Fast Error Analysis of Continuous GNSS Observations with Missing Data

M.S. Bos*[1], R.M.S. Fernandes†[2,3], S.D.P. Williams‡[4], and L. Bastos§[1,5]

[1]CIIMAR/CIMAR, University of Porto, Rua dos Bragas 289, 4050-123 Porto, Portugal.
[2]University of Beira Interior, IDL, R. Marquês d'Ávila e Boloma, 6201-001, Covilhã, Portugal.
[3]Delft Earth-Oriented Space Research (DEOS), DUT, Kluyverweg 1, 2629HS Delft, The Netherlands.
[4]National Oceanography Centre, Liverpool, 6 Brownlow Street, Liverpool, L3 5DA, United Kingdom.
[5]University of Porto, Faculty of Science, DGAOT, Rua do Campo Alegre, 4169-007 Porto, Portugal.

November 20, 2012

Abstract

One of the most widely used method for the time-series analysis of continuous Global Navigation Satellite System (GNSS) observations is Maximum Likelihood Estimation (MLE) which in most implementations requires $\mathcal{O}(n^3)$ operations for n observations. Previous research by the authors has shown that this amount of operations can be reduced to $\mathcal{O}(n^2)$ for observations without missing data. In the current research we present a reformulation of the equations that preserves this low amount of operations, even in the common situation of having some missing data.

Our reformulation assumes that the noise is stationary to ensure a Toeplitz covariance matrix. However, most GNSS time-series exhibit power-law noise which is weakly non-stationary. To overcome this problem, we present an Toeplitz covariance matrix that provides an approximation for power-law noise that is accurate for most GNSS time-series.

*mbos@ciimar.up.pt
†rmanuel@di.ubi.pt
‡sdwil@noc.ac.uk
§lcbastos@fc.up.pt

Numerical results are given for a set of synthetic data and a set of International GNSS Service (IGS) stations, demonstrating a reduction in computation time of a factor of 10-100 compared to the standard MLE method, depending on the length of the time-series and the amount of missing data.

1 Introduction

The methods of estimating a linear trend from continuous GNSS observations have received considerable attention in the last decades (Johnson and Agnew, 1995; Zhang et al, 1997; Williams, 2003). This interest stems mostly from the power spectral density of the noise at the low frequencies where it can be described by a power-law model (Williams et al, 2004; Santamaría-Gómez et al, 2011). Mao et al (1999) have shown that this spectral density behaviour of the noise must be taken into account in the analysis to avoid underestimating the trend error by a factor of 5-11.

Therefore, not only the trend, but also the noise properties need to be estimated from the data. The Maximum Likelihood Estimation (MLE) method is widely used for this type of analysis (Mao et al, 1999; Langbein, 2004; Williams, 2008). This is an iterative minimization process and each step requires computing the inverse of the covariance matrix. For power-law noise this matrix is full because the autocorrelation between the observations decreases slowly with increasing separation in time. As a result, calculating the inverse of the covariance matrix is computational demanding.

Williams (2003) noted that the computation time for analysing 10 years of data was around 6 hours. Computers have become much more powerful since then but Hackl et al (2011) still mention a computation time of several days to analyse 50 IGS stations using the CATS software (Williams, 2008) that employs the MLE method. They use this result as an argument in favour of their more rapid Allan Variance of the Rate method.

However, Hackl et al (2011) also mention that MLE gives more robust results. Therefore, it is desirable to lower the computation time of MLE to have the best results in a time comparable to heuristic methods. This becomes particularly important when global networks with hundreds of stations are being analysed.

Improved performance of MLE was already presented by Bos et al (2008) who reduced the complexity of taking the inverse of the covariance matrix from $\mathcal{O}(n^3)$ operations to $\mathcal{O}(n^2)$ operations, where n is the number of observations. They achieved this result by assuming that there are no missing data in the evenly spaced time-series and by taking the first difference of the data to make the noise stationary. The latter operation creates a Toeplitz covariance matrix that can be inverted using fast algorithms.

Unfortunately, most evenly spaced time-series contain missing data that destroy the symmetry and thereby the Toeplitz nature of the covariance

matrix. Simple linear interpolation or ignoring the presence of missing data in the construction of the covariance matrix are two ad hoc solutions that give good results (Bos et al, 2008). However, it is difficult to predict under which circumstances these two solutions can no longer be applied. Therefore, we have reformulated our equations without any loss of accuracy to avoid any type of interpolation while maintaining a Toeplitz structure for the covariance matrix.

2 Reformulation of MLE

The equations used in the MLE have already been explained in detail by Langbein (2004), Bos et al (2008) and Williams (2008) but to explain our modifications, we will summarize the steps.

Using the notation of Bos et al (2008), the GNSS observation vector $\mathbf{x}$ minus the estimated signal gives us the residual vector $\mathbf{r}$ that represent the noise in the data. This signal contains a linear trend, representing for instance the motion of the tectonic plates, a nominal bias, possible seasonal signals and other parameters such as offsets.

The $n \times n$ covariance matrix $\mathbf{C}$ that is used in the weighted least-squares estimation and in the likelihood function is constructed by first assuming that there are no missing data and afterwards by eliminating all rows and columns that correspond to the location of the missing data in the observations. The resulting matrix is indicated by $\breve{\mathbf{C}}$ and, assuming that there are m missing data points and n observations, has dimensions $(n-m) \times (n-m)$. Its inversion, and construction, takes around $\mathcal{O}(n-m)^3$ operations. The missing data must also be eliminated from the observation vector $\mathbf{x}$ and residual vector $\mathbf{r}$, resulting in observation and residual vectors $\breve{\mathbf{x}}$ and $\breve{\mathbf{r}}$ of lengths $n-m$. If we denote the $n \times p$ design matrix by $\mathbf{H}$, containing the above mentioned trend, bias, seasonal signals and offsets, then we can apply the same process to get a new design matrix $\breve{\mathbf{H}}$.

Using these matrices and vectors, the parameter p-vector $\hat{\theta}$ estimated with weighted least-squares equation is given by:

$$\hat{\theta} = \left(\breve{\mathbf{H}}^T \breve{\mathbf{C}}^{-1} \breve{\mathbf{H}}\right)^{-1} \breve{\mathbf{H}}^T \breve{\mathbf{C}}^{-1} \breve{\mathbf{x}} \tag{1}$$

The residuals are given by:

$$\breve{\mathbf{r}} = \breve{\mathbf{x}} - \breve{\mathbf{H}}\hat{\theta} \tag{2}$$

The properties of the covariance matrix $\mathbf{C}$ are normally not known beforehand and must also be estimated from the observations. As explained in section 1, we will discuss MLE which maximizes the logarithm of the likelihood function:

$$\ln(p(\mathbf{r})) = -\frac{1}{2}\left[(n-m)\ln(2\pi) + \ln\det(\breve{\mathbf{C}}) + \breve{\mathbf{r}}^T \breve{\mathbf{C}}^{-1} \breve{\mathbf{r}}\right] \tag{3}$$

The values of the parameters that describe the noise model, and thus the covariance matrix $\breve{\mathbf{C}}$, are varied until the most likely value of Eq. (3) has been found. For each variation of the parameters, Eqs. (1) and (2) need to be used to update the values of the residuals.

Eqs. (1) to (3), with the before mentioned procedure of dealing with missing data, are implemented in the CATS software (Williams, 2008) and in the program of Langbein (2010) and we will call this the standard method.

Next, it will be convenient to separate the residuals into observed and missing residuals:

$$\mathbf{r} = \mathbf{r}_o + \mathbf{r}_m = \begin{pmatrix} r_o^1 \\ 0 \\ \vdots \\ r_o^{n-1} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ r_m^2 \\ \vdots \\ 0 \\ r_m^n \end{pmatrix} \tag{4}$$

For example, assume that we have made five observations but that the first, second and fourth observation failed. Our vector $\mathbf{r}_o$ is:

$$\mathbf{r}_o = \begin{pmatrix} 0 \\ 0 \\ r_o^3 \\ 0 \\ r_o^5 \end{pmatrix} \tag{5}$$

Matrix $\mathbf{F}$ is now defined to have a column for each missing point and n rows. Each column is filled with zeros except for a one on the row that corresponds to the location of the missing point. For our example we have:

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{6}$$

The product $\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}$ will select the rows and columns of the inverse of the covariance matrix for which observations are missing. Using this matrix $\mathbf{F}$ and some algebra, see appendix A, we obtain the following relation for the last expression of Eq. (3) :

$$\breve{\mathbf{r}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{r}} = \mathbf{r}_o^T\left(\mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{C}^{-1}\right)\mathbf{r}_o \tag{7}$$

Similar relations exist for $\breve{\mathbf{H}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{x}}$ and $\breve{\mathbf{H}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{H}}$ that can be inserted in Eq. (1). Furthermore, we have:

$$\ln\det(\breve{\mathbf{C}}) = \ln\det(\mathbf{C}) + \ln\det(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}) \tag{8}$$

that can used to compute the second term on the right side of Eq. (3). Eqs. (7) and (8) are the main result of this research. Since the full covariance matrix $\mathbf{C}$ is used, the Toeplitz nature is preserved and its inverse can be computed in $\mathcal{O}(n^2)$ operations. The inversion of $\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}$, which is a $m \times m$ matrix, will take $\mathcal{O}(m^3)$ operations. We will call the MLE method based on Eqs. (7) and (8) the reformulated method. The reformulated method is faster if the fraction of missing data points in the data is less than around 50%, otherwise the standard method is faster. From an analysis of nearly 3000 time series from publicly available datasets we found the median number of missing data to be 9% with a 95th percentile of 41%. Only 1 in 30 time series are likely to have missing data over 50%. Therefore, using the reformulated method is more efficient with the advantage that no approximations are introduced into the equations when stationary noise models are used.

We will assume that the noise properties are constant over time. In reality they probably vary slightly because GNSS receivers and the computed GNSS satellite orbits, among others, continue to improve in accuracy over the years. Nevertheless, this topic falls outside the scope of the current research except for the fact that if the amplitude of the standard deviation of the noise varies by a known amount over time, then this variation can be incorporated into the design matrix $\mathbf{H}$ and observation vector $\mathbf{x}$ (by scaling both $\mathbf{H}$ and $\mathbf{x}$ by the inverside of this variation). In this way, we can continue to use a constant covariance matrix $\mathbf{C}$.

3 Taking the first difference of the data

So far we have assumed that the noise is stationary. However, as was noted in section 1, the noise in GNSS data is weakly non-stationary which means that the variance grows without bounds over time. To overcome this problem, Bos et al (2008) took the first difference of the observations which creates a stationary data set and they developed the corresponding covariance matrix for power-law noise. This makes it possible to use combination of power-law noises, such as white, flicker and random walk noise (Langbein, 2012), in the equations presented in section 2.

The disadvantage of using first differenced data is that a single missing data point will cause two missing first differenced data points. Even in the case of a sequence of missing data points, there are less first differenced observations than undifferenced ones. The result is a larger uncertainty in the estimated trend. In principle one could create a new first differenced observation using the two values around the sequence of missing data points and compute the associated covariance value. However, this would in general not restore the Toeplitz nature of the covariance matrix and we will therefore still not be able to use the fast method discussed in the previous section.

For a few missing data points this extra loss of observations might be acceptable but in this section we will investigate if the weakly non-stationary covariance matrix cannot be approximated by a stationary covariance matrix. To do so, we first have another look at the underlying assumptions that are used to compute the covariance matrix. Mao et al (1999), Williams (2003) and Langbein (2010) assume that the power-law noise starts to develop at the same time when one starts the measurements. On the other hand, Bos et al (2008) assume that the power-law noise is always present, independent of the fact if measurements are taken or not.

Using the uncertainty about what is the right approach, we can define that the power-law noise started at some arbitrary time in the past. Since flicker noise is weakly non-stationary, the variance grows only with the logarithm of time and it quickly approximates a quasi-stationary situation. In Fig. 1 the covariance matrix $\mathbf{C}$ for flicker noise with unit innovation variance is plotted. As such, it is without dimensions. Furthermore, we will omit any scaling factors related to the sampling period, see Williams (2003) and Bos et al (2008). In the left panel one can see that the covariance starts to grow for increasing row and column index. On the first diagonal, $\mathbf{C}(1,1) = 1$ while on the last diagonal $\mathbf{C}(100,100) = 2.53$. Thus, the variance in the observations has grown by 253% since the start of the measurements.

In the right panel the covariance matrix for flicker noise is again plotted but now starting at row and column 1000. For this case, the matrix looks much closer to a Toeplitz matrix. The values on the first and last diagonal, entry 1001 and 1100, are 3.27 and 3.30 respectively which corresponds to only a 1% difference. This example demonstrates that by putting the time when the noise starts to develop far enough in the past, a near Toeplitz covariance matrix for power-law noise can be obtained. For strong non-stationary noise the covariance on the diagonal grows linearly and this approximation becomes less accurate. However, even for an spectral index α of 1.6 the difference in estimated trend error using the full and the approximated covariance matrix is less than 10%.

For a spectral index α smaller than one, the noise is stationary and the covariance matrix will be exactly Toeplitz. This matrix can be constructed using the analytical expressions given by Hosking (1981).

The covariance matrix for power-law noise, that is assumed to have started at some specified time in the past, can be constructed by computing the matrix product $\mathbf{U}^T\mathbf{U}$ where $\mathbf{U}$ is a $n \times n$ upper triangular matrix, filled with coefficients h_i (Kasdin, 1995; Williams, 2003):

$$\mathbf{U} = \begin{pmatrix} 1 & h_1 & h_2 & \dots & h_n \\ 0 & 1 & h_1 & & h_{n-1} \\ \vdots & & & \ddots & \\ 0 & & 0 & \dots & 1 \end{pmatrix} \tag{9}$$

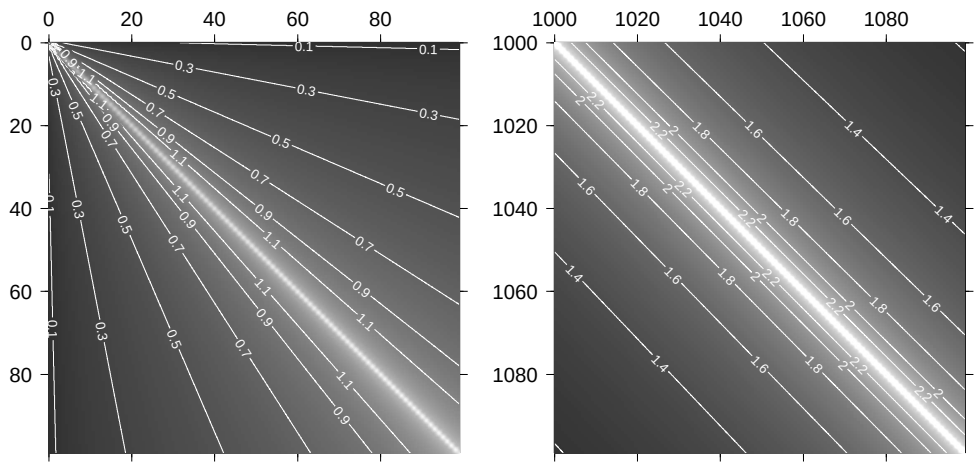

Figure 1: The covariance matrix for flicker noise. The left panel shows the covariance matrix for the first 100×100 elements. The right panel shows the same but starting at row and column 1000.

The last column of the product of $\mathbf{U}^T\mathbf{U}$ is our Toeplitz approximation of matrix $\mathbf{C}$ which can be computed using only $\mathcal{O}(n \log_2 n)$ operations, see Bos et al (2008). The results of using this approximation of the covariance matrix are discussed in sections 5 and 6.

4 Fast Toeplitz solver

Before discussing the numerical results using Eqs. (7) and (8), we introduce some new auxiliary matrices and vectors that facilitate their implementation into a computational application. Our starting point is the method described by Bos et al (2008) that involves performing a Cholesky decomposition of the covariance matrix in $\mathcal{O}(n^2)$ operations:

$$\mathbf{C} = \mathbf{U}^T\mathbf{U} \tag{10}$$

where $\mathbf{U}$ is a $n \times n$ upper triangular matrix. This decomposition also allows us to compute the determinant of matrix $\mathbf{C}$ quickly. At the same time, a set of linear equations are solved using back substitution:

We next introduce a $n \times p$ matrix $\mathbf{A}$ and a p-term vector $\mathbf{y}$, defined by

$$\mathbf{U}^T\mathbf{A} = \mathbf{H}_o \tag{11}$$

$$\mathbf{U}^T\mathbf{y} = \mathbf{x}_o \tag{12}$$

so, $\mathbf{A}$ and $\mathbf{y}$ can be computed by back substitution. Because $\mathbf{C}$ is Toeplitz, we do not actually need to compute $\mathbf{U}$, because we can do the back substitution as part of the Cholesky decomposition. This saves on memory space and increases the computation speed, see Bos et al (2008).

Note that we have to repeat this back substitution, which requires $\mathcal{O}(n^2)$ operations, for each column separately.

Using these matrices and vectors and assuming for the moment that there are no missing data, the equation for weighted least-squares, Eq. (1), reduces to:

$$\hat{\theta} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y} \tag{13}$$

which is the well known equation for ordinary least-squares because the multiplication with matrix $\mathbf{U}^{-T}$ can be viewed as an inverse filter operation that makes the noise in the observations white. If we now define the n-vector $\mathbf{t}$:

$$\mathbf{t} = \mathbf{y} - \mathbf{A}\hat{\theta} \tag{14}$$

then Eq. (7) reduces to:

$$\begin{aligned}\breve{\mathbf{r}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{r}} &= \mathbf{r}_o^T\mathbf{C}^{-1}\mathbf{r}_o \\ &= (\mathbf{x}_o - \mathbf{H}\hat{\theta})^T\mathbf{U}^{-1}\mathbf{U}^{-T}(\mathbf{x}_o - \mathbf{H}\hat{\theta}) \\ &= (\mathbf{y} - \mathbf{A}\hat{\theta})^T(\mathbf{y} - \mathbf{A}\hat{\theta}) \\ &= \mathbf{t}^T\mathbf{t}\end{aligned} \tag{15}$$

If there are missing data, then we also need to compute the $n \times m$ matrix $\mathbf{G}$:

$$\mathbf{U}^T\mathbf{G} = \mathbf{F} \tag{16}$$

Again, each column of matrix $\mathbf{G}$ can be computed with back substitution. One can already see that for large number m of missing data this will slow down the computation.

To make full use of the symmetry in Eq. (7), it will be convenient to define the $m \times m$ matrix $\mathbf{M}$, which comes from the Cholesky decomposition of

$$\mathbf{M} = \mathrm{chol}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}) = \mathrm{chol}(\mathbf{G}^T\mathbf{G}) \tag{17}$$

This also facilitates the computation of the determinant in Eq. (8). Next, we introduce the $m \times p$ matrix $\mathbf{Q}_A$ and m-vector $\mathbf{Q}_y$:

$$\mathbf{M}^T\mathbf{Q}_A = \mathbf{G}^T\mathbf{A} \tag{18}$$

$$\mathbf{M}^T\mathbf{Q}_y = \mathbf{G}^T\mathbf{y} \tag{19}$$

Eqs. (18) and (19) can again be solved using back substitution because $\mathbf{M}$ is a triangular matrix.

Looking back at the equation for weighted least-squares, Eq. (1), one can see that it contains the matrix product $\breve{\mathbf{H}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{H}}$. Using our new matrices

and Eq. (7) this can be rewritten as:

$$\begin{aligned}
&\breve{\mathbf{H}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{H}} = \\
&\quad \mathbf{H}^T\mathbf{C}^{-1}\mathbf{H} - \mathbf{H}^T\mathbf{C}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{C}^{-1}\mathbf{H} = \\
&\quad \mathbf{A}^T\mathbf{A} - \mathbf{A}^T\mathbf{G}\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{A} = \\
&\quad \mathbf{A}^T\mathbf{A} - (\mathbf{A}^T\mathbf{G}\mathbf{M}^{-1})(\mathbf{M}^{-T}\mathbf{G}^T\mathbf{A}) = \\
&\quad \mathbf{A}^T\mathbf{A} - \mathbf{Q}_A^T\mathbf{Q}_A
\end{aligned} \tag{20}$$

A similar expression exists for $\breve{\mathbf{H}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{x}}$ and we can write Eq. (1) as:

$$\hat{\theta} = \left(\mathbf{A}^T\mathbf{A} - \mathbf{Q}_A^T\mathbf{Q}_A\right)^{-1}\left(\mathbf{A}^T\mathbf{y} - \mathbf{Q}_A^T\mathbf{Q}_y\right) \tag{21}$$

Comparing Eq. (13) with Eq. (21) one can see that the matrix $\mathbf{Q}_A$ and vector $\mathbf{Q}_y$ are corrections that take into account the presence of missing data. To compute the weighted sum of squares of the residuals, we need to introduce the m-vector $\mathbf{Q}_t$:

$$\mathbf{M}^T\mathbf{Q}_t = \mathbf{G}^T\mathbf{t} \tag{22}$$

This gives us:

$$\breve{\mathbf{r}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{r}} = \mathbf{t}^T\mathbf{t} - \mathbf{Q}_t^T\mathbf{Q}_t \tag{23}$$

The total number of operation for solving Eq. (16) is of the order of $\mathcal{O}(mn^2)$, which becomes large quickly for increasing number of missing data points m. In fact, numerical experiments showed that already for 1-2% missing data the standard method is faster than using Eqs. (10) to (22), which defies the objective of this research.

This problem has been solved by using the results of Ammar and Gragg (1988) who describe the first step of a Levinson's algorithm (Szegö recursions) that decompose the inverse of our matrix $\mathbf{C}$ as:

$$\mathbf{C}^{-1} = \mathbf{R}\mathbf{D}^{-1}\mathbf{R}^T \tag{24}$$

where $\mathbf{R}$ is a $n \times n$ unit upper triangular matrix and $\mathbf{D}$ a $n \times n$ diagonal matrix that can be easily inverted. The determinant of matrix $\mathbf{D}$ is also easily calculated and corresponds to the determinant of matrix $\mathbf{C}$. The computation of this decomposition takes $\mathcal{O}(n^2)$ operations. However, once it is done we only need the last column of $\mathbf{R}$, which we will denote by $\mathbf{l} = (\mathbf{1}, \mathbf{l_1}, \ldots, \mathbf{l_{n-1}})$, and the last entry of $\mathbf{D} = \delta_n$ to compute the inverse of $\mathbf{C}$:

$$\mathbf{C}^{-1} = \frac{1}{\delta_n}\left(\mathbf{L}_1\mathbf{L}_1^T - \mathbf{L}_2\mathbf{L}_2^T\right) \tag{25}$$

where

$$\mathbf{L}_1 = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ l_{n-1} & 1 & & 0 \\ \vdots & & \ddots & 0 \\ l_1 & \ldots & l_{n-1} & 1 \end{pmatrix} \quad \mathbf{L}_2 = \begin{pmatrix} 0 & 0 & \ldots & 0 \\ l_1 & 0 & & 0 \\ \vdots & & \ddots & 0 \\ l_{n-1} & \ldots & l_1 & 0 \end{pmatrix} \tag{26}$$

Eq. (25) was derived by Trench (1964) and Gohberg and Semencul (1972), and can be used the solve each unknown column vector in Eqs. (11) to (16) using only $\mathcal{O}(n \log_2 n)$ operations because the $\mathbf{L}_1$ and $\mathbf{L}_2$ can be embedded into circulant matrices. The product with circulant matrices can be performed using the Fast Fourier Transform (FFT) which results in the given number of numerical operations.

To keep the equations concise, the factor $1/\sqrt{\delta_n}$ in Eq. (25) will from now on be included in the matrices $\mathbf{L}_1$ and $\mathbf{L}_2$.

Almost equal to what we did before, we will define the following matrices:

$$\mathbf{A}_1 = \mathbf{L}_1^T \mathbf{H}_o \tag{27}$$

$$\mathbf{A}_2 = \mathbf{L}_2^T \mathbf{H}_o \tag{28}$$

The vectors and matrices $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{t}_1$ and $\mathbf{t}_2$ are defined in an analogous manner. We also have:

$$\mathbf{M}^T \mathbf{Q}_A = \mathbf{G}_1^T \mathbf{A}_1 - \mathbf{G}_2^T \mathbf{A}_2 \tag{29}$$

where $\mathbf{M}$ is again the Cholesky decomposition of the matrix product $\mathbf{G}_1^T \mathbf{G}_1 - \mathbf{G}_2^T \mathbf{G}_2$. The definitions of matrices $\mathbf{Q}_y$ and $\mathbf{Q}_t$ are similar. The weighted least-squares equation, Eq. (1), becomes:

$$\hat{\theta} = \left(\mathbf{A}_1^T \mathbf{A}_1 - \mathbf{A}_2^T \mathbf{A}_2 - \mathbf{Q}_A^T \mathbf{Q}_A\right)^{-1} \left(\mathbf{A}_1^T \mathbf{y}_1 - \mathbf{A}_2^T \mathbf{y}_2 - \mathbf{Q}_A^T \mathbf{Q}_y\right) \tag{30}$$

and Eq. (7) can be written as:

$$\breve{\mathbf{r}}^T \breve{\mathbf{C}}^{-1} \breve{\mathbf{r}} = \mathbf{t}_1^T \mathbf{t}_1 - \mathbf{t}_2^T \mathbf{t}_2 - \mathbf{Q}_t^T \mathbf{Q}_t \tag{31}$$

To understand the gain in speed we obtained it must be realised that Eq. (25) is only computed once, requiring $\mathcal{O}(n^2)$ operations. The construction of matrices $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{G}_1$ and $\mathbf{G}_2$ requires in total $\mathcal{O}(2(1+p+m)n \log_2 n)$ operations which is still less than $\mathcal{O}(n^2)$ for sufficient large value of n. The construction of the other auxiliary matrices and the other matrix operations needed to compute the weighted least-squares equation and the MLE cost-function require less than $\mathcal{O}(m^3)$ operations.

Eqs. (30) and (31) are used to compute Eq. (3) and have been implemented in a C++ program that was used to produce the results presented in sections 5 and 6.

Ammar and Gragg (1988) also provide an algorithm that performs Eq. (24) in $\mathcal{O}(n \log_2^2 n)$ operations. However, numerical experiments showed us that, for the time-series discussed in this research, it was slower than the $\mathcal{O}(n^2)$ method due to the larger constant that is hidden in the $\mathcal{O}$ notation. In other words, the algorithm of Ammar and Gragg (1988) requires more intermediate steps and the fact that the algorithm is faster in the limit, only start to pay off at large value of n. Nevertheless, when time-series are used that are about ten times longer, it will be advantageous to use this alternative.

5 Results with synthetic data

This section describes results obtained from analysing synthetic data. Due to the large amount of matrix operations, it is essential to use an optimized BLAS/LAPACK library such as ATLAS or the Intel^{TM} Math Kernel Library. Without such an optimized library of subroutines, the computation time will be at least a factor of ten longer.

Our choice of noise model will be the standard power-law plus white noise model which exist in many GNSS observations (Williams et al, 2004):

$$\mathbf{C} = \sigma^2 \left(\sin^2(\phi)\mathbf{I} + \cos^2(\phi)\mathbf{E}(\alpha)\right) \tag{32}$$

where σ is the innovation noise and $\mathbf{I}$ is the unit matrix, which is the covariance matrix for white noise, and $\mathbf{E}$ is the unit covariance matrix for power-law that depends on the spectral index α $(=-\kappa)$. We have followed Williams (2008) by writing the variance in front of the sum of the two matrices and introducing an angle ϕ which determines the distribution of white and power-law noise. For our simulations we used the value of ϕ that we observed in the GNSS time-series of Kootwijk (KOSG) in The Netherlands, see also Bos et al (2008). However, the spectral index α of the power-law was set to one, to have exact flicker noise.

The time span of the observations was 1000 and 4000 days and have been computed using CATS, version 3.0.1 and a C++ implementation of the Eqs. (30) and (31), using the Toeplitz approximation described in section 3. It was assumed that the noise started to develop 1000 days before the first observation was taken.

The computation time, or more precisely, the wall-clock time has been plotted for various percentages of missing data points in Fig. 2. The location of the missing data have been selected randomly and for each percentage of missing data 10 different simulations were performed to get an idea of the variations in the result. The computer used had a dual-core Intel Core i3 processor, 2.26 GHz, with 4 GByte of memory.

Fig. 2 shows that the analysis of a time-series of 1000 days, which corresponds to roughly 3 years of data, requires less than a minute for both methods. However, the analysis of 4000 days, or roughly 11 years of data, takes 25 minutes using the standard method. Analysing the East, North and Up component for 50 stations would require 3 days of computations. Assuming an average of 3% of missing data, the reformulated method requires 12 seconds for each separate analysis, corresponding to a total computation time of half an hour, or 125 times faster than the standard method.

The estimated spectral index α estimated by CATS minus the index estimated using the reformulated MLE program have been plotted in Fig. 3. One can see that the new method lightly underestimates the spectral index but for the longer time-series the difference is minimal. The reason is the use of our Toeplitz approximation of the covariance matrix. We tested this by

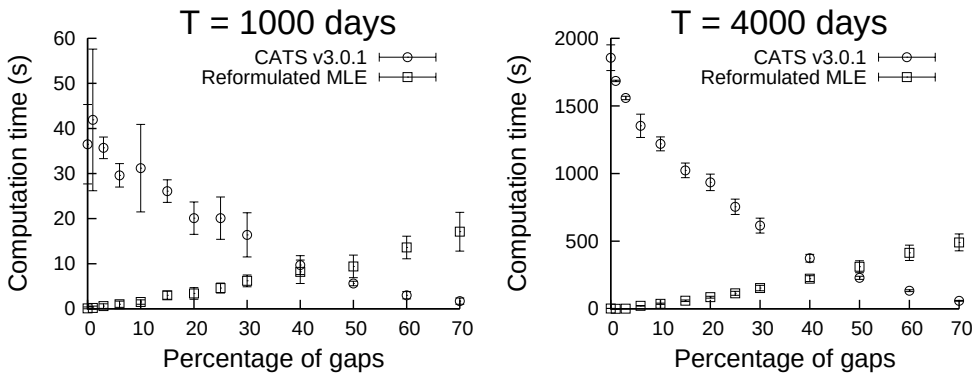

Figure 2: The computation time required by CATS and our new program using the reformulated MLE equations for synthetic time-series with a data span of 1000 and 4000 days and varying amount of missing data points.

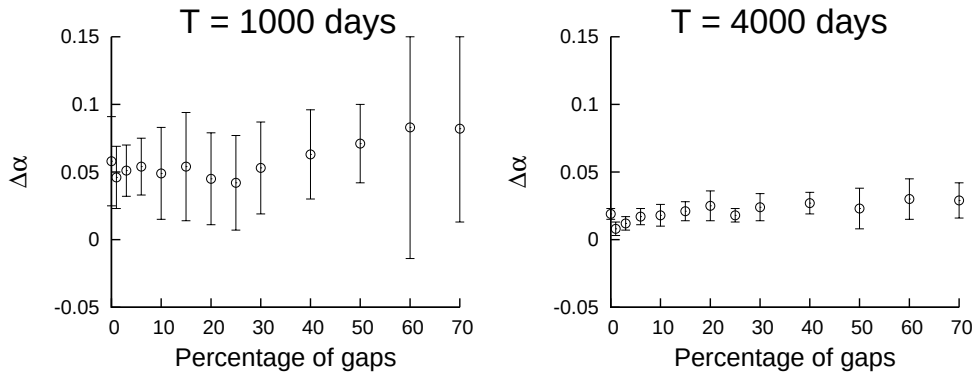

Figure 3: The difference in the estimated spectral index α of CATS minus our reformulated MLE program.

also implementing the standard method, using the non-Toeplitz covariance matrix in our program. The results were in very close agreement with the results from CATS.

It is of course essential that both methods give the same value for the estimated trend. Their difference has been plotted in Fig. 4 which shows that there is no significant discrepancy. The same has been done for the estimated trend error which is plotted in Fig. 5. It can be seen that the use of our covariance matrix gives a slightly larger error bar because the noise has had more time to develop, it is non-stationary, to larger values. Bos et al (2008) explain this phenomenon with more detail.

So far we have assumed that the missing data points are randomly distributed over the time-series. To investigate if the location of the missing data points has an influence on the computation time, we have repeated the

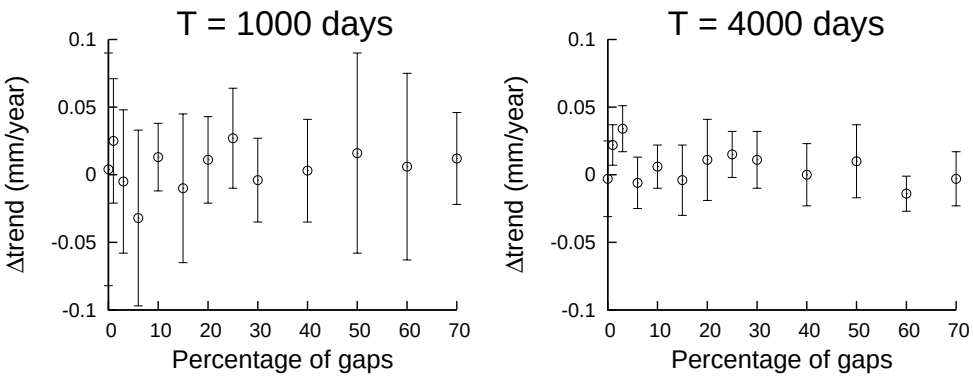

Figure 4: The same as Fig. 3 but for the estimated trend value. Shown is CATS minus our reformulated MLE program.

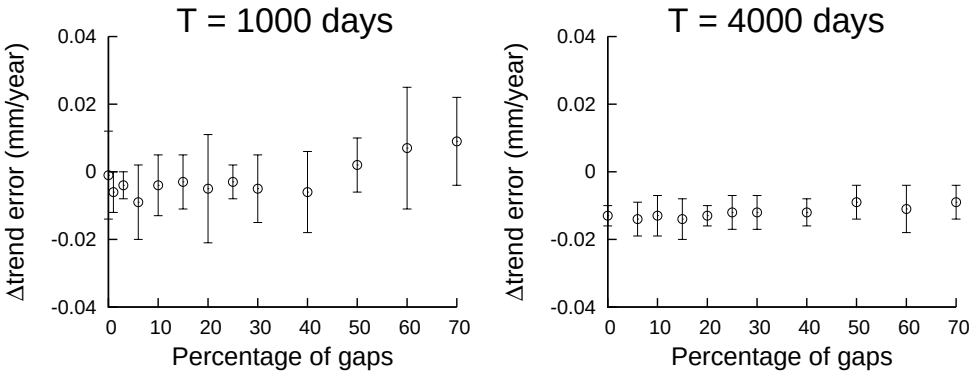

Figure 5: The same as Fig. 3 but for the estimated trend error. Shown is CATS minus our reformulated MLE program.

above described Monte Carlo simulations but with all missing data points put together into one sequence of missing data. Two runs were computed: one run with this large sequence of missing data put in the middle and one run with this sequence put at the end of the time-series, just before the last observation. In both cases no noticeable difference in computation time was observed.

6 Results with real data

In the previous section we analysed synthetic data but the new algorithm also performs well on real GNSS data as we will show in this section. We have selected 16 global stations of the International GNSS Service (Dow et al, 2009) that have between 8 and 15 years of good observations. Their names, data span in days and the percentage of missing data are given in the first three columns of Table 1. The raw GNSS observations were processed with the GIPSY-OASIS II software (Webb and Zumberge, 1995) using the PPP method (Zumberge et al, 1997) to produce daily solutions that were subsequently mapped into ITRF2008 (Altamimi et al, 2011).

Besides missing data, these time-series contain outliers that were eliminated by removing all observations for which the residual value, obtained by subtraction of the modelled signal, fell outside our chosen threshold of three times the interquartile range. This modelled signal was estimated by ordinary least-squares (thus assuming white noise) and repeated until there were no more outliers left in the time-series. The cleaned data were then analysed both with CATS and the new algorithm. The results for the East and North component are listed in columns 4 to 11 of Table 1. For these computations a quad-core Intel Core i3 processor, 2.1 Ghz, with 4 GByte of memory was used.

Table 1 clearly shows that a factor of 10-100 in computation time is gained with the new algorithm, while similar results are obtained. The highest gain is obtained for series with the smallest amount of missing data which is in agreement with the results of section 5. The speed up factors, thus the computation times of CATS divided by those of the new algorithm, for the North and East component have been plotted as function of the amount of percentage of missing data in Fig. 6. In this figure also the speed up factors obtained for the synthetic data with data spans of 1000 and 4000 days are shown.

The slight differences, although well within the uncertainties, in estimated trend and trend error values in Table 1 are caused by our approximation of the covariance matrix. Since MLE is an iterative method that converges to a solution, one has to decide when a good enough solution has been obtained. We used the default thresholds of CATS and we confirmed that our threshold was good enough by decreasing it by a factor of 10 and re-

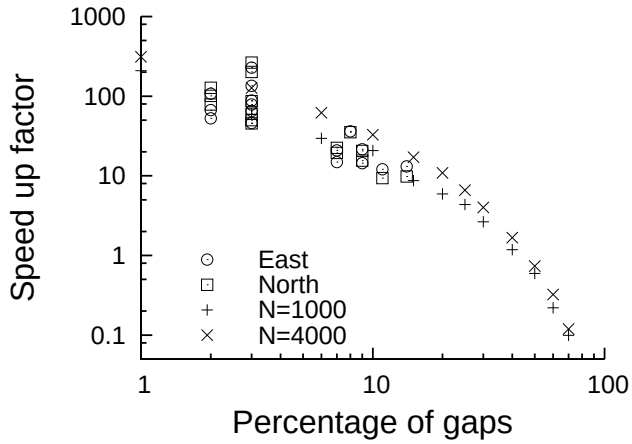

Figure 6: The speed up factor (CATS computation time divided by the time needed by the new algorithm) for various percentages of missing data. Also shown are the speed up factors for the analysis of synthetic data shown in Fig. 2.

computing the results. We observed only some changes in the third decimal value which indicates our chosen threshold is sufficient.

For the results computed with our new algorithm, it was assumed that the noise started to develop 5000 days before the first observations. To verify that this assumption does not have a large influence on the results, we also performed the same analysis with the noise starting at the same time as the first observation. The differences were only noticeable in the second and third decimal of the estimated spectral index α and of the estimated trend and trend error. Since the spectral index α ranged from 0.8 to 1.5, this shows that our approximation of the covariance matrix gives very satisfactory results.

7 Discussion and Conclusions

We have shown how the MLE method can be reformulated to gain a factor of 10-100 in computation time compared to the standard MLE method for the estimation of the linear trend in GNSS time-series with up to 20% of missing data points. This is an improvement with respect to the results of Bos et al (2008) because it is no longer necessary to fill the missing data by interpolation.

The influence of time correlated noise on time series and the presence of reasonably fast method to include coloured noise in the analysis of time-series, such as shown in this paper and by Hackl et al (2011), strongly suggest that people should routinely add time correlated noise analysis in

Table 1: The results of the time-series analysis using both CATS and the new algorithm, East and North component. Given are the estimated trends with the standard deviations in $mm/year$ and the computation times t in seconds.

			East				North			
	n	miss	CATS		new algorithm		CATS		new algorithm	
Station	(days)	(%)	trend $\pm$ σ	t	trend $\pm$ σ	t	trend $\pm$ σ	t	trend $\pm$ σ	t
ALRT	2890	3	-21.71 $\pm$ 0.84	406	-21.52 $\pm$ 0.96	3	5.90 $\pm$ 0.94	408	5.44 $\pm$ 1.00	2
BOR1	5332	3	20.18 $\pm$ 0.09	2111	20.18 $\pm$ 0.11	24	14.86 $\pm$ 0.05	2077	14.82 $\pm$ 0.06	24
BRUS	5294	3	17.68 $\pm$ 0.11	1882	17.71 $\pm$ 0.13	41	15.68 $\pm$ 0.14	2134	15.71 $\pm$ 0.17	47
DRAO	5369	2	-13.05 $\pm$ 0.06	1873	-13.04 $\pm$ 0.07	28	-10.38 $\pm$ 0.04	2162	-10.38 $\pm$ 0.04	17
HLFX	2861	2	-15.19 $\pm$ 0.05	751	-15.19 $\pm$ 0.05	7	8.77 $\pm$ 0.05	705	8.76 $\pm$ 0.06	7
HNLC	4236	14	-62.55 $\pm$ 0.07	1206	-62.55 $\pm$ 0.07	92	34.56 $\pm$ 0.05	2104	34.56 $\pm$ 0.05	214
HYDE	2777	7	42.15 $\pm$ 0.45	313	42.20 $\pm$ 0.52	15	33.34 $\pm$ 0.76	291	33.33 $\pm$ 0.84	13
MATE	5356	2	23.76 $\pm$ 0.17	1687	23.80 $\pm$ 0.20	32	19.29 $\pm$ 0.09	2046	19.25 $\pm$ 0.10	26
METS	4980	9	19.84 $\pm$ 0.08	1612	19.87 $\pm$ 0.09	75	12.85 $\pm$ 0.06	1686	12.85 $\pm$ 0.07	82
NRIL	3465	8	22.08 $\pm$ 0.06	542	22.09 $\pm$ 0.06	15	-2.09 $\pm$ 0.11	601	-2.07 $\pm$ 0.12	17
NYAL	4851	11	10.23 $\pm$ 0.09	1555	10.22 $\pm$ 0.10	129	14.18 $\pm$ 0.11	1323	14.14 $\pm$ 0.13	141
ONSA	5310	3	17.08 $\pm$ 0.04	2126	17.07 $\pm$ 0.05	27	14.73 $\pm$ 0.05	1861	14.72 $\pm$ 0.05	31
PERT	4979	9	39.08 $\pm$ 0.13	1646	39.04 $\pm$ 0.15	114	57.78 $\pm$ 0.15	1566	57.75 $\pm$ 0.18	101
TIXI	4169	7	16.92 $\pm$ 0.22	909	16.94 $\pm$ 0.26	61	-11.75 $\pm$ 0.20	988	-11.73 $\pm$ 0.24	51
WSRT	4773	3	17.57 $\pm$ 0.05	1632	17.55 $\pm$ 0.06	25	16.38 $\pm$ 0.05	1403	16.40 $\pm$ 0.06	28
WTZR	5322	3	20.50 $\pm$ 0.06	2055	20.51 $\pm$ 0.07	9	15.47 $\pm$ 0.04	2115	15.47 $\pm$ 0.05	8

the uncertainties estimation.

The new algorithm is highly parallelizable which is beneficial to run it on a cluster computer or for grid computing. For longer time-series one could implement the super-fast Toeplitz solver to reduce the number of operations for the inverse Cholesky decomposition, Eq. (24), even further from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log_2^2 n)$.

Although the processing speed of new computers continues to grow exponentially over the year, this new algorithm opens the possibility to analyse even more challenging problems such as the simultaneous time-series analysis of several stations, allowing both for spatial and temporal correlations (Amiri-Simkooei, 2009).

Agnew (1992) noted that most geophysical signals exhibit temporal correlation, mostly in the form of some kind of power-law, and therefore the algorithm we described can also be helpful to analyse tide gauge, strain and gravity data.

8 acknowledgements

The authors would like to thank the assistant editor Jeff Freymueller and three anonymous reviewers for their thorough and constructive review of our manuscript.

A Derivation of key equations

In this appendix we give the derivations of Eqs. (7) and (8) that were presented in section 2. These derivation make use of Jacobi's determinant identity and the formula for the inverse of a block matrix (Brualdi and Schneider, 1983). To start, assume that our square covariance matrix $\mathbf{C}$, size $n \times n$, can be partitioned into four sub-matrices:

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{oo} & \mathbf{C}_{om} \\ \mathbf{C}_{mo} & \mathbf{C}_{mm} \end{pmatrix} \tag{33}$$

where the subscripts m and o represent the rows and columns of missing and observed data respectively. Consequently, $\mathbf{C}_{mm}$ is a square $m \times m$ matrix.

Using Gaussian elimination we can reduce $\mathbf{C}_{mo}$ to a zero matrix:

$$\begin{pmatrix} \mathbf{I}^{n-m} & \mathbf{0} \\ -\mathbf{C}_{mo}\mathbf{C}_{oo}^{-1} & \mathbf{I}^{m} \end{pmatrix} \begin{pmatrix} \mathbf{C}_{oo} & \mathbf{C}_{om} \\ \mathbf{C}_{mo} & \mathbf{C}_{mm} \end{pmatrix} = \begin{pmatrix} \mathbf{C}_{oo} & \mathbf{C}_{om} \\ \mathbf{0} & \mathbf{C}'_{mm} \end{pmatrix} \tag{34}$$

where $\mathbf{C}'_{mm} = \mathbf{C}_{mm} - \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}\mathbf{C}_{om}$ which is called the Schur complement of $\mathbf{C}_{oo}$ in $\mathbf{C}$. When we take the determinant of Eq. (34), we obtain:

$$\det(\mathbf{C}) = \det(\mathbf{C}_{oo}) \det(\mathbf{C}'_{mm}) \tag{35}$$

Using the matrices $\breve{\mathbf{C}}$ ($=\mathbf{C}_{oo}$) and $\mathbf{F}$ that were introduced in section 2, we can rewrite Eq. (35) as:

$$\det(\mathbf{C}) = \det(\breve{\mathbf{C}}) \det\left((\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\right) \tag{36}$$

The last term of Eq. (36) can be obtained by realising that the two multiplications with matrix $\mathbf{F}$ do nothing more than selecting a submatrix of matrix $\mathbf{C}^{-1}$. The relation with $\mathbf{C}'_{mm}$ can be found by looking at the formula for the matrix inverse for a block matrix (Brualdi and Schneider, 1983):

$$\mathbf{C}^{-1} = \begin{pmatrix} \mathbf{C}_{oo}^{-1} + \mathbf{C}_{oo}^{-1}\mathbf{C}_{om}(\mathbf{C}'_{mm})^{-1}\mathbf{C}_{mo}\mathbf{C}_{oo}^{-1} & -\mathbf{C}_{oo}^{-1}\mathbf{C}_{om}(\mathbf{C}'_{mm})^{-1} \\ -(\mathbf{C}'_{mm})^{-1}\mathbf{C}_{mo}\mathbf{C}_{oo}^{-1} & (\mathbf{C}'_{mm})^{-1} \end{pmatrix} \tag{37}$$

one will note the equivalence of this submatrix with the inverse of the Schur complement. Thus, we have:

$$\mathbf{C}'_{mm} = \left(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F}\right)^{-1} \tag{38}$$

By taking the logarithm of Eq. (36), and using the relation $\det(\mathbf{C}) = 1/\det(\mathbf{C}^{-1})$, Eq. (8) of section 2 is obtained.

Of course the missing data will normally not occur after all observations have been made. Determinants have the property that rows and columns of the matrix can be interchanged without changing the value except for a change in sign. However, when rows i and j are swapped of the covariance matrix, the columns i and j are swapped at the same time which means that the sign does not change. Therefore, Eq. (8) is valid for any sequence of missing data.

Next, using Eq. (38) we have:

$$\mathbf{F}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^T = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}'_{mm} \end{pmatrix} \tag{39}$$

Using Eq. (37), this leads to the following result:

$$\mathbf{C}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{C}^{-1} = \begin{pmatrix} \mathbf{C}_{oo}^{-1}\mathbf{C}_{om}(\mathbf{C}'_{mm})^{-1}\mathbf{C}_{mo}\mathbf{C}_{oo}^{-1} & -\mathbf{C}_{oo}^{-1}\mathbf{C}_{om}(\mathbf{C}'_{mm})^{-1} \\ -(\mathbf{C}'_{mm})^{-1}\mathbf{C}_{mo}\mathbf{C}_{oo}^{-1} & (\mathbf{C}'_{mm})^{-1} \end{pmatrix} \tag{40}$$

Changing the sign of Eq. (40) and adding $\mathbf{C}^{-1}$ finally gives us:

$$\mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{C}^{-1} = \begin{pmatrix} \mathbf{C}_{oo}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{41}$$

This provides us the proof for Eq. (7) in section 2:

$$\breve{\mathbf{r}}^T\breve{\mathbf{C}}^{-1}\breve{\mathbf{r}} = \mathbf{r}_o^T\left(\mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{F}(\mathbf{F}^T\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{C}^{-1}\right)\mathbf{r}_o \tag{42}$$

Swapping rows i and j of the covariance matrix $\mathbf{C}$ will cause a swap of columns i and j of matrix $\mathbf{C}^{-1}$. Swapping the columns of $\mathbf{C}$ will cause a swap of the rows of $\mathbf{C}^{-1}$ in a similar way. Thus, we can reshuffle our set of observations and missing data, while at the same time adjusting matrix $\mathbf{F}$, to obtain the form of Eq. (34) which proves the general validity of Eq. (42) for any sequence of missing data.

References

Agnew DC (1992) The time-domain behaviour of power-law noises. Geophys Res Letters 19(4):333–336

Altamimi Z, Collilieux X, Mtivier L (2011) ITRF2008: an improved solution of the international terrestrial reference frame. J Geod 85(8):457–473, DOI 10.1007/s00190-011-0444-4

Amiri-Simkooei AR (2009) Noise in multivariate GPS position time-series. J Geod 83:175–187, DOI 10.1007/s00190-008-0251-8

Ammar GS, Gragg WB (1988) Superfast solution of real positive definite Toeplitz systems. SIAM J Matrix Anal Appl 9:61–76

Bos MS, Fernandes RMS, Williams SDP, Bastos L (2008) Fast error analysis of continuous GPS observations. J Geod 82:157–166, DOI 10.1007/s00190-007-0165-x

Brualdi RA, Schneider H (1983) Determinantal Identities:Gauss, Schur, Cauchy, Sylvester, Kronecker, Jacobi, Binet, Laplace, Muir, and Cayley. Linear Algebra and its Applications 52/53:769–791

Dow JM, Neilan RE, Rizos C (2009) The International GNSS Service in a changing landscape of Global Navigation Satellite Systems. J Geod 83:191–198, DOI 10.1007/s00190-008-0300-3

Gohberg IC, Semencul AA (1972) On the inversion of finite Toeplitz matrices and their continuous analogs. Mat Issled 7(2):201–223

Hackl M, Malservisi R, Hugentobler U, Wonnacott R (2011) Estimation of velocity uncertainties from GPS time series: Examples from the analysis of the South African TrigNet network. J Geophys Res 116(B15):B11404, DOI 10.1029/2010JB008142

Hosking JRM (1981) Fractional differencing. Biometrika 68:165–176

Johnson HO, Agnew DC (1995) Monument motion and measurements of crustal velocities. Geophys Res Letters 22(21):2905–2908

Kasdin NJ (1995) Discrete simulation of colored noise and stochastic processes and $1/f^{\alpha}$ power-law noise generation. Proc IEEE 83(5):802–827

Langbein J (2004) Noise in two-color electronic distance meter measurements revisited. J Geophys Res 109(B04406), DOI 10.1029/2003JB002819

Langbein J (2010) Computer algorithm for analyzing and processing borehole strainmeter data. Computers & Geosciences 36(5):611–619, DOI 10.1016/j.cageo.2009.08.011

Langbein J (2012) Estimating rate uncertainty with maximum likelihood: differences between power-law and flicker–random-walk models. Journal of Geodesy 86:775–783, DOI 10.1007/s00190-012-0556-5

Mao A, Harrison CGA, Dixon TH (1999) Noise in GPS coordinate time series. J Geophys Res 104(B2):2797–2816, DOI doi:10.1029/1998JB900033

Santamaría-Gómez A, Bouin MN, Collilieux X, Wöppelmann G (2011) Correlated errors in GPS position time series: Implications for velocity estimates. J Geophys Res 116(B15):B01405, DOI 10.1029/2010JB007701

Trench WF (1964) An algorithm for the inversion of finite Toeplitz matrices. J Soc Indust Appl Math 12:515–522

Webb FH, Zumberge JF (1995) An introduction to GIPSY/OASIS-II. Tech. Rep. JPL D-11088, California Institute of Technology, Pasadena, CA.

Williams SDP (2003) The effect of coloured noise on the uncertainties of rates from geodetic time series. J Geod 76(9-10):483–494, DOI 10.1007/s00190-002-0283-4

Williams SDP (2008) CATS : GPS coordinate time series analysis software. GPS Solutions 12(2):147–153, DOI 10.1007/s10291-007-0086-4

Williams SDP, Bock Y, Fang P, Jamason P, Nikolaidis RM, Prawirodirdjo L, Miller M, Johnson DJ (2004) Error analysis of continuous GPS position time series. J Geophys Res 109(B03412), DOI 10.1029/2003JB002741

Zhang J, Bock Y, Johnson H, Fang P, Williams S, Genrich J, Wdowinski S, Behr J (1997) Southern California Permanent GPS Geodetic Array: Error analysis of daily position estimates and site velocities. J Geophys Res 102(B8):18,035–18,055

Zumberge JF, Heflin MB, Jefferson DC, Watkins MM, Webb FH (1997) Precise point positioning for the efficient and robust analysis of GPS data from large networks. J Geophys Res 102(B3):5005–5018, DOI 10.1029/96JB03860