Advances in Water Resources 134 (2019) 103429

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Adaptation of the visibility graph algorithm for detecting time lag between rainfall and water level fluctuations in Lake Okeechobee

Rahul John[a,b,1], Majnu John[c,d,1,*]

[a] Water Science Associates, Inc., Fort Myers, FL, United States
[b] Department of Earth and Environmental Sciences, University of New Orleans, New Orleans, LA, United States
[c] Department of Mathematics, 308 Roosevelt Hall, 130 Hofstra University, Hempstead, NY 11549, United States
[d] The Feinstein Institute of Medical Research, Northwell Health System, Manhasset, NY, United States

ARTICLE INFO

Keywords:
Time series
Visibility graph algorithm
Cross correlogram
Water levels
Precipitation

ABSTRACT

Identifying time-lag between two hydrogeological time series for planning and management of water resources has a long history and is of continuing research interest. Many hydrogeological studies in the past have used visual inspection and cross-correlogram techniques for quantifying the time lag. Cross-correlogram techniques, if not done under the transfer function framework, could lead to ambiguous results. In order to conduct cross-correlogram analysis under the transfer function framework, careful pre-processing steps have to be undertaken, which are often ignored in practice. In this paper, we propose a new approach to compare two sets of hydrogeological time series data using the visibility graph algorithm and show the advantages of using the new approach over the traditional approach. Application of the new approach is demonstrated by assessing the lags between rainfall and water level fluctuation in Lake Okeechobee, Florida. We also present simulation studies to better understand the performance of the method for different sample sizes, different underlying models and in the presence of missing values.

1. Introduction

Long term planning of water resources often requires an understanding of time lag between a precipitation event and corresponding water level or flow response in a lake, stream, or an aquifer. Considering precipitation events are the primary source of recharge for groundwater and surface water resources, it is critical from an operational standpoint to quantify the time lag and responses due to precipitation in these water bodies, especially for lakes whose levels are artificially controlled for flood management and other ecological reasons.

Although existence of time lag between precipitation events and water level responses is supported by empirical data, quantitative assessments of time lags are typically done by visual inspection on a graphical plot (e.g. Westoff et al., 2010) or using cross-correlation techniques (e.g. Levanon et al., 2016). Cross-correlation method, although useful in many cases, could lead to ambiguous results if not done under the transfer function framework. If cross-correlation is done under the transfer function framework, certain assumptions (such as joint bivariate stationarity of the two-time series) have to be met (Wei, 2006; Box et al., 2008). The method also requires diagnostic checking for model adequacy, which is rarely done in practice. In this paper, we present a non-parametric method to quantify the time lag using a simple adaptation of the visibility graph algorithm (VGA). This algorithm converts a time series into a graph. Although originally developed by physicists (Lacasa et al., 2008; Lacasa and Luque, 2010; Nuñez et al., 2012), it has found wide applications outside the physics literature. In our adaptation, we consider one of the time series (e.g. water levels) as a reference time series and create time shifted copies of the other time series of interest (e.g. precipitation). The time series (original, copies and the reference) are then converted to graphs and their corresponding adjacency matrices calculated using VGA, and then compared. The VGA method is described in detail in Section 2.

1.1. Data selection

To illustrate the VGA based approach, we compiled long-term hydrological time series data, which include water level data of a surface water reservoir and corresponding precipitation data from nearby stations. The criteria used for data selection was that the time series should have quality continuous data without gaps and real world environmental or anthropogenic significance.

* Corresponding author.
E-mail address: majnu.john@hofstra.edu (M. John).
[1] Contributed equally.

https://doi.org/10.1016/j.advwatres.2019.103429
Received 29 January 2019; Received in revised form 14 September 2019; Accepted 30 September 2019
Available online 30 September 2019

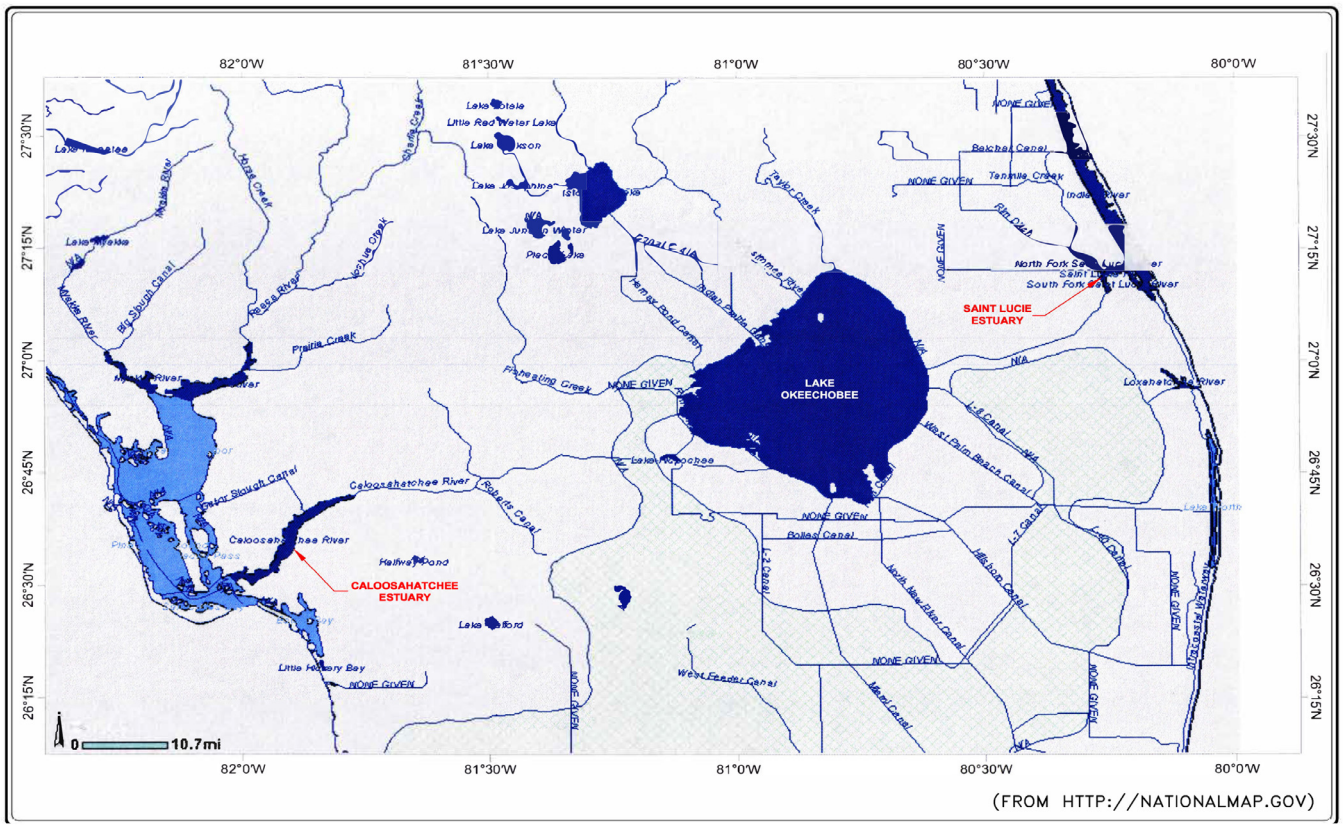

Fig. 1. Map showing the locations of Lake Okeechobee, and the Saint Lucie and the Caloosahatchie estuaries, obtained from http://nationalmap.gov.

The surface water reservoir lake selected for analysis is Lake Okeechobee, which is the second largest natural freshwater lake within the contiguous United States, covering approximately 730 square miles. The primary inflows into the lake are Kissimmee River and Taylor Creek located north of the lake, Fisheating Creek located west of the Lake, and the primary outflows are the Everglades, Caloosahatchee River and the St. Lucie River (Fig. 1). Water flowing in and out of the lake is controlled by human decisions which includes determining the time and frequency of opening and closing of numerous gates and locks. The South Florida Water Management District (SFWMD) and the US Army Corps of Engineers (USACE) jointly operate the lake's water control structures to achieve water levels in Lake Okeechobee that balance water supply, flood protection, and environmental health (Audobon Florida Naturalist Magazine,Fall, 2005a; Audobon Florida Naturalist Magazine,Writer, 2005b). The Comprehensive Everglades Restoration Plan (CERP) identifies ideal Lake Okeechobee water level range to be between 12 feet NGVD (at the end of dry season) and 15 feet NGVD (at the end of wet season). Any rise in lake level exceeding 18.5 feet NGVD may compromise the structural integrity of the Herbert Hoover Dike surrounding the lake; hence, as lake levels approach 17 feet NGVD, large freshwater discharges are made to the St.Lucie estuary to the east and Caloosahatchee Estuary to the west disrupting the natural salinity patterns and water chemistry of these estuaries and impacting its flora and fauna. Lake levels going below 12 feet NGVD can cause water shortages especially during drought years.

2. Method

Let us denote the two hydrogeological time series that we are interested in, namely precipitation and water levels, by $P(t)$ and $WL(t)$ (or simply P and WL), respectively. In order to find the time lag between the two time series, as a first step we fix one of the series, say WL, and obtain time-shifted copies of the other series, $P_{\tau_1}, \ldots, P_{\tau_K}$. The key step in our methodology is the conversion of all the above time series into graphs based on the visibility graph algorithm. Graphs are mathematical constructs that are used to study relationships among various objects. In graph models the objects of interest are modeled as nodes or vertices and the relationships among the objects are modeled using edges or lines connecting the vertices.

Visibility graph algorithm (VGA) (Lacasa et al., 2008; Lacasa and Luque, 2010; Nuñez et al., 2012) is a method that extends usefulness of the techniques and focus of mathematical graph theory to characterize time series. It has been shown that the visibility graph inherits several properties of the time series, and its study reveals nontrivial information about the time series itself. VGA has become very popular [e.g. Xu et al., 2008; Marwan et al., 2009; Donner et al., 2010; Luque et al., 2009; Ahmadlou et al., 2010; Gao et al., 2015; Ahmadlou et al., 2012; Elsner et al., 2009; Zhu et al., 2014; Yang et al., 2009; Donges et al., 2012; Donner and Donges, 2012; Zhang, 2017] and has found wide applications outside the physics literature as evidenced by the >700 citations of the original paper. The applications have ranged from health applications related to Alzheimer's disease (Ahmadlou et al., 2010), Autism disorders (Ahmadlou et al., 2012) and sleep studies (Zhu et al., 2014) to geophysical studies (Donner and Donges, 2012) such as hurricanes (Elsner et al., 2009), to financial applications (Yang et al., 2009). However, to the best of our knowledge, our paper is the first paper in which VGA has been used for time lag detection.

Fig. 2 top panel illustrates how the visibility algorithm works. The time series plotted in the upper panel is an approximate sine series; specifically, a sine series with Gaussian white noise added. The values at 24 time points are plotted as vertical bars. One may imagine these vertical bars as, for example, buildings along a straight line in a city landscape (i.e. a city block). Each node in the associated visibility graph (shown in the bottom panel) corresponds to each time point in the series. So, the graph in Fig. 2 has 24 nodes. We draw a link or an edge between a pair of nodes, say t_i and t_j, if the visual line of sight from the

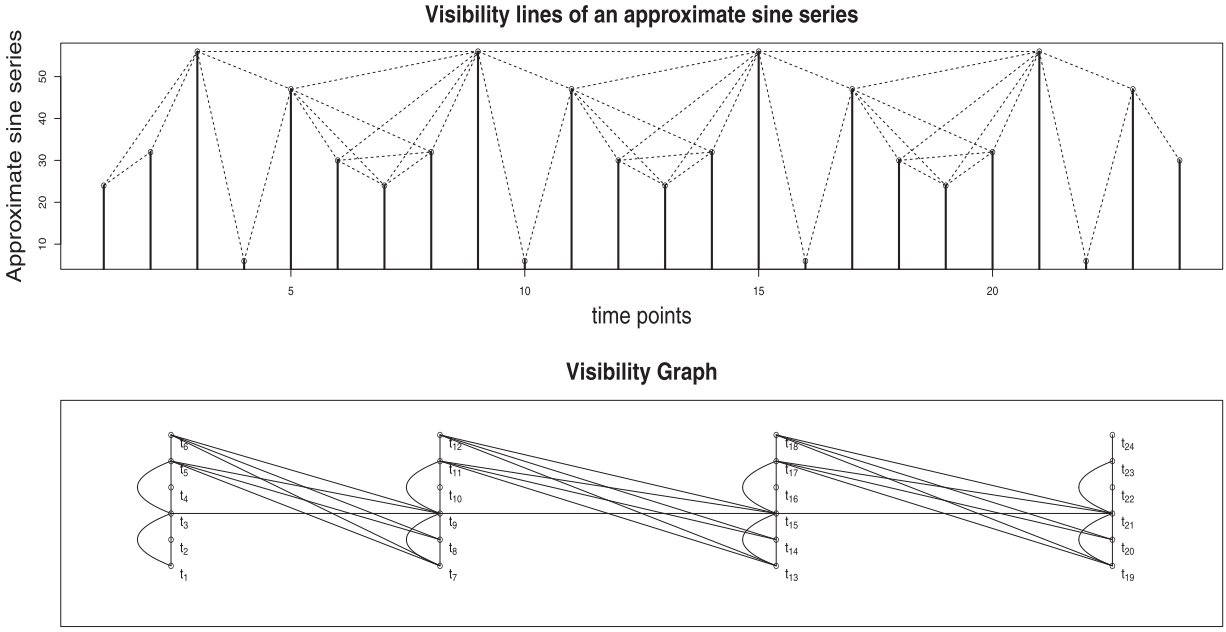

Fig. 2. A time series and the corresponding visibility graph. t_1, t_2 etc. denote the time points as well as the corresponding nodes in the visibility graph.

top of the building (vertical bar) situated at t_i towards the top of the building/bar at t_j is not blocked by any intermediate buildings - that is, if we were to draw a line from the top of the vertical bar at t_i to the top of the vertical bar at t_j, it should not intersect any intermediate vertical bars. Visibility lines corresponding to the edges in the graph are plotted as dotted lines in the figure in the upper panel. For example, there is no edge between t_2 and t_4 since the line of sight (not shown) between the top points of the vertical bars at these two time points is blocked by the vertical bar at t_3. On the other hand, there is an edge between t_1 and t_3 since the corresponding visibility line (shown as a dotted line) does not intersect the vertical bar at t_2.

More formally, the following visibility criteria can be established: two arbitrary data values (t_q, y_q) and (t_s, y_s) will have visibility, and consequently will become two connected nodes of the associated graph, if any other data (t_r, y_r) placed between them fulfills:

$$y_r < y_s + (y_q - y_s)\frac{t_s - t_r}{t_s - t_q}.$$

This simple intuitive idea has been proven useful practically because of certain features exhibited by the graphs generated by this algorithm. First of all they are connected, since each node is connected to at least its neighbors. Secondly, there is no directionality between the edges, so that the graph obtained is undirected. In addition, the visibility graph is invariant under rescaling of the horizontal and vertical axes and under horizontal and vertical translations. In other words, the graph is invariant under affine transformations of the original time series data.

In mathematical notation any graph with n nodes could be represented by its $n \times n$ adjacency matrix A which consists of 0's and 1's. The (i, j)th element of A is 1 if there is an edge connecting the ith and the jth node, 0 otherwise. Two graphs, G_1 and G_2, can be be compared by the metric "distance", $\|A_{G_1} - A_{G_2}\|_2$ between their corresponding adjacency matrices, A_{G_1} and A_{G_1}. Here, $\|\cdot\|_2$, called the Frobenius norm of a matrix, is the square root of the sum of the squares of the elements of the matrix; that is, the square root of the trace of the product of the matrix with itself. In mathematical notation, if D denotes the matrix $A_{G_1} - A_{G_2}$ and D_{ij} the ijth element of D (with $i, j = 1, \ldots, T$), then

$$\|A_{G_1} - A_{G_2}\|_2 = \|D\|_2 = \sqrt{\mathrm{Trace}(DD^t)} = \sqrt{\sum_{i=1}^{T}\sum_{j=1}^{T} D_{ij}^2}.$$

Our proposed method to assess the time lag between the two hydrogeological time series P and WL using the visibility graph approach is as follows: Convert the WL time series into a visibility graph and obtain its corresponding adjacency matrix, A_{WL}. Consider time-shifted copies of the P time series, $P_{\tau_1}, \ldots, P_{\tau_\kappa}$, each shifted in time by a lag from the set $\{\tau_1, \ldots \tau_\kappa\}$. Convert these time-shifted copies of P into their visibility graphs and obtain the corresponding adjacency matrices $A_{P_{\tau_1}}, \ldots, A_{P_{\tau_\kappa}}$. We determine the copy $A_{P_{\tau_s}}$ for which the Frobenius norm $\|A_{WL} - A_{P_{\tau_s}}\|_2$ is minimized. The time lag between the two original hydrogeological series is then taken as τ_s.

We further illustrate our method using the plots in Fig. 3. The time series in the top panel, *ts.a* is a series of 50 values approximately based on a sine function (that is, a sine series with some white noise added)

$$ts.a[t] = 100\sin(2\pi f t) + w_t, \text{ where } f = (80/1000),\ w_t \sim N(0, 25^2).$$

The time series, *ts.b*, plotted in the middle panel of Fig. 2 is derived from *ts.a* as follows:

$$ts.b[t] = (1/3)ts.a[t-2] + e_t, \text{where } e_t \sim N(0, 5^2).$$

That is, *ts.b* is derived by shifting *ts.a* to the right by two units, by reducing the amplitude to one-third that of *ts.a*, and adding some white noise. In other words, *ts.a* and *ts.b* have roughly the same shape although their amplitudes are different and one is shifted by two time units relative to the other as seen in the figure. One may think of *ts.a* and *ts.b* as two time series one affecting the other (since, *ts.b* is shifted to the left, physically we would think of *ts.b* affecting *ts.a*); e.g. *ts.b* as precipitation and *ts.a* as water levels. Physically, water levels and precipitation never take negative values; so, if one really wants to think of *ts.a* and *ts.b* as water levels and precipitation, one could think of them as mean-subtracted and scaled appropriately.

We considered time-shifted copies of *ts.b* with time-shifts from the following set: $\{0, 1, 2, \ldots, 20\}$. VGA was applied and adjacency matrices for the corresponding graphs were obtained. Distance-measure based on the Frobenius norm for the time-shifted copies of *ts.b* compared to the reference *ts.a*, are plotted in the bottom panel of Fig. 2. The distance-measure is minimized at 2, which was the lag that we set *a priori*. Thus, in this illustrative example, the lag was correctly identified by the method that we proposed.

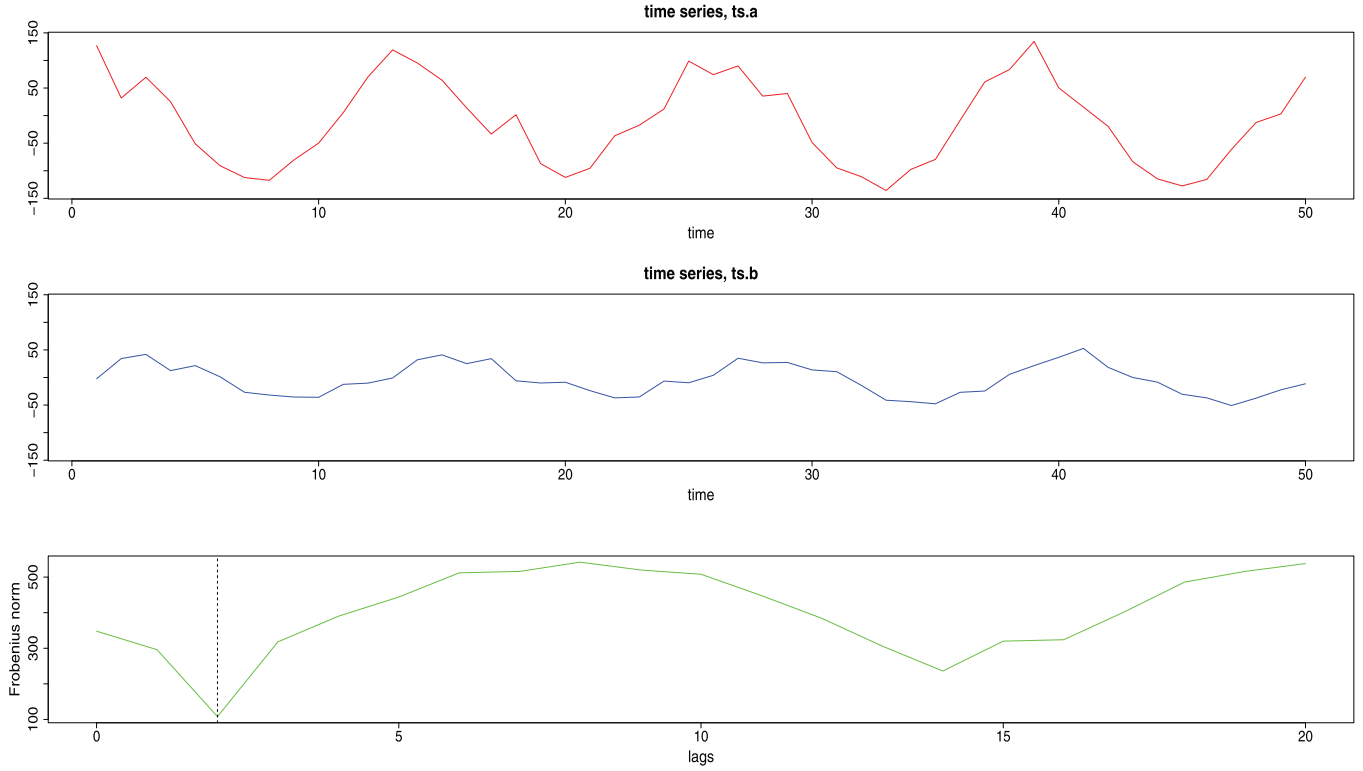

Fig. 3. Illustration of our method. Top two panels show time series, one shifted by a lag of two from the other. Bottom panel shows the distance-measure based on Frobenius norm for different time lags; minimum is achieved for the time lag 2.

3. Comparison with existing method

Currently existing method for detecting lags between two time series is based on cross-correlograms. Often in practice, cross-correlograms are applied to the original two time series and lag is determined as the point corresponding to the peak (maximum positive or maximum negative) correlation. However, this approach could lead to ambiguities, and is not the best recommended approach from a statistical point of view. A better approach involves a pre-processing step which assures that the two time series used in the cross-correlogram analysis are both stationary and white noise. The justification for this pre-processing step can be understood under the transfer function framework. Consider two time series y_t and x_t, with y_t affected by lagged values of x_t. The relationship between the two series may be modeled under the transfer function framework as

$$y_t = \nu(B)x_t, \text{ where } \nu(B) = \sum_{j=0}^{\infty} \nu_j B^j.$$

Here B denotes the back-shift operator $B^j x_t = x_{t-j}$; $\nu(B)$ is referred to as the transfer function. For finite samples from real-life examples a rational form is assumed for $\nu(B)$:

$$\nu(B) = \frac{\omega(B)B^b}{\delta(B)}. \tag{1}$$

Here $\omega(\cdot)$ and $\delta(\cdot)$ are polynomial functions with finite number of non-zero coefficients and b represents the actual lag between the two series. Our main goal is to estimate b - the time delay. Rewriting Eq. (1) as

$$\delta(B)\nu(B) = \omega(B)B^b$$

and expanding we will see that $\nu_j = 0$ for $j = 0, \ldots, b-1$ and $\nu_j \neq 0$ for $j = b$. Thus, in this framework, b is determined as the index of the first non-zero coefficient ν_j. The above procedure is easy if there is a way to estimate the coefficients ν_j's. As described below, cross-correlations provide a way to estimate the coefficients ν_j's.

Cross-correlation between the two series at lag k is defined as

$$\rho_{xy}(k) = \frac{\gamma_{xy}(k)}{\sigma_x \sigma_y},$$

where $\gamma_{xy}(k)$ is the cross-covariance between x_t and y_t, σ_x and σ_y are standard deviations of x_t and y_t respectively. If the following two conditions

(C1): x_t and y_t are jointly bivariate stationary,
(C2): x_t is white noise series

are simultaneously met, then there exists a scaled relationship between ν_k and $\rho_{xy}(k)$:

$$\nu_k = \frac{\sigma_y}{\sigma_x}\rho_{xy}(k). \tag{2}$$

Thus, if C1 and C2 are met, then based on Eq. (2) we may determine which coefficients ν_j are zero and which are non-zero; this in turn will help us to determine the lag b. Putting it all together, we may summarize that if conditions C1 and C2 are met, then the lag b is the index of the first statistically non-zero cross-correlation term. In other words, we may use a cross-correlogram to determine the lag.

The key point here is that the conditions C1 and C2 have to be met in order to apply the cross-correlogram method. C1 and C2 are simultaneously met if both y_t and x_t are white noise series. One way to assure this in practice is to fit appropriate models (AR, MA, ARMA or ARIMA) separately to x_t and y_t and use the corresponding residuals to plot the cross-correlogram because if the fitted model is accurate then the residuals are white noise. This necessitates an extra step in the process: considering several models for x_t and y_t and fitting the most appropriate model - at least appropriate enough to generate white noise as residuals. This key step (known as pre-whitening x_t and y_t) is often ignored in practice when cross-correlograms are used, which in turn could lead to erroneous conclusions. One advantage of using the VGA based method proposed in this paper is that the pre-whitening step is

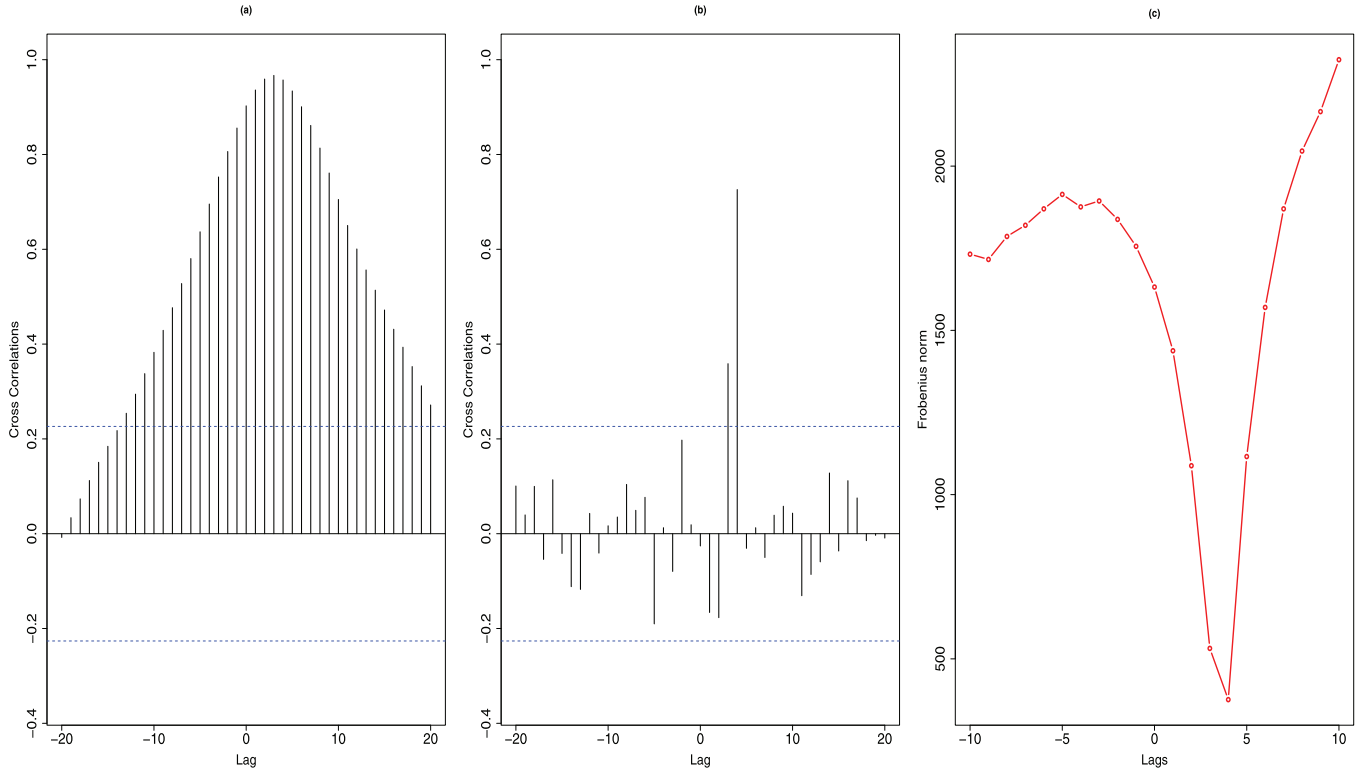

Fig. 4. Panel (a): CCF of original pair of time series. Panel (b): CCF of differenced pair of time series. Panel (c): Frobenius norm from VGA based method.

not necessary for the new method. We illustrate all the above concepts using the following simulated example.

We simulated x_t from a ARIMA (1,1,0) model; that is, one AR coefficient, one degree of differencing and no MA coefficients. AR coefficient was set to 0.7. The sample size was 75. y_t was generated as

$$y_t = 16 + 0.85x_{t-3} + 1.4x_{t-4} + e_t,\ e_t \sim N(0, 0.02).$$

Note that there are two *a priori* set lags (i.e. lags at 3 and 4) for this example. Pre-whitening was done for both series by fitting an ARIMA (1,1,0) model and using the residuals. Autocorrelogram of the residuals showed correlation of one at lag 0 and zero everywhere else, indicating that the residuals were indeed white noise. The left and middle panels in Fig. 4 below show cross-correlograms without and with pre-whitening, and the right panel shows the Frobenius norm from the VGA-based method at various lags. In the first correlogram, the maximum correlation is at lag 3 ($\rho_{xy}(3) = 0.967$) with next two highest correlations at lag 2 ($\rho_{xy}(2) = 0.959$) and lag 4 ($\rho_{xy}(4) = 0.957$) respectively. First of all, there is ambiguity about the number of lags to be picked based on the first correlogram. If we decide to pick lags with the two largest correlations, then we will correctly pick lag 3, but incorrectly pick lag 2. Note that in the model for y_t the effect for lag 4 (1.4) was higher than the effect for lag 3; however the first correlogram ranked lag 4 at the third place. Cross-correlogram based on pre-whitened series (middle panel) gives a more clear cut answer. Only correlations at lags 3 and 4 are statistically different from zero, with the one at lag 4 prominently greater than the one at lag 3. In this case, the correlogram identified the lags correctly.

VGA-based method's Frobenius norm is lowest at lag 4, and second lowest at lag 3. The norm value at lag 3 is relatively much closer to that at lag 4, compared to the next nearest values. Thus, in this case also we correctly picked the *a priori* set lags. Note that VGA-based method identified the correct lag without pre-whitening, while the pre-whitening step was necessary for the cross-correlogram based method.

4. Simulations

4.1. Sample size

We conducted Monte Carlo simulations to assess the performance of the VGA-based method as we varied some of the parameters of the two time series *ts.a* and *ts.b* considered in the second section. The parameters that we considered were *a*) the ratio of the amplitudes between the two simulated series *ts.a* and *ts.b, b*) the variance for the noise term '`rnorm(n, 0, *)`' in the series *ts.a* (indicated by $*$) and *c*) the variance for the noise term '`rnorm(n, 0, *)`' in the series *ts.b*. (Note that we did all the simulations in the R statistical software and above we borrowed from R language the term '`rnorm(n, 0, sd)`' which stands for 'n' data points from the normal density with mean 0 and standard deviation = 'sd'.) For each simulation scenario considered in this section (that is, for each set of the above parameters), 1000 pairs of *ts.a* and *ts.b* were generated, and for each pair time lag was assessed based on the proposed method and compared with the lag that was set *a priori*. The performance of the method was assessed based on the percentage of times that the *a priori* lag was correctly identified. The *a priori* lags that we considered for each scenario were 2, 5, 10 and 15; we assumed that in typical examples from hydrogeology, 2 will be a small lag and 15 will be a very large lag.

The reason for considering the ratio of amplitudes was that even if two hydrogeological time series are roughly of the same shape with only a lag between them, their amplitudes (i.e. roughly their 'sizes') are often vastly different. For *ts.a* and *ts.b* used in the introductory illustrative example in the second section, the ratio of their amplitudes was 1/3. One of the questions that was addressed in our simulations was whether our method was still good if we changed this ratio drastically, e.g. to 1/9. Another question that we thought should be addressed is that whether the proposed method works only for smooth periodic time series such as the 'sine series'. Increasing the variance for the noise term in *ts.a* makes it less like a 'sine series'. Finally, increasing the variance

Table 1
Performance of the VGA-adapted method when the ratio of amplitudes between *ts.a* and *ts.b* was 1/3, the noise term in *ts.a* was *rnorm*(*n*, 0, 25) and the noise term for *ts.b* was *rnorm*(*n*, 0, 5).

	a priori set lag			
	2	5	10	15
$n = 25$	94.0%	99.0%	93.0%	97.0%
$n = 50$	100.0%	100.0%	100.0%	100.0%

Table 2
Performance of the VGA-adapted method when sample size is fixed at $n = 50$ and various parameters are changed one at a time from the values for the illustrative example.

	a priori set lag			
	2	5	10	15
amplitude ratio = 1/9	71.0%	64.0%	64.0%	64.0%
ts.a noise: *rnorm*(*n*, 0, 50)	100.0%	100.0%	100.0%	100.0%
ts.b noise: *rnorm*(*n*, 0, 10)	94.0%	92.0%	91.0%	92.0%

of the noise term in *ts.b* makes the shape of *ts.b* quite different from that of *ts.a*, and by doing so in our simulations we also addressed the performance of the method in such scenarios.

Our hypothesis was that if we changed the above-mentioned parameters to make the relationship between *ts.a* and *ts.b* less ideal than in the illustrative example in the second section, the performance of the method will be worse. In that sense, essentially the purpose of our simulation was to see whether increasing the sample size will improve the performance in such 'bad scenarios', and if so, what would be a recommended minimum sample size that would hedge against such scenarios. In order to do that, we need a reference sample size; that is, a sample size for which the method's performance was excellent when the relationship between *ts.a* and *ts.b* was reasonably good (by reasonably good we mean roughly the same shape and size with only a lag in between them). In Table 1 above we present the performance of the method for sample sizes 25 and 50, when the ratio of the amplitudes and the noise terms were kept exactly the same as in the illustrative example. Since Table 1 shows that the performance was excellent for $n = 50$, we consider 50 as a good sample size choice if we have reasons to believe (may be by visual inspection) that there is a nice relationship between *ts.a* and *ts.b*.

For the next set of simulations, we fixed the sample size to be 50, and varied the above-mentioned parameters one at a time. We varied the parameters one a time rather than simultaneously in order to achieve a meaningful representation of two hydrogeological time series, one affecting the other. The results of this set of simulations are presented in Table 2. The first row presents the performance when the noise terms are kept the same as in the introductory illustrative example, but the ratio of the amplitudes was reduced to 1/9. In this case the performance of the method became drastically worse as seen from the table. In the next row, we present the results when the standard deviation for the noise term for *ts.a* was changed to 50 (-for the illustrative example, it was 25), but the other two parameters were kept the same. This was to check whether the performance became worse if the shape of both time series was not roughly like a sine series. Results from Table 2 show that the performance is not affected in this case. These results give reasons to believe that our initial choice of a sine series shape did not matter; in other words, we would think that the method will perform well no matter what the shapes of the two series are as long as both the series are roughly of the same shape and size. The third row in Table 2 shows the results when only the noise term for *ts.b* was changed. This would correspond to making the shape of *ts.b* quite different from that of *ts.a*. In this case, the performance is affected but not very much as the percentages in the third row are all still above 90%. Thus, based on Table 2, the factors that affected the performance was the ratio of the amplitudes and the noise term for *ts.b* and among these two, the effect of the former was much more severe than the latter.

Table 3
Performance of the VGA-adapted method when the ratio of amplitudes between *ts.a* and *ts.b* was 1/9, the noise term in *ts.a* was *rnorm*(*n*, 0, 25) and the noise term for *ts.b* was *rnorm*(*n*, 0, 5).

	a priori set lag			
	2	5	10	15
$n = 90$	95.0%	91.0%	91.0%	93.0%
$n = 180$	98.0%	100.0%	99.0%	100.0%
$n = 365$	100.0%	100.0%	100.0%	100.0%

Table 4
Performance of the VGA-adapted method when the amplitude for the middle 1/3rd section of *ts.a* was changed from 100 to 30. All the other parameters were retained exactly the same as in Table 3, including the ratio of amplitudes between *ts.a* and *ts.b* to be equal to 1/9.

	a priori set lag			
	2	5	10	15
$n = 90$	94.0%	93.0%	94.0%	95.0%
$n = 180$	99.0%	100.0%	100.0%	100.0%
$n = 360$	100.0%	100.0%	100.0%	100.0%

Next, we checked whether the performance of the method corresponding to the scenario in the first row in Table 2 (that is, ratio of amplitudes equals 1/9 and noise terms for *ts.a* and *ts.b* kept exactly the same as in the illustrative example) improved with sample and if so, what could be a recommended minimum sample size. The results from this set of simulations are presented in Table 3. As noted in the table, the performance increases very much when the sample size is increased to 90, and is near perfect when the sample size is 180. The percentages for all *a priori* lags are 100% when the sample size is 365.

Finally, we used simulations to also check the performance of the method when the amplitude of the time series varied 'seasonally'. It is well-known that the average accumulative precipitation varies seasonally; typically, for several consecutive months the average accumulative precipitation is high and for several other consecutive months the average accumulative precipitation is low, and similarly for water levels. In order to mimic this scenario somewhat, we generated *ts.a* series using the same parameters as in Table 3, except for the amplitude. We divided the set of time points into three equal sets of consecutive time points so that, when e.g. $n = 90$, we have the initial 30 time points $t_1 - t_{30}$, the middle 30 time points $t_{31} - t_{60}$ and finally the last 30 time points $t_{61} - t_{90}$. For the first and last one-thirds of the time points an amplitude of 100 and for the middle one-third an amplitude of 30 was used in the simulations for *ts.a*. *ts.b* was generated with 1/9 as the ratio of amplitudes, as in Table 3. The results for this new set of simulations are presented in Table 4. The results look very similar to that seen in Table 3, indicating that the additional seasonal variation of the amplitudes did not have any effect on the performance of the method.

The take home message from all the simulations results presented above is that if we are considering hydrogeological time series for which measurements were made daily, then an year's worth of data will be more than sufficient for the proposed method, although the method will work quite well even with 6 months' worth of data. Visually, if the two time series looks clearly to be one affecting the other and of roughly the same shape and size, then even 2 months worth of data will suffice. Note that in this section, we used the term *sample size* to refer to the number of time points in the time series, and throughout the paper, we implicitly assume that the data points in all the time series are measured at equal time intervals.

4.2. Missing values

Missing values are common in all time series measurements for hydrogeological phenomena. In this section we assess, via simulations, the performance of the proposed method in the presence of missing values for two different types of imputation methods - least observation carried forward (LOCF) and mean imputation. In LOCF imputation, if a value is missing at any time point, we carry forward the previous non-missing value; in mean value imputation we impute the average of the prior and the subsequent non-missing values. The missing value mechanism that we considered for our simulations was Missing Completely At Random (MCAR) which means that the missing values are missing exactly as the name implies (completely at random). There is another commonly considered (for example, in clinical studies) missing value mechanism - Missing At Random (MAR) - under which the missingness may depend on the previously observed outcomes. Under this mechanism, both LOCF and mean imputation are known to be biased. But, we consider that the missingness in hydrogeological time series do not depend on the previously observed outcomes, and hence the MAR assumption is unrealistic, and thus we did not consider MAR for our simulations. As a matter of fact, the missingness that we have seen for real hydrogeological time series is as follows - for example, for a time series in which measurements are made daily, non-missing measurements are seen for a large chunk of consecutive time points (6 months to 2 years) followed by a large chunk of data missing at a stretch (several weeks or months), which again is followed by a large chunk of non-missing data, and so on. When a significant amount of data is missing for a large number of consecutive time points, none of the existing imputation methods will work very well. In such cases, the best strategy is to analyze separately the large chunks of data with no missing values at all. Nevertheless, we conducted the following simulations for hypothetical scenarios.

In all the simulations reported in this section, we fixed the sample size to be 180, and we used the same noise terms for *ts.a* and *ts.b* as in the illustrative example in the second section. The ratio of amplitudes was set to be 1/9 as in the simulations for Table 3. In order to adhere to the MCAR mechanism we randomly set either 9 or 18 or 27 or 36 values to be missing; 9, 18, 27 and 36 correspond to 5%, 10%, 15% and 20% of 180. Furthermore, we considered scenarios where the values were set to be missing for only one time series (*ts.a*) or for both. If it were set to be missing for both, then it was at the same time points for both, which we think is the more realistic scenario.

The performance of the proposed method with both LOCF and mean imputation was near perfect when only 5% of the values (that is, 9 out of 180) were missing (Table 5). This was true regardless of whether the values were missing for only one time series or for both, and also true across all *a priori* set lags, 2, 5, 10 and 15. When 10% of the values (that is, 18 out of 180) were missing for only one time series, the method did very well under both LOCF and mean imputation for all lags. When 10% of the values were missing for both time series, the performance was still very good when the lags were large (10 or 15); when the lags were small (2 or 5), the performance with both imputation methods was still good but not as good as when the lags were large. For example, when 10% values were missing and when the lag was 2, the performance with LOCF was 97% and 92%, respectively, depending on whether the values were missing for only one time series or both; the corresponding values for lag 10, on the other hand, were even better: 98% and 97%.

With 15% missing values (27 out of 180), the performance was still good (that is, in the range $90\% - 97\%$) with LOCF and mean imputation, for lags 2, 5, and 10, irrespective of whether it was missing for only one or for both time series (although, of course, if it was missing only for one time series, it was better). However, when the *a priori* set lag was 15, the performance with LOCF was weak (84%), when 15% values were missing for both time series; it was still good (96%) with LOCF when only one time series had 15% missing values, and with mean imputation also (92% and 98%). With 20% missing values the method worked well under both types of imputations and for all lags, only when one time series had missing values. When both time series had 20% missing values, the performance of LOCF was not good with small lags (84% for lag 2 and 86% for lag 5) and got worse for larger lags (77% for lag 10 and 79% for lag 15). The performance with mean imputation was slightly better (82%, 90%, 81% and 88%, for lags 2, 5, 10 and 15, respectively) but still not quite up to the mark.

Table 5
Performance of the VGA-adapted method with imputation methods for missing values. The sample size was fixed to be 180. All the other parameters were retained exactly the same as in Table 3, including the ratio of amplitudes between *ts.a* and *ts.b* to be equal to 1/9.

a priori set lag	No. missing	LOCF		Mean Imputation	
		both TS	only 1 TS	both TS	only 1 TS
2	9	97.0%	99.0%	100.0%	100.0%
	18	92.0%	97.0%	95.0%	98.0%
	27	91.0%	96.0%	93.0%	98.0%
	36	84.0%	92.0%	82.0%	92.0%
5	9	100.0%	99.0%	99.0%	100.0%
	18	94.0%	98.0%	94.0%	98.0%
	27	95.0%	95.0%	91.0%	97.0%
	36	86.0%	94.0%	90.0%	98.0%
10	9	100.0%	99.0%	98.0%	99.0%
	18	97.0%	98.0%	97.0%	99.0%
	27	92.0%	96.0%	92.0%	97.0%
	36	77.0%	93.0%	81.0%	94.0%
15	9	98.0%	100.0%	100.0%	100.0%
	18	99.0%	98.0%	98.0%	97.0%
	27	84.0%	96.0%	92.0%	98.0%
	36	79.0%	94.0%	88.0%	94.0%

In summary, based on the above simulation results, we consider it acceptable to use the proposed method in conjunction with either of the imputation methods if it is only 5% values missing for only one time series or for both. With $6 - 15\%$ values missing, the imputation methods give good results only if it is missing for one time series. With about 20% of the values missing for both time series, it is definitely not recommended to use the proposed method with either of the imputations although it may be somewhat acceptable if it is missing for only one time series. Also, in general, we observed that the performance with mean imputation was slightly better except for one or two scenarios. If the statistical practitioner has a preference of one method over the other, it may still be recommended to use both for the proposed method, at least as a sensitivity analysis. Finally, we emphasize again the point made in the beginning of the section, that if large chunks of data are missing at a stretch then the imputation methods are not likely to work; in such cases, it is better to focus the analysis on other chunks of data with no or very sparse missing values.

4.3. Multivariate simulations

In all the simulations done so far, the first time series was simulated using a univariate model and the second time series was generated by obtaining a lagged copy of the first series and then adding random noise to it. A better strategy, technically, is to generate both the time series from a multivariate model and shift the second series to set a lag between the two series *a priori*. In this subsection we present results from such simulation strategies. Multivariate time series that we considered were all generated from vector ARMA processes which include vector AR processes and vector MA processes as special cases. The general form for a n-vector ARMA (p, q) process z_t is given by

$$z_t + A_1 z_{t-1} + \ldots A_p z_{t-p} = \varepsilon_t + B_1 \varepsilon_{t-1} + \ldots + B_q \varepsilon_{t-q},$$

where A_i's and B_j's are $n \times n$ matrices $i = 1, \ldots, p$ and $j = 1, \ldots, q$ and z_t's and ε_t's are vectors with n-elements. Elements of ε_t are white noise processes (i.e. serially uncorrelated across time); however, at each time point there might be correlation among the elements. In this paper, since we consider only bivariate time series, n was set equal to 2 for all

Table 6
Performance of the three methods under different multivariate models. Sample sizes of 100 and 500, and *a priori* set lags of 3 and 7 were considered.

Sample size, $n = 100$					Sample size, $n = 500$				
Model	lag	VGA	CCF1	CCF2	Model	lag	VGA	CCF1	CCF2
M1	$\Delta = 3$	99.7%	100%	100%	M1	$\Delta = 3$	100%	100%	100%
	$\Delta = 7$	100%	100%	100%		$\Delta = 7$	100%	100%	100%
M2	$\Delta = 3$	96.1%	93.9%	95.6%	M2	$\Delta = 3$	99.6%	100%	100%
	$\Delta = 7$	95.3%	94.1%	95.0%		$\Delta = 7$	99.7%	100%	100%
M3	$\Delta = 3$	91.9%	98.7%	100%	M3	$\Delta = 3$	100%	100%	100%
	$\Delta = 7$	91.8%	98.4%	100%		$\Delta = 7$	100%	100%	100%
M4	$\Delta = 3$	81.2%	42.9%	100%	M4	$\Delta = 3$	100%	52.7%	100%
	$\Delta = 7$	82.2%	42.2%	100%		$\Delta = 7$	100%	63.6%	100%

the simulations in this subsection. All the matrices that we considered were diagonal; however we did include a correlation among the elements within each ε_t. Following are the models that we considered:

(M1): $x_t = 0.65x_{t-1} - 0.38\varepsilon_{t-1}^{(1)} + \varepsilon_t^{(1)}$, $y_t = 0.95y_{t-1} - 0.62\varepsilon_{t-1}^{(2)} + \varepsilon_t^{(2)}$; $y_{t-\Delta} = x_t$, or in matrix notation $z_t + Az_{t-1} = \varepsilon_t + B\varepsilon_{t-1}$ where $z_t = [x_t, y_t]'$, $\varepsilon_t = [\varepsilon_t^{(1)}, \varepsilon_t^{(2)}]'$, $A = \text{diag}(-0.65, -0.95)$, $B = \text{diag}(-0.38, -0.62)$.

(M2): $x_t = 0.40x_{t-1} + 1.20x_{t-2} + 0.05x_{t-3} - 0.35x_{t-4} + \varepsilon_t^{(1)}$, $y_t = 0.30y_{t-1} + 0.85y_{t-2} - 0.04y_{t-3} - 0.45y_{t-4} + \varepsilon_t^{(2)}$; $y_{t-\Delta} = x_t$. In matrix notation, this will be $z_t - A_1z_{t-1} - A_2z_{t-2} - A_3z_{t-3} - A_4z_{t-4} = \varepsilon_t$ where $A_1 = \text{diag}(-0.40, -0.30)$, $A_2 = \text{diag}(-1.20, -0.85)$, $A_3 = \text{diag}(-0.05, 0.04)$ and $A_4 = \text{diag}(0.35, 0.45)$.

(M3): $x_t = 0.40\varepsilon_{t-1}^{(1)} + \varepsilon_t^{(1)}$, $y_t = -0.40y_{t-1} + \varepsilon_t^{(2)}$; $y_{t-\Delta} = x_t$. In matrix notation $z_t + Az_{t-1} = \varepsilon_t + B\varepsilon_{t-1}$ where $A = \text{diag}(0, 0.40)$, $B = \text{diag}(0.40, 0)$.

(M4): $x_t = 0.95x_{t-1} + \varepsilon_t^{(1)}$, $y_t = 0.95y_{t-1} - 0.95\varepsilon_{t-1}^{(2)} + \varepsilon_t^{(2)}$; $y_{t-\Delta} = x_t$. In matrix notation $z_t + Az_{t-1} = \varepsilon_t + B\varepsilon_{t-1}$ where $A = \text{diag}(-0.95, -0.95)$, $B = \text{diag}(0, -0.95)$.

The pair of time series within M1 were same as the best fitted models for the respective time series (i.e. rainfall and water level fluctuation) from the Lake Okeechobee data and analysis presented below in Section 5. Thus, model M1 was included primarily to assess the validity of the methods for the Lake Okeechobee data analysis and its conclusions. The correlation ρ among the elements of ε_t used for generating all the above bivariate series was 0.6. For each model, we considered separate simulations with lags 3 and 7 set *a priori*. For most hydrogeological time series pairs, a lag of 3 would be considered as small and a lag of 7 considered large. Bivariate time series based on models 1, 3 and 4 were generated using the 'varma' command within the 'multiwave' package in R (Achard and Gannaz, 2019). For bivariate time series based on model 2, 'mAr.sim' command within the 'mAr' package (Barbosa, 2015) in R was used. The results from the simulations studies corresponding to various scenarios are presented in Table 6 below. The results are based on 1000 iterations for each simulation scenario. CCF1 and CCF2 in Table 6 correspond to cross-correlation method without and with pre-processing step, respectively.

In all scenarios, there was no substantial difference between the accuracies obtained in the lag 3 setting compared to that with lag 7. All the methods worked very well for time series pairs generated from M1, even with a sample size of 100. This justifies our use of the methods, particularly the VGA-based method, for the analysis of Lake Okeechobee data. For simulations under M2, all three methods were comparable for both sample sizes. The accuracy of all three methods were not perfect (but only approximately 95%) when the sample size was 100, but it reached 100% or near 100% with a sample size of 500. For the next two models, only CCF2 performed very well for a sample size of 100. Under M3, the accuracy of VGA-based method was approximately 92% and that of CCF1 was approximately 98.5%. Although the performances of VGA-based method and CCF1 were not optimal with the smaller sample size, they improved substantially when a sample size of 500 was used. Under M4, the accuracy of VGA-based method was approximately 81% and that of CCF1 was approximately 43% when the *a priori* lag was 3. However, with a sample size of 500, the performance of VGA-based improved substantially to 100%. The performance of CCF1 method also improved, but not as much as for the VGA method. The results for the simulations based on M3 and M4 bring out a limitation of the VGA based method - that its performance is dependent on sample size.

M3 and M4 are different from M1 and M2 in the following sense. The pair of time series in M1 and M2 were both from the same type of time series models (ARMA(1,1) for M1 and AR(4) for M2) although the coefficients differed. The two time series within M3 were from different types (MA(1) for the first time series and AR(1) for the second time series). The two time series within M4 were also from different types (AR(1) for the first time series and ARMA(1,1) for the second time series). The above set of simulations seems to suggest that when the underlying pair of time series under consideration are of two different types, then VGA-adapted method performs very well only with large sample sizes. Thus, if there are reasons to believe that the underlying time series are from two different types of models, then the VGA-based method presented in this paper should be used with caution, especially for smaller sample sizes.

Next we present analysis of lag between rainfall and water level fluctuation in Lake Okeechobee to illustrate the methods further.

5. Lake Okeechobee data

For this analysis, we selected daily water level and rainfall data from two monitoring stations located on Lake Okeechobee. The Lake and hydrologic features connected to it are one of the most studied and monitored watershed systems in United States. Because of its significance in regional flood management and water supply and considering it is a vital fresh water resource for Florida, the Lake and its surrounding wetlands have a suite of monitoring stations that collect water level, water quality, meteorological and flow data throughout the year. The Lake system is part of the Greater Everglades watershed that stretches from the Kissimmee River to Florida Bay with significant flows occurring through the Everglades. The SFWMD and ACOE takes extreme measures to constantly monitor the lake levels and maintain it at optimal levels taking into consideration public's safety, water demands, and the health of flora and fauna in the estuaries downgradient from the Lake.

Daily rainfall and water level data used for this analysis was collected between January 1, 2000 and December 31, 2018 (19 years). Rainfall data was highly skewed; so, a fourth root transformation was done before all analyses. There were no missing values for water levels, and less than 0.5% values missing for rainfall data. Since the percent of missing values was very small, we used imputation based on least observation carried forward (LOCF) before conducting the analysis. The primary goal of the analysis was to detect the time lag(s) between daily rainfall and daily changes in water level. Note that we used daily changes in water level rather than daily water level itself because lake water levels are more complex and tied to many hydrogeological

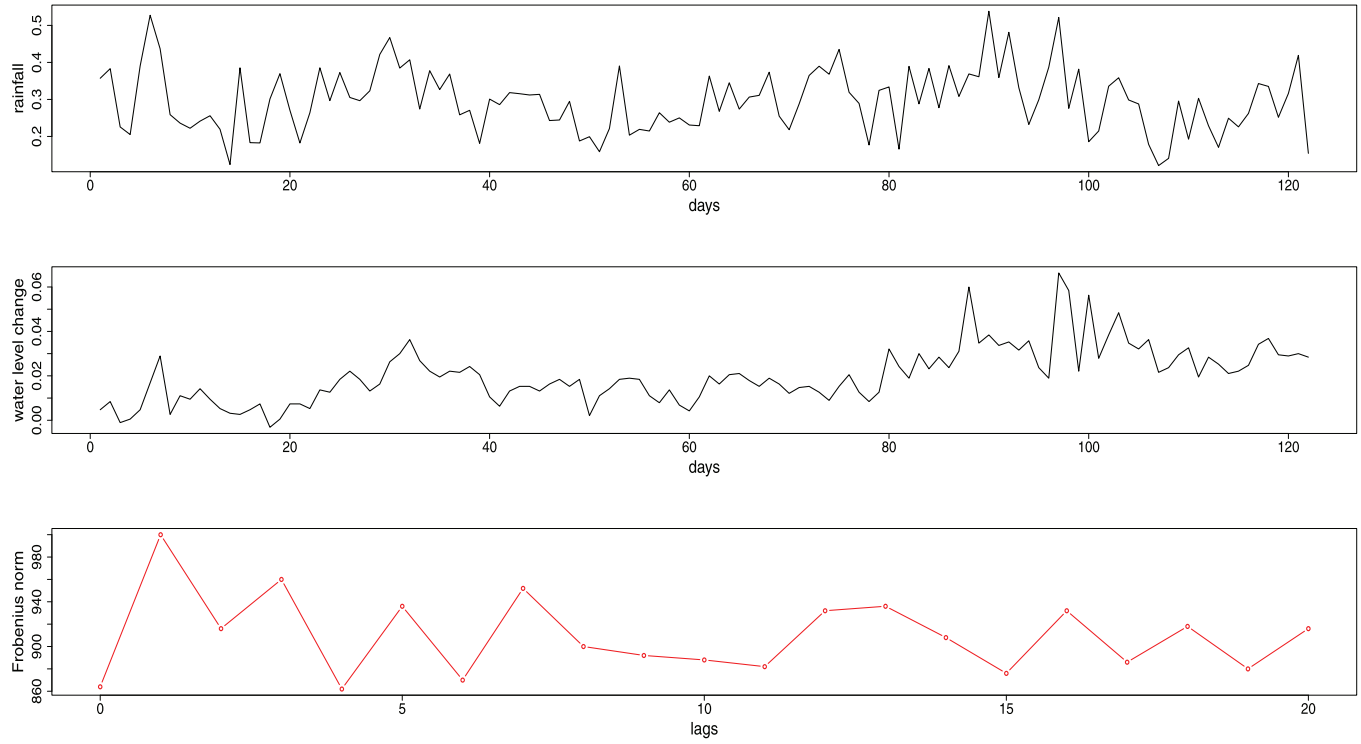

Fig. 5. Daily rainfall data averaged across 19 years (2000–2018) for months of June, July, August and September are shown in the top panel. Daily water level change in Lake Okeechobee averaged across 19 years for the same months are shown in the middle panel. Frobenius norm from the VGA based analysis are shown in the bottom panel.

factors in addition to rainfall such as evaporation, surface water and groundwater inflow and outflows.

The number of days of rainfall for each month from January to December, averaged across all 19 years were 5.1, 4.8, 6.2, 5.4, 8.1, 14.2, 13.5, 16.1, 13.2, 7.1, 5.4, and 5.7 respectively. The average number of days with rainfall was substantially higher for the months of June, July, August and September; these four months represent the "wet" season for the region. Further, we estimated the average annual rainfall for each of the 19 years and noticed that 2014 was the wettest year. The first analysis below is mainly for illustration purposes only - we applied our proposed method to the wet season (June through September) of the wettest year (2014).

The lowest value of the Frobenius norm was at lag 4 (norm value 706), the second lowest value at lag 6 (norm value 724). The third and fourth lowest values were at lags 15 and 0 (norm values were close to each other: 742 and 746). It is interesting to note that our analysis indicates a primary lag of about 4 or 6 days and a secondary lag of about 15 days. The primary lag may be a reflection of immediate water level response to rainfall events occurring in and around the Lake while the secondary lag may be a response to rainfall occurring in the Lake vicinity and an increase in inflows via the lake inlet streams caused due to rainfall events occurring upstream from the Lake. Although not prominent, there is also some indication of an immediate effect shown by the lag at 0.

The above analysis was conducted for the wettest year. We repeated the analysis after averaging the data across all 19 years. Averaging the data will be more statistically appropriate when the results are used for prediction. Again, this analysis was done for illustrative purposes only. The averaged data for rainfall and water level change are shown in the upper panel and middle panels of Fig. 5, respectively. The Frobenius norm from VGA based analysis for different lags are plotted in the bottom panel of Fig. 5. The lowest value of the norm was at lag 4, followed by lags at 0 and 6 (norm values 862, 864 and 870, respectively). The fourth ranked lag was at lag 15 (norm value 876). It is interesting to note that the top four lags were same as that for the wettest year, although in the analysis for the averaged data the immediate effect (i.e. lag at 0) became more prominent.

We also conducted cross-correlogram analysis for detecting lag. An ARMA model fit with one AR coefficient and one MA coefficient (i.e. ARIMA(1,0,1)) yielded white noise residuals for time series. (See Appendix Fig. A1). The coefficients obtained with this ARMA(1,1) fit were the same as the ones used in the two series in the model M1 for multivariate simulations above. AR alone or MA alone did not produce white noise residuals. The CCF based on the residuals from the ARMA models are plotted in Fig. 6 below. The significant lags detected in this case are at 0 and 1. Thus CCF based analysis detects only immediate effects due to rainfall.

One of the primary reasons for detecting lags is to use them for prediction based on, for example, a time series regression. In order to compare the predictive performance of the lags obtained via VGA versus those obtained via CCF, we fitted time series regressions with daily water level change as the dependent variable, no intercept term and lagged rainfall data as independent variables. In order to accommodate lag terms for rainfall, we used only water level data for months of July, August and September in the regressions. To be precise, if June water level data was included and lag 15 (for example) for rainfall was modelled in the regression, then that would have required rainfall data from the month of May, which was not used in the original VGA or CCF analyses. Hence the prediction analysis was restricted to water level data from July, August and September (total 92 days). Having no intercept is justified as follows. A regression with no intercept term, in the current context, interprets as 'zero rainfall implies zero change in water level', or more technically correctly - 'when there is no rainfall in the recent past, the only change in water level is due to random variation'. For each regression, fitted values and 95% confidence intervals for fitted values were calculated. Accuracy for each regression was determined as the number of actual water-level-change values that were within the 95% confidence interval band of the fitted values.

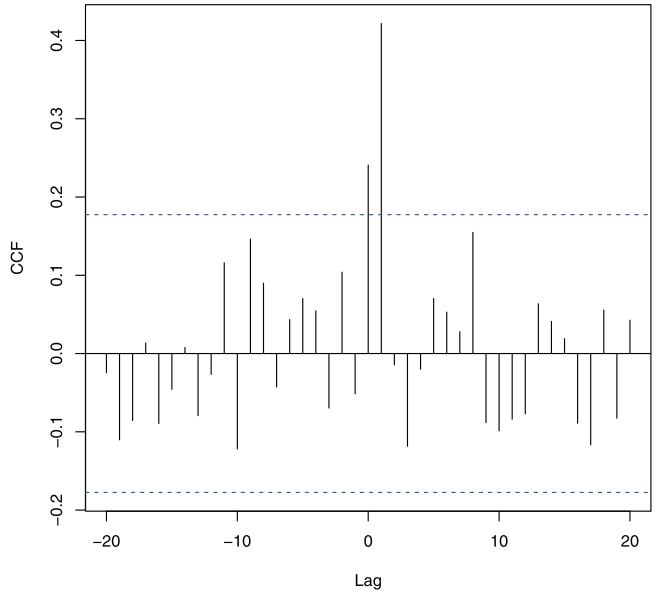

Fig. 6. Cross-correlation function of pre-whitened rainfall and water-level change series from the top panels in Fig. 5.

A time series regression with no intercept and VGA-based lags (that is, lags at 4, 0 and 6) predicted water level change for 20 out of 92 days accurately within statistical error (i.e. within the 95% confidence intervals of fitted values). A similar time series regression with CCF-based lags (that is, lags at 0 and 1) showed slight improvement (- 23 out of 92 days) but was comparable to the predictive performance of VGA-based lags. When the fourth ranked lag from VGA analysis (that is, lag at 15) was also included in a regression with the first three lags, the regression predicted 28 out of 92 days correctly, a slight improvement from CCF-lags based prediction. Including lagged rainfall only in a model will not predict optimally water level change as can be seen from the predictive accuracy values of the above regressions. The purpose of the above analysis to compare the performance between the VGA-based and CCF-based methods, and we do see that they are comparable.

In order to predict daily changes in water level better we have to consider variables other than lagged variables. Including an indicator variable for month and another indicator variable for the week along with lagged rainfall data improved the predictive performance dramatically. Month and week indicators with the top three VGA lags predicted 71 out of 92 days (77%) within statistical error while as the indicator variables along with CCF lags predicted 68 days (74%). Accuracy values reported for both regressions above are considered reasonably good. In order to add more flexibility to modeling (that is, in order to capture rapid fluctuations), we also ran regressions with a four-day indicator variable instead of the week indicator variable. Month indicator variable was retained as in the previous regressions. A regression with month and 4-day indicator variables and the three VGA-based lags predicted 80 days (87%) correctly, while as the same indicator variables with CCF lags predicted 78 days (85%) correctly. The predictive accuracy for the above models are very good. The fitted values and 95% confidence intervals from the above models are plotted in Fig. 7 below.

In summary, the predictive performance of VGA-based lags are comparable to CCF-based lags in this example. Lagged rainfall data by itself does not have very good predictive performance for daily changes in water level data. Other variables such as month and week or four-day indicator variables have much more stronger predictive capability. The best prediction accuracy (87%) was obtained when VGA based lags were included with month and four-day indicator variables.

6. Discussion

Quantifying time lags between two hydrogeological time series is of significance in many modeling contexts. There are several examples in hydrogeological literature where one time series is affected by

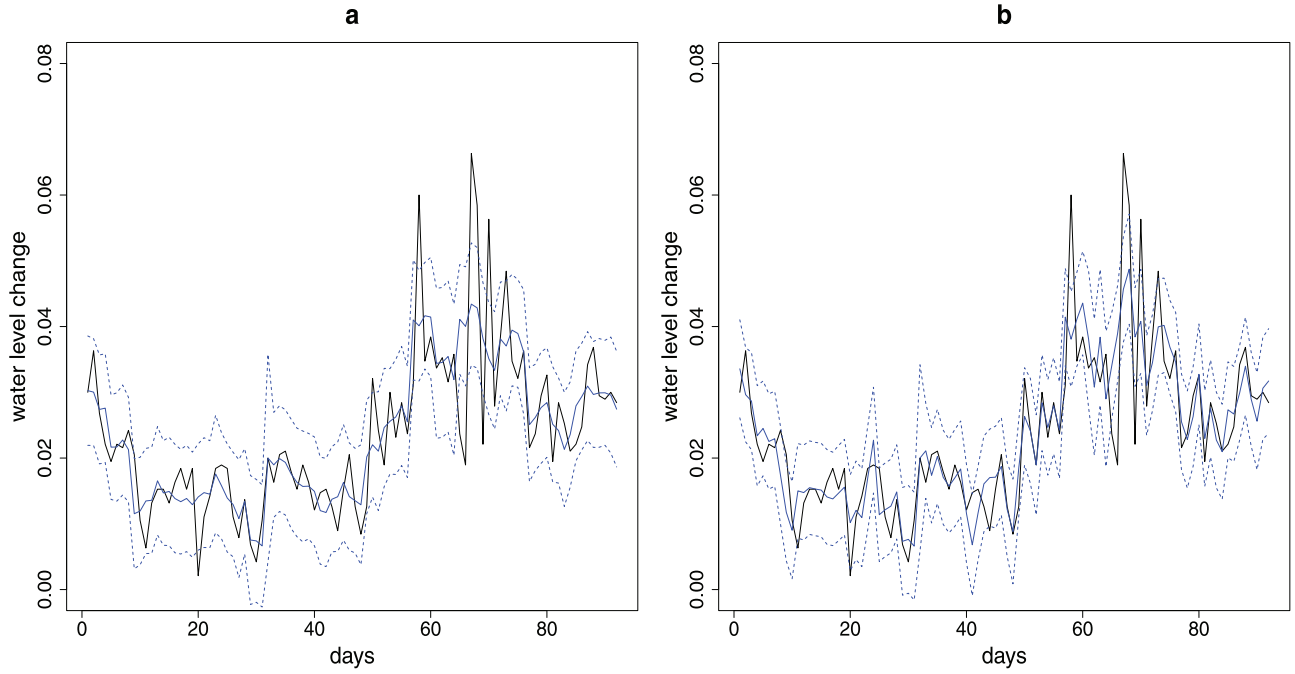

Fig. 7. Water level change and predicted water level change for the months of July, August and September based on the VGA method (left panel) and CCF method (right panel). The black solid lines in each panel shows the original water level change and blue solid lines the predicted values. The blue dotted lines represent the 95% confidence intervals for the predicted values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

another after a time lag. For example, it is often hypothesized that time lag between net precipitation and water level changes in a seepage lake is significantly different from a drainage lake. Seepage lakes are hydraulically isolated from surface water features and primarily fed by groundwater and direct precipitation. Drainage lakes are typically connected to a network of streams and rivers (Wisconsin Department of Natural Resources, 2009). Another example is in the study of karst systems where the pair of hydrological time series could be discharge-discharge or rainfall-discharge or water level-discharge. For instance, Bailly-Comte *et al* Bailey-Comte et al. (2008) studied the karst/river interactions during the flooding of Coulazou river in southern France, and detected time lags that explained the influence of the river on the water-level elevation in a karst aquifer.

Yet another example is the relationship between precipitation and water levels of a shallow well in an unconfined aquifer versus water levels in a relatively deeper well in a semi-confined aquifer. This relationship is particularly important to water resource managers and groundwater modelers who need to accurately quantify groundwater recharge into aquifers, for developing water-supply-plans for sustainable use of aquifers. Groundwater recharge, defined as entry of water into the saturated zone, is influenced by a wide variety of factors including vegetation, topography, geology, climate, and soils (Dripps, 2003; Dripps et al., 2006). Groundwater recharge, which is a small percentage of the precipitation that eventually reaches the water table, is one of the most difficult parameters to quantify. This is because processes such as evaporation, transpiration and infiltration through unsaturated subsurface must first be estimated to determine the amount of water lost after a rainfall event. Often times, groundwater models are developed by estimating the groundwater recharge using empirical relationships or as a percentage of precipitation. It is a common practice to use groundwater recharge as a calibration parameter, meaning the recharge value that provides the best calibration to the model is selected as representative for the watershed simulated. For temporal simulations, the lag time between a rainfall event and groundwater recharge into deeper aquifers are often ignored.

Currently used methods in hydrogeological literature to detect time lags between a pair of time series is based on simple visual inspection or on cross-correlograms. The latter approach, although substantially better than the former, if used without pre-whitening could lead to ambiguous results as exhibited in a simulated example in the paper. A better way to conduct cross-correlogram analysis is under the transfer function framework. In this paper, we briefly reviewed the transfer function framework, and showed for the above-mentioned simulated example how cross-correlogram under this framework (i.e. after pre-whitening) gives the correct results without ambiguity. However, pre-whitening the series requires careful model fitting in order to obtain the residuals as white noise. There could be examples where even the best fit models may not completely yield white noise as residuals. In this paper, we present an alternate method to detect time lags based on the visibility graph algorithm (VGA) which is a method developed by physicists to convert a time series into a mathematical graph. VGA has become highly popular in various scientific disciplines and have found wide applications. The method for time lag detection proposed in this paper is based on a simple extension of VGA. In the simulated example mentioned above, we showed how the VGA based method detects the lag correctly and unambiguously without having to do a pre-whitening process as in the transfer function framework. However, simulations based on multivariate models revealed that when the pair of time series are from two different types of underlying models, the new approach performs well only for large sample sizes.

7. Conclusions

The primary objective of the paper was to demonstrate that the VGA method is a useful tool that can be applied on hydrogeological data to calculate lags between time series. Understanding the hydrogeological cycle and water balance within any watershed has practical value for water resource managers from an operational and planning perspective. We selected two time series data that are fundamental to hydrogeological science to demonstrate the application of the VGA method, one is rainfall which is the primary source of recharge for a watershed, and the other is water levels which dictate how much water can be used for human consumption or needs to be released for flood management. We applied the methods discussed in this paper to detect the time lags between rainfall and water level fluctuation in Lake Okkechobee in Florida. VGA method detected lags at days 4, 0, 6 and 15, with the most prominent lag at day 4. CCF based method detected lags at days 0 and 1. One of the purposes of detecting lags is to use them in a regression model for prediction. A regression analysis which included month and four-day indicator variable in addition to the lags for rainfall data predicted water level fluctuations with 87% accuracy when VGA lags were used and with 85% accuracy when CCF lags were used. Thus the prediction accuracies based on both approaches were comparable in this case. Both VGA and CCF detected an immediate effect due to rainfall (lag at 0). However, from a scientific perspective, lags at 4 or 6 days and at 15 days are important considering they are reflective of inflows from lake inlet streams. Only VGA based method detected these longer lags. Multivariate simulations study based on the fitted models for the two time series related to Lake Okeechobee validated the use of the proposed methods for this particular analysis, since both time series were from the same type of models. Predictive accuracy based on regression models also justify the use of the proposed method for Lake Okeechobee analysis. However, for datasets where the two time series are from two completely different type of underlying models, larger sample sizes are necessary for the VGA-based approach presented in this paper.

Declaration of Competing Interest

None.

Appendix

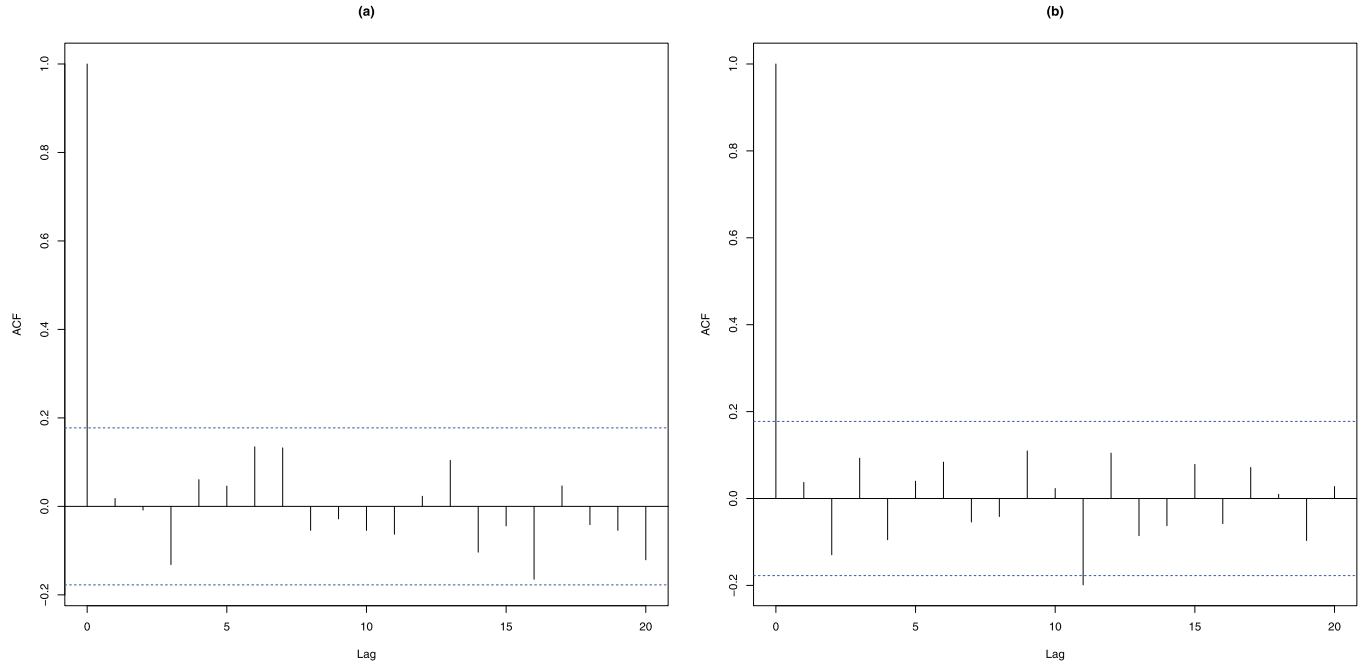

Fig. A1. ACF of residuals from ARMA(1,1) fits for (a) rainfall series after 4th root transformation and (b) water level change series from Lake Okeechobee data analysis. The plots look very similar to the ACF of white noise series, justifying the ARMA(1,1) fit. The coefficients for fitted models were the same as used in model M1 for multivariate simulations. AR-coefficient = 0.65 and MA-coefficient = −0.38 for the rain fall series; AR-coefficient = 0.95 and MA-coefficient = −0.62 for the water level change series.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.advwatres.2019.103429.

References

Achard, S., Gannaz, I., 2019. Wavelet-based and fourier-based multivariate whittle estimation: multiwave. J. Stat. Soft. 89 (6), 1–39.

Ahmadlou, M., Adeli, H., Adeli, A., 2010. New diagnostic EEG markers of the Alzheimer's disease using visibility graph. J. Neural Transm. 117 (9).

Ahmadlou, M., Adeli, H., Adeli, A., 2012. Improved visibility graph fractality with application for the diagnosis of Autism Spectrum Disorder. Phys. A 391 (20).

Audobon Florida Naturalist Magazine, Fall, 2005. http://fl.audubon.org/sites/g/files/amh666/f/pubs-naturalist-fall05.pdf.

Audobon Florida Naturalist Magazine, Writer, 2005. http://fl.audubon.org/sites/g/files/amh666/f/pubs-naturalist-winter05.pdf.

Bailey-Comte, V., Jourde, H., Roesch, A., Pistre, S., Batiot-Guilhe, C., 2008. Time series analyses for Karst/River interactions assessment: case of the Coulazou river (southern France). J. Hydrol. 349, 98–114.

Barbosa, S. M., 2015. mAr: Multivariate Autoregressive Analysis. R package version 1.1-2. https://CRAN.R-project.org/package=mAr.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., 2008. Time Series Analysis: Forecasting and Control, 4th ed. Wiley.

Donges, J.F., Heitzig, J., Donner, R.V., Kurths, J., 2012. Analytical framework for recurrence network analysis of time series. Phys. Rev. E 85 (4).

Donner, R.V., Donges, J.F., 2012. Visibility graph analysis of geophysical time series: potentials and possible pitfalls. Acta Geophys. 60 (3), 589–623.

Donner, R.V., Zou, Y., Donges, J.F., Marwan, N., Kurths, J., 2010. Recurrence networks-a novel paradigm for nonlinear time series analysis. J. New J. Phys. 12 (3).

Dripps, W.R., 2003. The Spatial and Temporal Variability of Ground Water Recharge. Department of Geology and Geophysics, University of Wisconsin-Madison Ph.d. diss..

Dripps, W.R., Hunt, R.J., Anderson, M., 2006. Estimating recharge rates with analytic element models and parameter estimation. Ground Water 44, 47–55.

Elsner, J.B., Jagger, T.H., Fogarty, E.A., 2009. Visibility network of united states hurricanes. Geophys Res Lett 36 (16).

Gao, Z.K., Yang, Y.X., Fang, P.C., Jin, N.D., Xia, C.Y., Hu, L.D., 2015. Multi-frequency complex network from time series for uncovering oil-water flow structure. Sci. Rep. 5 (1).

Lacasa, L., Luque, B., 2010. Mapping time series to networks: a brief overview of visibility algorithms. Comput. Sci. Res. Technol. 3. (edited by Nova Publisher), ISBN: 978-1-61122-074-2.

Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuño, J.C., 2008. From time series to complex networks: the visibilty graph. Proc. Natl. Acad. Sci. USA 105 (13), 4972–4975.

Levanon, E., Shalev, E., Yechieli, Y., Gvirtzman, H., 2016. Fluctuations of fresh-saline water interface and of water table induced by sea tides in unconfined aquifers. Adv. Water Resour. 96, 34–42.

Luque, B., Lacasa, L., Ballesteros, F., Luque, J., 2009. Horizontal visibility graphs: exact results for random time series. Phys. Rev. E 80 (4).

Marwan, N., Donges, J.F., Zou, Y., et al., 2009. Complex network approach for recurrence analysis of time series. Phys. Lett. A 373 (46).

Nuñez, A., Lacasa, L., Luque, B., 2012. Visibility algorithms: a short review. Graph Theory (edited by Intech). ISBN 979-953-307-303-2.

Wei, W.W.S., 2006. Time series analysis: Univariate and Multivariate Methods, 2nd ed. Pearson.

Westoff, M.C., Bogaard, T.A., Savenije, H.H.G., 2010. Quantifying the effect of in-stream rock clasts on the retardation of heat along a stream. Adv. Water Resour. 33 (11), 1417–1425.

Wisconsin Department of Natural Resources, 2009. PUB-FH-800 "Wisconsin Lakes".

Xu, X., Zhang, J., Small, M., 2008. Superfamily phenomena and motifs of networks induced from time series. In: Proceedings of the National Academy of Sciences, Vol. 105.

Yang, Y., Wang, J., Yang, H., Mang, J., 2009. Visibility graph approach to exchange rate series. Phys. A 388 (20).

Zhang, R., et al., 2017. Visibility graph analysis for re-sampled time series from auto-regressive stochastic processes. Commun. Nonlinear Sci. Numer. Simul. 42, 396–403.

Zhu, G., Li, Y., Wen, P.P., 2014. Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal. IEEE J. Biomed. Health Inform. 18 (6).