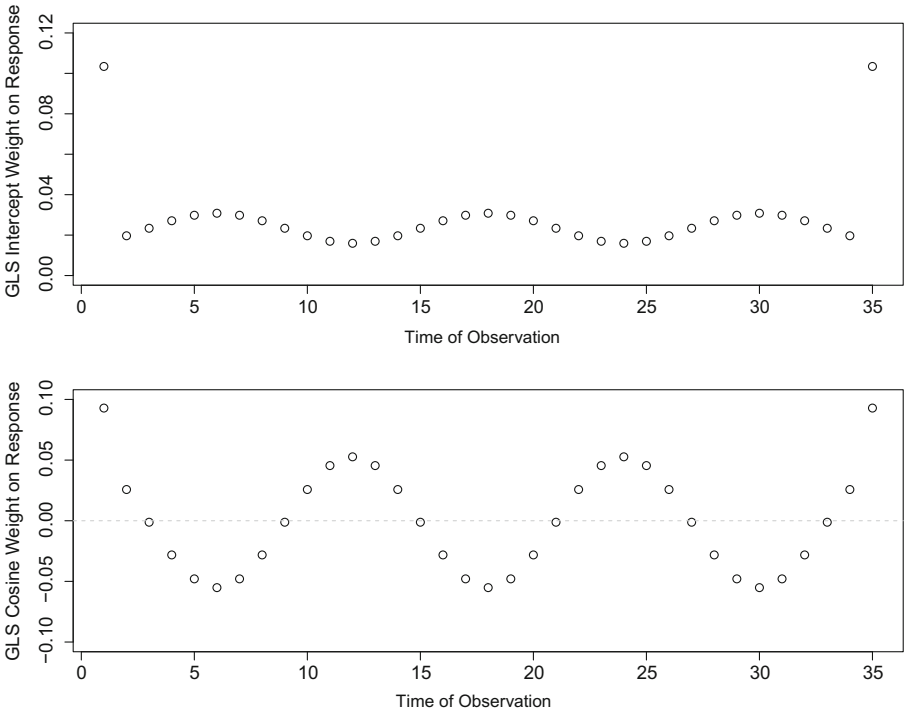

Fig. 1 The weight of the GLS intercept parameter β_0 estimate (*top*) and weight of GLS covariate parameter β_1 estimate (*bottom*) on the response series in Model 1 with AR parameter $\phi = 0.8$

The resulting GLS estimator from the model (6) is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}_{\mathrm{r}}^{\mathrm{T}}\boldsymbol{\Gamma}_{\mathrm{r}}^{-1}\mathbf{X}_{\mathrm{r}})^{-1}\mathbf{X}_{\mathrm{r}}^{\mathrm{T}}\boldsymbol{\Gamma}_{\mathrm{r}}^{-1}\mathbf{y}_{\mathrm{r}} \tag{7}$$

with variance

$$\mathrm{var}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}_{\mathrm{r}}^{\mathrm{T}}\boldsymbol{\Gamma}_{\mathrm{r}}^{-1}\mathbf{X}_{\mathrm{r}})^{-1}. \tag{8}$$

The dimension-reduced GLS expressions in (7) and (8) produce identical results with the original GLS expressions in (2) and (3).

Note that the matching matrix $\mathbf{D}$ is used to reduce our effort in GLS computation. The number of rows in $\mathbf{D}$ is only 8 for $T = 12$, not proportional to n. Multiplying the $8 \times n$ dimensional $\mathbf{D}$ to $\mathbf{y}$, $\mathbf{X}$, and $\boldsymbol{\Gamma}$ (for $\boldsymbol{\Gamma}_{\mathrm{r}} = \mathbf{D}\boldsymbol{\Gamma}\mathbf{D}^{\mathrm{T}}$) and then inverting only 8×8 dimensional matrix $\boldsymbol{\Gamma}_{\mathrm{r}}$ comprise our computational cost to obtain the equivalent GLS estimator and variance in (7) and (8). This new algorithm is substantially more efficient than directly inverting $n \times n$ dimensional $\boldsymbol{\Gamma}$ as n gets larger. We also emphasize that the matrix inverse dimension for GLS computation is a small constant in our algorithm for any $C \in \{2, 3, \ldots\}$, implying that the inverse computation complexity via our algorithm is *always* constant and *not* monotonically increasing with sample size n unlike other existing GLS methods. The matching matrix $\mathbf{D}$ for different values of n can be found in the same manner. We present the matching matrices for $n = CT$ and $n = CT + 1$ in Appendix for further illustration.

Model 2 We consider the following periodic regression model with AR(1) errors:

$$y_t = \beta_0 + \beta_1 \sin(2\pi t/T) + \varepsilon_t$$

for $t \in \{1, \ldots, n\}$ with $n = CT + T/2 - 1$. We use $T = 12$ for illustration purposes. The GLS estimator can be expressed as (5). Finding an explicit expression for the weights $\{w_{0t}\}$ and $\{w_{1t}\}$ can be laborious. However, $\{w_{0t}\}$ and $\{w_{1t}\}$ for this model also show a periodic pattern as depicted in Fig. 2 when $C = 3$. Use of this periodic pattern leads to the following matching matrix:

$$\mathbf{D} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0
\end{pmatrix}.$$

Extension to a general cycle $C \in \{3, 4, \ldots\}$ is simple. We expand $\mathbf{D}$ by repeatedly inserting $\mathbf{D}^*$ and $\mathbf{D}_*$ for C times:

$$\mathbf{D} = (\mathbf{e}_1, \mathbf{D}_*, \mathbf{D}^*, \mathbf{D}_*, \cdots, \mathbf{D}^*, \mathbf{D}_*, \mathbf{e}_1),$$

where $\mathbf{e}_i$ is an 8-dimensional column vector having one in the ith element and zeros in the other elements for $i \in \{1, \ldots, 8\}$, and $\mathbf{D}^*$ and $\mathbf{D}_*$ are identical to those of Model 1. After multiplying $\mathbf{D}$ to $\mathbf{y}$, $\mathbf{X}$, and $\mathbf{\Gamma}$ to obtain the dimension-reduced $\mathbf{y}_\mathrm{r}$, $\mathbf{X}_\mathrm{r}$, and $\mathbf{\Gamma}_\mathrm{r}$, we compute the GLS estimator and variance using the expressions in (7) and (8).

Our algorithm requires only a few simple steps: (i) multiplying the matching matrix $\mathbf{D}$ to the response vector $\mathbf{y}$ and the covariate matrix $\mathbf{X}$, (ii) computing the compressed 8×8 variance-covariance matrix $\mathbf{\Gamma}_\mathrm{r}$, and (iii) inverting $\mathbf{\Gamma}_\mathrm{r}$ for GLS results. For different periods and sample sizes, our algorithm can be similarly applied to produce equivalent results.

Model 3 Consider another periodic trended regression model with AR(1) errors:

$$y_t = \beta_0 + \beta_1 \cos(2\pi t/T) + \beta_2 \sin(2\pi t/T) + \varepsilon_t$$

for $t \in \{1, \ldots, n\}$ with $n = CT + T/6$. The periodicity of $T = 12$ is used again for our illustration. The GLS estimator can be expressed as:

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)^\mathrm{T} = \left(\sum_{t=1}^{n} w_{0t} y_t, \sum_{t=1}^{n} w_{1t} y_t, \sum_{t=1}^{n} w_{2t} y_t \right)^\mathrm{T}.$$

The weight $\{w_{0t}\}$ behaves similarly to those of Model 1 and Model 2. The weights $\{w_{1t}\}$ and $\{w_{2t}\}$ also vary periodically. Figure 3 shows these periodic patterns in $\{w_{1t}\}$ and $\{w_{2t}\}$ for $C = 3$. Interestingly, these weights are symmetric to each other:

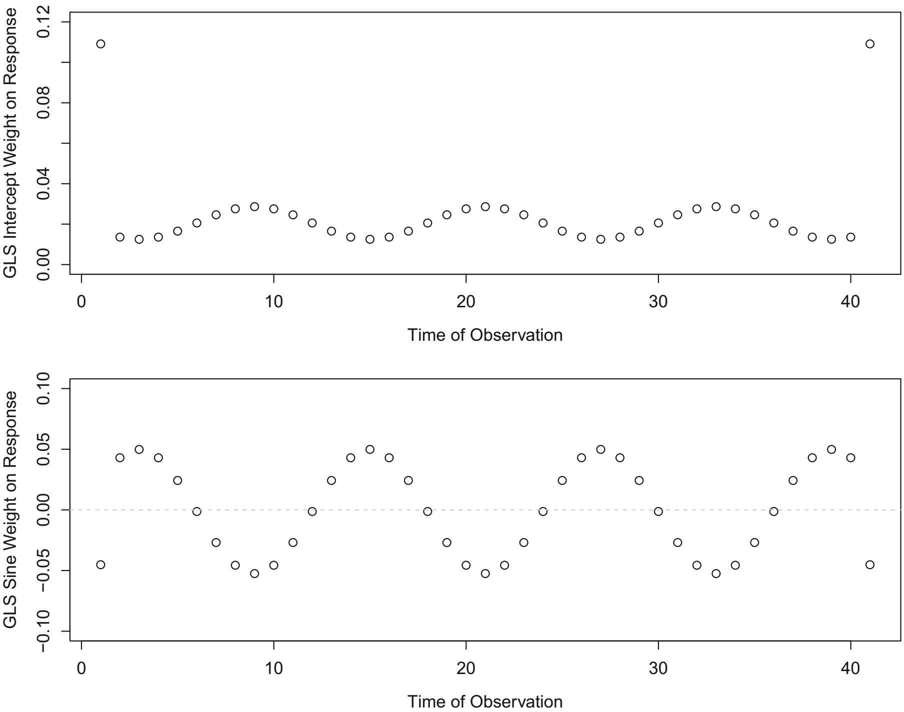

Fig. 2 The weight of the GLS intercept parameter β_0 estimate (*top*) and weight of GLS covariate parameter β_1 estimate (*bottom*) on the response series in Model 2 with AR parameter $\phi = 0.8$

$w_{2,t} = w_{1,n+1-t}$. Using this symmetric property, we need to coin a matching matrix associated with either of $\{w_{1t}\}$ or $\{w_{2t}\}$, resulting in the following:

$$\mathbf{D} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

For larger cycles, we just need to repeat the following pattern matrix $\mathbf{E}$ by C times to obtain $\mathbf{D}$:

$$\mathbf{D} = (\mathbf{e}_1, \mathbf{E}, \cdots, \mathbf{E}, \mathbf{e}_{14}), \qquad \mathbf{E} = (\mathbf{e}_2, \cdots, \mathbf{e}_{13}). \tag{9}$$

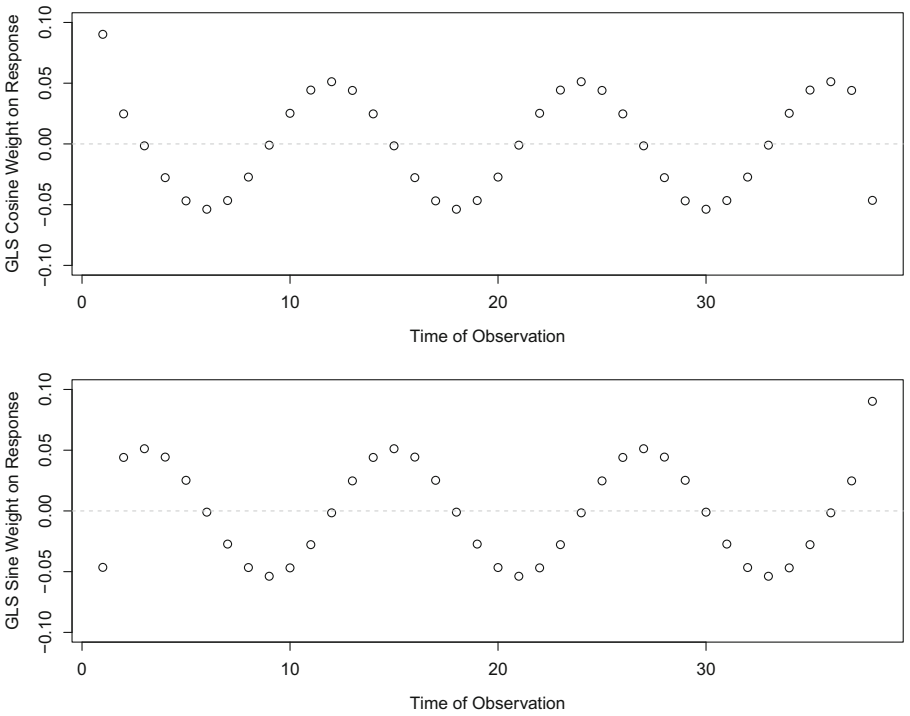

Fig. 3 The weight of the GLS cosine parameter β_1 estimate (*top*) and weight of GLS sine parameter β_2 estimate (*bottom*) on the response series in Model 3 with AR parameter $\phi = 0.8$

Here, $\mathbf{e}_i$ is a 14-dimensional column vector with one in the ith element and zero in the others for $i \in \{1, \ldots, 14\}$. Therefore, the matching matrix $\mathbf{D}$ has a dimension of $14 \times n$. We do not need a separate matching matrix for the intercept weight $\{w_{0t}\}$ as this matrix is in fact linearly dependent on $\mathbf{D}$.

Appreciably, the $14 \times n$ dimensional matching matrix $\mathbf{D}$ in (9) is also applicable to Model 1 and Model 2 for the sample size $n = CT + T/6$. The resulting compressed variance-covariance matrix $\mathbf{\Gamma}_{\mathrm{r}}$ has a dimension of 14×14, which is slightly larger than the 8×8 variance-covariance matrix in Model 1 and Model 2, but this small increase in dimension does not add any meaningful complexity in GLS computation.

Model 4 We consider a multiple periodic trended regression model with AR(1) errors:

$$y_t = \beta_0 + \sum_{i=1}^{2} \{\beta_{i1} \cos(2\pi t/T_i) + \beta_{i2} \sin(2\pi t/T_i)\} + \varepsilon_t \tag{10}$$

for $t \in \{1, \ldots, n\}$ with $n = CT_1 + T_1/6$. This model has two different periods at T_1 and T_2. We assume that C is an even integer. The GLS estimator can be expressed as:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\hat{\beta}_0, \hat{\beta}_{11}, \hat{\beta}_{12}, \hat{\beta}_{21}, \hat{\beta}_{22})^{\mathrm{T}} \\ &= \left(\sum_{t=1}^{n} w_{0t} y_t, \sum_{t=1}^{n} w_{11t} y_t, \sum_{t=1}^{n} w_{12t} y_t, \sum_{t=1}^{n} w_{21t} y_t, \sum_{t=1}^{n} w_{22t} y_t \right)^{\mathrm{T}}.\end{aligned}$$

Figure 4 displays the weights $\{w_{11t}\}$, $\{w_{12t}\}$, $\{w_{21t}\}$, and $\{w_{22t}\}$ when $C = 6$, $T_1 = 12$, and $T_2 = 24$ for our illustration. These weights exhibit a periodic pattern, similar to that of Model 1, Model 2, and Model 3. Using the same approach used in the previous models, we obtain the matching matrix as follows:

$$\mathbf{D} = (\mathbf{e}_1, \mathbf{F}, \cdots, \mathbf{F}, \mathbf{e}_{26}), \qquad \mathbf{F} = (\mathbf{e}_2, \cdots, \mathbf{e}_{25}). \tag{11}$$

Here, $\mathbf{e}_i$ is a 26-dimensional column vector with one in the ith element and zeros in the others for $i \in \{1, \ldots, 26\}$. The 26×24 dimensional pattern matrix $\mathbf{F}$ is repeated $C/2$ times to form the matching matrix $\mathbf{D}$, setting $\mathbf{D}$ to be $26 \times n$ dimensional. The inverse of $\boldsymbol{\Gamma}_{\mathrm{r}} = \mathbf{D}\boldsymbol{\Gamma}\mathbf{D}^{\mathrm{T}}$ should be comfortably calculable at a very fast speed without any numerical issues. Again, we do not need to include a separate matching matrix for $\{w_{0t}\}$ as this matrix is linearly dependent on $\mathbf{D}$.

Our algorithm can be easily adapted to the general case that there are g periods $T_1, \ldots, T_g$ in the multiple periodic trended regression with AR(1) errors in (10). We only need to configure the matching matrix $\mathbf{D}$, akin to the matrix (11), from a smaller sample size and extend this matching matrix to the full sample size. Multiplying this $\mathbf{D}$ to $\mathbf{y}$ and $\mathbf{X}$, and computing $\boldsymbol{\Gamma}_{\mathrm{r}} = \mathbf{D}\boldsymbol{\Gamma}\mathbf{D}^{\mathrm{T}}$ and its inverse $\boldsymbol{\Gamma}_{\mathrm{r}}^{-1}$ are all computational tasks to obtain the GLS results in (7) and (8). These matrix calculations are very straightforward; there are no computational issues and/or numerical inaccuracies in any of these matrix operations. Recall that the dimension of $\boldsymbol{\Gamma}_{\mathrm{r}}$ is very small ($\ll n$); inverting $\boldsymbol{\Gamma}_{\mathrm{r}}$ is not an issue.

3 Numerical comparisons

This section compares our GLS computation methods with other methods: GLS via a direct inverse, GLS via the Cholesky decomposition, and GLS via the Durbin-Levinson algorithm. To be specific, we briefly describe the other three methods below.

First, the GLS method via a direct inverse uses the expression in (2). This method directly computes $\boldsymbol{\Gamma}^{-1}$ and uses this inverse matrix to compute GLS estimates. Second, the GLS method via the Cholesky decomposition decomposes the variance-covariance matrix $\boldsymbol{\Gamma} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ and then multiplies the $n \times n$ dimensional $\mathbf{Q}^{-1}$ by $\mathbf{y}$ and $\mathbf{X}$, rewriting the linear model (1) as follows [5, pp. 329–330]:

$$\mathbf{y}_{\mathrm{q}} = \mathbf{X}_{\mathrm{q}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_{\mathrm{q}}, \tag{12}$$

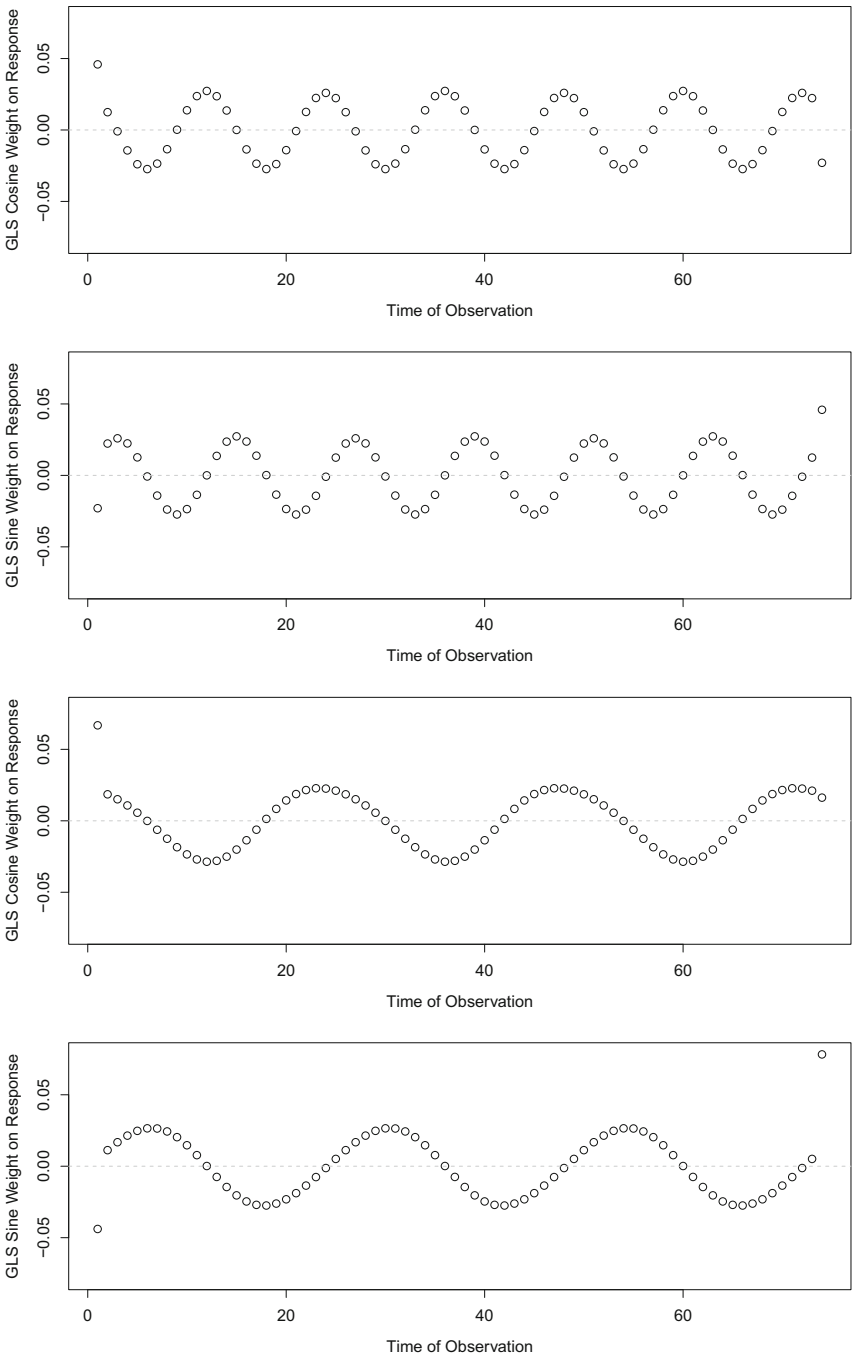

Fig. 4 The GLS parameter estimate weights on the response variable in Model 4 with AR parameter $\phi = 0.8$: period-12 cosine (*first*), period-12 sine (*second*), period-24 cosine (*third*), and period-24 sine (*last*)

where $\mathbf{y}_\mathrm{q} = \mathbf{Q}^{-1}\mathbf{y}$, $\mathbf{X}_\mathrm{q} = \mathbf{Q}^{-1}\mathbf{X}$, and $\boldsymbol{\varepsilon}_\mathrm{q} = \mathbf{Q}^{-1}\boldsymbol{\varepsilon}$. As $\boldsymbol{\varepsilon}_\mathrm{q}$ is a vector of independent errors (note that $\mathrm{var}(\boldsymbol{\varepsilon}_\mathrm{q}) = \mathbf{Q}^{-1}\boldsymbol{\Gamma}\mathbf{Q}^{-1\mathrm{T}} = \mathbf{Q}^{-1}\mathbf{Q}\mathbf{Q}^\mathrm{T}\mathbf{Q}^{\mathrm{T}-1} = \mathbf{I}$), the GLS estimates are equivalent to the OLS estimates for the model (12), giving us:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}_\mathrm{q}^\mathrm{T}\mathbf{X}_\mathrm{q})^{-1}\mathbf{X}_\mathrm{q}^\mathrm{T}\mathbf{y}_\mathrm{q} = (\mathbf{X}^\mathrm{T}\mathbf{Q}^{-1\mathrm{T}}\mathbf{Q}^{-1}\mathbf{X})^{-1}\mathbf{X}^\mathrm{T}\mathbf{Q}^{-1\mathrm{T}}\mathbf{Q}^{-1}\mathbf{y}.$$

This Cholesky decomposition GLS method can be faster than the direct inverse GLS method. A caveat for this Cholesky GLS method is, however, that it still requires computing an inverse matrix of $n \times n$ dimensional matrix $\mathbf{Q}$. Third, the GLS method via the Durbin-Levinson algorithm directly finds $n \times n$ dimensional matrix $\mathbf{P}$ akin to $\mathbf{Q}^{-1}$ as in (12) by the Durbin-Levinson algorithm [2, pp. 169–170]. To elaborate, we predict the error ε_{t+1} by the one-step ahead best linear predictor, denoted by $\hat{\varepsilon}_{t+1} = P(\varepsilon_{t+1}|\varepsilon_t, \ldots, \varepsilon_1)$, as follows: begin with the start-up value $\hat{\varepsilon}_1 = 0$ and compute

$$\hat{\varepsilon}_{t+1} = \sum_{j=1}^{t} \phi_{t,j}\varepsilon_{t+1-j}$$

for $t \in \{1, \ldots, n-1\}$. The mean squared prediction error for $\hat{\varepsilon}_{t+1}$ is denoted by $v_t = E[(\varepsilon_{t+1} - \hat{\varepsilon}_{t+1})^2]$. The terms $\{\phi_{t,j}\}$ and $\{v_t\}$ can be expressed as a simple form for AR(1) autocorrelation, but we consider the Durbin-Levinson algorithm for general stationary $\{\varepsilon_t\}$ as this general setting is coded in most software packages. Specifically, the Durbin-Levinson algorithm recursively computes $\{\phi_{t,j}\}$ and $\{v_t\}$ as follows: begin with the start-up values $v_0 = \gamma(0)$ and $\phi_{1,1} = \gamma(1)/\gamma(0)$ and find

$$\begin{aligned}
\phi_{t,t} &= v_{t-1}^{-1}\left(\gamma(t) - \sum_{j=1}^{t-1}\phi_{t-1,j}\gamma(n-j)\right),\\
\phi_{t,j} &= \phi_{t-1,j} - \phi_{t,t}\phi_{t-1,t-j}, \quad j \in \{1, \ldots, t-1\},\\
v_t &= v_{t-1}(1-\phi_{t,t}^2),
\end{aligned}$$

where $\gamma(h) = \mathrm{cov}(\varepsilon_t, \varepsilon_{t+h})$ for $h \in \{0, \ldots, n-1\}$. These calculations are then repeated for each successive t. Using these $\{\phi_{t,j}\}$ and $\{v_t\}$, we construct

$$\mathbf{P} = \begin{pmatrix}
\frac{1}{\sqrt{v_0}} & 0 & 0 & \cdots & 0 & 0\\
\frac{-\phi_{1,1}}{\sqrt{v_1}} & \frac{1}{\sqrt{v_1}} & 0 & \cdots & 0 & 0\\
\frac{-\phi_{2,2}}{\sqrt{v_2}} & \frac{-\phi_{2,1}}{\sqrt{v_2}} & \frac{1}{\sqrt{v_2}} & \cdots & 0 & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\
\frac{-\phi_{n-2,n-2}}{\sqrt{v_{n-2}}} & \frac{-\phi_{n-2,n-3}}{\sqrt{v_{n-2}}} & \frac{-\phi_{n-2,n-4}}{\sqrt{v_{n-2}}} & \cdots & \frac{1}{\sqrt{v_{n-2}}} & 0\\
\frac{-\phi_{n-1,n-1}}{\sqrt{v_{n-1}}} & \frac{-\phi_{n-1,n-2}}{\sqrt{v_{n-1}}} & \frac{-\phi_{n-1,n-3}}{\sqrt{v_{n-1}}} & \cdots & \frac{-\phi_{n-1,1}}{\sqrt{v_{n-1}}} & \frac{1}{\sqrt{v_{n-1}}}
\end{pmatrix}$$

and multiply this $\mathbf{P}$ to the linear model (1) to obtain:

$$\mathbf{y}_\mathrm{p} = \mathbf{X}_\mathrm{p}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_\mathrm{p}, \tag{13}$$

where $\mathbf{y}_\mathrm{p} = \mathbf{P}\mathbf{y}$, $\mathbf{X}_\mathrm{p} = \mathbf{P}\mathbf{X}$, and $\boldsymbol{\varepsilon}_\mathrm{p} = \mathbf{P}\boldsymbol{\varepsilon}$. Note that the tth element in $\boldsymbol{\varepsilon}_\mathrm{p}$ is a standardized prediction error $(\varepsilon_t - \hat{\varepsilon}_t)/\sqrt{v_{t-1}}$ for $t \in \{1, \ldots, n\}$. These prediction errors

are mutually independent by the projection theorem [2, pp. 53–54], and therefore the GLS estimates are equivalent to the OLS estimates for the model (13):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}_{\mathrm{p}}^{\mathrm{T}}\mathbf{X}_{\mathrm{p}})^{-1}\mathbf{X}_{\mathrm{p}}^{\mathrm{T}}\mathbf{y}_{\mathrm{p}} = (\mathbf{X}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{y}.$$

Note that this technique does not require any large matrix inversion for the GLS estimates.

We now consider the running time of our algorithm. For this, we simulated time series from Model 4 with $\beta_0 = 10$, $\beta_{11} = 1$, $\beta_{12} = 1.5$, $\beta_{21} = 0.5$, $\beta_{22} = 2$, $\phi = 0.8$, and $\sigma^2 = 1$. Simulation results will be invariant to these parameter selections. GLS regression parameter estimators depend on the AR(1) parameter ϕ and the white noise variance σ^2 associated with $\{\varepsilon_t\}$. These error parameters must be estimated in practice. For this, we iterate a recursive two-stage method to convergence to obtain optimal GLS estimates for $\boldsymbol{\beta} = (\beta_0, \beta_{11}, \beta_{12}, \beta_{21}, \beta_{22})^{\mathrm{T}}$. Elaborating, OLS estimates for $\boldsymbol{\beta}$ are initially calculated. Residuals from this OLS fit, denoted by r_t, are computed by $r_t = y_t - \hat{\beta}_0 - \sum_{i=1}^{2}(\hat{\beta}_{i1}\cos(2\pi t/T_i) + \hat{\beta}_{i2}\sin(2\pi t/T_i))$. We then estimate ϕ by the lag-1 sample autocorrelation from the residual series $\{r_t\}$ and σ^2 by $(1-\hat{\phi}^2)\sum_{t=1}^{n} r_t^2/n$. Using these error parameter estimates, we estimate the variance-covariance matrix $\boldsymbol{\Gamma}$, and then compute the GLS regression parameter estimates. These GLS estimates are used to obtain the residual series of this GLS fit, and new estimates for ϕ and σ^2 are calculated. These procedures are iterated recursively until convergence meets a relative tolerance of 0.0001.

All simulations were run using a PC with 2.7 GHz Intel Core i5, and 8 GB 1600 MHz DDR3 under OS X 10.8.5. R (version 3.0.1) was used. We considered various cycles $C \in \{100, 200, \ldots, 1400\}$ or equivalently the corresponding sample sizes $n \in \{1202, 2402, \ldots, 16802\}$. Each cycle (sample size) is repeated ten times to obtain accurate running times. As all GLS methods considered in our simulation, excluding the Durbin-Levinson algorithm GLS method, requires a laborious construction of the $n \times n$ variance-covariance matrix $\boldsymbol{\Gamma}$, we employed a truncation technique without a substantial loss of accuracy. Specifically, the autocovariances $\{\gamma(0), \gamma(1), \ldots, \gamma(n-1)\}$ are truncated to $\{\gamma(0), \gamma(1), \ldots, \gamma(n_\tau - 1), 0, \ldots, 0\}$ for a large value of n_τ. We used $n_\tau = 200$. Applying this truncation technique to each GLS method, except the Durbin-Levinson algorithm GLS method, produces seven different methods: GLS with full autocovariance via a direct inverse (FINV), GLS with truncated autocovariance via a direct inverse (TINV), GLS with full autocovariance via Cholesky decomposition (FCHO), GLS with truncated autocovariance via Cholesky decomposition (TCHO), GLS via the Durbin-Levinson algorithm (DLA), our matching GLS with full autocovariance (FMAT), and our matching GLS with truncated autocovariance (TMAT).

All methods in our simulation study gave us approximately identical GLS estimates, but the running times are quite different. Figure 5 summarizes running times for these seven GLS methods. As expected, the GLS methods (FINV, TINV, FCHO, and TCHO) based on a direct inverse and the Cholesky decomposition get drastically slower as n gets larger. The Durbin-Levinson algorithm GLS method (DLA) runs fast but experiences an increasingly slow performance for large n. Appealingly, GLS methods (FMAT and TMAT) using our algorithm are fastest; the truncation technique makes the computation run slightly faster with a more noticeable difference for larger

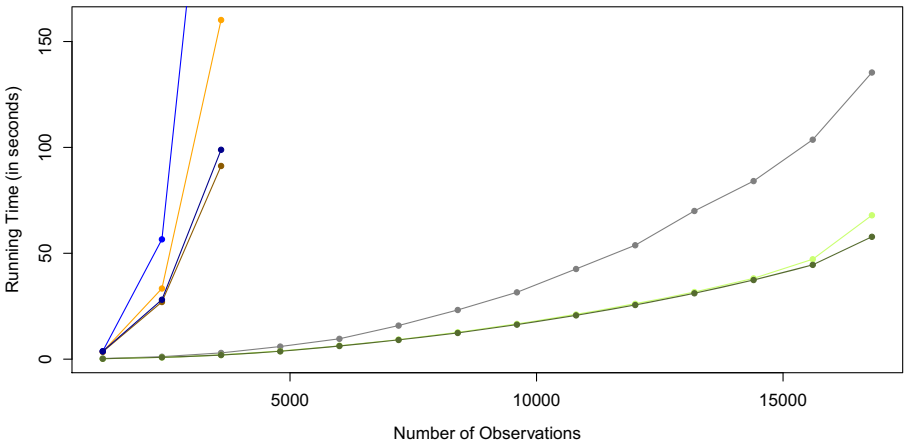

Fig. 5 Comparison of GLS computation running times: FINV (*blue*), TINV (*dark blue*), FCHO (*orange*), TCHO (*dark orange*), DLA (*grey*), FMAT (*green*), and TMAT (*dark green*)

n. We also comment that unlike other methods, our methods FMAT and TMAT run closely at a constant rate as n increases.

The theoretical numbers of numerical operations in these methods also can be compared. As the dominating part of computation in FINV is $\boldsymbol{\Gamma}^{-1}$, FINV has a computation complexity of order $O(n^3)$. Similarly, FCHO is of $O(n^3)$ as the Cholesky decomposition involves $O(n^3)$ computations. The DLA method needs approximately $n^2 + O(n)$ additions/subtractions and $n^2 + O(n)$ multiplications in the Durbin-Levinson algorithm, $n^2/2 + O(n)$ additions and $n^2/2 + O(n)$ multiplications in $\mathbf{Py}$, and $5n^2/2 + O(n)$ additions and $5n^2/2 + O(n)$ multiplications in $\mathbf{PX}$, resulting in a total complexity of $8n^2 + O(n)$. On the other hand, FMAT requires less complexity as follows: the dominating part in FMAT is $\mathbf{D}\boldsymbol{\Gamma}\mathbf{D}^{\mathrm{T}}$ which involves $2n^2 + O(n)$ additions. Note that in FMAT, $\mathbf{Dy}$ and $\mathbf{Dy}$ are of order $O(n)$, which does not add significant complexity in computation. In addition, FMAT requires $(\mathbf{D}\boldsymbol{\Gamma}\mathbf{D}^{\mathrm{T}})^{-1}$, but this nearly does not increase complexity. As a result, FMAT is of order $2n^2 + O(n)$, about a quarter of the DLA method. Figure 5 also verifies these complexities.

4 Concluding remarks

We develop an efficient GLS algorithm for periodic trended regression with first-order autoregressive errors. For each model considered in this paper, we find a periodic matching matrix that drastically reduces the dimension of the linear model into a small constant dimension, not dependent on sample size unlike existing GLS estimation algorithms. The new algorithm is similar to the real-valued fast Fourier transform (FFT) algorithms in that these FFT algorithms use Hermitian symmetry pertinent to the real-valued data at every stage to remove redundant computations of repetitive parts [1, 14]. However, our algorithm uses

specific periodic pattern in GLS estimators in one single step to reduce computation time. This paper considered two periods, $T \in \{12, 24\}$, but our algorithm can be easily adapted to different periods. Our algorithm should be greatly appreciated if the data are big or the GLS computation is repeated in computationally demanding procedures, including the FGLS estimation in unknown regression error parameter settings and/or unknown multiple changepoint estimation problems via the Generic Algorithm [8]. Our method can be applied to estimate periodic trends of time series data in diverse applications, including signal processing [6], two-photon imaging and fMRI data [9, 12], microarray data [15], climatic time series [10], environmental data [11, pp. 243–246], and economic seasonal data [3].

There are a few things that can make our algorithm more useful in practice. We considered the AR(1) error model in this study, but more general error models, including a stationary autoregressive moving-average process, can be studied. We can also consider our algorithm for periodic models with stochastic periodicity [3], including periodic trended models with seasonal AR errors or periodic trended models with periodic AR errors, to accurately model more complicated periodic autocorrelations in actual data. Further study is needed for this possibility. One issue in our algorithm is somewhat increasing running times for very large n. This could be addressed if the compressed variance-covariance matrix $\mathbf{\Gamma}_{\mathrm{r}}$ is more efficiently calculated. Specifically, we calculated this matrix by $\mathbf{\Gamma}_{\mathrm{r}} = \mathbf{D}\mathbf{\Gamma}\mathbf{D}^{\mathrm{T}}$, which can require laborious matrix operations. This calculation seems to be replaceable by a more efficient method that simplifies the matrix operations by using either periodic pattern in $\mathbf{D}$ or Toeplitz structure in $\mathbf{\Gamma}$. With this simplification, our algorithm can run faster with a more constant increasing rate of running times as n gets larger.

Acknowledgments Lee's research was supported by NSF Grant DMS 1107225. Dini's and Negri's research was supported by NSF Grant DUE 0856815. The authors thank the two anonymous referees for their valuable comments that significantly improved this paper.

Conflict of interests The authors declare that they have no conflict of interest.

Appendix

Explicit expression for (5) We obtain an explicit expression of the GLS weights in (5) when $C = 3$ for simplicity. For this, we rewrite (5) as:

$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = \begin{pmatrix} \sum_{t=1}^{n} w_{0t}\, y_t \\ \sum_{t=1}^{n} w_{1t}\, y_t \end{pmatrix} = \frac{1}{v^*} \begin{pmatrix} \sum_{t=1}^{n} w^*_{0t}\, y_t \\ \sum_{t=1}^{n} w^*_{1t}\, y_t \end{pmatrix},$$

where these weights are

$$w_{0t}^* = \begin{cases} 17 - 2\phi + 16\phi^2 + (1 - 33\phi + \phi^2)\cos(\pi/6), & \text{if } t \in R_1; \\ \frac{1}{2}(35 - 35\phi + 35\phi^2 - 32\phi^3) - \phi(34 - 33\phi + \phi^2)\cos(\pi/6), & \text{if } t \in R_2; \\ (1-\phi)[17 + \frac{31}{2}\phi^2 - 32\phi\cos(\pi/6)], & \text{if } t \in R_3; \\ \frac{1}{2}(33 - 33\phi + 27\phi^2 - 30\phi^3) - \phi(30 - 31\phi - \phi^2)\cos(\pi/6), & \text{if } t \in R_4; \\ 17 - 14\phi + 14\phi^2 - 14\phi^3 - (1 + 31\phi - 28\phi^2 - \phi^3)\cos(\pi/6), & \text{if } t \in R_5; \\ 16 - 16\phi + \frac{23}{2}\phi^2 - \frac{29}{2}\phi^3 - 2\phi(14 - 15\phi - \phi^2)\cos(\pi/6), & \text{if } t \in R_6; \\ 17 - 20\phi + 17\phi^2 - 17\phi^3 + (1 - 33\phi + 36\phi^2 - \phi^3)\cos(\pi/6), & \text{if } t \in R_7; \\ 18 - 18\phi + \frac{39}{2}\phi^2 - \frac{33}{2}\phi^3 - 2\phi(18 - 17\phi + \phi^2)\cos(\pi/6), & \text{if } t \in R_8 \end{cases}$$

and

$$w_{1t}^* = \begin{cases} 1 - \frac{39}{2}\phi + \frac{35}{2}\phi^2 + (35 - 35\phi + 2\phi^2)\cos(\pi/6), & \text{if } t \in R_1; \\ \frac{1}{2}(37 - 39\phi + 41\phi^2 - 35\phi^3) - \phi(37 - 37\phi + 2\phi^2)\cos(\pi/6), & \text{if } t \in R_2; \\ (1-\phi)[(1-\phi)^2 - 2\phi(1-\phi)\cos(\pi/6)], & \text{if } t \in R_3; \\ -\frac{1}{2}(33 - 27\phi + 29\phi^2 - 31\phi^3) + \phi(33 - 29\phi - 2\phi^2)\cos(\pi/6), & \text{if } t \in R_4; \\ 1 + \frac{99}{2}\phi - \frac{93}{2}\phi^2 - \phi^3 - (35 - 31\phi + 31\phi^2 - 31\phi^3)\cos(\pi/6), & \text{if } t \in R_5; \\ -34 + 30\phi - 32\phi^2 + 32\phi^3 + 2\phi(34 - 31\phi - \phi^2)\cos(\pi/6), & \text{if } t \in R_6; \\ 1 - \frac{111}{2}\phi + \frac{105}{2}\phi^2 - \phi^3 + (35 - 35\phi + 39\phi^2 - 35\phi^3)\cos(\pi/6), & \text{if } t \in R_7; \\ 36 - 36\phi + 38\phi^2 - 34\phi^3 - 2\phi(36 - 35\phi + \phi^2)\cos(\pi/6), & \text{if } t \in R_8 \end{cases}$$

with

$$v^* = 594 - 558\phi + \frac{1073}{2}\phi^2 - \frac{1015}{2}\phi^3 + 4\phi(\phi^2 + 262\phi - 279)\cos(\pi/6).$$

Here, $R_1, \ldots, R_8$ denote the time points where the corresponding GLS weights are identical, given by

$$R_\ell = \begin{cases} \{1, 35\}, & \text{if } \ell = 1; \\ \{2, 10, 14, 22, 26, 34\}, & \text{if } \ell = 2; \\ \{3, 9, 15, 21, 27, 33\}, & \text{if } \ell = 3; \\ \{4, 8, 16, 20, 28, 32\}, & \text{if } \ell = 4; \\ \{5, 7, 17, 19, 29, 31\}, & \text{if } \ell = 5; \\ \{6, 18, 30\}, & \text{if } \ell = 6; \\ \{11, 13, 23, 25\}, & \text{if } \ell = 7; \\ \{12, 24\}, & \text{if } \ell = 8. \end{cases}$$

Matching matrices for $n = CT$ and $n = CT + 1$ in Model 1 Here, we choose $C = 3$ for simplicity. Our approach of constructing $\mathbf{D}$ can be easily extended to any larger C as illustrated in Section 2. As $\{w_{0t}\}$ has the same periodic pattern as $\{w_{1t}\}$, we consider only $\{w_{1t}\}$. Plots like Fig. 1 show a periodic pattern (similar to the pattern for $n = CT - 1$) in $\{w_{1t}\}$. For $n = CT$, nine different values of $\{w_{1t}\}$ are identified, giving us the following $9 \times n$ matching matrix:

$$\mathbf{D} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

This matrix is a straightforward extension from the case $n = CT - 1$. As $w_{11} \neq w_{1n}$ for $n = CT$, we use 9-dimensional $\mathbf{e}_i$'s ($i \in \{1, \ldots, 9\}$) to trace the periodic pattern in $\{w_{1t}\}$ with $\mathbf{e}_9$ as the last column of $\mathbf{D}$. When $n = CT + 1$, this approach gives us:

$$\mathbf{D} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

This $\mathbf{D}$ follows the same pattern in the matching matrix for $n = CT$ with the successive pattern vector $\mathbf{e}_8$ inserted before the last column vector $\mathbf{e}_9$.

References

1. Bergland, G.D.: A fast Fourier transform algorithm for real-valued series. Commun. ACM **11**, 703–710 (1968)
2. Brockwell, P.J., Davis, R.A. Time Series: Theory and Methods, 2nd edn. Springer, New York (1991)
3. Ghysels, E., Osborn, D.: The Econometric Analysis of Seasonal Time Series. Cambridge University Press, Cambridge (2001)
4. Grenander, U., Rosenblatt, M.: Statistical Analysis of Stationary Time Series. Wiley, New York (1957)
5. Judge, G.G., Hill, R.C., Griffiths, W.E., Lütkepohl, H., Lee, T.C.: Introduction to the Theory and Practice of Econometrics, 2nd edn. Wiley, New York (1988)
6. Kay, S.M.: Maximum likelihood estimation of signals in autoregressive noise. Amer. Statist. **42**, 88–101 (1994)
7. Lee, J.: A reformulation of weighted least squares estimators. Amer. Statist. **63**, 49–55 (2009)
8. Li, S., Lund, R.: Multiple change point detection via Genetic Algorithms. J. Climate **25**, 674–686 (2012)
9. Malik, W.Q., Schummers, J., Sur, M., Brown, E.N.: Denoising two-photon calcium imaging data. PLoS ONE **6**, e20490 (2011). doi:10.1371/journal.pone.0020490
10. Mann, M.E., Lees, J.M.: Robust estimation of background noise and signal detection in climate time series. Clim. Chang. **33**, 409–445 (1996)
11. Piegorsch, W.W., Bailer, A.J.: Analyzing Environmental Data. Wiley, Chichester (2005)
12. Purdon, P.L., Solo, V., Weisskoff, R.M., Brown, E.N.: Locally regularized spatiotemporal modeling and model comparison for functional MRI. NeuroImage **14**, 912–923 (2001)
13. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and its Applications with R Examples, 3rd edn. Springer, New York (2011)

14. Sorensen, H.V., Jones, D.L., Heideman, M.T., Burrus, C.S.: Real-valued fast Fourier transform algorithms. IEEE Trans. Acoust. Speech Signal Processing **35**, 849–863 (1987)
15. Yang, R., Su, Z.: Analyzing circadian expression data by harmonic regression based on autoregressive spectral estimation. Bioinformatics **26**, i168–i174 (2010). doi:10.1093/bioinformatics/btq189
16. Zinde-Walsh, V., Galbraith, J.W.: Estimation of a linear regression model with stationary ARMA(p, q) errors. J. Economet. **47**, 333–357 (1991)