Chapter 2
Introduction to Geodetic Time Series Analysis

Machiel S. Bos, Jean-Philippe Montillet, Simon D. P. Williams and Rui M. S. Fernandes

Abstract The previous chapter gave various examples of geophysical time series and the various trajectory models that can be fitted to them. In this chapter we will focus on how the parameters of the trajectory model can be estimated. It is meant to give researchers new to this topic an easy introduction to the theory with references to key books and articles where more details can be found. In addition, we hope that it refreshes some of the details for the more experienced readers. We pay special attention to the modelling of the noise which has received much attention in the literature in the last years and highlight some of the numerical aspects. The subsequent chapters will go deeper into the theory, explore different aspects and describe the state of art of this area of research.

Keywords GNSS time series · Linear model · Power-law noise · Bayesian analysis · Maximum likelihood estimation

The original version of this chapter was revised: Electronic Supplementary Materials have been added. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-21718-1_14

Electronic supplementary material
The online version of this chapter (https://doi.org/10.1007/978-3-030-21718-1_2) contains supplementary material, which is available to authorized users.

M. S. Bos (✉) · R. M. S. Fernandes
Instituto Dom Luiz, Universidade da Beira Interior, Covilha, Portugal
e-mail: machiel@segal.ubi.pt

R. M. S. Fernandes
e-mail: rui@segal.ubi.pt

J.-P. Montillet
Space and Earth Geodetic Analysis Laboratory, Universidade da Beira Interior, Covilha, Portugal
e-mail: jpmontillet@segal.ubi.pt

Institute of Earth Surface Dynamics, University of Lausanne, Neuchatel, Lausanne, Switzerland

S. D. P. Williams
National Oceanographic Centre, Liverpool, United Kingdom
e-mail: sdwil@noc.ac.uk

J.-P. Montillet and M. S. Bos (eds.), *Geodetic Time Series Analysis in Earth Sciences*, Springer Geophysics,
https://doi.org/10.1007/978-3-030-21718-1_2

2.1 Gaussian Noise and the Likelihood Function

Geodetic time series consist out of a set observations at various epochs. These observations, stored in a vector $\mathbf{y}$, are not perfect but contain noise which can be described as a set of multivariate random variables. Let us define this as the vector $\mathbf{w} = [W_1, W_2, W_3, \dots, W_N]$ where each W_i is a random variable. If $f(w)$ is the associated probability density function, then the first moment μ_1, the mean of the noise, is defined as Casella and Berger (2001):

$$\mu_1 = E[W] = \int_{-\infty}^{\infty} w f(w)\, dw \tag{2.1}$$

where E is the expectation operator. It assigns to each possible value of random variable w a weight $f(w)$ over an infinitely small interval of dw, sums each of them to obtain the mean expected value $E[W]$. The second moment μ_2 is defined in a similar manner:

$$\mu_2 = E[W^2] = \int_{-\infty}^{\infty} w^2 f(w)\, dw = \int_{-\infty}^{\infty} w^2 dF(w) \tag{2.2}$$

The last term F is the cumulative distribution. For zero mean, the second moment is better known as the variance. Since we have N random variables, we can compute variances for $E[W_i W_j]$, where both i and j range from 1 to N. The result is called the covariance matrix. In this book, the probability density function $f(w)$ is assumed to be a Gaussian:

$$f(w|\mu_1, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(w-\mu_1)^2}{2\sigma^2}\right] \tag{2.3}$$

where σ is the standard deviation, the square-root of the variance of random variable w. This function is very well known and is shown in Fig. 2.1 for zero μ_1.

The standard error is defined as the 1-σ interval and contains on average 68% of the observed values of w. The reason why it is so often encountered in observations is that the central limit theorem states that the sum of various continuous probability distributions always tends to the Gaussian one. An additional property of the Gaussian probability density function is that all its moments higher than two (μ_3, $\mu_4, \dots$) are zero. Therefore, the mean and the covariance matrix provide a complete description of the stochastic properties. Actually, we will always assume that the mean of the noise is zero and therefore only need the covariance matrix. The term in front of the exponential is needed to ensure that the integral of $f(x)$ from $-\infty$ to ∞ produces 1. That is, the total probability of observing a value between these limits is 1, as it should be. We have not one, but several observations with noise in our time series. The probability density function of the multi-variate noise is:

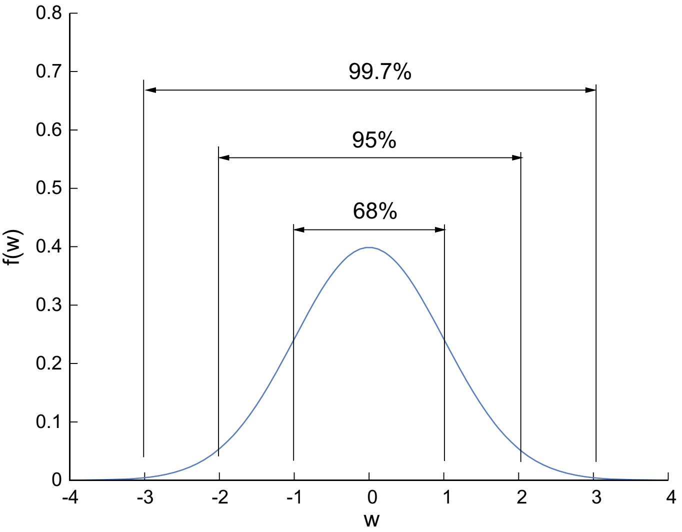

Fig. 2.1 The Gaussian probability density function, together with the 1, 2 and 3 σ intervals

$$f(\mathbf{w}|\mathbf{C}) = \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{C})}} \exp\left[-\tfrac{1}{2}\mathbf{w}^T\mathbf{C}^{-1}\mathbf{w}\right] \tag{2.4}$$

We assumed that the covariance matrix **C** is known. The expression $f(\mathbf{w}|\mathbf{C})$ should be read as the probability density function f for variable w, for given and fixed covariance matrix **C**. Next, we assume that our observations can be described by our model $\mathbf{g}(\mathbf{x}, t)$, where **x** are the parameters of the model and t the time. The observations are the sum of our model plus the noise:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, t) + \mathbf{w} \quad \text{or} \quad \mathbf{w} = \mathbf{y} - \mathbf{g}(\mathbf{x}, t) \tag{2.5}$$

The noise **w** is described by our Gaussian probability density function with zero mean and covariance matrix **C**. The probability that we obtained the actual values of our observations is:

$$f(\mathbf{y}|\mathbf{x}, \mathbf{C}) = \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{C})}} \exp\left[-\tfrac{1}{2}(\mathbf{y} - \mathbf{g}(\mathbf{x}, t))^T\mathbf{C}^{-1}(\mathbf{y} - \mathbf{g}(\mathbf{x}, t))\right] \tag{2.6}$$

However, we don't know the true values of **x** or the covariance matrix **C**. We only know the observations. Consequently, we need to rephrase our problem as follows: what values of **x** and **C** would produce the largest probability that we observe **y**? Thus, we are maximising $f(\mathbf{x}, \mathbf{C}|\mathbf{y})$ which we call the likelihood function L. Furthermore, we normally work with the logarithm of it which is called the log-likelihood:

$$\ln(L) = -\frac{1}{2}\left[N\ln(2\pi) + \ln(\det(\mathbf{C})) + (\mathbf{y} - \mathbf{g}(\mathbf{x}, t))^T \mathbf{C}^{-1}(\mathbf{y} - \mathbf{g}(\mathbf{x}, t))\right] \tag{2.7}$$

We need to find value of $\mathbf{x}$ to maximise this function and the method is therefore called Maximum Likelihood Estimation (MLE). The change from $f(\mathbf{y}|\mathbf{x}, \mathbf{C})$ to $f(\mathbf{x}, \mathbf{C}|\mathbf{y})$ is subtle. Assume that the covariance matrix $\mathbf{C}$ also depends on parameters that we store in vector $\mathbf{x}$. In this way, we can simplify the expression $f(\mathbf{y}|\mathbf{x}, \mathbf{C})$ to $f(\mathbf{y}|\mathbf{x})$. Bayes' Theorem, expressed in terms of probability distributions gives us:

$$f(\mathbf{x}|\mathbf{y}) = \frac{f(\mathbf{y}|\mathbf{x})f(\mathbf{x})}{f(\mathbf{y})} \tag{2.8}$$

where $f(\mathbf{y})$ and $f(\mathbf{x})$ are our prior probability density function for the observations $\mathbf{y}$ and parameters $\mathbf{x}$, respectively. These represent our knowledge about what observations and parameter values we expect before the measurements were made. If we don't prefer any particular values, these prior probability density functions can be constants and they will have no influence on the maximising of the likelihood function $f(\mathbf{x}|\mathbf{y}) = L$.

Another subtlety is that we changed from random noise and fixed parameter values of the trajectory model $f(\mathbf{y}|\mathbf{x})$ to fixed noise and random parameters of the trajectory model $f(\mathbf{x}|\mathbf{y})$. If the trajectory model is for example a linear tectonic motion then this is a deterministic, fixed velocity, not a random one. However, one should interpret $f(\mathbf{x}|\mathbf{y})$ as our degree of trust, our confidence that the estimated parameters $\mathbf{x}$ are correct. See also Koch (1990, 2007) and Jaynes (2003). The last one is particularly recommended to learn more about Bayesian statistics.

2.2 Linear Models

So far we simply defined our trajectory model as $\mathbf{g}(\mathbf{x}, t)$. An important class of models that are fitted to the observations are linear models. These are defined as:

$$\mathbf{g}(\mathbf{x}, t) = x_1 g_1(t) + x_2 g_2(t) + \cdots + x_M g_M(t) \tag{2.9}$$

where x_1 to x_M are assumed to be constants. We can rewrite this in matrix form as follows:

$$\mathbf{g}(\mathbf{x}, t) = \begin{pmatrix} g_1(t_1) & g_2(t_1) & \dots & g_M(t_1) \\ g_1(t_2) & g_2(t_2) & & g_M(t_2) \\ \vdots & & \vdots & \\ g_1(t_N) & g_2(t_N) & & g_M(t_N) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_M \end{pmatrix} = \mathbf{A}\mathbf{x} \tag{2.10}$$

Matrix $\mathbf{A}$ is called the design matrix. From Chap. 1 we know that tectonic motion or sea level rise can be modelled by a linear trend (i.e. the Standard Linear Trajectory Model). Thus $g_1(t)$ is a constant and $g_2(t)$ a linear trend. This can be extended to a

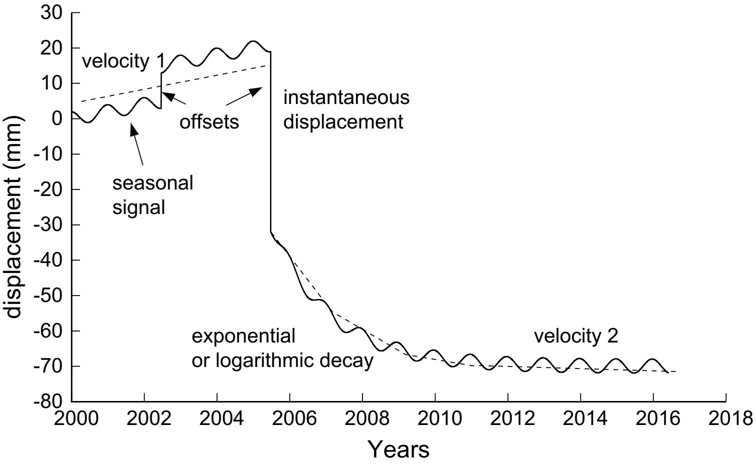

Fig. 2.2 Sketch of a trajectory model containing common phenomena

higher degree polynomial to model acceleration for example. Next, in many cases an annual and semi-annual signal is included as well. A periodic signal can be described by its amplitude b_k and its phase-lag ψ_k with respect to some reference epoch:

$$\begin{aligned} g(t) &= b_k \cos(\omega_k t - \psi_k) \\ &= b_k \cos \psi \cos(\omega_k t) + b_k \sin \psi_k \sin(\omega_k t) \\ &= c_k \cos(\omega_k t) + s_k \sin(\omega_k t) \end{aligned} \tag{2.11}$$

Since the unknown phase-lag ψ_k makes the function non-linear, one must almost always estimate the amplitudes c_k and s_k, see Chap. 1. These parameters are linear with functions cos and sin, and derive from these values the amplitude b_k and phase-lag ψ_k.

Other models that can be included in $g(t)$ are offsets and post-seismic relaxation functions, see Chap. 1. An example of a combination of all these models into a single trajectory model is shown in Fig. 2.2.

For linear models, the log-likelihood can be rewritten as:

$$\ln(L) = -\frac{1}{2}\left[N \ln(2\pi) + \ln(\det(\mathbf{C})) + (\mathbf{y} - \mathbf{A}\mathbf{x})^T \mathbf{C}^{-1}(\mathbf{y} - \mathbf{A}\mathbf{x})\right] \tag{2.12}$$

This function must be maximised. Assuming that the covariance matrix is known, then it is a constant and does not influence finding the maximum. Next, the term $(\mathbf{y} - \mathbf{A}\mathbf{x})$ represent the observations minus the fitted model and are normally called the residuals $\mathbf{r}$. It is desirable to choose the parameters $\mathbf{x}$ in such a way to make these residuals small. The last term can be written as $\mathbf{r}^T \mathbf{C}^{-1}\mathbf{r}$ and it is a quadratic function, weighted by the inverse of matrix $\mathbf{C}$.

Now let us compute the derivative of $\ln(L)$:

$$\frac{d\ln(L)}{d\mathbf{x}} = \mathbf{A}^T\mathbf{C}^{-1}\mathbf{y} - \mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\mathbf{x} \tag{2.13}$$

The minimum of $\ln(L)$ occurs when this derivative is zero. Thus:

$$\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{C}^{-1}\mathbf{y} \quad \rightarrow \quad \mathbf{x} = \left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}^{-1}\mathbf{y} \tag{2.14}$$

This is the celebrated weighted least-squares equation to estimate the parameters $\mathbf{x}$. Most derivations of this equation focus on the minimisation of the quadratic cost function. However, here we highlight the fact that for observations that contain Gaussian multivariate noise, the weighted least-squares estimator is a maximum likelihood estimator (MLE). From Eq. (2.14) it can also be deduced that vector $\mathbf{x}$, like the observation vector $\mathbf{y}$, follows a multi-variate Gaussian probability density function.

The variance of the estimated parameters estimated is:

$$\begin{aligned} var(\mathbf{x}) &= var\left(\left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}^{-1}\mathbf{y}\right) \\ &= \left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}^{-1}var(\mathbf{y})\ \mathbf{C}^{-1}\mathbf{A}\left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1} \\ &= \left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{C}^{-1}\mathbf{C}\ \mathbf{C}^{-1}\mathbf{A}\left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1} \\ &= \left(\mathbf{A}^T\mathbf{C}^{-1}\mathbf{A}\right)^{-1} \end{aligned} \tag{2.15}$$

Next, define the following matrix $\mathcal{I}(\mathbf{x})$:

$$\mathcal{I}(\mathbf{x}) = -E\left[\frac{\partial^2}{\partial\mathbf{x}^2}\ln(L)\right] = -\int\left(\frac{\partial^2}{\partial\mathbf{x}^2}\ln(f)\right)\ f\ d\mathbf{x} \tag{2.16}$$

It is called the Fisher Information matrix. As in Eqs. (2.1) and (2.2), we use the expectation operator E. Remember that we simply called f our likelihood L but these are the same. We already used the fact that the log-likelihood as function of $\mathbf{x}$ is horizontal at the maximum value. Let us call this $\hat{\mathbf{x}}$. The second derivative is related to the curvature of the log-likelihood function. The sharper the peak near its maximum, the more accurate we can estimate the parameters $\mathbf{x}$ and therefore the smaller their variance will be.

Next, it can be shown that the following inequality holds:

$$1 \leq \int(\hat{\mathbf{x}} - \mathbf{x})^2\ f\ d\mathbf{x}\int\left(\frac{\partial\ln(f)}{\partial\mathbf{x}}\right)^2\ f\ d\mathbf{x} \tag{2.17}$$

The first integral represents the variance of $\mathbf{x}$, see Eq. (2.2). The second one, after some rewriting, is equal to the Fisher information matrix. This gives us, for any unbiased estimator, the following Cramér–Rao Lower Bound (Kay 1993):

$$var(\hat{\mathbf{x}}) \geq \frac{1}{\mathcal{I}(\mathbf{x})} \tag{2.18}$$

Equation (2.18) predicts the minimum variance of the estimated parameters $\mathbf{x}$ for given probability density function f and its relation with the parameters $\mathbf{x}$ that we want to estimate. If we use Eq. (2.13) to compute the second derivative of the log-likelihood, then we obtain:

$$\mathcal{I}(\mathbf{x}) = \mathbf{A}^T\mathbf{C}^{-1}\mathbf{A} \tag{2.19}$$

Comparing this with Eq. (2.15), one can see that for the case of the weighted least-square estimator, the Cramér–Rao Lower Bound is achieved. Therefore, it is an optimal estimator. Because we also need to estimate the parameters of the covariance matrix $\mathbf{C}$, we shall use MLE which approximates this lower bound for increasing number of observations. Therefore, one can be sure that out of all existing estimation methods, none of them will produce a more accurate result than MLE, only equal or worse. For more details, see Kay (1993).

2.3 Models for the Covariance Matrix

Least-squares and maximum likelihood estimation are well known techniques in various branches of science. In recent years much attention has been paid by geodesists to the structure of the covariance matrix. If there was no relation between each noise value, then these would be independent random variables and the covariance matrix $\mathbf{C}$ would be zero except for values on its diagonal. However, in almost all geodetical time series these are dependent random variables. In statistics this is called temporal correlation and we should consider a full covariance matrix:

$$\mathbf{C} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1N}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & & \sigma_{2N}^2 \\ \vdots & & \ddots & \vdots \\ \sigma_{N1}^2 & \cdots & \sigma_{NN-1}^2 & \sigma_{NN}^2 \end{pmatrix} \tag{2.20}$$

where σ_{12}^2 is the covariance between random variables w_1 and w_2. If we assume that the properties of the noise are constant over time, then we have the same covariance between w_2 and w_3, w_3 and w_4 and all other correlations with 1 time step separation. As a result, $\sigma_{12}^2, \sigma_{23}^2, \ldots, \sigma_{(N-1)N}^2$ are all equal. A simple estimator for it is:

$$\sigma_{12}^2 = \sigma_{23}^2 = \cdots = \sigma_{(N-1)N}^2 = \frac{1}{N-1}\sum_{i=1}^{N-1} w_i w_{i+1} \tag{2.21}$$

This is an approximation of the formula to compute the second moment, see Eq. (2.2), and it called the empirical or sample covariance matrix. Therefore, one could try the following iterate scheme: fit the linear model to the observations some a priori covariance matrix, compute the residuals and use this to estimate a more realistic covariance matrix using Eq. (2.20) and fit again the linear model to the observations until all estimated parameters have converged.

The previous chapter demonstrated that one of the purpose of the trajectory models is to estimate the linear or secular trend. For time series longer than 2 years, the uncertainty of this trend depends mainly on the noise at the lowest observed periods (Bos et al. 2008; He et al. 2019). However, the empirical covariance matrix estimation of Eq. (2.20) does not result in an accurate estimate of the noise at long periods because only a few observations are used in the computation. In fact, only the first and last observation are used to compute the variance of the noise at the longest observed period (i.e. σ_{1N}^2).

This problem has been solved by defining a model of the noise and estimating the parameters of this noise model. The estimation of the noise model parameters can be achieved using the log-likelihood with a numerical maximisation scheme but other methods exist such as least-squares variance component estimation (see Chap. 6).

The development of a good noise model started with the paper of Hurst (1957) who discovered that the cumulative water flow of the Nile river depended on the previous years. The influence of the previous years decayed according a power-law. This inspired Mandelbrot and van Ness (1968) to define the fractional Brownian motion model which includes both the power-law and fractional Gaussian noises, see also Beran (1994) and Graves et al. (2017). While this research was well known in hydrology and in econometry, it was not until the publication by Agnew (1992), who demonstrated that most geophysical time series exhibit power-law noise behaviour, that this type of noise modelling started to be applied to geodetic time series. In hindsight, Press (1978) had already demonstrated similar results but this work has not received much attention in geodesy. That the noise in GNSS time series also falls in this category was demonstrated by Johnson and Agnew (1995). Power-law noise has the property that the power spectral density of the noise follows a power-law curve. On a log-log plot, it converts into a straight line. The equation for power-law noise is:

$$P(f) = P_0\,(f/f_s)^{\kappa} \tag{2.22}$$

where f is the frequency, P_0 is a constant, f_s the sampling frequency and the exponent κ is called the spectral index.

Granger (1980), Granger and Joyeux (1980) and Hosking (1981) demonstrated that power-law noise can be achieved using fractional differencing of Gaussian noise:

$$(1 - B)^{-\kappa/2}\mathbf{v} = \mathbf{w} \tag{2.23}$$

where B is the backward-shift operator ($Bv_i = v_{i-1}$) and $\mathbf{v}$ a vector with independent and identically distributed (IID) Gaussian noise. Hosking and Granger used the parameter d for the fraction $-\kappa/2$ which is more concise when one focusses on the fractional differencing aspect. It has been adopted by people studying general statistics (Sowell 1992; Beran 1995). However, in geodesy the spectral index κ is used in the equations. Hosking's definition of the fractional differencing is:

$$\begin{aligned}(1-B)^{-\kappa/2} &= \sum_{i=0}^{\infty} \binom{-\kappa/2}{i} (-B)^i \\ &= 1 - \frac{\kappa}{2}B - \frac{1}{2}\frac{\kappa}{2}(1-\frac{\kappa}{2})B^2 + \cdots \\ &= \sum_{i=0}^{\infty} h_i\end{aligned} \tag{2.24}$$

The coefficients h_i can be viewed as a filter that is applied to the independent white noise. These coefficients can be conveniently computed using the following recurrence relation (Kasdin 1995):

$$\begin{aligned}h_0 &= 1 \\ h_i &= (i - \frac{\kappa}{2} - 1)\frac{h_{i-1}}{i} \qquad \text{for } i > 0\end{aligned} \tag{2.25}$$

One can see that for increasing i, the fraction $(i - \kappa/2 - 1)/i$ is slightly less than 1. Thus, the coefficients h_i only decrease very slowly to zero. This implies that the current noise value w_i depends on many previous values of $\mathbf{v}$. In other words, the noise has a long memory. Actually, the model of fractional Gaussian noise defined by Hosking (1981) is the basic definition of the general class of processes called Auto Regressive Integrated moving Average (Taqqu et al. 1995). If we ignore the Integrated part, then we obtain the Auto Regressive Moving Average (ARMA) model (Box et al. 2015; Brockwell and Davis 2002) which are short-memory noise models. The original definition of the ARIMA processes only considers the value of the power $\kappa/2$ in Eq. (2.24) as an integer value. Granger and Joyeux (1980) further extended the definition to a class of fractionally integrated models called FARIMA or ARFIMA, where κ is a floating value, generally in the range of $-2 < i < 2$. Montillet and Yu (2015) discussed the application of the ARMA and FARIMA models in modelling GNSS daily position time series and concluded that the FARIMA is only suitable in the presence of a large amplitude coloured noise capable of generating a distribution with large tails (i.e. random-walk, aggregations).

Equation (2.25) also shows that when the spectral index $\kappa = 0$, then all coefficients h_i are zero except for h_0. This implies that there is no temporal correlation between the noise values. In addition, Eq. (2.22) shows that this corresponds to a horizontal line in the power spectral density domain. Using the analogy of the visible light spectrum, this situation of equal power at all frequencies produces white light and it is therefore called white noise. For $\kappa \neq 0$, some values have received a specific colour. For example, $\kappa = -1$ is known as pink noise. Another name is flicker noise which seems to have originated in the study of noise of electronic devices. Red noise is defined as power-law noise with $\kappa = -2$ and produces $h_i = 1$ for all values of i.

Thus, this noise is a simple sum of all its previous values plus a new random step and is better known as random walk (Mandelbrot 1999). However, note that the spectral index κ does not need to be an integer value (Williams 2003).

One normally assumes that $v_i = 0$ for $i < 0$. With this assumption, the unit covariance between w_k and w_l with $l > k$ is:

$$C(w_k, w_l) = \sum_{i=0}^{k} h_i h_{i+(l-k)} \tag{2.26}$$

Since $\kappa = 0$ produces an identity matrix, the associated white noise covariance matrix is represented by unit matrix $\mathbf{I}$. The general power-law covariance matrix is represented by the matrix $\mathbf{J}$. The sum of white and power-law noise can be written as Williams (2003):

$$\mathbf{C} = \sigma_{pl}^2 \mathbf{J}(\kappa) + \sigma_w^2 \mathbf{I} \tag{2.27}$$

where σ_{pl} and σ_w are the noise amplitudes. It is a widely used combination of noise models to describe the noise in GNSS time series (Williams et al. 2014). Besides the parameters of the linear model (i.e. the trajectory model), maximum likelihood estimation can be used to also estimate the parameters κ, σ_{pl} and σ_w. This approach has been implemented various software packages such as CATS (Williams 2008), est_noise (Langbein 2010) and Hector (Bos et al. 2013). In recent years one also has detected random walk noise in the time series and this type has been included as well in the covariance matrix (Langbein 2012; Dmitrieva et al. 2015).

We assumed that $v_i = 0$ for $i < 0$ which corresponds to no noise before the first observation. This is an important assumption that has been introduced for a practical reason. For a spectral index κ smaller than -1, the noise becomes non-stationary. That is, the variance of the noise increases over time. If it is assumed that the noise was always present, then the variance would be infinite.

Most GNSS time series contain flicker noise which is just non-stationary. Using the assumption of zero noise before the first observation, the covariance matrix still increases over time but remains finite.

For some geodetic time series, such as tide gauge observations, the power-law behaviour in the frequency domain shows a flattening below some threshold frequency. To model such behaviour, Langbein (2004) introduced the Generalised Gauss–Markov (GGM) noise model which is defined as:

$$(1 - \phi B)^{-\kappa/2} \mathbf{v} = \mathbf{w} \tag{2.28}$$

The only new parameter is ϕ. The associated recurrence relation to compute the new coefficients h_i is:

$$\begin{aligned} h_0 &= 1 \\ h_i &= (i - \frac{\kappa}{2} - 1)\phi \frac{h_{i-1}}{i} \qquad \text{for } i > 0 \end{aligned} \tag{2.29}$$

If $\phi = 1$, then we obtain again our pure power-law noise model. For any value of ϕ slightly smaller than one, this term helps to shorten the memory of noise which makes it stationary. That is, the temporal correlation decreases faster to zero for increasing lag between the noise values. The power-spectrum of this noise model shows a flattening below some threshold frequency which guarantees that the variance is finite and that the noise is stationary. Finally, it is even possible to generalise this a bit more to a fractionally integrated generalised Gauss–Markov model (FIGGM):

$$\begin{gathered}(1-\phi B)^{-\kappa_1/2}(1-B)^{\kappa_2/2}\mathbf{v} = \mathbf{w} \\ (1-\phi B)^{-\kappa_1/2}\mathbf{u} = \mathbf{w}\end{gathered} \tag{2.30}$$

This is just a combination of the two previous models. One can first apply the power-law filter to **v** to obtain **u** and afterwards apply the GGM filter on it to obtain **w**. Other models will be discussed in this book, such as ARMA (Box et al. 2015; Brockwell and Davis 2002), but the power-law, GGM and FIGGM capture nicely the long memory property that is present in most geodetic time series. A list of all these noise models and their abbreviation is given in Table 2.1.

2.4 Power Spectral Density

Figure 2.3 shows examples of white, flicker and random walk noise for a displacement time series. One can see that the white noise varies around a stable mean while the random walk is clearly non-stationary and deviates away from its initial position.

In the previous section we mentioned that power-law noise has a specific curve in the power spectral density plots. Methods to compute those plots are given by Buttkus (2000). A simple but effective method is based on the Fourier transform that states that each time series with finite variance can be written as a sum of periodic signals:

Table 2.1 Common abbreviation of noise models

Noise model	Abbreviation
Auto-Regressive Moving Average	ARMA
Auto-Regressive Fractionally Integrated Moving Average	ARFIMA or FARIMA
Flicker Noise	FN
Fractionally Integrated GGM	FIGGM
Generalised Gauss Markov	GGM
Power-Law	PL
Random Walk	RW
White Noise	WN

$$y_n = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} Y_k \cdot e^{i2\pi kn/N} \quad \text{for } n = [0, \ldots, N-1] \tag{2.31}$$

Actually, this is called the inverse discrete Fourier transform. Y_k are complex numbers, denoting the amplitude and phase of the periodic signal with period $k/(NT)$ where T is the observation span. An attentive reader will remember that flicker and random walk noise are non-stationary while the Fourier transform requires time series with finite variance. However, we never have infinitely long time series which guarantees the variance remains within some limit. The coefficients can be computed as follows:

$$Y_k = \sum_{n=0}^{N-1} y_n \cdot e^{-i2\pi kn/N} \quad \text{for } k = [-N/2+1, \ldots, N/2] \tag{2.32}$$

The transformation to the frequency domain provides insight which periodic signals are present in the signal and in our case, insight about the noise amplitude at the various frequencies. This is a classic topic and more details can be found in the books by Bracewell (1978) and Buttkus (2000). The one-sided power spectral density S_k is defined as:

$$\begin{aligned} S_0 &= |Y_0|^2/f_s \\ S_{N/2} &= |Y_{N/2}|^2/f_s \\ S_k &= 2|Y_k|^2/f_s \quad \text{for } k = [1, \ldots, N/2-1] \end{aligned} \tag{2.33}$$

The frequency f_k associated to each S_k is:

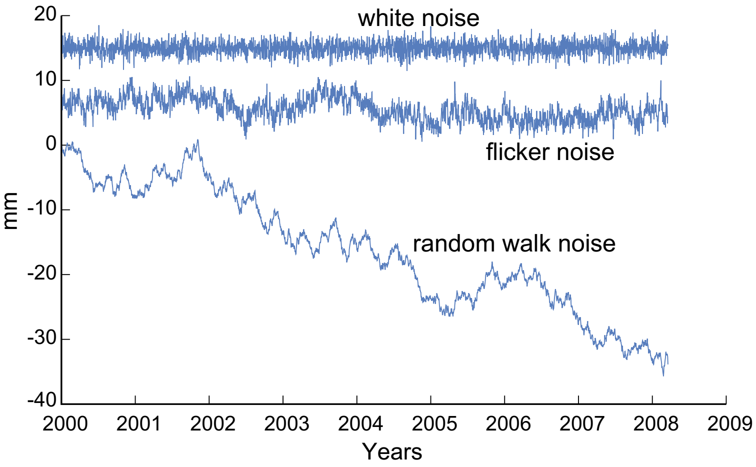

Fig. 2.3 Examples of white, flicker and random walk noise

$$f_k = \frac{kf_s}{N} \quad \text{for } k = [0, \ldots, N/2] \tag{2.34}$$

The highest frequency is half the sampling frequency, $f_s/2$, which is called the Nyquist frequency. The power spectral density (PSD) computed in this manner is called a periodogram. There are many refinements, such as applying window functions and cutting the time series in segments and averaging the resulting set of PSD's. However, a detail that normally receives little attention is that the Fourier transform produces positive and negative frequencies. Time only increases and there are no negative frequencies. Therefore, one always uses the one-sided power spectral density. Another useful relation is that of Parseval (Buttkus 2000):

$$\frac{1}{N}\sum_{n=0}^{N-1} |y_n|^2 = \frac{1}{N^2} \sum_{k=-N/2+1}^{N/2} |Y_k|^2 \tag{2.35}$$

Thus, the variance of the noise should be equal to the sum of all S_k values (and an extra f_s/N^2 scale). The one-sided power spectral density of the three time series of Fig. 2.3 are plotted in Fig. 2.4. It shows that power-law noise indeed follows a straight line in the power spectral density plots if a log-log scale is used. In fact, the properties of the power-law noise can also be estimated by fitting a line to the power spectral density estimates (Mao et al. 1999; Caporali 2003).

The PSD of power-law noise generated by fractionally differenced Gaussian noise is Kasdin (1995):

$$\begin{aligned} S(f) &= \frac{2\sigma^2}{f_s}(2\sin(\pi f/f_s))^\kappa \\ &\approx \frac{2\sigma^2}{f_s}(\pi f/f_s))^\kappa = P_0(f/f_s)^\kappa \quad \text{for } f \ll f_s \end{aligned} \tag{2.36}$$

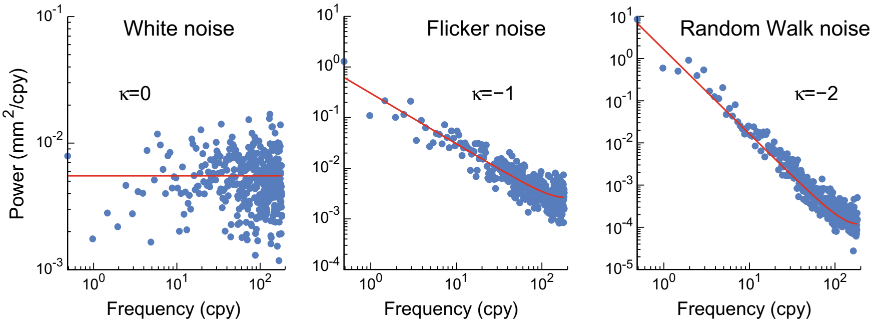

Fig. 2.4 One-sided power spectral density for white, flicker and random walk noise. The blue dots are the computed periodogram (Welch's method) while the solid red line is the fitted power-law model

For small value of f, this approximates $P_0(f/f_s)^\kappa$. The sine function is the result of having discrete data (Kasdin 1995). The PSD for GGM noise is:

$$S(f) = \frac{2\sigma^2}{f_s}\left[1 + \phi^2 - 2\phi\cos(2\pi f/f_s)\right]^{\kappa/2} \tag{2.37}$$

For $\phi = 1$, it converts to the pure power-law noise PSD. The Fourier transform, and especially the Fast Fourier Transform, can also be used to filter a time series. For example, Eqs. (2.23) and (2.24) represent a filtering of white noise vector $\mathbf{v}$ to produce coloured noise vector $\mathbf{w}$:

$$w_i = \sum_{j=0}^{i-1} h_{i-j}\ v_j \tag{2.38}$$

Let us now extend the time series $\mathbf{y}$ and the vector $\mathbf{h}$ containing the filter coefficients with N zeros. This zero padding allows us to extend the summation to $2N$. Using Eq. (2.32), their Fourier transforms, Y_k and H_k, can be computed. In the frequency domain, convolution becomes multiplication and we have Press et al. (2007):

$$W_k = H_k\ Y_k \quad \text{for } k = [-N, \ldots, N] \tag{2.39}$$

Using Eq. (2.31) and only using the first N elements, the vector $\mathbf{w}$ with the coloured noise can be obtained.

2.5 Numerical Examples

To explain the principle of maximum likelihood, this section will show some examples of the numerical method using Python 3. For some years Matlab has been the number one choice to analyse and visualise time series. However, in recent years Python has grown in popularity, due to the fact that it is open source and has many powerful libraries. The following examples are made in IPython (https://ipython.org), using the Jupyter notebook webapplication. How to install this program is described on the afore mentioned website. The examples shown here can be downloaded from the publisher website. The first step is to import the libraries:

```
import math
import numpy as np
from matplotlib import pyplot as plt
from scipy.optimize import minimize
from numpy.linalg import inv
```

Next step is to create some data which we will store in Numpy arrays. As in Matlab, the ‘linspace’ operator creates a simple array on integers. Furthermore, as

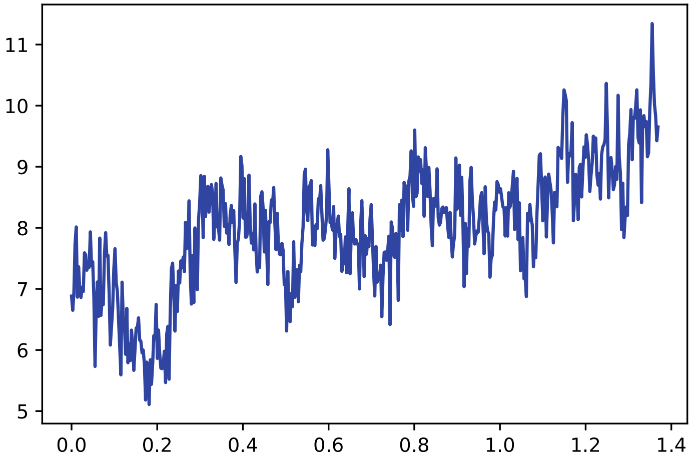

Fig. 2.5 Our synthetic time series containing a simple line plus flicker noise

the name implies 'random.normal' creates an array of Gaussian distributed random numbers. We create a line **y** with slope 2 and offset 6 on which we superimpose the noise **w** that were created using a standard deviation $\sigma_{pl} = 0.5$ for vector **v**, see Eq. (2.23). This time series is plotted in Fig. 2.5.

```
N = 500                         # Number of daily observations
t = np.linspace(0,N/365.25,N)   # time in years
np.random.seed(0)               # Assure we always get the same noise

kappa = -1               #  Flicker noise
h     = np.zeros(2*N)    #  Note the size : 2N
h[0]  = 1                #  Eq. (25)
for i in range(1,N):
    h[i] = (i-kappa/2-1)/i * h[i-1]

v      = np.zeros(2*N)   # Again zero-padded N:2N
v[0:N] = np.random.normal(loc = 0.0, scale = 0.5, size = N)

w = np.real(fft.ifft(fft.fft(v) * fft.fft(h)))  # Eq. (39)

y = (6 + 3*t) + w[0:N]   # trajectory model + noise

plt.plot(t, y, 'b-')     # plot the time series
```

Of course the normal situation is that we are given a set observations and that we need to estimate the parameters of the trajectory model $y(t) = a + bt$. However, creating synthetic time series is a very good method to test if your estimation procedures are correct.

First we will estimate the trajectory assuming white noise in the data:

```
#--- The design matrix
A = np.empty((N,2))
```

```
for i in range(0,N):
    A[i,0] = 1
    A[i,1] = t[i]

#--- Old white noise method
C     = np.identity(N)
x     = inv(A.T @ inv(C) @ A) @ (A.T @ inv(C) @ y)  # Eq. (14)
y_hat = A @ x
r     = y - y_hat                                   # residuals
C_x   = np.var(r)* inv(A.T @ inv(C) @ A)            # Eq. (15)
print(’White noise approximation’)
print(’a = {0:6.3f} +/- {1:5.3f} mm’.format(x[0],\
                                          math.sqrt(C_x[0,0])))
print(’b = {0:6.3f} +/- {1:5.3f} mm/yr’.format(x[1],\
                                          math.sqrt(C_x[1,1])))
```

The result should be:

```
White noise approximation
a =  6.728 +/- 0.064 mm
b =  1.829 +/- 0.080 mm/yr
```

What we have done here is using weighted least-squares with a white noise model that has unit variance. The real variance of the noise has been estimated from the residuals and the uncertainty of the estimated parameters **x** have been scaled with it.

At this point the reader will realise that this approach is not justified because the noise is temporally correlated. It will be convenient to define the following two functions that will create the covariance matrix for power-law noise and apply weighted least-squares (Williams 2003; Bos et al. 2008):

```
#--- power-law noise covariance matrix
def create_C(sigma_pl,kappa):
    U      = np.identity(N)
    h_prev = 1
    for i in range(1,N):
        h = (i-kappa/2-1)/i * h_prev #  Eq. (25)
        for j in range(0,N-i):
            U[j,j+i] = h
        h_prev = h
    U *= sigma_pl                    # scale noise
    return U.T @ U                   # Eq. (26)


#--- weighted least-squares
def leastsquares(C,A,y):
    U     = np.linalg.cholesky(C).T
    U_inv = inv(U)
    B     = U_inv.T @ A
    z     = U_inv.T @ y
    x     = inv(B.T @ B) @ B.T @ z           # Eq. (14)

    #--- variance of the estimated parameters
    C_x = inv(B.T @ B)                           # Eq. (15)

    #--- Compute log of determinant of C
    ln_det_C = 0.0
    for i in range(0,N):
        ln_det_C += 2*math.log(U[i,i])

    return [x,C_x,ln_det_C]
```

The function that creates the covariance matrix for power-law noise has been discussed in Sect. 2.3 and uses Eqs. (2.25) and (2.26). The weighted least-squares function contains some numerical tricks. First, the Cholesky decomposition is applied to the covariance matrix (Bos et al. 2008):

$$\mathbf{C} = \mathbf{U}^T\mathbf{U} \tag{2.40}$$

where $\mathbf{U}$ is an upper triangle matrix. That is, only the elements above the diagonal are non-zero. A covariance matrix is a positive definite matrix which ensures that the Cholesky decomposition always exists. The most important advantage it that one can compute the logarithm of the determinant of matrix $\mathbf{C}$ by just summing the logarithm of each element on the diagonal of matrix $\mathbf{U}$. The factor two is needed because matrix $\mathbf{C}$ is the product of $\mathbf{U}^T\mathbf{U}$. Using these two functions, we can compute the correct parameters $\mathbf{x}$:

```
#--- The correct flicker noise covariance matrix
sigma_pl         = 4
kappa            = -1
C                = create_C(sigma_pl,kappa)
[x,C_x,ln_det_C] = leastsquares(C,A,y)
print(’Correct Flicker noise’)
print(’a = {0:6.3f} +/- {1:5.3f} mm’.format(x[0],\
                                        math.sqrt(C_x[0,0])))
print(’b = {0:6.3f} +/- {1:5.3f} mm/yr’.format(x[1],\
                                        math.sqrt(C_x[1,1])))
```

The result is:

```
Correct Flicker noise
a =  6.854 +/- 2.575 mm
b =  1.865 +/- 4.112 mm/yr
```

If one compares the two estimates, one assuming white noise and the other assuming flicker noise, then one can verify that the estimates themselves are similar. The largest difference occurs for the estimated errors which are 5 times larger for the latter. This also happens in real geodetic time series. Mao et al. (1999) concluded that the velocity error in GNSS time-series could be underestimated by factors of 5–11 if a pure white noise model is assumed. Langbein (2012) demonstrated that an additional factor of two might be needed if there is also random walk noise present.

For sea level time series (Bos et al. 2014) obtained a more moderate factor of 1.5–2 but still, white noise underestimates the true uncertainty of the estimated linear trend. Williams et al. (2014) estimated a factor 6 for the GRACE gravity time series. As discussed in Sect. 2.3, most geodetic time series are temporally correlated and therefore one nowadays avoids the white noise model.

So far we have assumed that we knew the true value of the spectral index κ and the noise amplitude σ_{pl}. Using MLE, we can estimate these parameters from the data:

```
#--- Log-likelihood (with opposite sign)
def log_likelihood(x_noise):
    sigma_pl = x_noise[0]
    kappa    = x_noise[1]
    C        = create_C(sigma_pl,kappa)
    [x,C_x,ln_det_C] = leastsquares(C,A,y)
    r        = y - A @ x  # residuals

    #--- Eq. (12)
    logL     = -0.5*(N*math.log(2*math.pi) + ln_det_C \
                                        + r.T @ inv(C) @ r)
    return -logL


x_noise0 = np.array([1,1]) # sigma_pl and kappa guesses
res = minimize(log_likelihood, x_noise0, \
               method=’nelder-mead’, options={’xatol’:0.01})

print(’sigma_pl={0:6.3f}, kappa={1:6.3f}’.\
                                  format(res.x[0],res.x[1]))
```

Note that we inverted the sign of the log-likelihood function because most software libraries provide minimization subroutines, not maximisation. In addition, it is in this function that we need the logarithm of the determinant of matrix **C**. If one tries to compute it directly from matrix **C**, then one quickly encounters too large numbers that create numerical overflow. This function also shows that we use weighted least-squares to estimate the parameters of the trajectory model while the numerical minimization algorithm (i.e. Nelder–Mead), is only used the compute the noise parameters. The reason for using weighted least-squares, also a maximum likelihood estimator as we have shown in Sect. 2.2, is solely for speed. Numerical minimization is a slow process which becomes worse for each additional parameter we need to estimate. The results is:

```
sigma_pl= 0.495, kappa=-1.004
```

These values are close to the true values of $\sigma_{pl} = 0.5$ and $\kappa = -1$. The following code can be used to plot the log-likelihood as function of κ and σ_{pl}:

```
S = np.empty((21,21))
for i in range(0,21):
    sigma_pl = 1.2 - 0.05*i
    for j in range(0,21):
        kappa = -1.9 + 0.1*j
        x_noise0 = [sigma_pl,kappa]
        S[i,j] = math.log(log_likelihood(x_noise0))

plt.imshow(S,extent=[-1.9,0.1,0.2,1.2], cmap='nipy_spectral', \
                                               aspect='auto');
plt.colorbar()
plt.ylabel('sigma_pl')
plt.xlabel('kappa')
plt.show()
```

The result is shown in Fig. 2.6 which indeed shows a minimum around $\sigma_{pl} = 0.5$ and $\kappa = -1$. Depending on the computer power, it might take some time to produce the values for this figure.

In Sect. 2.3 we noted that for GNSS time series the power-law plus white noise model is common. Thus, one must add the covariance matrix for white noise, $\sigma_w^2\mathbf{I}$, to the covariance matrix we discussed in the examples. In addition, it is more efficient to write the covariance matrix of the sum of power-law and white noise as follows:

$$\mathbf{C} = \sigma_{pl}^2\mathbf{J}(\kappa) + \sigma_w^2\mathbf{I} = \sigma^2\,(\phi\;\mathbf{J}(\kappa) + (1-\phi)\mathbf{I}\;) \tag{2.41}$$

where σ can be computed using:

$$\sigma = \sqrt{\frac{\mathbf{r}^T\mathbf{C}^{-1}\mathbf{r}}{N}} \tag{2.42}$$

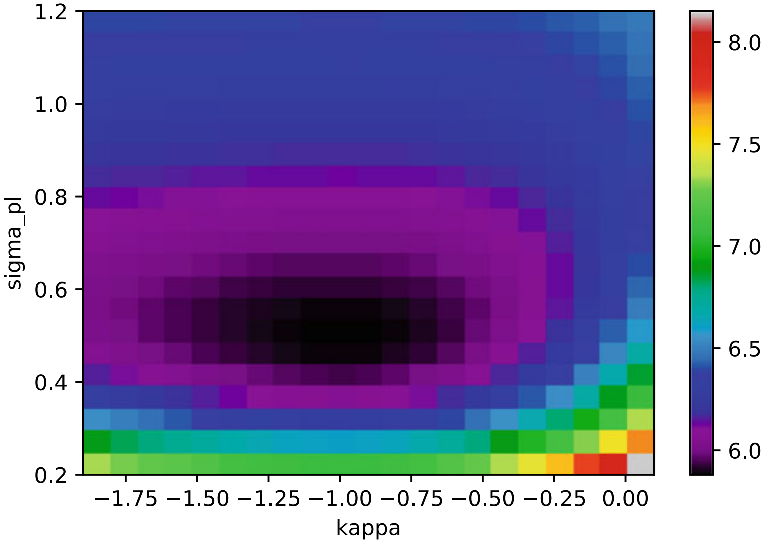

Fig. 2.6 The log of the $\log(L)$ function as function of κ and σ_{pl}

Since σ can be computed from the residuals, we only use our slow numerical minimization algorithm need to find the value of ϕ (Williams 2008).

Note that we only analysed 500 observations while nowadays time series with 7000 observations are not uncommon. If one tries to rerun our examples for this value of N, then one will note this takes an extremely long time. The main reason is that the inversion of matrix $\mathbf{C}$ requires $\mathcal{O}(N^3)$ operations. Bos et al. (2008, 2013) have investigated how the covariance matrix $\mathbf{C}$ can be approximated by a Toeplitz matrix. This is a special type of matrix which has a constant value on each diagonal and one can compute its inverse using only $\mathcal{O}(N^2)$ operations. This method has been implemented in the Hector software package that is available from http://segal.ubi.pt/hector.

The Hector software was used to create time series with a length of 5000 daily observations (around 13.7 years) for 20 GNSS stations which we will call the Benchmark Synthetic GNSS (BSG). This was done for the the horizontal and vertical components, producing 60 time series in total. Each contains a linear trend, an annual and a semi-annual signal. The sum of flicker and white noise, w_i, was added to these trajectory models:

$$w_i = \sigma \left[\sqrt{\phi} \sum_{j=0}^{i-1} h_{i-j} v_j + \sqrt{1-\phi}\, u_i \right] \tag{2.43}$$

with both u_i and v_i are Gaussian noise variables with unit variance. The factor ϕ was defined in Eq. (2.41). To create our BSG time series we used $\sigma = 1.4$ mm, $\phi = 0.6$ and horizontal components and $\sigma = 4.8$ mm, $\phi = 0.7$ for the vertical component.

It is customary to scale the power-law noise amplitudes by $\Delta T^{-\kappa/4}$ where ΔT is the sampling period in years. For the vertical flicker noise amplitude we obtain:

$$\sigma_{pl} = \frac{\sigma\sqrt{\phi}}{\Delta T^{\kappa/4}} = \frac{4.8 \times \sqrt{0.7}}{(1/365.25)^{1/4}} = 17.6 \ \ \text{mm/yr}^{0.25} \tag{2.44}$$

The vertical white noise amplitude is 2.6 mm. For the horizontal component these values are $\sigma_{pl} = 4.7\,\text{mm/yr}^{0.25}$ and $\sigma_w = 0.9$ mm respectively. The BGS time series can be found on the Springer website for this book, and can be used to verify the algorithms developed by the reader. These series will also be compared with other methods in the following chapters.

2.6 Discussion

In this chapter we have given a brief introduction to the principles of time series analysis. We paid special attention to the maximum likelihood estimation (MLE) method and the modelling of power-law noise. We showed that with our assumptions on the stochastic noise properties, the estimated parameters have their variance bounded by the Cramer Rao lower bound. Therefore the MLE is an optimal estimator in the sense of asymptotically unbiased and efficient (minimum variance).

In this book we will present other estimators such as the Kalman filter, the Markov Chain Monte Carlo Algorithm and the Sigma-method. All have their advantages and disadvantages and to explain them was one of the reasons for writing this book. The other reason was to highlight the importance of temporal correlated noise. This phenomenon has been known for a long time but due to increased computer power, it has now become possible to include it in the analysis of geodetic time series. We illustrated how this can be done by various examples in Python 3 that highlighted some numerical aspects that will help the reader to implement their own algorithms.

References

Agnew DC (1992) The time-domain behaviour of power-law noises. Geophys Res Letters 19(4):333–336, https://doi.org/10.1029/91GL02832

Beran J (1994) Statistics for long-memory processes. Monographs on statistics and applied probability, Chapman & Hall, New York

Beran J (1995) Maximum Likelihood Estimation of the Differencing Parameter for Invertible Short and Long Memory Autoregressive Integrated Moving Average Models. Journal of the Royal Statistical Society Series B (Methodological) 57(4):659–672, http://www.jstor.org/stable/2345935

Bos MS, Williams SDP, Araújo IB, Bastos L (2008) Fast error analysis of continuous GPS observations. J Geodesy 82:157–166, https://doi.org/10.1007/s00190-007-0165-x

Bos MS, Williams SDP, Araújo IB, Bastos L (2013) Fast error analysis of continuous GNSS observations with missing data. J Geodesy 87:351–360, https://doi.org/10.1007/s00190-012-0605-0

Bos MS, Williams SDP, Araújo IB, Bastos L (2014) The effect of temporal correlated noise on the sea level rate and acceleration uncertainty. Geophys J Int 196:1423–1430, https://doi.org/10.1093/gji/ggt481

Box GEP, Jenkins GM, Reinsel GC, Ljung GM (2015) Time Series Analysis, Forecasting and Control, 5th edn. Wiley

Bracewell R (1978) The Fourier Transform and its Applications, 2nd edn. McGraw-Hill Kogakusha, Ltd., Tokyo

Brockwell P, Davis RA (2002) Introduction to Time Series and Forecasting, second edition edn. Springer-Verlag, New York

Buttkus B (2000) Spectral Analysis and Filter Theory in Applied Geophysics. Springer-Verlag Berlin Heidelberg

Caporali A (2003) Average strain rate in the Italian crust inferred from a permanent GPS network - I. Statistical analysis of the time-series of permanent GPS stations. Geophys J Int 155:241–253, https://doi.org/10.1046/j.1365-246X.2003.02034.x

Casella G, Berger R (2001) Statistical Inference, 2nd edn. Duxbury Resource Center

Dmitrieva K, Segall P, DeMets C (2015) Network-based estimation of time-dependent noise in GPS position time series. J Geodesy 89:591–606, https://doi.org/10.1007/s00190-015-0801-9

Granger C (1980) Long memory relationships and the aggregation of dynamic models. Journal of Econometrics 14(2):227 – 238, https://doi.org/10.1016/0304-4076(80)90092-5

Granger CWJ, Joyeux R (1980) An Introduction to Long-Memory Time Series Models and Fractional Differencing. Journal of Time Series Analysis 1(1):15–29, https://doi.org/10.1111/j.1467-9892.1980.tb00297.x

Graves T, Gramacy R, Watkins N, Franzke C (2017) A Brief History of Long Memory: Hurst, Mandelbrot and the Road to ARFIMA, 1951–1980. Entropy 19(9), https://doi.org/10.3390/e19090437, http://www.mdpi.com/1099-4300/19/9/437

He X, Bos MS, Montillet JP, Fernandes RMS (2019) Investigation of the noise properties at low frequencies in long GNSS time series. Journal of Geodesy https://doi.org/10.1007/s00190-019-01244-y

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

Hurst HE (1957) A Suggested Statistical Model of some Time Series which occur in Nature. Nature 180:494, https://doi.org/10.1038/180494a0

Jaynes ET (2003) Probability theory: The logic of science. Cambridge University Press, Cambridge

Johnson HO, Agnew DC (1995) Monument motion and measurements of crustal velocities. Geophys Res Letters 22(21):2905–2908, https://doi.org/10.1029/95GL02661

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

Kay SM (1993) Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice-Hall, Inc., Upper Saddle River, NJ, USA

Koch KR (1990) Bayesian Inference with Geodetic Applications. Lecture Notes in Earth Sciences, Springer-Verlag, https://doi.org/10.1007/BFb0048699

Koch KR (2007) Introduction to Bayesian Statistics, 2nd edn. Springer-Verlag, Berlin Heidelberg

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

Langbein J (2010) Computer algorithm for analyzing and processing borehole strainmeter data. Comput Geosci 36(5):611–619, https://doi.org/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. J Geodesy 86:775–783, https://doi.org/10.1007/s00190-012-0556-5

Mandelbrot BB (1999) Multifractals and $1/f$ Noise. Springer, https://doi.org/10.1007/978-1-4612-2150-0

Mandelbrot BB, van Ness JW (1968) Fractional Brownian motions, fractional noises and applications. SIAM Review 10:422–437

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

Montillet JP, Yu K (2015) Modeling Geodetic Processes with Levy α-Stable Distribution and FARIMA. Mathematical Geosciences 47(6):627–646

Press WH (1978) Flicker noises in astronomy and elsewhere. Comments on Astrophysics 7:103–119

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd edn. Cambridge University Press, New York, NY, USA

Sowell F (1992) Maximum likelihood estimation of stationary univariate fractionally integrated time series models. J Econom 53:165–188

Taqqu MS, Teverovsky V, Willinger W (1995) Estimators for long-range dependence: An empirical study. Fractals 3:785–798

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

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

Williams SD, Moore P, King MA, Whitehouse PL (2014) Revisiting grace antarctic ice mass trends and accelerations considering autocorrelation. Earth and Planetary Science Letters 385:12 – 21, https://doi.org/10.1016/j.epsl.2013.10.016, http://www.sciencedirect.com/science/article/pii/S0012821X13005797

Machiel S. Bos obtained in 1996 his M.Sc. from the Delft University of Technology. In 2001, he received his Ph.D. from the University of Liverpool. Afterward, he held various post-doc positions in Sweden, the Netherlands, and Portugal. Currently, he is a post-doc at Instituto Dom Luiz. His scientific interests include ocean tide loading, GPS time series analysis, and geoid computations.

Jean-Philippe Montillet (MS'03 (Aalborg), PhD'08 (Nottingham), SMIEEE'14) is a geoscientist working in GNSS technology with applications in mathematical geodesy, geophysics and civil engineering. His latest research projects include crustal deformation and sea-level rise in the Pacific Northwest. He has also been involved in the analysis of Earth observations (i.e., GRACE, Satellite altimetry) and climate change monitoring and climate of the past simulations (TRACE21K, PMIP). The past 5 years, he has extensively worked on geodetic time series analysis, including signal processing techniques to extract geophysical and transient signals, and several works on the stochastic and functional modeling. His work is generally funded by governments (EPSRC, Australia RC, NASA, Swiss CTI) and collaboration with safety authorities or government agencies.

Simon D. P. Williams has worked on sea and land level measurements in the Marine Physics and Ocean Climate Group at the National Oceanography Centre in Liverpool since 1999 and prior to that spent 4 years at the Scripps Institute of Oceanography, La Jolla in California. He received his PhD from Durham University on "Current Motion on Faults of the San Andreas System in Central California Inferred from Recent GPS and Terrestrial Survey Measurements". He has particular expertise in the area of stochastic modelling and uncertainty analysis of geophysical series including sea level, GNSS and gravity data. He is the author of the CATS software, which is used by researchers worldwide to analyse geophysical time series and several seminal papers on stochastic noise analysis of geophysical time series.

Rui M. S. Fernandes (male), has a doctoral degree in Earth and Space Sciences by Technical University of Delft (The Netherlands). He is Assistant Professor in the University of Beira Interior (UBI), Covilhã, Portugal and Associated Researcher of Institute Geophysical Infante D. Luíz (IDL), Lisbon, Portugal. He is the coordinator of C4G (Colaboratory For Geosciences), the research infrastructure for Geosciences in Portugal and President of the Interim Governing Board of the GNSS EPOS-ERIC (European Plate Observing System). His main areas of research are related with application to geosciences of rigorous positioning using space-geodetic techniques.