Superfast Inference for Stationary Gaussian Processes in Particle Tracking Microrheology

by

Yun Ling

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Member:

Derek Bingham
Professor, Dept. of Statistics and Actuarial Science
Simon Fraser University

Supervisor:

Martin Lysy
Associate Professor

Internal Members:

Marius Hofert
Associate Professor

Michael Wallace
Assistant Professor

Internal-External Member:

Anita Layton
Professor, Dept. of Applied Mathematics

Additional Member:

Mu Zhu
Professor

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I am the sole author of Chapters 1, 4, and 5.

Chapters 2 and 3 of this thesis have been prepared for submission to peer-reviewed journals. Writing contributions have been made by my supervisor Martin Lysy, and co-authors M. Gregory Forest and Ian Seim (University of North Carolina – Chapel Hill). All computations in these chapters were performed by me.

The data used in Chapters 3 and 4 has been provided by David B. Hill and Jeremy Cribb (University of North Carolina – Chapel Hill).

Abstract

Particle tracking of passive microscopic species has become the experimental measurement of choice in diverse applications, where either the material volumes are limited, or the materials themselves are so soft that they deform uncontrollably under the stresses and strains of traditional instruments. As such, the results of countless biological and rheological analyses hinge pivotally on extracting reliable dynamical information from large datasets of particle trajectory recordings. However, to do this in a statistically and computationally efficient manner presents a number of important challenges. Addressing some of these challenges is the focus of the present work.

In Chapter 2, we present a "superfast" set of tools for parametric inference in single-particle tracking. Parametric likelihoods for particle trajectory measurements typically consist of stationary Gaussian time series, for which traditional "fast" inference algorithms scale as $\mathcal{O}(N^2)$ in the number of observations. We present a superfast $\mathcal{O}(N \log^2 N)$ algorithm for parametric inference for stationary Gaussian processes and propose novel superfast algorithms for score and Hessian calculations. This effectively enables superfast inference for stationary Gaussian process via a wide array of frequentist and Bayesian methods. In Chapters 3 and 4, we use the superfast toolkit to address two outstanding problems prevalent in many particle tracking analyses. The first is that particle position measurements are generally contaminated by various forms of high-frequency errors. Failure to account for these errors leads to considerable bias in estimation results. In Chapter 3 we propose a novel strategy to filter high-frequency noise from measurements of particle positions. Our filters are shown theoretically to cover a vast range of high-frequency noise regimes and lead to an efficient computational estimator of model coefficients. Analyses of numerous experimental and simulated datasets suggest that our filtering approach performs remarkably well. The second problem we address is the considerable heterogeneity of typical biological fluids in which particle tracking experiments are conducted. In Chapter 4, we propose a simple metric by which to quantify the degree of heterogeneity of a fluid, along with a computationally efficient estimator and statistical test against the hypothesis that the fluid is homogeneous. The thesis is concluded by outlining several directions for future research.

Acknowledgments

First and most importantly, I want to express my sincere gratitude and appreciation to my supervisor Dr. Martin Lysy for the continuous support of my Ph.D study and related research. Without his patient guidances, this will not be possible.

The Department of Statistics and Actuarial Science is a wonderful family, where I met many friendly and talented colleagues. The professors in this department are all brilliant, admirable and helpful. Thank you for your teaching.

Finally, I thank my parents for their unconditional and selfless support. They provided me with endless power during my five years stay in Canada.

To my family

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past several decades, technological developments of optical spectroscopy have greatly advanced the field of bio-imaging, providing vast amounts of information about target particles at unprecedented accuracy and spatio-temporal resolution. In particular, the ability to record single particle trajectories – rather than summary statistics of a large ensemble – has become an invaluable approach to studying the dynamics of biological particles [van der Schaar et al., 2008], microrheology of complex fluids [Mason et al., 1997] and various mechanisms of drug delivery [Suh et al., 2005]. Here, a "particle" can be anything from a single molecule diffusing in a biological fluid [Saxton and Jacobson, 1997] to a probe used for detecting the rheological properties of diverse biomaterials [Mason et al., 1997].

Single-particle tracking generally starts from a sequence of microscope images of particles, which are usually taken at an evenly-distributed times $\boldsymbol{t} = [0, \Delta t, 2\Delta t, \ldots, N\Delta t]$ where $\Delta t = \frac{1}{\text{fps}}$ is the fixed interobservation time and $N\Delta t$ is the duration of the experiment. The task of particle tracking can then be divided into the following three steps:

1. **Detection**, which consists of identifying the "spots" that represent particles from the background and converting such spots into coordinates. In Figure 1.1 we show the picture of particles diffusing in different media, where various methods can be applied to locate the particles [Crocker and Grier, 1996, Newby et al., 2018] and

convert each image to a series of two-dimensional measurements, each corresponding to a particle located in the frame.

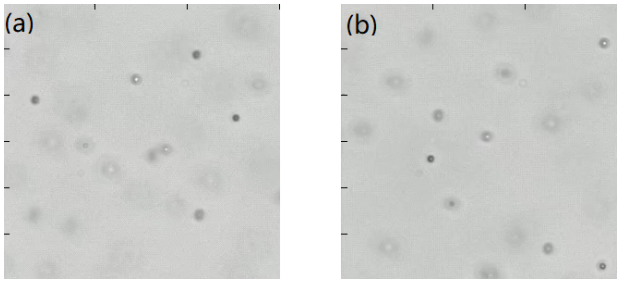


Figure 1.1: Pictures of 1 µm beads diffusing in (a) water and (b) biological mucus. Lighter and wider circles indicate particles that are further from the camera focal plane in the direction perpendicular to the image.
Source of pictures: Ian Seim, David Hill (University of North Carolina - Chapel Hill)

2. **Linking**, which means connecting the particle positions on all microscope images and then constructing a time-dependent particle trajectory. This step also involves determining which particle trajectories are shorter in duration than $T = N\Delta t$, because those particles have moved far enough in the direction perpendicular to the focal plane so as to no longer be detectable. In Figure 1.2 we demonstrate the trajectories of 1 µm beads diffusing in water.

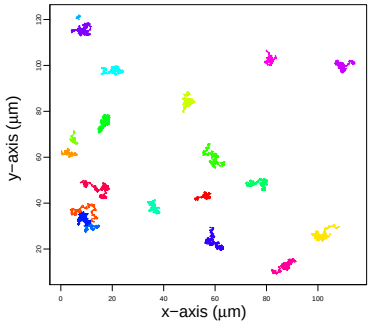


Figure 1.2: Trajectories of 1 µm beads diffusing in water.
The interobservation time is $\Delta t = 1/60(s)$ and the experiment lasts for $N\Delta t = 30(s)$.

3. **Analysis**, where particle trajectory data obtained from previous steps are studied to understand the dynamic of particles and properties of fluids. In many experiments, the resulting analysis hinges pivotally on the measurement of particles' mean square displacement (MSD), which for a 2-dimensional particle trajectory $\boldsymbol{X}(t)$ is given by

$$\text{MSD}_{\boldsymbol{X}}(t) = \frac{1}{2} \times E\big[\|\boldsymbol{X}(t) - \boldsymbol{X}(0)\|^2\big].$$

In Figure 1.3 we show the 2D particle trajectory of a particle diffusing in water, and its MSD in logarithm scale.

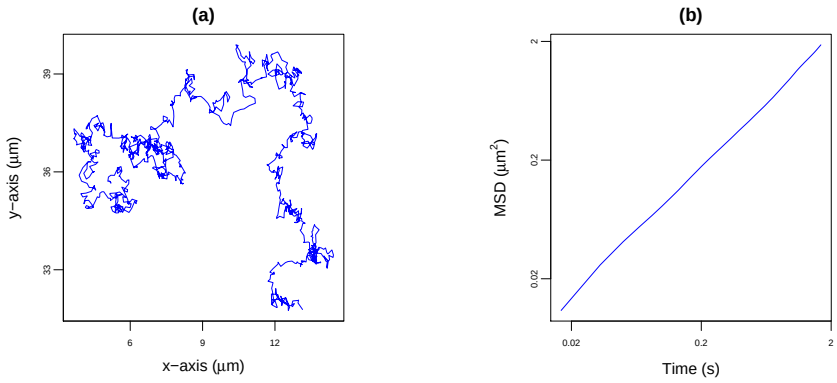


Figure 1.3: 2D Trajectory of a 1 µm bead diffusing in water and its MSD.
(a) 2D trajectory of a 1 µm diffusing in water.
(b) Non-parametric estimate of MSD using Equation (3.2.2).

For viscous fluids such as water and glycerol, the linear trend

$$\text{MSD}_{\boldsymbol{X}}(t) \quad \propto \quad t$$

seen in Figure 1.3 is predicted by Einstein's theory of Brownian motion [Einstein, 1956]. However, most biological fluids are viscoelastic, for which a nearly ubiquitous experimental finding is that the MSD has sublinear power-law scaling over a given range of timescales,

$$\text{MSD}_{\boldsymbol{X}}(t) \sim 2Dt^{\alpha}, \qquad t_{\min} < t < t_{\max}, \quad 0 < \alpha < 1.$$

This phenomenon is referred to as *subdiffusion*, and the interpretation of the parameters (α, D) has far-reaching consequences for numerous biological applications, which are discussed in Chapter 3.

At present there is no scientific theory by which to predict the subdiffusion parameters from the properties of a given fluid, such that (α, D) must be estimated from empirical data. However, obtaining reliable parameter estimates poses a number of statistical challenges:

1. The particle tracking literature predominantly uses a regression-based subdiffusion estimator constructed from the nonparametric MSD estimator displayed in Figure 1.3. However, this MSD estimator becomes extremely variable at longer lag times. Moreover, the nonparametric estimator suffers from considerable bias in the presence of the drift which commonly occurs in fluid media [Mellnik et al., 2016].

2. Subdiffusion estimators derived from fully-parametric models of the stochastic trajectory process are considerably more statistically efficient than their regression-based counterparts. However, likelihood calculations for these models at best scale as $O(N^2)$ in the number of observations, which in many practical applications is prohibitively expensive.

3. Particle trajectory measurements are almost invariably subjected to various sources of instrumental error [Savin and Doyle, 2005]. Failure to account for these errors can lead to considerable bias in most subdiffusion estimators.

4. Single-particle experiments typically track tens to thousands of particles simultaneously over the spatial domain defined by the camera's focal plane. Most biological fluids exhibit considerable spatial heterogeneity due to a number of factors, including the diverse molecular composition and the nonuniform distribution of biological substance like mucus. Consequently, summarizing the subdiffusion dynamics by a single pair of (α, D) can be highly misleading.

This thesis presents several contributions to addressing the challenges described above. In Chapter 2 we present a "superfast" set of tools for parametric inference in single-particle tracking, where stationary Gaussian processes are involved. With the superfast algorithm

that inverts the covariance matrices of stationary Gaussian processes [Ammar and Gragg, 1988], we reduce the total complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log^2 N)$ for particle trajectories of length N. Moreover, we present new superfast algorithms for the derivatives of the likelihood, thus providing tools for highly efficient inference for stationary Gaussian process via a broad array of frequentist and Bayesian methods.

In Chapter 3, we propose a likelihood-based method for correcting the instrumental errors that distort particles' behaviours in the high-frequency domain. We consider a family of high-frequency filters that can readily be applied to a wide range of parametric particle position models and show how to estimate all model parameters in a computationally efficient manner. Extensive simulations and analyses of experimental data suggest that our filter performs remarkably well.

In Chapter 4, we propose to quantify the degree of heterogeneity of a fluid with a readily interpretable metric. With a computationally efficient estimator for this metric and a statistical test against the hypothesis that the fluid is homogeneous, we can obtain an intuitive sense of the degree of heterogeneity of the medium and better understand the properties of the fluid in experiments. Chapter 5 presents the possible directions of future research in superfast inference of stationary Gaussian process and particle tracking areas.

Chapter 2

Superfast Inference for Stationary Gaussian Process

2.1 Introduction

Stationary Gaussian processes are widely used in a variety of statistical applications including time series modeling [Breidt et al., 1998, Harvey, 2002, Granger and Joyeux, 1980, Hosking, 1981], unsupervised function estimation [Smola and Bartlett, 2001], differential equation modeling [Archambeau et al., 2007, Calderhead et al., 2009] and signal filtering and smoothing [Särkkä et al., 2014]. They have convenient properties for various modeling tasks in machine learning [Williams and Rasmussen, 2006], examples range from regression over classification [Neal, 1997] to reinforcement learning [Engel et al., 2005].

Many of the applications listed above involve the estimation of the unknown parameters $\boldsymbol{\theta}$ of a stationary Gaussian time series from N consecutive equally spaced observations $\boldsymbol{x} = (x_1, \ldots, x_N)$. In the simplest case, we have

$$\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{V}_{\boldsymbol{\theta}}),$$

where

$$\boldsymbol{V}_{\boldsymbol{\theta}} = \begin{bmatrix} \gamma_{\boldsymbol{\theta}}(0) & \gamma_{\boldsymbol{\theta}}(1) & \gamma_{\boldsymbol{\theta}}(2) & \dots & \gamma_{\boldsymbol{\theta}}(N-1) \\ \gamma_{\boldsymbol{\theta}}(1) & \gamma_{\boldsymbol{\theta}}(0) & \gamma_{\boldsymbol{\theta}}(1) & \dots & \gamma_{\boldsymbol{\theta}}(N-2) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \gamma_{\boldsymbol{\theta}}(N-1) & \gamma_{\boldsymbol{\theta}}(N-2) & \gamma_{\boldsymbol{\theta}}(N-3) & \dots & \gamma_{\boldsymbol{\theta}}(0) \end{bmatrix}$$

is a Toeplitz matrix of which the elements $\gamma_{\boldsymbol{\theta}}(h) = \text{cov}(x_n, x_{n+h} \mid \boldsymbol{\theta})$ are parametrized by $\boldsymbol{\theta}$.

The log-likelihood for this problem is

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{x}) = -\frac{1}{2}\left[\boldsymbol{x}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{x} + \log|\boldsymbol{V}_{\boldsymbol{\theta}}|\right], \tag{2.1.1}$$

such that most approaches to parameter inference require repeatedly solving the Toeplitz system $\boldsymbol{V}_{\boldsymbol{\theta}} \cdot \boldsymbol{z} = \boldsymbol{x}$ and evaluating $\log|\boldsymbol{V}_{\boldsymbol{\theta}}|$ for different values of $\boldsymbol{\theta}$. Exploiting the Toeplitz structure of the variance matrix, "fast" algorithms for evaluating (2.1.1) require only $\mathcal{O}(N^2)$ operations [Levinson, 1946, Durbin, 1960, Trench, 1964, Zohar, 1969, Bareiss, 1969]. This is a massive computational improvement over unstructured variances, for which the corresponding calculations are $\mathcal{O}(N^3)$. However, the quadratic scaling of fast algorithms becomes a serious limitation when N is large.

Beginning with work on displacement ranks of Kailath et al. [1979], it was realized that Toeplitz systems could be solved by "superfast" FFT-based methods scaling as $\mathcal{O}(N \log^2 N)$ [Brent et al., 1980, Bitmead and Anderson, 1980, De Hoog, 1984, de Hoog, 1987, Musicus, 1988, Ammar and Gragg, 1988, 1987, Chandrasekaran et al., 2007]. However, these algorithms have yet to be leveraged for statistical analyses for a several reasons. For one, most of them do not provide direct means of calculating the log-determinant of $\boldsymbol{V}_{\boldsymbol{\theta}}$ (though superfast methods for this calculation do exist, e.g. Kravanja and Van Barel [2000]). Moreover, many superfast algorithms bury considerable overhead in the big-O notation [Sexton, 1982]. Third and perhaps most importantly, many superfast algorithms are numerically unstable [Bunch, 1985], prompting the developments of stable $\mathcal{O}(N \log^p N)$ solvers with $p > 2$ (e.g., the algorithms of Stewart [2003] and Chen et al. [2006] with $p = 3$ and $p = 5/2$, respectively).

In this chapter, we present a set of methods for superfast inference for stationary

Gaussian time series. They build upon the superfast Toeplitz solver of Ammar and Gragg [1988], the only algorithm of those mentioned above which provides the log-determinant as well. This algorithm has provably low overhead, crossing over Levinson's fast solver around $N = 260$. As the Ammar-Gragg algorithm is defined only for matrices of size $N = 2^K + 1$, we present a novel extension to arbitrary N with no additional overhead. Moreover, we present new superfast algorithms for the score and Hessian functions, thus providing tools for highly efficient inference for stationary Gaussian process via a broad array of frequentist and Bayesian methods. An implementation of our method is publicly available in the R/C++ library *SuperGauss*.

The remainder of the chapter is organized as follows. In Section 2.2, we provide an overview of the Ammar-Gragg superfast Toeplitz solver and its generalization to arbitrary N. In Section 2.3 we present superfast gradient and Hessian algorithms for (2.3.2) and show how to extend these algorithms to profile likelihoods where the mean of $\boldsymbol{x}$ is given by a regression equation (2.3.7). In Section 2.4, we provide speed and stability comparisons with several fast and superfast algorithms. It is noted that for a variety of commonly-used models for statistical inference, numerical instability does not appear to be a practical issue. In Section 2.5, we present an application to Gaussian process factor analysis (2.5.1). Concluding remarks are offered in Section 2.6.

2.2 The Generalized Schur Algorithm for Toeplitz Systems

In this section, we present the generalized Schur Algorithm proposed in [Ammar and Gragg, 1988]. It is an algorithm that computes the inverse and determinant of size $N = 2^k + 1$ Toeplitz matrix in $\mathcal{O}(N \log^2 N)$ steps. The Ammar-Gragg algorithm, including many of the algorithms mentioned in the introduction [Bareiss, 1969, Brent et al., 1980, Bitmead and Anderson, 1980, Sexton, 1982], are manifestations of Schur algorithm [Ammar, 1996]. The Ammar-Gragg divide-and-conquer version of Schur algorithm is referred by them as the "generalized" Schur algorithm. For consistency we keep this nomenclature, although "generalized Schur algorithm" is more widely accepted to refer to a fast algorithm for the

Cholesky factorization of positive-definite structured matrices [Chandrasekaran and Sayed, 1996]. In order to give a clearer picture of the algorithm as a whole, technical details are omitted whenever possible. For an in-depth discussion of the mathematics underlying the Ammar-Gragg algorithm, the reader is referred to Ammar and Gragg [1987, 1988].

2.2.1 Gohberg-Semencul Formula

Let $\boldsymbol{V}$ denote an $N \times N$ symmetric positive definite Toeplitz matrix. From the definition of $\boldsymbol{V}$, it is clear that all entries can be obtained from the first row (or column). A seminal result of Gohberg and Semencul [Gohberg and Semencul, 1972] is that the same is true for $\boldsymbol{V}^{-1}$. Namely, let $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_N]$ denote the first row of $\boldsymbol{V}^{-1}$. Then

$$\boldsymbol{V}^{-1} = \frac{1}{\delta_1} \left(\boldsymbol{L}_1 \boldsymbol{L}_1' - \boldsymbol{L}_2 \boldsymbol{L}_2' \right), \tag{2.2.1}$$

where

$$\boldsymbol{L}_1 = \begin{bmatrix} \delta_1 & & & \\ \delta_2 & \delta_1 & & \\ \vdots & \ddots & \ddots & \\ \delta_N & \ldots & \delta_2 & \delta_1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{L}_2 = \begin{bmatrix} 0 & & & \\ \delta_N & 0 & & \\ \vdots & \ddots & \ddots & \\ \delta_2 & \ldots & \delta_N & 0 \end{bmatrix}$$

are lower-triangular Toeplitz matrices [Gohberg and Semencul, 1972].

The Gohberg-Semencul formula not only reduces the storage from a whole matrix $\mathcal{O}(N^2)$ to a vector $\mathcal{O}(N)$, but also simplifies the computation that involves solving the Toeplitz system $\boldsymbol{V}\boldsymbol{x} = \boldsymbol{b}$. While matrix-vector multiplication generally takes $\mathcal{O}(N^2)$ steps, the matrix product $\boldsymbol{V}^{-1}\boldsymbol{b}$ can be computed as successive matrix-vector products with triangular Toeplitz matrices. It is well known that each of these multiplications can be obtained in $\mathcal{O}(N \log N)$ steps using the fast Fourier transform (FFT) [Kailath and Sayed, 1999b]. The exact algorithm is provided in Appendix A.

2.2.2 Generalized Schur Algorithm

For a size $N \times N$ Toeplitz covariance matrix $\boldsymbol{V}$ with first column $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_N]$, consider a rational function

$$\phi_0(x) = \frac{\sum_{j=1}^{N-1} -\gamma_{j+1} x^j}{\sum_{j=1}^{N-1} \gamma_j x^j}.$$

The Schur algorithm is an iterative procedure that can generate a rational function $\phi_n(x)$ from $\phi_0(x)$ using the following linear functional transformation

$$\phi_{i+1}(x) = \frac{1}{x} \cdot \frac{\phi_i(x) - \mu_i}{1 - \mu_i \cdot \phi_i(x)}, \quad \mu_i = \phi_i(0), \quad i = 0, \ldots, n-1,$$

where $\{\mu_i\}_{i=1}^n$ are the Schur parameters. For a given n-th order polynomial $\phi(x)$, let $\tilde{\phi}(x) = x^n \phi_n(1/x)$, which is also a polynomial of order at most n. Then the n-th step of Schur's algorithm can be expressed as $\phi_n(x) = T_n^{-1}(\phi_0(x))$, where the ϕ_n are rational functions and T_n has following representation

$$T_n(x) = \frac{\xi_n(x) + \tilde{\eta}_n(x) \cdot x}{\eta_n(x) + \tilde{\xi}_n(x) \cdot x},$$

where ξ_n and η_n are polynomials of degree $< n$ with coefficients depending on those of ϕ_0, such that $\phi_0(x) = T_n(\phi_n(x))$.

It was realized by Ammar and Gragg [1987] that the coefficients of the $N-1$ step of Schur's algorithm $T_{N-1} = \frac{\xi_{N-1}(x) + \tilde{\eta}_{N-1}(x) \cdot x}{\eta_{N-1}(x) + \tilde{\xi}_{N-1}(x) \cdot x}$ produce the first column of $\boldsymbol{V}^{-1}$ via

$$\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_N] = \frac{1}{\sigma_{N-1}^2}([\eta_{N-1}^{(1)}, \ldots, \eta_{N-1}^{(N-1)}, 0] + [0, \xi_{N-1}^{(1)}, \ldots, \xi_{N-1}^{(N-1)}]),$$

where $\sigma_{N-1}^2 = \gamma_1 \prod_{j=1}^{N-1}(1-\mu_j^2)$, $\eta_{N-1}^{(i)}$ and $\xi_{N-1}^{(i)}$ is the i-th coefficient of polynomials $\eta_{N-1}(x)$ and $\xi_{N-1}(x)$ respectively. And the determinant of the Toeplitz matrix is given by

$$|\boldsymbol{V}| = \gamma_1 \prod_{i=1}^{N-1} \sigma_i^2, \quad \sigma_i^2 = \gamma_1 \prod_{j=1}^{i} (1 - \mu_j^2).$$

While the sequential calculation of $\{T_1, \ldots, T_{N-1}\}$ requires $O(N^2)$ operations, Ammar & Gragg realized a doubling procedure avoiding most of the intermediary T_n. That is, if for $\phi_0 = T_m(\phi_m)$, define $T_{m,m}(x)$ as the rational function obtained from applying m steps of Schur algorithm to ϕ_m, such that

$$\phi_{2m} = T_{m,m}^{-1}(\phi_m) = T_{m,m}^{-1}(T_m^{-1}(\phi_0)) = T_{2m}^{-1}(\phi_0),$$

i.e. $T_{2m} = T_m \circ T_{m,m}$. The merged transformation $T_{2m} = T_m \circ T_{m,m}$ is generated through polynomial multiplications:

$$\xi_{2m} = \tilde{\eta}_m \xi_{m,m} + \xi_m \eta_{m,m}, \quad \eta_{2m} = \tilde{\xi}_m \xi_{m,m} + \eta_m \eta_{m,m}, \quad \gamma_{2m} = \gamma_m + \lambda^m \gamma_{m,m}.$$

By calculating these multiplications via FFT, the computational cost of calculating T_{2m} is $O(m \log m)$. Thus, for $N = 2^K + 1$, the cost of going through the entire doubling procedure is $\mathcal{O}(K^2 \cdot 2^K) = \mathcal{O}(N \log^2 N)$, which is demonstrated in Figure 2.1.

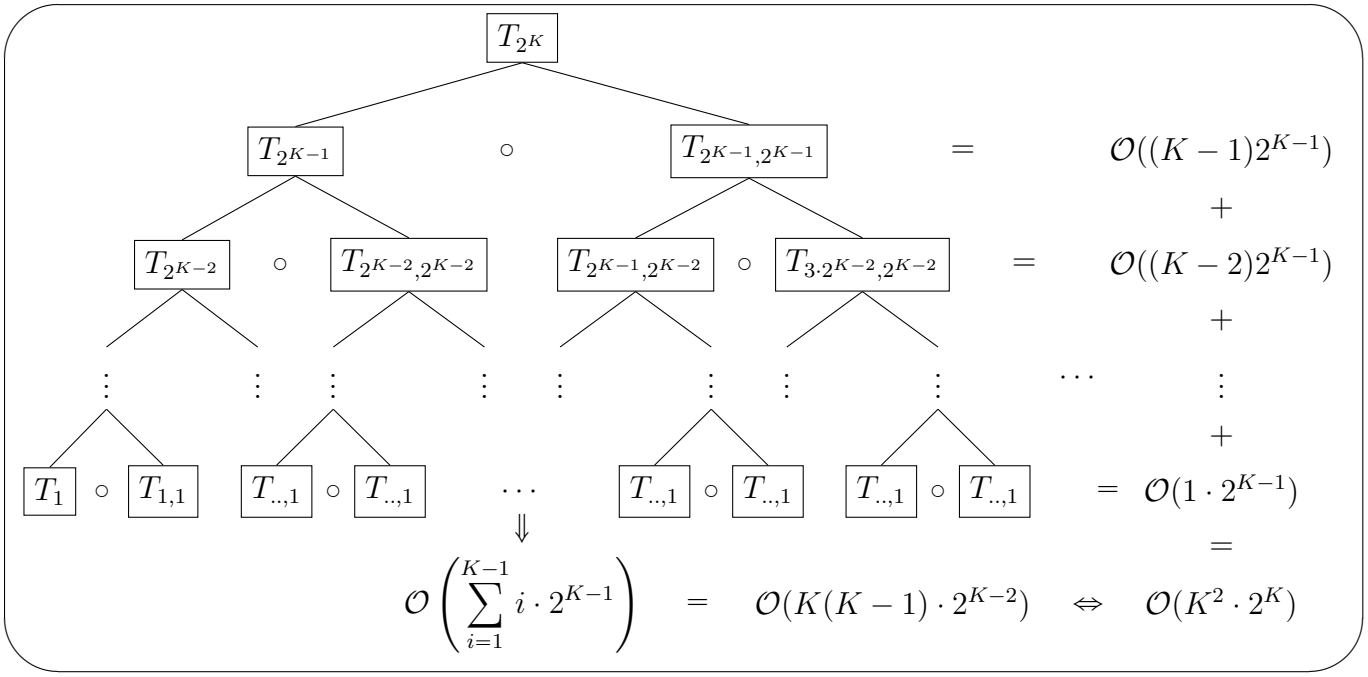


Figure 2.1: Tree diagram of the generalized Schur algorithm.

The generalized Schur algorithm for size $N = 2^K + 1$ is presented in Algorithm 1. The Ammar-Gragg version contains a number of technical accelerations, for which they prove that an exact operation count crosses over with Levinson's algorithm an $N = 2^8 + 1 = 257$ [Ammar and Gragg, 1989]. For ease of presentation we do not describe these technical improvements in Algorithm 1, although we do implement them in the *SuperGauss* library.

2.2.3 Extension of Generalized Schur Algorithm

Our extension stems from Ammar and Gragg's observation that for $n \neq m$, by defining $T_{m,n}$ as the n-steps of Schur's algorithm applied to ϕ_m, we have $T_{m+n} = T_m \circ T_{m,n}$, and

$$\xi_{n+m} = \tilde{\eta}_n \xi_{n,m} + \xi_n \eta_{n,m}, \quad \eta_{n+m} = \tilde{\xi}_n \xi_{n,m} + \eta_n \eta_{n,m}, \quad \gamma_{n+m} = \gamma_n + \lambda^n \gamma_{n,m}.$$

Assuming that N is the dimension of the target Toeplitz matrix and $M = N - 1$ is the step of the Schur algorithm, we can first decompose M into the summation of powers of 2:

$$M = \sum_{k=1}^{K} s_k, \quad s_1 < \ldots < s_K,$$

where s_k are powers of 2. For any positive integer M, the corresponding vector $\boldsymbol{s} = [s_1, \ldots, s_K]$ always exists and is unique. With this vector $\boldsymbol{s}$, we can decompose a size M rational function T_M into smaller ones with size being a power of 2

$$T_M = T_{s_1} \circ T_{c_1,s_2} \circ \ldots \circ T_{c_{K-1},s_K}, \quad c_k = \sum_{j=1}^{k} s_j \text{ for } k = 1, \ldots, K-1,$$

where each T_{c_{k-1},s_k} can be directly computed using the original generalized Schur algorithm with its input $\phi_{c_{k-1}}$ provided. We thus propose to calculate the coefficients of T_M by merging pieces of various sizes obtained from the original algorithm for powers of 2. The exact steps are given by Algorithm 2.

To better explain this procedure, we demonstrate how the extended generalized Schur

Algorithm 1 Generalized Schur Algorithm for $M = 2^k$

1: **function** GSCHUR($\alpha_0^{(M)}, \beta_0^{(M)}$)

▷ $\alpha_0^{(M)}$, $\beta_0^{(M)}$: polynomials of degree $M-1$

2: $$\begin{cases} \begin{bmatrix} \xi_{0,1} \\ \eta_{0,1} \end{bmatrix} \leftarrow \begin{bmatrix} \frac{\alpha_0^{(1)}}{\beta_0^{(1)}} \\ 1 \end{bmatrix} \\ \mu_{0,1} \leftarrow \xi_{0,1} \end{cases}$$

▷ $[T_{0,1} \leftarrow \text{GSCHUR}(\alpha_0^{(1)}, \beta_0^{(1)})]$

3: **for** $m = 1, 2, 4, \ldots, M/2$ **do**

4: $$\begin{bmatrix} \alpha_n^{(m)} \\ \beta_n^{(m)} \end{bmatrix} \leftarrow \frac{1}{x^m} \times \begin{bmatrix} \eta_{0,m} & -\xi_{0,m} \\ -\tilde{\xi}_{0,m} & \tilde{\eta}_{0,m} \end{bmatrix} \begin{bmatrix} \alpha_0^{(2m)} \\ \beta_0^{(2m)} \end{bmatrix}$$

▷ Truncate $\alpha_n^{(m)}$, $\beta_n^{(m)}$ to degree $m-1$

▷ $[\frac{\alpha_n^{(m)}}{\beta_n^{(m)}} = T_{0,m}^{-1}(\frac{\alpha_0^{(2m)}}{\beta_0^{(2m)}})]$

5: $\{\xi_{m,m}, \eta_{m,m}, \mu_{m,m}\} \leftarrow \text{GSCHUR}(\alpha_n^{(m)}, \beta_n^{(m)})$ ▷ $[T_{m,m} \leftarrow \text{GSCHUR}(\alpha_n^{(m)}, \beta_n^{(m)})]$

6: $$\begin{cases} \begin{bmatrix} \xi_{0,2m} \\ \eta_{0,2m} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{\eta}_{0,m} & \xi_{0,m} \\ \tilde{\xi}_{0,m} & \eta_{0,m} \end{bmatrix} \begin{bmatrix} \xi_{m,m} \\ \eta_{m,m} \end{bmatrix} \\ \mu_{0,2m} \leftarrow (\mu_{0,m}, \mu_{m,m}) \end{cases}$$

▷ $[T_{0,2m} = T_{0,m} \circ T_{m,m}]$

7: **end for**

8: **return** $\{\xi_{0,M}, \eta_{0,M}, \mu_{0,M}\}$

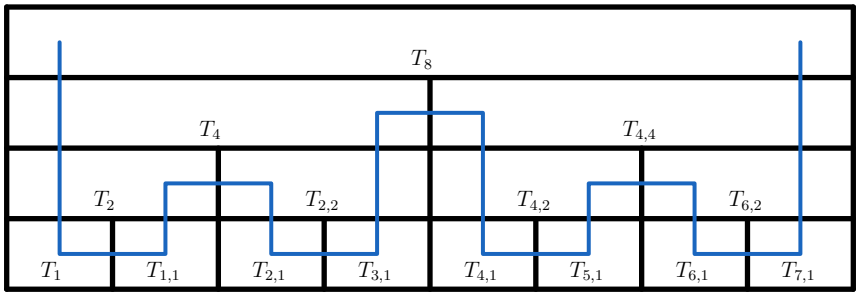
▷ $\mu_{0,M}$: vector of Schur parameters

9: **end function**

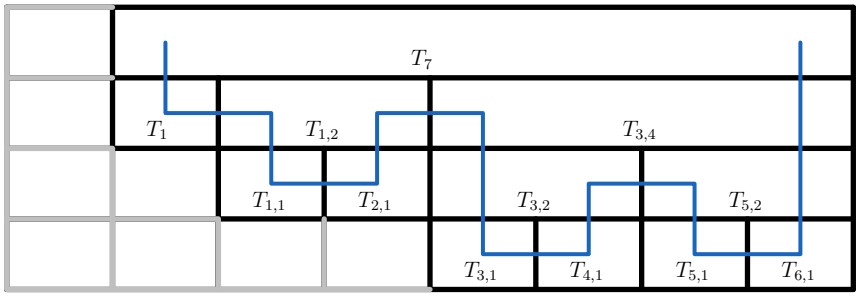
algorithm solves an $M = 7$ system in Figure 2.2(b), where

$$T_7 = T_1 \circ T_{1,2} \circ T_{3,4},$$

and compare it with the original generalized Schur algorithm for $M = 8$ in Figure 2.2(a).



(2.2(a)) Demonstration of the generalized Schur Algorithm when $M = 8$.



(2.2(b)) Demonstration of the generalized Schur Algorithm when $M = 7$.
Different rectangles stand for various transformations T, where the width of rectangles is the size of the transformation. The blue line stands for the sequence of mergence steps.

Algorithm 2 Generalized Schur Algorithm for arbitrary M

function GSCHUR2($\alpha_0^{(M)}, \beta_0^{(M)}$)

$\boldsymbol{s} = [s_1, s_2, \ldots, s_k]$, where $\text{sum}(s) = M$, s_i are power of 2 in ascending order.

$m = s_1$

$\{\xi_{0,m}, \eta_{0,m}, \mu_{0,m}\} \leftarrow \text{GSCHUR}(\alpha_0^{(m)}, \beta_0^{(m)})$

$\triangleright\ [T_{0,m} \leftarrow \text{GSCHUR}(\alpha_0^{(m)}, \beta_0^{(m)})]$

(**END** if $k == 1$)

for $n = 2, \ldots, k$ **do**

$$\begin{bmatrix} \alpha_m^{(s_n)} \\ \beta_m^{(s_n)} \end{bmatrix} \leftarrow \frac{1}{x^m} \times \begin{bmatrix} \eta_{0,m} & -\xi_{0,m} \\ -\tilde{\xi}_{0,m} & \tilde{\eta}_{0,m} \end{bmatrix} \begin{bmatrix} \alpha_0^{(m+s_n)} \\ \beta_0^{(m+s_n)} \end{bmatrix}$$

$\triangleright\ [\frac{\alpha_m^{(s_n)}}{\beta_{s_n}^{(m)}} = T_{0,m}^{-1}(\frac{\alpha_0^{(m+s_n)}}{\beta_0^{(m+s_n)}})]$

$\{\xi_{m,s_n}, \eta_{m,s_n}, \mu_{m,s_n}\} \leftarrow \text{GSCHUR}(\alpha_m^{(s_n)}, \beta_m^{(s_n)})$

$\triangleright\ [T_{m,s_n} \leftarrow \text{GSCHUR}(\alpha_m^{(s_n)}, \beta_m^{(s_n)})]$

$$\begin{cases} \begin{bmatrix} \xi_{0,m+s_n} \\ \eta_{0,m+s_n} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{\eta}_{0,m} & \xi_{0,m} \\ \tilde{\xi}_{0,m} & \eta_{0,m} \end{bmatrix} \begin{bmatrix} \xi_{m,s_n} \\ \eta_{m,s_n} \end{bmatrix} \\ \mu_{0,m+s_n} \leftarrow (\mu_{0,m}, \mu_{m,s_n}) \end{cases}$$

$\triangleright\ [T_{0,m+s_n} = T_{0,m} \circ T_{m,s_n}]$

$m = m + s_n$

end for

return $\{\xi_{0,N}, \eta_{0,N}, \mu_{0,N}\}$

end function

2.3 Inference for Stationary Gaussian Processes

In this section we present superfast algorithms for the log-likelihood and its derivatives for a general family of Gaussian observations with Toeplitz covariance structure. In addition, we show how to extend these algorithms to profile likelihood when the mean process of

target time series can be expressed in forms of a regression equation.

Let $\boldsymbol{X}(t) = \boldsymbol{\mu}_{\boldsymbol{\theta}}(t) + \boldsymbol{Z}_{\boldsymbol{\theta}}(t)$ be a d-dimensional stochastic process where $\boldsymbol{Z}_{\boldsymbol{\theta}}(t) = \begin{bmatrix} Z_1(t) & \cdots & Z_d(t) \end{bmatrix}$, $Z_i(t)$ is a mean-zero Gaussian process with separable stationary covariance structure, such that

$$\operatorname{cov}(Z_i(t), Z_j(s)) = \boldsymbol{\Sigma}_{\boldsymbol{\theta} ij} \cdot \gamma_{\boldsymbol{\theta}}(|t - s|),$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is a size $d \times d$ covariance matrix.

For equally spaced observations $[\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N], \boldsymbol{X}_n = \boldsymbol{X}(n \cdot \Delta t)$, the matrix $\boldsymbol{X}_{N \times d} = \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_N \end{bmatrix}$ has a matrix-normal distribution

$$\boldsymbol{X} \sim \operatorname{MatNorm}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{V}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}), \tag{2.3.1}$$

where $\boldsymbol{V}_{\boldsymbol{\theta}}$ is the among-column covariance matrix with elements $\boldsymbol{V}_{\boldsymbol{\theta} ij} = \gamma_{\boldsymbol{\theta}}(|i - j|\Delta t)$, $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is the among-row covariance. Its vectorized form $\operatorname{vec}(\boldsymbol{X})$ follows a multivariate normal distribution

$$\operatorname{vec}(\boldsymbol{X}) \sim \mathcal{N}(\operatorname{vec}(\boldsymbol{\mu}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}),$$

where $\operatorname{vec}(\boldsymbol{\mu})$ is the vectorized form of the mean process $\boldsymbol{\mu}_{\boldsymbol{\theta}}$, and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}$ is the Kronecker product between $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and $\boldsymbol{V}_{\boldsymbol{\theta}}$ and is a matrix of size $Nd \times Nd$.

In either a frequentist or Bayesian inference, the estimation of parameters $\boldsymbol{\theta}$ involves repeated evaluation of the log-likelihood

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{X}) = -\frac{1}{2}\operatorname{tr}\left\{\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{\theta}})' \boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right\} - \frac{d}{2}\log|\boldsymbol{V}_{\boldsymbol{\theta}}| - \frac{N}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\theta}}|, \tag{2.3.2}$$

which requires the inverse and determinant of a size N covariance matrix $\boldsymbol{V}_{\boldsymbol{\theta}}$. With the extended generalized Schur algorithm, both $\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}$ and determinant $|\boldsymbol{V}_{\boldsymbol{\theta}}|$ can be easily computed in superfast steps.

2.3.1 Superfast Computation of the Gradient

To estimate model parameters $\boldsymbol{\theta}$, one popular method is to find the maximum of the likelihood $\ell(\boldsymbol{\theta} \mid \boldsymbol{X})$ (2.3.2). Optimization methods that maximize the likelihood typically

require first order, even second order derivatives. In this section, we show that the first order derivative of $\ell(\boldsymbol{\theta} \mid \boldsymbol{X})$ with respect to parameter $\theta_i \in \boldsymbol{\theta}$ can also be computed within $\mathcal{O}(N \log^2 N)$ steps.

The first derivative of the log-likelihood with respect to parameter θ_i consists of the five parts

$$\frac{\partial}{\partial \theta_i}\ell(\boldsymbol{\theta} \mid \boldsymbol{X}) = -\frac{1}{2}\mathrm{tr}\{\underbrace{\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_i\boldsymbol{Z}_{\boldsymbol{\theta}}}_{A(\boldsymbol{\theta})} + \underbrace{2\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_i\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_{\boldsymbol{\theta}}}_{B(\boldsymbol{\theta})} + \underbrace{\boldsymbol{\Omega}_i\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_{\boldsymbol{\theta}}}_{C(\boldsymbol{\theta})}\} - \frac{d}{2}\mathrm{tr}\{\underbrace{\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_i}_{D(\boldsymbol{\theta})}\} - \frac{N}{2}\mathrm{tr}\{\underbrace{\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{\Sigma}_i}_{E(\boldsymbol{\theta})}\},$$

where

$$\boldsymbol{Z}_{\boldsymbol{\theta}} = \boldsymbol{X} - \boldsymbol{\mu}_{\boldsymbol{\theta}}, \quad \boldsymbol{Z}_i = \frac{\partial \boldsymbol{Z}_{\boldsymbol{\theta}}}{\partial \theta_i}, \quad \boldsymbol{\Sigma}_i = \frac{\partial \boldsymbol{\Sigma}_{\boldsymbol{\theta}}}{\partial \theta_i}, \quad \boldsymbol{V}_i = \frac{\partial \boldsymbol{V}_{\boldsymbol{\theta}}}{\partial \theta_i}, \quad \boldsymbol{\Omega}_{\boldsymbol{\theta}} = \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1},$$
$$\boldsymbol{\Omega}_i = \frac{\partial \boldsymbol{\Omega}_{\boldsymbol{\theta}}}{\partial \theta_i} = -\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}_i\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}, \quad \boldsymbol{\zeta}_{\boldsymbol{\theta}} = \boldsymbol{V}_{\boldsymbol{\theta}}^{-1}, \quad \boldsymbol{\zeta}_i = \frac{\partial \boldsymbol{\zeta}_{\boldsymbol{\theta}}}{\partial \theta_i} = -\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}$$

are the partial derivatives, and $\boldsymbol{V}_{\boldsymbol{\theta}}$'s partial derivative $\boldsymbol{V}_i$ is still a Toeplitz matrix.

With $\boldsymbol{\zeta}_{\boldsymbol{\theta}}$ given in terms of the Gohberg-Semencul formula (2.2.1), the computation of part $A(\boldsymbol{\theta})$, $B(\boldsymbol{\theta})$ and $C(\boldsymbol{\theta})$ only involves the multiplication between a Toeplitz matrix or its inverse ($\boldsymbol{V}_{\boldsymbol{\theta}}$, $\boldsymbol{\zeta}_{\boldsymbol{\theta}}$ or $\boldsymbol{V}_i$) and a size $N \times d$ matrix ($\boldsymbol{Z}_{\boldsymbol{\theta}}$ or $\boldsymbol{Z}_i$), which only costs $\mathcal{O}(N \log N)$ steps (since $d \ll N$ in applications, we typically ignore d when examining the complexity). As for the remaining terms, part $E(\boldsymbol{\theta})$ requires the inversion and multiplication of a size $d \times d$ matrix and takes $\mathcal{O}(d^3)$ steps, while the computation of $D(\boldsymbol{\theta})$ is non-trivial. The direct computation of part $\mathrm{tr}\{\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_i\}$ takes $\mathcal{O}(N^2)$ steps, and we here demonstrate how to obtain this term in $\mathcal{O}(N \log N)$ steps.

A Toeplitz covariance matrix $\boldsymbol{V}_i$ with first row $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_N]$ has displacement rank 2 [Kailath et al., 1979] and can be written as

$$\boldsymbol{V}_i = \frac{1}{\gamma_1}[\boldsymbol{U}_1\boldsymbol{U}'_1 - \boldsymbol{U}_2\boldsymbol{U}'_2], \tag{2.3.3}$$

where $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ are upper triangular Toeplitz matrices with first row being $[\gamma_1, \ldots, \gamma_N]$ and $[0, \gamma_2, \ldots, \gamma_N]$ respectively. Combining this with the Gohberg-Semencul representation

of $\boldsymbol{\zeta} = \boldsymbol{V}^{-1} = \frac{1}{\delta_1}\left[\boldsymbol{L}_1\boldsymbol{L}_1' - \boldsymbol{L}_2\boldsymbol{L}_2'\right]$, we have that

$$\begin{aligned}
\mathrm{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_i\} &= \frac{1}{\delta_1\gamma_1}\mathrm{tr}\{\left[\boldsymbol{L}_1\boldsymbol{L}_1' - \boldsymbol{L}_2\boldsymbol{L}_2'\right]\left[\boldsymbol{U}_1\boldsymbol{U}_1' - \boldsymbol{U}_2\boldsymbol{U}_2'\right]\} \\
&= \frac{1}{\delta_1\gamma_1}\mathrm{tr}\{\boldsymbol{L}_1\boldsymbol{L}_1'\boldsymbol{U}_1\boldsymbol{U}_1' - \boldsymbol{L}_2\boldsymbol{L}_2'\boldsymbol{U}_1\boldsymbol{U}_1' - \boldsymbol{L}_1\boldsymbol{L}_1'\boldsymbol{U}_2\boldsymbol{U}_2' + \boldsymbol{L}_2\boldsymbol{L}_2'\boldsymbol{U}_2\boldsymbol{U}_2'\} \\
&= \frac{1}{\delta_1\gamma_1}\left(\mathrm{tr}\{\boldsymbol{U}_1'\boldsymbol{L}_1\boldsymbol{L}_1'\boldsymbol{U}_1\} - \mathrm{tr}\{\boldsymbol{U}_1'\boldsymbol{L}_2\boldsymbol{L}_2'\boldsymbol{U}_1\} - \mathrm{tr}\{\boldsymbol{U}_2'\boldsymbol{L}_1\boldsymbol{L}_1'\boldsymbol{U}_2\} + \mathrm{tr}\{\boldsymbol{U}_2'\boldsymbol{L}_2\boldsymbol{L}_2'\boldsymbol{U}_2\}\right).
\end{aligned} \tag{2.3.4}$$

Since

$$\boldsymbol{A}_{ij} = \boldsymbol{U}_i'\boldsymbol{L}_j$$

is the product of two lower triangular Toeplitz matrices, we can verify that $\boldsymbol{A}_{ij}$ is also a lower triangular Toeplitz matrix that can be computed in $\mathcal{O}(N\log N)$ steps. The trace of $\boldsymbol{A}_{ij}\boldsymbol{A}_{ij}'$ can be determined in $\mathcal{O}(N)$ steps with following equation

$$\mathrm{tr}\{\boldsymbol{A}_{ij}\boldsymbol{A}_{ij}'\} = \sum_{i=1}^{N}\sum_{j=1}^{i} a_j^2 = \sum_{j=1}^{N}(n-j+1)a_j^2, \tag{2.3.5}$$

where $\boldsymbol{a}_{ij} = [a_1, \ldots, a_N]$ is the first column of $\boldsymbol{A}_{ij}$. Therefore the calculation of $\mathrm{tr}\{\boldsymbol{A}_{ij}\boldsymbol{A}_{ij}'\}$ is $\mathcal{O}(N\log N)$. All these together leads to the $\mathcal{O}(N\log N)$ complexity of $\mathrm{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_i\}$. In conclusion, the evaluation of the gradient of likelihood $\frac{\partial}{\partial\theta_i}\ell(\boldsymbol{\theta} \mid \boldsymbol{X})$ is superfast.

2.3.2 Automatic Differentiation

Given the dimension of unknown parameters p, the present algorithm for the gradient $\frac{\partial}{\partial\theta_i}\ell(\boldsymbol{\theta} \mid \boldsymbol{X})$ scales as $\mathcal{O}(p \cdot N\log^2 N)$, which is suitable for $p \ll N$. However, for $p \sim N$ repeated calculation of the trace formula above breaks the superfast scaling. This is an important restriction for applications of automatic differentiation, where derivatives with respect to each element of the autocorrelation function $\boldsymbol{\theta} = \boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_N]$ are desired. A superfast algorithm for this situation is presented here.

Consider the derivative of $\boldsymbol{V}$ with respect to γ_i, a sparse symmetric Toeplitz matrix:

$$\frac{\partial \boldsymbol{V}}{\partial \gamma_i} = \boldsymbol{I}^{(i)} = \begin{bmatrix} 0 & \cdots & \overbrace{0}^{i-1} & 1 & 0 & \cdots & \overbrace{0}^{N-i} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & 0 & \cdots & \cdots & 0 & 1 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

When $i =$, $\boldsymbol{I}^{(1)}$ is an identity matrix. For the derivative of likelihood (2.3.2) with respect to vector $\boldsymbol{\gamma}$, we have that $\frac{\partial}{\partial \boldsymbol{\gamma}} \ell(\boldsymbol{\theta} \mid \boldsymbol{X})$ is a vector with elements

$$\frac{\partial}{\partial \gamma_i} \ell(\boldsymbol{\theta} \mid \boldsymbol{X}) = -\frac{1}{2} \mathrm{tr}\{\boldsymbol{\Omega} \boldsymbol{Z}' \boldsymbol{\zeta}_i \boldsymbol{Z}\} - \frac{d}{2} \cdot \mathrm{tr}\{\boldsymbol{\zeta} \boldsymbol{I}^{(i)}\},$$

where $\boldsymbol{\zeta}_i = -\boldsymbol{\zeta} \boldsymbol{I}^{(i)} \boldsymbol{\zeta}$.

Recall that $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \omega_{11} & \cdots & \omega_{1d} \\ \vdots & \ddots & \vdots \\ \omega_{1d} & \cdots & \omega_{dd} \end{bmatrix}$ is a $d \times d$ symmetric matrix and $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_d \end{bmatrix}$ is a $N \times d$ matrix, we have

$$\begin{aligned} \mathrm{tr}\{\boldsymbol{\Omega} \boldsymbol{Z}' \boldsymbol{\zeta}_i \boldsymbol{Z}\} =& \mathrm{tr}\{\begin{bmatrix} \omega_{11} & \cdots & \omega_{1d} \\ \vdots & \ddots & \vdots \\ \omega_{d1} & \cdots & \omega_{dd} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z}_1' \\ \vdots \\ \boldsymbol{Z}_d' \end{bmatrix} \boldsymbol{\zeta}_i \begin{bmatrix} \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_d \end{bmatrix}\} \\ &= \sum_{n=1}^{d} \sum_{m=1}^{d} \omega_{nm} \cdot \boldsymbol{Z}_n' \boldsymbol{\zeta}_i \boldsymbol{Z}_m = -\sum_{n=1}^{d} \sum_{m=1}^{d} \omega_{nm} \cdot \boldsymbol{Z}_n' \boldsymbol{\zeta} \boldsymbol{I}^{(i)} \boldsymbol{\zeta} \boldsymbol{Z}_m \end{aligned}$$

For a vector $\boldsymbol{l}^{(n,m)} = [l_1, l_2, \ldots, l_N]$ of following form

$$\boldsymbol{l}^{(n,m)} = \boldsymbol{U}(\boldsymbol{a}_n) \times \boldsymbol{a}_m + \boldsymbol{U}(\boldsymbol{a}_m) \times \boldsymbol{a}_n$$

where $\boldsymbol{a}_n = \boldsymbol{\zeta} \boldsymbol{Z}_n$, $\boldsymbol{a}_m = \boldsymbol{\zeta} \boldsymbol{Z}_m$ and $\boldsymbol{U}(\boldsymbol{a}_n)$ is the upper triangular Toeplitz matrix with first row being $\boldsymbol{a}_n$, we can verify that $\boldsymbol{Z}_n' \boldsymbol{\zeta}_1 \boldsymbol{Z}_m = l_1/2$ and $\boldsymbol{Z}_n' \boldsymbol{\zeta}_i \boldsymbol{Z}_m = l_i$ for $i = 2, \ldots, N$. In other words, by putting $\tilde{\boldsymbol{l}}^{(n,m)} = [l_1/2, l_2, \ldots, l_N]$ we can obtain the vector of $\mathrm{tr}\{\boldsymbol{\Omega} \boldsymbol{Z}' \boldsymbol{\zeta}_i \boldsymbol{Z}\}$

immediately

$$\begin{bmatrix} \mathrm{tr}\{\boldsymbol{\Omega}\boldsymbol{Z}'\boldsymbol{\zeta}_1\boldsymbol{Z}\} \\ \dots \\ \mathrm{tr}\{\boldsymbol{\Omega}\boldsymbol{Z}'\boldsymbol{\zeta}_N\boldsymbol{Z}\} \end{bmatrix} = -\sum_{n=1}^{d}\sum_{m=1}^{d}\omega_{nm}\cdot\tilde{\boldsymbol{l}}^{(n,m)}$$

Similarly, considering the Gohberg-Semencul formula (2.2.1) that constructs the $\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}$ from its first column $\boldsymbol{\delta} = [\delta_1, \dots, \delta_N]$, we can define a vector $\boldsymbol{v} = [v_1, v_2, \dots, v_N]$ as

$$\boldsymbol{v} = \frac{1}{\delta_1}\{\begin{bmatrix} \delta_1 & \delta_2 & \delta_3 & \dots & \delta_N \\ 0 & \delta_1 & \delta_2 & \dots & \delta_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \delta_1 \end{bmatrix}\cdot\begin{bmatrix} N\cdot\delta_1 \\ (N-1)\cdot\delta_2 \\ \vdots \\ 1\cdot\delta_N \end{bmatrix} - \begin{bmatrix} 0 & \delta_N & \delta_{N-1} & \dots & \delta_2 \\ 0 & 0 & \delta_N & \dots & \delta_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}\cdot\begin{bmatrix} 0 \\ (N-1)\cdot\delta_N \\ \vdots \\ 1\cdot\delta_2 \end{bmatrix}\}.$$

such that $\mathrm{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{I}^{(1)}\} = v_1/2$ and $\mathrm{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{I}^{(i)}\} = v_i$ for $i = 2, \dots, N$. That is to say

$$\begin{bmatrix} \mathrm{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{I}^{(1)}\} \\ \dots \\ \mathrm{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{I}^{(N)}\} \end{bmatrix} = \tilde{\boldsymbol{v}}$$

where $\tilde{\boldsymbol{v}} = [v_1/2, v_2, \dots, v_N]$.

In conclusion, we have that

$$\frac{\partial}{\partial\boldsymbol{\gamma}}\ell(\boldsymbol{\theta} \mid \boldsymbol{X}) = \frac{1}{2}\sum_{n=1}^{d}\sum_{m=1}^{d}\omega_{nm}\cdot\tilde{\boldsymbol{l}}^{(n,m)} - \frac{d}{2}\tilde{\boldsymbol{v}}.$$

Since the vectors $\tilde{\boldsymbol{l}}$ and $\tilde{\boldsymbol{v}}$ can be computed in $\mathcal{O}(N\log N)$ steps with $\boldsymbol{\delta}$ provided, we can obtain the gradient vector $\frac{\partial}{\partial\boldsymbol{\gamma}}\ell(\boldsymbol{\theta} \mid \boldsymbol{z})$ in superfast speed.

2.3.3 Superfast Computation of the Hessian matrix

The second derivative of the log-likelihood with respect to parameter $\theta_i, \theta_j \in \boldsymbol{\theta}$ consists of six terms:

$$\begin{aligned}\frac{\partial^2}{\partial\theta_i\partial\theta_j}\ell(\boldsymbol{\theta} \mid \boldsymbol{X}) = &-\frac{1}{2}\mathrm{tr}\left\{\frac{\partial A(\boldsymbol{\theta})}{\partial\theta_j}+\frac{\partial B(\boldsymbol{\theta})}{\partial\theta_j}+\frac{\partial C(\boldsymbol{\theta})}{\partial\theta_j}\right\}-\frac{N}{2}\mathrm{tr}\left\{\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{\Sigma}_{ij}-\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{\Sigma}_j\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{\Sigma}_i\right\}\\ &-\frac{d}{2}\mathrm{tr}\left\{\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_{ij}\right\}+\frac{d}{2}\mathrm{tr}\left\{\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_j\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_i\right\},\end{aligned} \tag{2.3.6}$$

where

$$\begin{aligned}\frac{\partial A(\boldsymbol{\theta})}{\partial\theta_j} &= \boldsymbol{\Omega}_j\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_i\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_j\boldsymbol{\zeta}_i\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_{ij}\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_i\boldsymbol{Z}_j\\ \frac{\partial B(\boldsymbol{\theta})}{\partial\theta_j} &= 2\left(\boldsymbol{\Omega}_j\boldsymbol{Z}'_i\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_{ij}\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}\boldsymbol{Z}'_j\boldsymbol{\zeta}_j\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_{\boldsymbol{\theta}}\boldsymbol{Z}'_i\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_j\right)\\ \frac{\partial C(\boldsymbol{\theta})}{\partial\theta_j} &= \boldsymbol{\Omega}_{ij}\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_i\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_i\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_j\boldsymbol{Z}_{\boldsymbol{\theta}}+\boldsymbol{\Omega}_i\boldsymbol{Z}'_{\boldsymbol{\theta}}\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{Z}_j,\end{aligned}$$

and

$$\begin{aligned}&\boldsymbol{Z}_{ij}=\frac{\partial^2}{\partial\theta_i\partial\theta_j}\boldsymbol{Z}_{\boldsymbol{\theta}},\quad \boldsymbol{V}_{ij}=\frac{\partial^2}{\partial\theta_i\partial\theta_j}\boldsymbol{V}_{\boldsymbol{\theta}},\quad \boldsymbol{\Sigma}_{ij}=\frac{\partial^2}{\partial\theta_i\partial\theta_j}\boldsymbol{\Sigma}_{\boldsymbol{\theta}},\\ &\boldsymbol{\Omega}_{ij}=\frac{\partial^2}{\partial\theta_i\partial\theta_j}\boldsymbol{\Omega}_{\boldsymbol{\theta}}=\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}_j\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}_i\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}+\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}_i\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}_j\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}-\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}_{ij}\boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{-1}\\ &\boldsymbol{\zeta}_{ij}=\frac{\partial^2}{\partial\theta_i\partial\theta_j}\boldsymbol{\zeta}_{\boldsymbol{\theta}}=\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_j\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}+\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_j\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}-\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_{ij}\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\end{aligned}$$

are the second order partial derivatives, and $\boldsymbol{V}_{ij}$ is also a Toeplitz matrix.

The first three terms still consist of the multiplication between Toeplitz matrices or inverse ($\boldsymbol{V}_{\boldsymbol{\theta}}$, $\boldsymbol{\zeta}_{\boldsymbol{\theta}}$, $\boldsymbol{V}_i$, $\boldsymbol{V}_j$, $\boldsymbol{V}_{ij}$) and size $N \times d$ matrices ($\boldsymbol{Z}_{\boldsymbol{\theta}}$, $\boldsymbol{Z}_i$, $\boldsymbol{Z}_j$, $\boldsymbol{Z}_{ij}$), which takes $\mathcal{O}(N \log N)$ steps. The fourth term involves matrix computation of several size $d \times d$ matrices and is of complexity $\mathcal{O}(d^3)$. The fifth term can be computed in superfast steps using Equation (2.3.4). In the following we present the non-trivial superfast computation

for the last part $\mathrm{tr}\{\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_j\boldsymbol{\zeta}_{\boldsymbol{\theta}}\boldsymbol{V}_i\}$.

In addition to the chain rule, the partial derivative of $\boldsymbol{\zeta}_{\boldsymbol{\theta}}$ admits another computation via the Gohberg-Semencul formula

$$\boldsymbol{\zeta} = \boldsymbol{\zeta}(\boldsymbol{\delta}) = \frac{1}{\delta_1}\left[\boldsymbol{L}_1(\boldsymbol{\delta})\boldsymbol{L}_1(\boldsymbol{\delta})' - \boldsymbol{L}_2(\boldsymbol{\delta})\boldsymbol{L}_2(\boldsymbol{\delta})'\right],$$

where $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ are simple permutations of $\boldsymbol{\delta}$. Therefore we have

$$\begin{aligned}\boldsymbol{\zeta}_i =& -\frac{\delta_{1,(i)}}{\delta_1}\boldsymbol{\zeta} + \frac{1}{\delta_1}\left[\boldsymbol{L}_1(\boldsymbol{\delta}_{(i)})\boldsymbol{L}_1(\boldsymbol{\delta})' + \boldsymbol{L}_1(\boldsymbol{\delta})\boldsymbol{L}_1(\boldsymbol{\delta}_{(i)})' - \boldsymbol{L}_2(\boldsymbol{\delta}_{(i)})\boldsymbol{L}_2(\boldsymbol{\delta})' - \boldsymbol{L}_2(\boldsymbol{\delta})\boldsymbol{L}_2(\boldsymbol{\delta}_{(i)})'\right] \\ =& h(\boldsymbol{\delta}, \boldsymbol{\delta}_{(i)}),\end{aligned}$$

where $\boldsymbol{\delta}_{(i)} = \frac{\partial}{\partial\theta_i}\boldsymbol{\delta}$, $\delta_{1,(i)}$ is the first element of $\boldsymbol{\delta}_{(i)}$ and $\boldsymbol{L}_1(\boldsymbol{\delta}_{(i)}), \boldsymbol{L}_2(\boldsymbol{\delta}_{(i)})$ are lower triangular Toeplitz matrices constructed from $\boldsymbol{\delta}_{(i)}$ in the same way as $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$. To obtain this vector $\boldsymbol{\delta}_{(i)}$, recall that $\boldsymbol{\delta}$ is the first column of $\boldsymbol{\zeta}$, such that

$$\boldsymbol{V}\boldsymbol{\delta} = \boldsymbol{e}_1, \qquad \boldsymbol{e}_1 = [1, 0, \ldots, 0]'.$$

Taking derivatives on both sides, we have

$$\boldsymbol{V}_i\boldsymbol{\delta} + \boldsymbol{V}\boldsymbol{\delta}_{(i)} = 0 \Rightarrow \boldsymbol{\delta}_{(i)} = -\boldsymbol{\zeta}\boldsymbol{V}_i\boldsymbol{\delta}.$$

With $\boldsymbol{\delta}$ computed, the matrix-vector product $\boldsymbol{V}_i\boldsymbol{\delta}$ can be obtained in $\mathcal{O}(N\log N)$ steps, after which $\boldsymbol{\delta}_{(i)} = -\boldsymbol{\zeta}\boldsymbol{V}_i\boldsymbol{\delta}$ can also be obtained in $\mathcal{O}(N\log N)$ steps by applying the Gohberg-Semencul decomposition (2.2.1).

Thus the computation of $\mathrm{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_i\boldsymbol{\zeta}\boldsymbol{V}_j\} = -\mathrm{tr}\{\boldsymbol{\zeta}_i\boldsymbol{V}_j\}$ can be obtained from

$$\begin{aligned}\mathrm{tr}\{\boldsymbol{\zeta}_i\boldsymbol{V}_j\} =& \frac{1}{\delta_1}\mathrm{tr}\{\left[\boldsymbol{L}_1(\boldsymbol{\delta}_{(i)})\boldsymbol{L}_1(\boldsymbol{\delta})' + \boldsymbol{L}_1(\boldsymbol{\delta})\boldsymbol{L}_1(\boldsymbol{\delta}_{(i)})' - \boldsymbol{L}_2(\boldsymbol{\delta}_{(i)})\boldsymbol{L}_2(\boldsymbol{\delta})' - \boldsymbol{L}_2(\boldsymbol{\delta})\boldsymbol{L}_2(\boldsymbol{\delta}_{(i)})'\right]\boldsymbol{V}_j\} \\ & - \frac{\delta_{1,(i)}}{\delta_1}\mathrm{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_j\}.\end{aligned}$$

With the decomposition formula (2.3.3) of $\boldsymbol{V}_j$ and the efficient computation of the trace of product between lower and upper Toeplitz matrices (2.3.5), we can compute $\mathrm{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_i\boldsymbol{\zeta}\boldsymbol{V}_j\}$ in

$\mathcal{O}(N \log N)$ steps. In conclusion, the computation of the Hessian matrix (2.3.6) is superfast $\mathcal{O}(N \log^2 N)$.

2.3.4 Profile Likelihood

In many applications, the mean function $E[\boldsymbol{X}(t)]$ is modeled via a regression equation

$$E[\boldsymbol{X}(t)] = \boldsymbol{\mu}_{\boldsymbol{\theta}}(t) = \sum_{i=1}^{p} \boldsymbol{\beta}_i g_i(t), \tag{2.3.7}$$

and a separable covariance structure

$$\operatorname{cov}(X_i(t), X_j(s)) = \boldsymbol{\Sigma}_{ij} \cdot \gamma_{\boldsymbol{\theta}}(|t-s|).$$

In these cases, the distribution of observation matrix $\boldsymbol{X}_{N\times d}$ is given by

$$\boldsymbol{X}_{N\times d} \sim \operatorname{MatNorm}(\boldsymbol{G}\boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{V}_{\boldsymbol{\theta}}),$$

where $\boldsymbol{G}_{N\times p} = [\boldsymbol{g}_1, \cdots, \boldsymbol{g}_p]$ is the observation matrix of regression processes, $\boldsymbol{g}_i = [g_i(\Delta t), \ldots, g_i(N \cdot \Delta t)]$, $\boldsymbol{\beta}_{p\times d} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_p \end{bmatrix}$ is the vector of coefficients of the regression process, $\boldsymbol{\Sigma}_{d\times d}$ is the covariance matrix, and $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_m\}$ is the parameter set that determines the covariance matrix $\boldsymbol{V}_{\boldsymbol{\theta}}$.

In this case, for fixed $\boldsymbol{\theta}$ the condition maximum likelihood estimates

$$(\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}) = \arg\max_{\boldsymbol{\beta},\boldsymbol{\Sigma}} \left\{ -\frac{d}{2} \log |\boldsymbol{V}_{\boldsymbol{\theta}}| - \frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \operatorname{tr}[\boldsymbol{\Sigma}^{-1} (\boldsymbol{X} - \boldsymbol{G}\boldsymbol{\beta})' \boldsymbol{V}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{X} - \boldsymbol{G}\boldsymbol{\beta})] \right\}$$

are given by [Jones et al., 1998, Lysy et al., 2016] as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} = (\boldsymbol{G}' \boldsymbol{V}_{\boldsymbol{\theta}}^{-1} \boldsymbol{G})^{-1} \boldsymbol{G}' \boldsymbol{V}_{\boldsymbol{\theta}}^{-1} \boldsymbol{X}, \quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} = \frac{1}{N} (\boldsymbol{X} - \boldsymbol{G}\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})' \boldsymbol{V}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{X} - \boldsymbol{G}\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}),$$

leading to a profile likelihood function

$$\ell_{\text{prof}}(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{G}) = -\frac{Nd}{2}\log(2\pi) - \frac{d}{2}\log|\boldsymbol{V}_{\boldsymbol{\theta}}| - \frac{N}{2}\log|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}| - \frac{N}{2}. \tag{2.3.8}$$

Optimization over this function greatly reduces the dimensionality. Gradient and Hessian algorithms for the profile likelihood are provided in Appendix B.

2.4 Numerical Experiments

For the purpose of efficient inference of stationary Gaussian processes, we implemented the extended generalized Schur algorithm in **C++** in forms of a head-only library called *SuperGauss*, where the Fast Fourier transformation is implemented using the *fftw* library [Frigo and Johnson, 2005]. An **R** library, *SuperGauss* [Ling and Lysy, 2017] is also provided.

To compare the computation speed of the generalized Schur algorithm, we implemented the Levinson's algorithm in C++ and use the *fftw* library for the FFT computations. The theoretical cross-over point between the generalized Schur algorithm and Levinson algorithm is $N = 257$ [Ammar and Gragg, 1989]. In our implementation the real cross-over point is around $N = 260$. We also look into the performance of the Fortran implementation of the Hierarchical Structured Solver (HSS) algorithm of Xia et al. [2012], Xi et al. [2014], an $\mathcal{O}(N\log^2 N)$ stable algorithm that works for asymmetric and complex Toeplitz matrices as well. In Figure 2.3 we present the computation time for the extended Generalized Schur algorithm and Levinson algorithm for matrices of sizes ranging from 100×100 to $10^5\times 10^5$. To better measure the computation time, we repeat each trial 100 times and record the average value.

2.4.1 Numerical Stability Experiments

The main concern about the generalized Schur algorithm is its numerical stability. According to Stewart and Van Dooren [1997], Chandrasekaran and Sayed [1998], the Schur algorithm for Toeplitz matrix inversion is stable, and extensive numerical experiments on

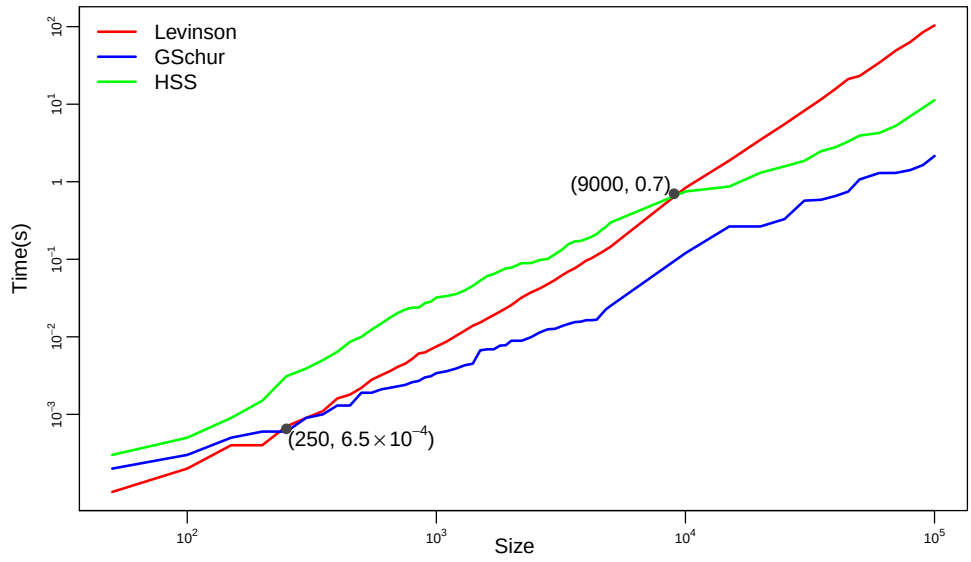


Figure 2.3: Average time for solving Toeplitz systems $\boldsymbol{V} \cdot \boldsymbol{x} = \boldsymbol{y}$ using different algorithms. A size N Toeplitz system $\boldsymbol{V} \cdot \boldsymbol{x} = \boldsymbol{y}$ comes with an $N \times N$ Toeplitz covariance matrix $\boldsymbol{V}$ and a length-N vector $\boldsymbol{y}$. The first column of $\boldsymbol{V}$ is the ACF of a fractional Gaussian noise process (2.4.1) with $\alpha = 0.8$. The vector $\boldsymbol{y}$ has elements $\boldsymbol{y}_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$.

the generalized Schur algorithm display that its growth rates of computation errors are comparable with those of the Szegö recursions, which is equivalent with algorithms like the Levinson algorithm [Ammar and Gragg, 1989]. However, numerical methods that are based on explicit inversions are usually unstable [Higham, 2002], which is exactly the case with the generalized Schur algorithm. In extreme cases where ill-conditioned Toeplitz covariance matrices are generated, the generalized Schur algorithm has worse performance than the Levinson algorithm [Stewart, 2003, Chen et al., 2006]. In this section, some numerical experiments are conducted to examine the performance of the generalized Schur algorithm with respect to various kinds of stationary Gaussian processes.

For a Toeplitz covariance matrix $\boldsymbol{V}$, its condition number is defined as

$$\kappa(\boldsymbol{V}) = ||\boldsymbol{V}||_p \cdot ||\boldsymbol{V}^{-1}||_p,$$

where matrix norm p can be arbitrary. In this chapter, we choose $p = \infty$, whose corresponding matrix norm $||\boldsymbol{V}||_\infty = \max_{1 \leqslant i \leqslant N} \sum_{j=1}^{N} |\boldsymbol{V}_{ij}|$ is the maximum absolute row sum of the matrix, and $||\boldsymbol{x}||_\infty = \max_{1 \leqslant j \leqslant N} |\boldsymbol{x}_i|$ is the maximum element of the vector. The condition number is the index of the singularity of matrices. A matrix with a high condition number κ is viewed as ill-conditioned and for singular matrices, its $\kappa = \infty$.

Given an $N \times N$ Toeplitz covariance matrix $\boldsymbol{V}$, its measurement error is defined and estimated in the following steps:

1. Simulate a length N vector $\boldsymbol{y} = [y_1, \ldots, y_N]$ whose elements are i.i.d. $y_i \sim \mathcal{N}(0, 1)$.
2. Solve the Toeplitz system $\boldsymbol{V} \cdot \boldsymbol{x} = \boldsymbol{y}$ and obtain the estimation $\hat{\boldsymbol{x}}$
3. Check the measurement error $r(\boldsymbol{V}, \hat{\boldsymbol{x}}, \boldsymbol{y}) = \frac{||\boldsymbol{V}\hat{\boldsymbol{x}} - \boldsymbol{y}||}{||\boldsymbol{V}|| \cdot ||\hat{\boldsymbol{x}}|| + ||\boldsymbol{y}||}$.

where the matrix supremum norm is applied here.

Stewart's Example

We can generate an arbitrary $N \times N$ Toeplitz covariance matrix $\boldsymbol{V}$ (with first element $\boldsymbol{V}_{1,1} = 1$) for given Schur parameters $\{\mu_k\}_{k=1}^{N-1}$ using Szegö recurrence [Ammar and Gragg, 1987].

In a numerical experiment [Stewart, 2003], ill-conditioned Toeplitz covariance matrices are generated by manipulating the Schur parameters $\{\mu_k\}_{k=1}^{N-1}$ in the particular way explained in Figure 2.4.

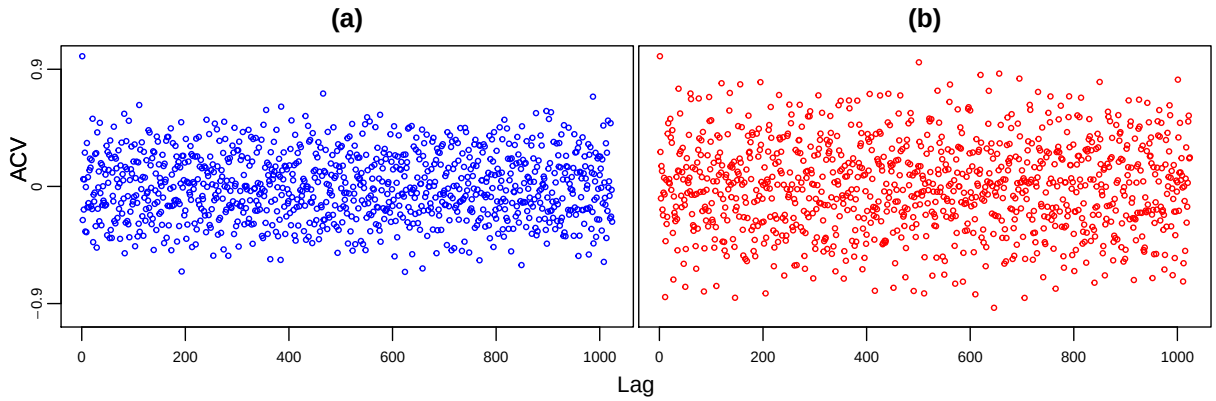


Figure 2.4: Auto-covariance of Stewart's experiments
(a) Experiment 1, where $\mu_i \sim \text{Unif}(-0.5, 0.5), i = 1, 2, \ldots, N-1$.
(b) Experiment 2, where $\mu_{10} = 1 - 10^{-6}$, $\mu_{15} = -0.99$ and $\mu_i \sim \text{Unif}(-0.3, 0.3)$ for remaining i.

In Table 2.1 we demonstrate the estimation errors of 4 different algorithms: the generalized Schur algorithm, Levinson algorithm, Cholesky decomposition, HSS algorithm and an $\mathcal{O}(N \log^{5/2} N)$ preconditioned conjugate gradient algorithm (PCG) developed by Chen et al. [2006] for long-memory processes. In both experimental setups, $\kappa(\boldsymbol{V})$ grows rapidly as matrix size N increases. For the generalized Schur algorithm, its relative error r_{GSchur} grows at a similar rate as $\kappa(\boldsymbol{V})$. On the contrary, the result of LTZ and Cholesky algorithm is accurate and robust against the conditions of $\boldsymbol{V}$. As for the PCG method, its relative error is stable but constantly large.

Despite the performance of the generalized Schur algorithm in this numerical experiment, from Figure 2.4 we can see that the auto-covariance generated in such a way hardly exists in real applications. Since our Toeplitz-system solver is developed for statistical applications, we are more interested in the performance of the generalized Schur algorithms under ill-conditioned statistical models. Time series models are roughly categorized into two types for their decay speed: short-memory processes and long-memory process. Sce-

Table 2.1: Measurement errors of different algorithms in Stewart's examples.
Experiment (a), where $\mu_i \sim \text{Unif}(-0.5, 0.5), i = 1, \cdots, N-1$.
Experiment (b), where $\mu_i \sim \text{Unif}(-0.3, 0.3), i \in \{1, 2, \ldots, N-1\} \backslash \{10, 15\}$, $\mu_{10} = 1 - 10^{-6}$, $\mu_{15} = -0.99$.

	Experiment (a)			Experiment (b)	
	$N = 64$	$N = 128$	$N = 256$	$N = 64$	$N = 128$
$\kappa(\boldsymbol{V})$	2.9×10^{4}	1.1×10^{9}	3.5×10^{15}	8.9×10^{13}	1.8×10^{17}
r_{GSchur}	8.1×10^{-13}	6.3×10^{-7}	1.3×10^{-2}	2.7×10^{-4}	2.1×10^{-1}
r_{LTZ}	7.4×10^{-16}	1.7×10^{-14}	6.2×10^{-14}	4.2×10^{-15}	2.8×10^{-14}
r_{Chol}	4.9×10^{-16}	5.6×10^{-16}	8.7×10^{-16}	9.2×10^{-16}	1.0×10^{-15}
r_{PCG}	1.5×10^{-4}	5.7×10^{-5}	4.5×10^{-5}	5.4×10^{-8}	1.2×10^{-8}
r_{HSS}	5.1×10^{-16}	4.5×10^{-10}	1.2×10^{-8}	7.1×10^{-9}	6.3×10^{-9}

narios of both types are simulated in the following sections, and the performance of the generalized Schur algorithms is evaluated.

Long-Memory process

Models for long-memory time series are believed to have ill-conditioned covariance matrices because their auto-covariances decline slowly at a power law rate [Chen et al., 2006]. Therefore they are inappropriate for the generalized Schur algorithm. In this section, two well-known models for long-memory processes are applied to study their condition numbers $\kappa(\boldsymbol{V})$ and the corresponding impact on the measurement errors of the generalized Schur algorithm.

The autoregressive fractionally integrated moving average (ARFIMA) model [Granger and Joyeux, 1980] measures the persistence of shocks by introducing fractional differentiation into autoregressive moving average models. An ARFIMA(p, d, q) model has the following form

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d X_n = (1 + \sum_{i=1}^{q} \theta_i B^i)\varepsilon_n, \quad \varepsilon_n \overset{\text{iid}}{\sim} N(0, \sigma^2),$$

where B is the lag operator such that $B^k \cdot X_n = X_{n-k}$. The ARFIMA$(0, d, 0)$ model has the following auto-correlation function (ACF)

$$\rho_n = \frac{\Gamma(n+d)\Gamma(1-d)}{\Gamma(n-d+1)\Gamma(d)} \approx n^{2d-1}, \quad n = 1, 2, \ldots.$$

When $d \in (0, 0.5)$, we have that $\sum_{n=1}^{\infty} \rho_n = \infty$, meaning that X_n has long-range persistence.

Another popular stationary model with long-range dependence is the fractional Gaussian noise (fGn), the increment process of fractional Brownian motion $B^\alpha(t)$

$$X_n = B_\alpha(n+1) - B_\alpha(n),$$

whose ACF is

$$\rho_n = \frac{1}{2}\left[(n+1)^\alpha + |n-1|^\alpha - 2 \cdot n^\alpha\right]. \tag{2.4.1}$$

For $\alpha \in (1, 2)$, we also have that $\sum_{i=1}^{\infty} \rho_i = \infty$, indicating its long-memory property. In Figure 2.5 we show the ACF of two long-memory models with different parameters. For the ARFIMA model, the long range dependence is more significant for d closer to 1. For the fGn model, the long-memory property is more obvious for α closer to 2.

In order to verify the degree of ill-conditioning for long-memory processes, we generate the covariance matrices for ARFIMA$(0, 0.49, 0)$ and fGn with $\alpha = 1.9$ of different sizes, ranging from 2000 to 10^5. By repeating the procedures of the previous section, we compute the condition number $\kappa(\boldsymbol{V})$ and measure the relative errors of the generalized Schur algorithm, LTZ and PCG algorithm (Cholesky decomposition is a $\mathcal{O}(N^3)$ algorithm, we are not going to apply it for time series longer than 1000). In Table 2.2 we show the relative errors for different long-memory processes. Judging from the condition number $\kappa(\boldsymbol{V})$, we discover that the covariance matrices of long-memory processes are ill-conditioned, but not to an extreme degree like Stewart's examples. The relative errors of the generalized Schur algorithm are systematically larger than the result of the LTZ algorithm but still within a tolerable range.

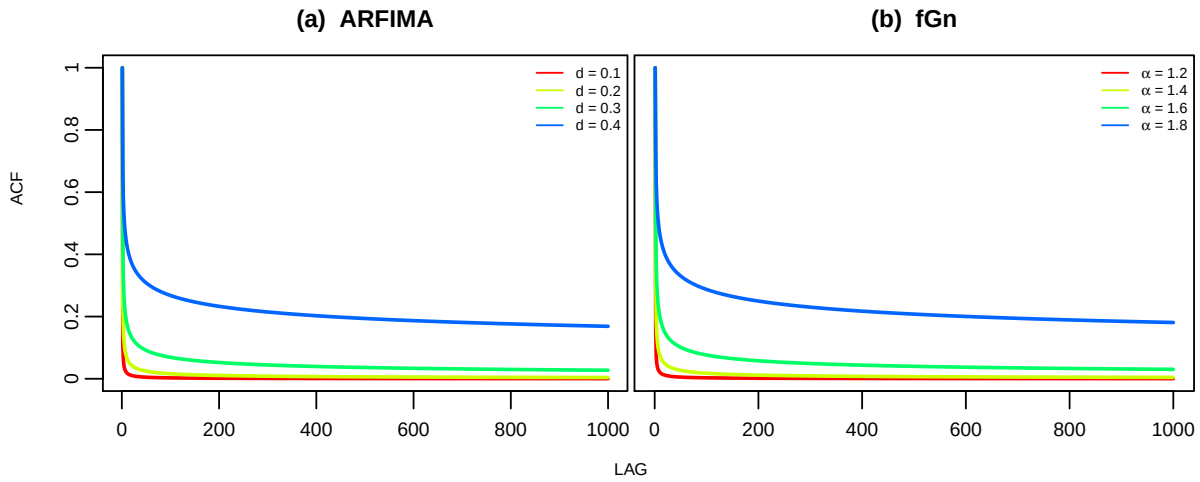


Figure 2.5: Auto-correlation function of long-memory models.
(a) ARFIMA$(0, d, 0)$ where $d = 0.1, 0.2, 0.3, 0.4$.
(b) fGn model where $\alpha = 1.2, 1.4, 1.6, 1.8$.

Model	Size N	$\kappa(\boldsymbol{V})$	r_{GSchur}	r_{LTZ}	r_{PCG}	r_{HSS}
ARFIMA(0,d,0) d = 0.49	2000	8.2×10^3	9.9×10^{-13}	6.6×10^{-15}	5.6×10^{-16}	1.3×10^{-15}
	5000	1.9×10^4	1.9×10^{-12}	9.3×10^{-15}	3.1×10^{-15}	1.3×10^{-14}
	10^4	3.5×10^4	1.2×10^{-12}	1.5×10^{-14}	6.7×10^{-16}	1.1×10^{-15}
	10^5	4.1×10^5	2.1×10^{-11}	3.6×10^{-13}	1.3×10^{-15}	1.6×10^{-14}
fBM $\alpha = 1.9$	2000	2.9×10^3	4.1×10^{-14}	8.0×10^{-15}	4.6×10^{-16}	4.9×10^{-16}
	5000	5.9×10^3	2.6×10^{-13}	1.1×10^{-14}	7.6×10^{-16}	8.9×10^{-15}
	10^4	1.0×10^4	2.0×10^{-13}	9.2×10^{-15}	7.3×10^{-16}	6.7×10^{-15}
	10^5	2.1×10^5	3.6×10^{-12}	1.2×10^{-14}	8.1×10^{-16}	3.1×10^{-14}

Table 2.2: Measurement errors of different algorithms, long-memory models.

Short Memory Process

Time series with exponential decay ACF, as an important model for short-memory processes, is established to be well-conditioned. These time series are also stationary Gaussian processes and are widely applied in applications including social communication [Karagiannis et al., 2010] and neuronal performance [Stein, 1965, Byron et al., 2009]. In this section, the performance of the generalized Schur algorithm is evaluated when analyzing processes with exponential decay ACF of the following form

$$\gamma_n = \exp\{-\lambda \cdot n^d\}, \quad n = 1, 2, \ldots, N, \tag{2.4.2}$$

where parameters λ and d determine the decreasing speed of γ_n. For larger λ and d, γ_n drops more rapidly. In Table 2.3 we show the measurement error of three different Toeplitz system solvers, where all algorithms have similarly good performance.

d	$\kappa(\boldsymbol{V})$	r_{GSchur}	r_{LTZ}	r_{PCG}	r_{HSS}
1	6.3	8.4×10^{-16}	3.2×10^{-16}	4.0×10^{-16}	5.3×10^{-16}
2	2.3	7.0×10^{-16}	4.9×10^{-16}	4.2×10^{-16}	6.2×10^{-16}
3	2.0	9.9×10^{-16}	5.4×10^{-16}	4.0×10^{-16}	4.9×10^{-16}
4	2.0	8.8×10^{-16}	3.6×10^{-16}	5.4×10^{-16}	5.6×10^{-16}

Table 2.3: Measurement errors of different algorithms, exponential decay models. Data size $N = 10^5$, $\lambda = 1$.

It is worthwhile mentioning that PCG algorithm solves the Toeplitz system by recursively updating its output, where the number of iterations is related to the condition number of Toeplitz matrix $\kappa(\boldsymbol{V})$. During our experiments with the PCG algorithm, we discovered that this algorithm solves a long-memory Toeplitz system much faster than a short-memory system. More specifically, let $\boldsymbol{V}_1$ be the covariance matrix of a long-memory process and $\boldsymbol{V}_2$ be the covariance matrix of an exponential decay process. If $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are equivalently ill-conditioned (we can achieve this by having a very small λ in (2.4.2)), PCG algorithm will take many more iterations to invert matrix $\boldsymbol{V}_2$ than $\boldsymbol{V}_1$.

2.4.2 Parameter Estimation for Long-Memory Models

One main purpose of the evaluation of the log-likelihood for stationary Gaussian processes is to estimate the parameters. In the previous section we investigate the measurement error of the generalized Schur algorithm with respect to various time processes but are still in lack of a straightforward impression of the impact of these errors. In order to reveal a potential bias when applying the generalized Schur algorithm, we design a numerical experiment to measure the accuracy of estimation results using the generalized Schur algorithm. Among all common statistical models, long-memory series are established to be ill-conditioned and turn out to be most unsuitable for the generalized Schur algorithm. In order to see the limitation of the generalized Schur algorithm, M long-dependency time series $\boldsymbol{X} = \left[\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(M)}\right]$ are generated, where $\boldsymbol{X}^{(m)} = [X_1^{(m)}, \ldots, X_N^{(m)}]$ i.i.d. follows ARFIMA$(0, d, 0)$ model for $m = 1, \ldots, M$, i.e.

$$(1 - B)^d X_n^{(m)} = \varepsilon_n, \quad \varepsilon \overset{\text{iid}}{\sim} N(0, \sigma^2), \quad n = 1, 2, \ldots, N.$$

In the simulation, we generate $M = 500$ time series with true parameters $d = 0.45, \sigma = 1$ and length of data $N = 10^4$. This is a long-memory time series model with only two unknown parameters $\boldsymbol{\theta} = \{d, \sigma\}$, which can be estimated by maximizing the following likelihood

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{X}^{(m)}) = -\frac{1}{2}\left[\boldsymbol{X}^{(m)'}\boldsymbol{V}_{\boldsymbol{\theta}}\boldsymbol{X}^{(m)} + \log|\boldsymbol{V}_{\boldsymbol{\theta}}| + N\log(2\pi)\right],$$

where $\boldsymbol{V}_{\boldsymbol{\theta}}$ is the Toeplitz covariance matrix whose first column is the auto-covariance of ARFIMA$(0, d, 0)$ multiplying σ^2.

In addition to the MLE estimates

$$\hat{\boldsymbol{\theta}}_m = \arg\max_{\boldsymbol{\theta}}\{\ell(\boldsymbol{\theta} \mid \boldsymbol{X}^{(m)})\},$$

we can also compute their covariance matrix using the observed Fisher information

$$\operatorname{cov}(\hat{\boldsymbol{\theta}}_m) = -\left[\frac{\partial^2}{\partial\boldsymbol{\theta}^2}\ell(\boldsymbol{\theta} \mid \boldsymbol{X}_i)\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_m}\right]^{-1}.$$

Based on $\hat{\boldsymbol{\theta}}_m$ and $\text{cov}(\hat{\boldsymbol{\theta}}_m)$, we evaluate the quality of estimation using three statistics of the estimator, including the bias, the MSE and the true coverage rate:

- Bias, $\frac{1}{M}\sum_{m=1}^{M}\hat{\boldsymbol{\theta}}_m - \boldsymbol{\theta}_0$ where $\boldsymbol{\theta}_0 = \{0.45, 1\}$ is the true parameter value.
- MSE, $\frac{1}{M}\sum_{m=1}^{M}(\hat{\boldsymbol{\theta}}_m - \boldsymbol{\theta}_0)^2$ measures the average of squares of errors.
- True coverage $\text{P}_\alpha(\hat{\boldsymbol{\theta}})$, for different confidence intervals for each parametric estimator it is calculated as

$$\text{P}_\alpha(\hat{\boldsymbol{\theta}}) = \frac{1}{M}\sum_{m=1}^{M}\mathbf{1}\{\boldsymbol{\theta}_0 \in \hat{\boldsymbol{\theta}}_m \pm q_\alpha \cdot \text{se}(\hat{\boldsymbol{\theta}}_m)\},$$

  where q_α is the normal quantile for significance level α, $\text{se}(\hat{\boldsymbol{\theta}}_m)$ is the square root of the diagonal elements of the covariance matrix $\text{cov}(\hat{\boldsymbol{\theta}}_m)$. When correct models are applied, their corresponding coverage rate should be close to the theoretical coverage rate, which is the significance level α.

In Table 2.4 we show the estimation results using the generalized Schur algorithm, Levinson algorithm and PCG. For all methods the estimation errors in both parameters $\{\sigma, d\}$ are negligible, and the coverage rate r_α suggests that the estimations of confidence intervals for various significant levels $\alpha = 90\%, 95\%, 99\%$ are also accurate.

	algorithm	bias	MSE	$\text{P}_{90\%}$	$\text{P}_{95\%}$	$\text{P}_{99\%}$
σ	GSchur	-8.4×10^{-5}	9.9×10^{-5}	90	95	99
	Levinson	1.5×10^{-5}	1.3×10^{-4}	91	95	99
	PCG	6.9×10^{-6}	2.6×10^{-5}	92	97	100
d	GSchur	-1.2×10^{-3}	1.0×10^{-4}	93	97	98
	Levinson	-7.4×10^{-4}	8.9×10^{-5}	93	96	99
	PCG	1.0×10^{-3}	7.4×10^{-5}	91	94	98

Table 2.4: Estimation results of different algorithms, ARFIMA$(0, d, 0)$ model. The true parameterare $d = 0.45$, $\sigma = 1$.

To conclude the experiments on numerical stability, we first propose an empirical relation between the measurement errors of the generalized Schur algorithm and the condition

number of Toeplitz matrices

$$r_{\text{GSchur}} \quad \propto \quad \kappa(\boldsymbol{V}),$$

which explains the concerns about the numerical stability of the generalized Schur algorithm. However, with further investigation of the covariance matrices involved in statistical applications, including long-memory processes whose covariance matrix is most ill-conditioned, their condition number is still within a tolerable degree such that the relative error of the superfast Toeplitz-system solver is acceptable. The result of parametric estimation experiment also supports this point of view. In general, the generalized Schur algorithm is applicable in the majority of statistical applications, and its disadvantage in numerical stability will hardly hinder the correct inference of models.

2.5 Application: Gaussian Process Factor Analysis

Summarizing a high dimensional data set with a low dimensional embedding is a standard approach for exploring the data structure. Typical techniques which can be used for dimensionality reduction includes linear discriminant analysis, principal component analysis (PCA) and factor analysis. The Gaussian process factor analysis (GPFA), whose motivation can be traced back to the use of PCA for extracting informative low dimensional views of high-dimensional neural data [Byron et al., 2009], actually accomplishes the dimensionality reduction and smoothing operations in a common probabilistic framework. In this section we describe the GPFA model and later propose a superfast Gibbs sampling for the inference.

Let $\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) & \cdots & y_D(t) \end{bmatrix} \in \mathbb{R}^{1\times D}$ be the vector of the high-dimensional processes recorded at time t, where D is the number of processes recorded. In the framework of GPFA, we try to extract a corresponding low-dimensional signal $\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) & \cdots & x_K(t) \end{bmatrix} \in \mathbb{R}^{1\times K}$ at time t, where K is the number of factors used to explain $\boldsymbol{y}$. Each factor $x_k(t)$ has mean 0 and a stationary covariance function

$$\text{cov}\left(x_k(t), x_k(s)\right) = f_k(|t-s|, \boldsymbol{\theta}_k).$$

For the discrete observation $\boldsymbol{y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_N] \in \mathbb{R}^{N\times D}, \boldsymbol{y}_n = \boldsymbol{y}(n\Delta t)$ at evenly distributed time series $\boldsymbol{t} = [\Delta t, \ldots, N\Delta t]$, we define a conditional Gaussian distribution of $\boldsymbol{y}_n$ given $\boldsymbol{x}(t)$:

$$\boldsymbol{y}_n \mid \boldsymbol{x}(t) \sim \mathcal{N}\left(\boldsymbol{x}(n\Delta t)\boldsymbol{\beta}, \boldsymbol{\Sigma}\right),$$

where $\boldsymbol{\beta}_{K\times D}$ is the coefficient of factors and $\boldsymbol{\Sigma}_{D\times D} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_D^2 \end{bmatrix}$ is the diagonal covariance matrix.

The matrix form of the GPFA model is

$$\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\boldsymbol{\Sigma}^{1/2}, \tag{2.5.1}$$

where the signal matrix $\boldsymbol{x} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{N\times K}, \boldsymbol{x}_n = \begin{bmatrix} x_1(n\Delta t) & \cdots & x_K(n\Delta t) \end{bmatrix}$, $\boldsymbol{\varepsilon}_{N\times D}$ is the matrix of white noises $\boldsymbol{\varepsilon}_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and the factor observations $\boldsymbol{x}_k$ are assumed to follow independent multivariate normal distributions

$$\boldsymbol{x}_k \overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}), \tag{2.5.2}$$

where $\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}$ is the covariance matrix with elements $\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}(n,m) = f_k(|n-m|\Delta t, \boldsymbol{\theta}_k)$.

Normally the parameters of the GPFA model are learnt in a straightforward way using the expectation-maximization (EM) algorithm, where the conditional distribution $p(\boldsymbol{x} \mid \boldsymbol{y})$ is required in the E-step:

$$\boldsymbol{x}_k \mid \boldsymbol{y} \sim \mathcal{N}\left((\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\boldsymbol{\Omega}^{-1}\text{VEC}(\boldsymbol{y}), \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)^{-1}} - (\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\boldsymbol{\Omega}^{-1}(\boldsymbol{\beta}_{k:}^T \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\right) \tag{2.5.3}$$

where $\boldsymbol{\beta}_{k:}$ is the k-th row of $\boldsymbol{\beta}$, $\boldsymbol{\Omega} = \sum_{k=1}^{K} \boldsymbol{\beta}_{k:}^T\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)} + \boldsymbol{\Sigma} \otimes \mathrm{I}_N$ is a $Nd \times Nd$ matrix. The evaluation of $E_{\boldsymbol{x}|\boldsymbol{y}}\left[\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y})\right]$, where $\ell(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y})$ is the log-likelihood of (2.5.1), requires the inversion of $\boldsymbol{\Omega}$, which can only be achieved with the Cholesky decomposition in $\mathcal{O}(d^3N^3)$ steps. The overall computation cost of the EM algorithm is too expensive.

In order to reduce the computational cost, we propose the following superfast Gibbs sampling for parameter estimation. Each step of sampling can be efficiently done in

$\mathcal{O}(N \log^2 N)$ steps. For a prior

$$\sigma_d^2 \sim \text{Inv-Gamma}(\alpha_d, \beta_d), \quad \boldsymbol{\beta}_{:d} \sim \mathcal{N}(\boldsymbol{\Psi}_d, \boldsymbol{S}_d), \quad d = 1, 2, \ldots, D,$$

the Gibbs sampling updates its various components using the analytical distributions

$$\begin{aligned} \boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\theta} &\sim p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\theta}) \\ \sigma_d^2 \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta} &\sim \text{Inv-Gamma}\left(\alpha_d^\star, \beta_d^\star\right) \\ \boldsymbol{\beta}_{:d} \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\Sigma} &\sim \mathcal{N}\left(\boldsymbol{\Psi}_d^\star, \boldsymbol{S}_d^\star\right) \\ \boldsymbol{\theta} \mid \boldsymbol{x} &\sim q(\boldsymbol{\theta} \mid \boldsymbol{x}) \end{aligned}$$

where $\alpha_d^\star = \alpha_d + \dfrac{N}{2}$, $\beta_d^\star = \beta_d + \dfrac{(\boldsymbol{y}_d - \boldsymbol{x}\boldsymbol{\beta}_{:d})^T(\boldsymbol{y}_d - \boldsymbol{x}\boldsymbol{\beta}_{:d})}{2}$, $\boldsymbol{\Psi}_d^\star = \left[\boldsymbol{S}_d^{-1} + \dfrac{\boldsymbol{x}'\boldsymbol{x}}{\sigma_d^2}\right]^{-1}\left[\boldsymbol{S}_d^{-1}\boldsymbol{\Psi}_d + \dfrac{\boldsymbol{x}'\boldsymbol{y}_d}{\sigma_d^2}\right]$ and $\boldsymbol{S}_d^\star = \left[\boldsymbol{S}_d^{-1} + \dfrac{\boldsymbol{x}'\boldsymbol{x}}{\sigma_d^2}\right]^{-1}$.

The conditional distribution $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\theta})$ is not trivial. Consider

$$\boldsymbol{y}_k^\star = \boldsymbol{y} - \sum_{i \neq k} \boldsymbol{x}_i \boldsymbol{\beta}_{i:} = \boldsymbol{x}_k \boldsymbol{\beta}_{k:} + \boldsymbol{\varepsilon}\boldsymbol{\Sigma}^{1/2}.$$

We find that $\boldsymbol{x}_k \mid \boldsymbol{y}_k^\star$ follows a multivariate normal distribution

$$\boldsymbol{x}_k \mid \boldsymbol{y}_k^\star \sim \mathcal{N}\left((\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\boldsymbol{\Omega}_k^{-1}\text{vec}(\boldsymbol{y}_k^\star), \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)^{-1}} - (\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\boldsymbol{\Omega}_k^{-1}(\boldsymbol{\beta}_{k:}^T \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\right)$$

where $\boldsymbol{\Omega}_k = \boldsymbol{\beta}_{k:}^T\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)} + \boldsymbol{\Sigma} \otimes \mathrm{I}_N$ is also an $Nd \times Nd$ matrix. Unlike the computation that requires $p(\boldsymbol{x}_k \mid \boldsymbol{y})$ (2.5.3), the calculation involving $\boldsymbol{\Omega}_k$ can be greatly simplified with the Woodbury matrix identity [Higham, 2002]

$$\boldsymbol{\Omega}_k^{-1} = \boldsymbol{\Sigma}^{-1} \otimes \mathrm{I}_N - (\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_{k:}^T\boldsymbol{\beta}_{k:}\boldsymbol{\Sigma}^{-1}) \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}\boldsymbol{Q}^{-1}$$

where $c = \boldsymbol{\beta}_{k:}\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_{k:}^T$ is a scale and $\boldsymbol{Q} = c \cdot \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)} + \mathrm{I}_N$ is an $N \times N$ Toeplitz matrix. We

find that

$$(\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\boldsymbol{\Omega}_k^{-1}\mathrm{vec}(\boldsymbol{y}_k^{\star}) = \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}\boldsymbol{Q}^{-1}\boldsymbol{y}_k^{\star}\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_{k:}^{T}$$

and

$$\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)} - (\boldsymbol{\beta}_{k:} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)})\boldsymbol{\Omega}_k^{-1}(\boldsymbol{\beta}_{k:}^{T} \otimes \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}) = \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}\boldsymbol{Q}^{-1},$$

that is to say

$$\boldsymbol{x}_k \mid \boldsymbol{y}_k^{\star} \sim \mathcal{N}\left(\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}\boldsymbol{Q}^{-1}\boldsymbol{y}_k^{\star}\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_{k:}^{T}, \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}\boldsymbol{Q}^{-1}\right),$$

whose simulation can be achieved efficiently by generating

$$\boldsymbol{x}_k = \boldsymbol{Q}^{-1}\boldsymbol{\varepsilon}_1 + \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}\boldsymbol{Q}^{-1}\boldsymbol{\varepsilon}_2,$$

where

$$\boldsymbol{\varepsilon}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}), \qquad \boldsymbol{\varepsilon}_2 \sim \mathcal{N}(\boldsymbol{y}_k^{\star}\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_{k:}^{T}, c \cdot \mathrm{I}_N).$$

Since both $\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}$ and $\boldsymbol{Q}$ are size $N \times N$ Toeplitz matrices, sampling from $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\Sigma}, \boldsymbol{\theta})$ is superfast.

As for the posterior $q(\boldsymbol{\theta} \mid \boldsymbol{x})$, we have that

$$q(\boldsymbol{\theta} \mid \boldsymbol{x}) = \prod_{k=1}^{K} q(\boldsymbol{\theta}_k \mid \boldsymbol{x}_k), \qquad q(\boldsymbol{\theta}_k \mid \boldsymbol{x}_k) \quad \propto \quad L(\boldsymbol{\theta}_k \mid \boldsymbol{x}_k)$$

where $L(\boldsymbol{\theta}_k \mid \boldsymbol{x}_k)$ is the likelihood of (2.5.2)

$$L(\boldsymbol{\theta}_k \mid \boldsymbol{x}_k) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{x}_k {\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}}^{-1} \boldsymbol{x}_k\right)}{\sqrt{|2\pi\boldsymbol{V}_{\boldsymbol{\theta}}^{(k)}|}},$$

To verify the quality of the proposed Gibbs sampler, we simulated a length $N = 2000$, $d = 10$ dimensional data $\boldsymbol{y}$ containing two factors $\boldsymbol{x}_1, \boldsymbol{x}_2$, where $\boldsymbol{x}_1$ is a short-memory

process with exponential decay ACF and $\boldsymbol{x}_2$ is a long-memory fGn process

$$\begin{aligned}\text{ACF}_{\boldsymbol{x}_1}(n) &= \exp\{-\lambda \cdot n^2\} \\ \text{ACF}_{\boldsymbol{x}_2}(n) &= \frac{1}{2}\left[(n+1)^\alpha + |n-1|^\alpha - 2 \cdot n^\alpha\right].\end{aligned}$$

For $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$, their elements are drawn from uniform distributions

$$\boldsymbol{\beta}_{ij} \overset{\text{iid}}{\sim} \text{Unif}(-10, 10), \quad \sigma_j \overset{\text{iid}}{\sim} \text{Unif}(0, 3), \quad 1 \leqslant i \leqslant 2, 1 \leqslant j \leqslant 10.$$

In Figure 2.6 we show the posterior distribution of α and λ, and in Table 2.4 we demonstrate the point estimation of $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ with the standard deviation. The estimated coefficients $\{\hat{\alpha}, \hat{\lambda}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Sigma}}\}$ are very close to their true value, indicating that the result of the proposed Gibbs sampling procedure for GPFA model (2.5.1) is consistent and asymptotically unbiased.

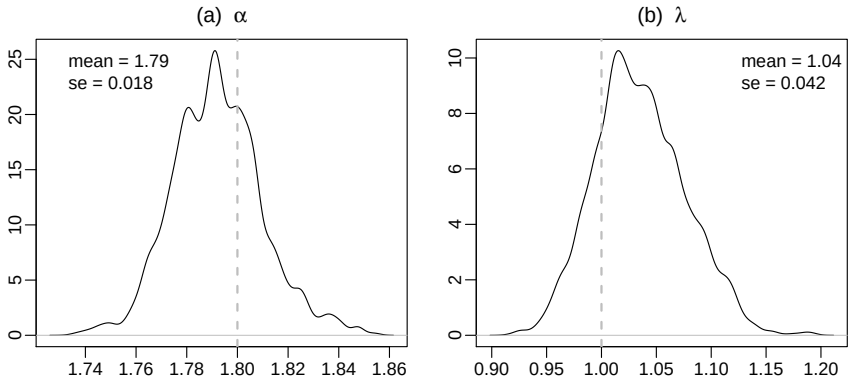


Figure 2.6: Posterior distribution of the factor parameters $\{\alpha, \lambda\}$ using the proposed Gibbs sampling procedure.
(a) The posterior density of estimated α.
(b) The posterior density of estimated λ.

	$\boldsymbol{\beta}_{1,1}$	$\boldsymbol{\beta}_{1,2}$	$\boldsymbol{\beta}_{1,3}$	$\boldsymbol{\beta}_{1,4}$	$\boldsymbol{\beta}_{1,5}$	$\boldsymbol{\beta}_{1,6}$	$\boldsymbol{\beta}_{1,7}$	$\boldsymbol{\beta}_{1,8}$	$\boldsymbol{\beta}_{1,9}$	$\boldsymbol{\beta}_{1,10}$
True	9.2	6.4	1.2	−3.9	4.0	−2.2	−1.1	5.2	4.1	−5.2
Mean	9.2	6.2	0.98	−4.0	4.1	−2.4	−1.2	5.0	4.3	−5.2
SD	0.16	0.093	0.064	0.11	0.10	0.11	0.082	0.10	0.13	0.11
	$\boldsymbol{\beta}_{2,1}$	$\boldsymbol{\beta}_{2,2}$	$\boldsymbol{\beta}_{2,3}$	$\boldsymbol{\beta}_{2,4}$	$\boldsymbol{\beta}_{2,5}$	$\boldsymbol{\beta}_{2,6}$	$\boldsymbol{\beta}_{2,7}$	$\boldsymbol{\beta}_{2,8}$	$\boldsymbol{\beta}_{2,9}$	$\boldsymbol{\beta}_{2,10}$
True	−7.9	3.4	4.2	7.0	−6.8	7.3	5.5	5.5	−8.5	6.2
Mean	−7.6	3.5	4.2	6.9	−6.7	7.2	5.4	5.5	−8.4	6.1
SD	0.35	0.19	0.15	0.26	0.25	0.26	0.20	0.23	0.32	0.25
	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8	σ_9	σ_{10}
True	2.0	2.7	0.71	1.8	0.81	2.9	0.55	2.6	1.7	2.3
Mean	2.0	2.6	0.78	1.9	0.96	2.9	0.61	2.4	1.5	2.3
SD	0.077	0.091	0.035	0.037	0.036	0.038	0.031	0.091	0.044	0.039

Table 2.5: Estimated $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ for the simulated GPFA data.

2.6 Discussion

In this chapter, we have provided computationally efficient algorithms for the inference of stationary Gaussian processes. Realizing that the covariance matrix for stationary Gaussian process is Toeplitz, we implemented the generalized Schur algorithm that solves the Toeplitz systems in superfast steps ($\mathcal{O}(N \log^2 N)$) and extend the range of this algorithm from $N = 2^K + 1$ to an arbitrary N. With a superfast solution to the Toeplitz system, the evaluation of the log-likelihood (2.3.2) and its derivatives for a general family of Gaussian observations with Toeplitz covariance structure can also be accomplished in superfast speed, which greatly reduce the time cost for parameter estimation, in either frequentist or Bayesian approaches. Profile likelihood for a special condition (2.3.7) is also provided for dimensionality reduction.

Based on our **R**/**C++** implementation of the extended generalized Schur algorithm, extensive numerical experiments are conducted to compare the superfast method and other distinguished Toeplitz-system solves in aspects of overall computation speed and numerical stability. Despite that the generalized Schur algorithm can be unstable in very extreme

cases, we show that for statistical applications its measurement error remains within a tolerable extent and will return unbiased estimates. Finally, we introduce a GPFA model for smoothing and dimensionality reduction and propose a superfast Gibbs sampling procedure that returns consistent and asymptotically unbiased estimates.

Chapter 3

Camera Error Correction for Particle Tracking

3.1 Introduction

With the development of high-resolution microscopy, single-particle tracking has emerged as an invaluable tool in the study of biophysical and transport properties of diverse soft materials. Examples of applications include cellular membrane dynamics [Saxton and Jacobson, 1997], drug delivery mechanisms [Suh et al., 2005], properties of colloidal particles [Lee et al., 2007], mechanisms of virus infection [van der Schaar et al., 2008], microrheology of complex fluids and living cells [Mason et al., 1997, Wirtz, 2009] and functional analyses of the cytoskeleton [Gal et al., 2013].

Passive single-particle tracking refers to experiments in which microscale probes and/or pathogens (e.g., viruses) are recorded without external forcing, producing high-resolution time series of particle positions from which dynamical properties of the transport medium are inferred. In many of these experiments, the resulting analysis hinges pivotally on the measurement of particles' mean square displacement (MSD), which for a k-dimensional particle trajectory $\boldsymbol{X}(t) = \big(X_1(t), \ldots, X_k(t)\big)$ (with $k \in \{1, 2, 3\}$ depending on the experi-

ment) is given by

$$\text{MSD}_{\boldsymbol{X}}(t) = \frac{1}{k} \times E\big[\|\boldsymbol{X}(t) - \boldsymbol{X}(0)\|^2\big] = \frac{1}{k} \times \sum_{j=1}^{k} E\big[|X_j(t) - X_j(0)|^2\big]. \tag{3.1.1}$$

For particles diffusing in viscous media, empirical evidence suggests that dynamics are accurately modeled by Brownian motion [Einstein, 1956]. The MSD is then linear in time,

$$\text{MSD}_{\boldsymbol{X}}(t) = 2Dt,$$

and the diffusion coefficient D is determined by the Stokes-Einstein relation [Einstein, 1956, Edward, 1970]

$$D = \frac{k_B T}{6\pi\eta r}, \tag{3.1.2}$$

where r is the particle radius, T is temperature, η is the viscosity of the medium, and k_B is the Boltzmann constant.

However, due to the microstructure of large molecular weight biopolymers (e.g., mucins in mucosal layers), most biological fluids are *viscoelastic*. Unlike viscous fluids that have constant resistance to deformation, viscoelastic fluids exhibit time-dependent strain. In such fluids, a nearly ubiquitous experimental finding has been that the MSD has sublinear power-law scaling over a given range of timescales,

$$\text{MSD}_{\boldsymbol{X}}(t) \sim 2Dt^{\alpha}, \qquad t_{\min} < t < t_{\max}, \quad 0 < \alpha < 1. \tag{3.1.3}$$

which is referred to as subdiffusion. Due to its pervasiveness, interpretation of the subdiffusion parameters (α, D) has far-reaching consequences for numerous biological applications, for example: distinguishing signatures of healthy versus pathological human bronchial epithelial mucus [Hill et al., 2014]; cytoplasmic crowding [Weiss et al., 2004]; local viscoelasticity in protein networks [Amblard et al., 1996]; dynamics of telomeres in the nucleus of mammalian cells [Bronstein et al., 2009]; and microstructure dynamics of entangled F-Actin networks [Wong et al., 2004].

Unlike for viscous fluids exhibiting ordinary (linear) diffusion, the precise manner in

which the properties of a viscoelastic fluid determine its subdiffusion parameters (α, D) is unknown, such that (α, D) must be estimated from particle tracking data. To this end, a widely-used approach is to apply ordinary least-squares to a non-parametric estimate of the MSD against time on the log-log scale [e.g., Qian et al., 1991]. While minimal modeling assumptions suffice to make this estimator consistent [Michalet, 2010] for finite-length trajectories, the non-parametric MSD estimator at longer timescales is severely biased [Mellnik et al., 2016]. Thus, in practice the information about longer timescales is typically discarded, at the expense of considerable loss in statistical efficiency. In contrast, fully parametric subdiffusion estimators specify a complete stochastic process for $\boldsymbol{X}(t)$ as a function of (α, D) [e.g., Berglund, 2010, Lysy et al., 2016, Mellnik et al., 2016], whereby optimal statistical efficiency is achieved via likelihood-based inference. However, the accuracy of these parametric estimators critically depends on the adequacy of the parametric model, and particle tracking measurements are well known to be corrupted by various sources of experimental noise.

Noise in single-particle tracking experiments can be categorized roughly into two types. Low-frequency noise, originating primarily from slow drift currents in the fluid itself, is typically removed from particle trajectories by way of various linear detrending methods [e.g., Fong et al., 2013, Rowlands and So, 2013, Koslover et al., 2016, Mellnik et al., 2016]. In contrast, high-frequency noise can be due to a variety of reasons: mechanical vibrations of the instrumental setup; particle displacement while the camera shutter is open; noisy estimation of true position from the pixelated microscopy image; error-prone tracking of particle positions when they are out of the camera focal plane. A systematic review of high-frequency or *localization* errors in single-particle tracking is given by Deschout et al. [2014]. The effect of such noise is to distort the MSD at the shortest observation timescales. Since fully-parametric models extract far more information about (α, D) from short timescales than long ones, their accuracy in the presence of high-frequency noise can suffer considerably.

In a seminal work, Savin and Doyle [2005] present a theoretical model for localization error, encompassing most of the approaches reviewed by Deschout et al. [2014]. The parameters of the Savin-Doyle model can be derived either from first-principles [for instance, by analyzing uncertainty in position-extraction algorithms, e.g., Mortensen et al., 2010,

Chenouard et al., 2014, Kowalczyk et al., 2014, Burov et al., 2017], or empirically [via signal-free control experiments, e.g., Savin and Doyle, 2005, Deschout et al., 2014]. Model-based methods for estimating localization error have also been proposed, under the assumption of ordinary diffusion $\alpha = 1$ [e.g., Michalet, 2010, Berglund, 2010, Michalet and Berglund, 2012, Vestergaard et al., 2014, Ashley and Andersson, 2015, Calderon, 2016].

The Savin-Doyle theoretical framework accounts for a wide range of experimental errors. However, due to the extreme complexity and inter-dependence between various sources of localization error, the Savin-Doyle model cannot account for them all. This is illustrated in the control experiment of Figure 3.1(a), where trajectories of 1 µm diameter tracer particles are recorded in water, for which it is known that $\alpha = 1$, and for which D may be determined theoretically by the Stokes-Einstein relation (3.1.2). However, the Savin-Doyle model estimates both of these parameters with considerable bias (Figure 3.1(b)).

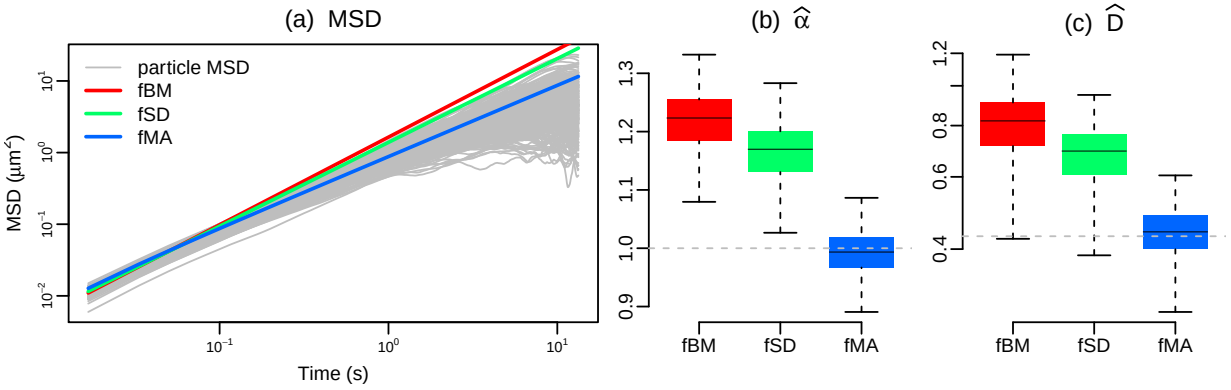


Figure 3.1: (a) Pathwise empirical MSD for 1931 particles of diameter 1 µm recorded at $\Delta t = 1/60$ s, and fitted MSD for three parametric models: fractional Brownian Motion (fBM); fBM with Savin-Doyle noise correction (fSD), and fBM with the noise correction proposed in this chapter (fMA). (b-c) Estimated values of α and D for each particle and parametric model. The predicted values from Stokes-Einstein theory are given by the horizontal dashed lines.

In this chapter, we propose a likelihood-based method for correcting localization errors, complementing the theoretical Savin-Doyle approach. We consider a family of high-frequency filters that can readily be applied to a wide range of parametric particle position

models. We show how to combine the high-frequency filter with parametric approaches to correct for low-frequency drift, and estimate all model parameters in a computationally efficient manner. Extensive simulations and analyses of experimental data suggest that a one-parameter version of our filter performs remarkably well, both for estimating the true values of (α, D), and compared to the state-of-the-art in denoising procedures (e.g., Figure 3.1(c)).

The remainder of the chapter is organized as follows. In Section 3.2 we review a number of existing subdiffusion estimators and high-frequency error-correction techniques. In Section 3.3 we present our family of high-frequency filters, along with some theoretical justification for the proposed construction. Sections 3.4-3.5 contain simulation results and analyses of numerous viscous and viscoelastic particle tracking experiments comparing our estimator with existing alternatives. Section 3.6 offers concluding remarks and directions for further work.

3.2 Existing Subdiffusion Estimators

3.2.1 Semiparametric Least-Squares Estimator

Let $\boldsymbol{X} = (\boldsymbol{X}_0, \ldots, \boldsymbol{X}_N)$, $\boldsymbol{X}_n = \big(X_1(n \cdot \Delta t), \ldots, X_k(n \cdot \Delta t)\big)$ denote the discrete-time observations of a given particle recorded at frequency $1/\Delta t$. Assuming that the position process $\boldsymbol{X}(t)$ has second order stationary increments

$$E\big[\|\boldsymbol{X}(s+t) - \boldsymbol{X}(s)\|^2\big] = E\big[\|\boldsymbol{X}(t) - \boldsymbol{X}(0)\|^2\big], \tag{3.2.1}$$

a standard nonparametric estimator for the particle MSD is given by

$$\widehat{\mathrm{MSD}}_{\boldsymbol{X}}(n \cdot \Delta t) = \frac{1}{k \cdot (N-n+1)} \sum_{i=0}^{N-n} \|\boldsymbol{X}_{n+i} - \boldsymbol{X}_i\|^2. \tag{3.2.2}$$

Based on the linear relation (3.1.3)

$$\log \mathrm{MSD}_{\boldsymbol{X}}(t) = \log 2D + \alpha \log t \tag{3.2.3}$$

over the subdiffusion timescale $t \in (t_{\min}, t_{\max})$, perhaps the most commonly-used subdiffusion estimator is the least-squares solution regressing $y_n = \log\big(\widehat{\mathrm{MSD}}_{\boldsymbol{X}}(n \cdot \Delta t)\big)$ onto $x_n = \log(n \cdot \Delta t)$, namely

$$\hat{\alpha} = \frac{\sum_{n=0}^{N}(y_n - \bar{y})(x_n - \bar{x})}{\sum_{n=0}^{N}(x_n - \bar{x})^2}, \qquad \hat{D} = \tfrac{1}{2}\exp(\bar{y} - \hat{\alpha}\bar{x}). \tag{3.2.4}$$

The least-squares subdiffusion estimator is easy to implement and consistent under the minimal assumption of (3.2.1) and $t_{\max} = \infty$ [Sikora et al., 2017]. However, it also presents two major drawbacks. First, the errors underlying the regression (3.2.3) are neither homoscedastic nor uncorrelated [Sikora et al., 2017], such that (3.2.4) is statistically inefficient. Second, it is common practice to account for low-frequency noise by calculating the empirical MSD (3.2.2) from the drift-subtracted positions

$$\tilde{\boldsymbol{X}}_n = (\boldsymbol{X}_n - \boldsymbol{X}_0) - n \cdot \overline{\Delta \boldsymbol{X}},$$

where $\overline{\Delta \boldsymbol{X}} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{X}_n - \boldsymbol{X}_{n-1})$ is the average displacement over the interobservation time Δt. However, a straightforward calculation [Mellnik et al., 2016] shows that $\tilde{\boldsymbol{X}}_N = 0$, such that $\widehat{\mathrm{MSD}}_X(n \cdot \Delta t)$ becomes increasingly biased towards zero as n approaches N. Consequently, a widely-reported figure [e.g., Weihs et al., 2007] suggests that, prior to fitting (3.2.4), the largest 30% of MSD lag times are discarded, thus severely compounding the inefficiency of the least-squares subdiffusion estimator when low-frequency noise correction is applied.

3.2.2 Fully-Parametric Subdiffusion Estimators

While the semiparametric estimator (3.2.4) operates under minimal modeling assumptions, complete specification of the stochastic process $\boldsymbol{X}(t)$ provides not only a considerable increase in statistical efficiency [e.g., Mellnik et al., 2016], but in fact is necessary to establish dynamical properties of particle-fluid interactions which cannot be determined from second-order moments (such as the MSD) alone [Gal et al., 2013, Lysy et al., 2016]. A convenient framework for stochastic subdiffusion modeling is the location-scale model

of Lysy et al. [2016],

$$\boldsymbol{X}(t) = \sum_{j=1}^{d} \boldsymbol{\beta}_j f_j(t) + \boldsymbol{\Sigma}^{1/2} \boldsymbol{Z}(t), \tag{3.2.5}$$

where $f_1(t), \dots f_d(t)$ are known functions accounting for low-frequency drift (typically linear, $f_1(t) = t$, and occasionally quadratic, $f_2(t) = t^2$), $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_d \in \mathbb{R}^k$ are regression coefficients, $\boldsymbol{\Sigma}_{k\times k}$ is a variance matrix, and $\boldsymbol{Z}(t) = \big(Z_1(t), \dots, Z_k(t)\big)$ are iid continuous stationary-increments (CSI) Gaussian processes with mean zero and MSD parametrized by $\boldsymbol{\varphi}$,

$$\text{MSD}_Z(t) = E\big[\|Z_j(t) - Z_j(0)\|^2\big] = \eta(t \mid \boldsymbol{\varphi}),$$

such that the MSD of the drift-subtracted process $\tilde{\boldsymbol{X}}(t) = \boldsymbol{X}(t) - \sum_{j=1}^{d} \boldsymbol{\beta}_j f_j(t)$ is given by

$$\text{MSD}_{\tilde{\boldsymbol{X}}}(t) = \tfrac{1}{k}\text{tr}(\boldsymbol{\Sigma}) \cdot \eta(t \mid \boldsymbol{\varphi}).$$

Perhaps the simplest parametric subdiffusion model sets $Z_j(t) = B_\alpha(t)$ to be fractional Brownian Motion (fBM) [e.g., Szymanski and Weiss, 2009, Weiss, 2013], a mean-zero CSI Gaussian process with covariance function

$$\text{cov}\big(B_\alpha(t), B_\alpha(s)\big) = \tfrac{1}{2}(|t|^\alpha + |s|^\alpha - |t-s|^\alpha), \qquad 0 < \alpha < 2.$$

Indeed, as the covariance function of a CSI process is completely determined by its MSD, fBM is the only (mean-zero) CSI Gaussian process exhibiting *uniform* subdiffusion,

$$\text{MSD}_{B_\alpha}(t) = t^\alpha, \qquad 0 < t < \infty,$$

in which case the diffusivity coefficient is given by

$$D = \frac{1}{2k} \times \text{tr}(\boldsymbol{\Sigma}).$$

Other examples of driving CSI processes are the confined diffusion model of Ernst et al. [2017] and the viscoelastic Generalized Langevin Equation (GLE) of McKinley et al. [2009],

both of which exhibit *transient* (anomalous) subdiffusion, i.e., power-law scaling only on a given timescale $t \in (t_{\min}, t_{\max})$. In this case, the subdiffusion parameters (α, D) become functions of the other parameters, namely $\alpha = \alpha(\boldsymbol{\varphi})$ and $D = D(\boldsymbol{\varphi}, \boldsymbol{\Sigma})$. We shall revisit these transient subdiffusion models in Section 3.4.

Parameter estimation for the location-scale model (3.2.5) can be done by maximum likelihood. Let $\Delta \boldsymbol{X}_n = \boldsymbol{X}_{n+1} - \boldsymbol{X}_n$ denote the nth trajectory increment, and $\Delta \boldsymbol{X} = (\Delta \boldsymbol{X}_0, \ldots, \Delta \boldsymbol{X}_{N-1})$. Then $\Delta \boldsymbol{X}$ are consecutive observations of a stationary Gaussian time series with autocorrelation function

$$\text{ACF}_{\Delta \boldsymbol{X}}(h) = \text{cov}(\Delta \boldsymbol{X}_n, \Delta \boldsymbol{X}_{n+h}) = \boldsymbol{\Sigma} \times \gamma(h \mid \boldsymbol{\varphi}),$$

where

$$\gamma(n \mid \boldsymbol{\varphi}) = \tfrac{1}{2} \times \Big\{ \eta(|n-1| \cdot \Delta t \mid \boldsymbol{\varphi}) + \eta(|n+1| \cdot \Delta t \mid \boldsymbol{\varphi}) - 2\eta(|n| \cdot \Delta t \mid \boldsymbol{\varphi}) \Big\},$$

such that the increments follow a matrix-normal distribution,

$$\Delta \boldsymbol{X}_{N \times k} \sim \text{MatNorm}(\boldsymbol{F}\boldsymbol{\beta}, \boldsymbol{V}_{\boldsymbol{\varphi}}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{\beta}_{d \times k} = [\boldsymbol{\beta}_1 \mid \cdots \mid \boldsymbol{\beta}_d]'$, $\boldsymbol{F}_{N \times d}$ is a matrix with elements $F_{nm} = f_m((n+1) \cdot \Delta t) - f_m(n \cdot \Delta t)$, and $\boldsymbol{V}_{\boldsymbol{\varphi}}$ is an $N \times N$ Toeplitz matrix with element (n, m) given by $V_{\boldsymbol{\varphi}}^{(n,m)} = \gamma(n - m \mid \boldsymbol{\varphi})$, such that the log-likelihood function is given by

$$\begin{aligned} \ell(\boldsymbol{\varphi}, \boldsymbol{\beta}, \boldsymbol{\Sigma} \mid \Delta \boldsymbol{X}) = &-\frac{1}{2} \text{tr} \left\{ \boldsymbol{\Sigma}^{-1} (\Delta \boldsymbol{X} - \boldsymbol{F}\boldsymbol{\beta})' \boldsymbol{V}_{\boldsymbol{\varphi}}^{-1} (\Delta \boldsymbol{X} - \boldsymbol{F}\boldsymbol{\beta}) \right\} \\ &- \frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{k}{2} \log |\boldsymbol{V}_{\boldsymbol{\varphi}}|. \end{aligned}$$

In order to calculate the MLE of $\boldsymbol{\theta} = (\boldsymbol{\varphi}, \boldsymbol{\beta}, \boldsymbol{\Sigma})$, model (3.2.5) has two appealing properties. First, for given $\boldsymbol{\varphi}$, the conditional MLEs of $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ can be obtained analytically using the profile likelihood described in Section 2.3.4, such that the optimization problem can be reduced by $2k + \binom{k}{2}$ dimensions by calculating the profile likelihood $\ell_{\text{prof}}(\boldsymbol{\varphi} \mid \Delta \boldsymbol{X}) = \max_{\boldsymbol{\beta}, \boldsymbol{\Sigma}} \ell(\boldsymbol{\varphi}, \boldsymbol{\beta}, \boldsymbol{\Sigma} \mid \Delta \boldsymbol{X})$. Second, we show in Chapter 2 that the computa-

tional bottleneck in $\ell_{\text{prof}}(\boldsymbol{\varphi} \mid \Delta \boldsymbol{X})$ involves the calculation of $\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}$ and its log-determinant. While the computational cost of these operations is $\mathcal{O}(N^3)$ for general variance matrices, for Toeplitz matrices it is only $\mathcal{O}(N^2)$ using the Durbin-Levinson algorithm [Levinson, 1947, Durbin, 1960], or more recently, only $\mathcal{O}(N \log^2 N)$ using the Generalized Schur algorithm [Kailath et al., 1979, Ammar and Gragg, 1988, Ling and Lysy, 2017].

3.2.3 Savin-Doyle Noise Model

In order to characterize high-frequency noise in particle tracking experiments, Savin and Doyle [2005] decompose it into so-called *static* and *dynamic* sources. Static noise is due to measurement error in the recording of the position of the particle at a given time. Thus, if $\boldsymbol{X}_n$ denotes the true particle position at time $t = n \cdot \Delta t$, and $\boldsymbol{Y}_n$ is its recorded value, then Savin and Doyle suggest the additive error model

$$\boldsymbol{Y}_n = \boldsymbol{X}_n + \varepsilon_n, \tag{3.2.6}$$

where ε_n is a k-dimensional stationary process independent of $\boldsymbol{X}(t)$. Thus, if the autocorrelation of the static noise is denoted as

$$\text{ACF}_\varepsilon(n) = \text{cov}(\varepsilon_m, \varepsilon_{m+n}),$$

the MSD of the observations becomes

$$\begin{aligned} \text{MSD}_{\boldsymbol{Y}}(n) &= \tfrac{1}{k} \times E\big[\|\boldsymbol{Y}_n - \boldsymbol{Y}_0\|^2\big] \\ &= \text{MSD}_{\boldsymbol{X}}(n) + \tfrac{1}{k} \times 2 \cdot \text{tr}\big(\text{ACF}_\varepsilon(0) - \text{ACF}_\varepsilon(n)\big). \end{aligned}$$

Savin and Doyle describe how to estimate the temporal dynamics of ε_n by recording immobilized particles, i.e., for which it is known that $\boldsymbol{X}_n \equiv 0$. Over a wide range of signal-to-noise ratios, they report that ε_n is effectively white noise,

$$\text{ACF}_\varepsilon(n) = \boldsymbol{\Sigma}_\varepsilon \cdot \mathbb{1}(n = 0),$$

a result corroborated by many other experiments [Deschout et al., 2014]. For the canonical trajectory model of fractional Brownian motion, $\mathrm{MSD}_{\boldsymbol{X}}(t) = 2Dt^{\alpha}$, white static noise has the effect of inflating the MSD at the shortest timescales, as seen in Figure 3.2(b).

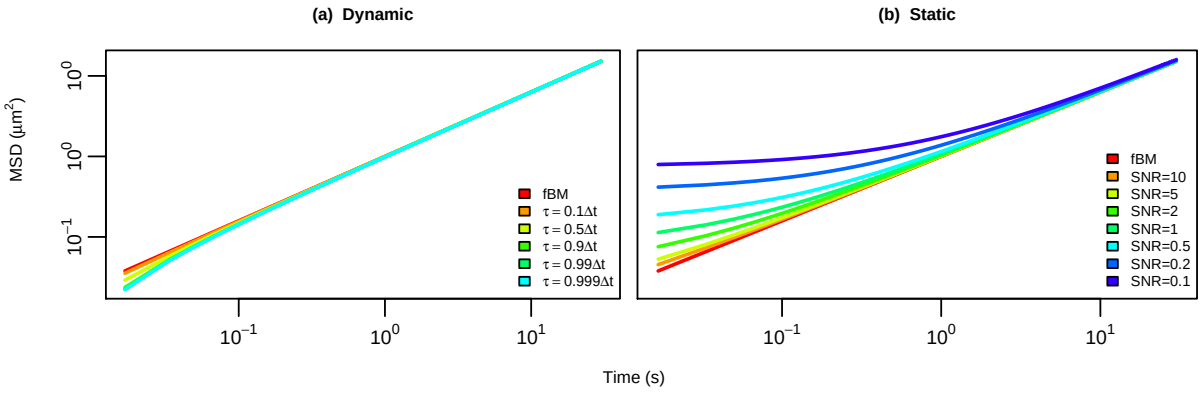


Figure 3.2: Effect of localization error on the MSD of an fBM process $X(t) = B_t^{\alpha}$ with $\alpha = 0.8$ and $\Delta t = 1/60$. (a) Dynamic error, as a function of exposure time τ. (b) Static error, as a function of the signal-to-noise ratio, $\mathrm{SNR} = \mathrm{var}(\Delta B_n^{\alpha})/\mathrm{var}(\varepsilon_n)$.

In contrast to static noise, Savin and Doyle define dynamic noise as originating from movement of the particle during the camera frame exposure time. Thus, if the camera exposure time is $\tau < \Delta t$ (as it must be less than the framerate), the recorded position of the particle at time $t = n \cdot \Delta t$ is

$$\boldsymbol{Y}_n = \frac{1}{\tau} \int_0^{\tau} \boldsymbol{X}(n \cdot \Delta t - s)\, \mathrm{d}s.$$

The dynamic-error MSD for an fBM process $X(t) = B_t^{\alpha}$ is given in Appendix C. Larger values of τ have the effect of depressing the MSD at the shortest timescales, as seen in Figure 3.2(a).

Combining static and dynamic models, the Savin-Doyle localization error model is

$$\boldsymbol{Y}_n = \frac{1}{\tau} \int_0^{\tau} \boldsymbol{X}(n \cdot \Delta t - s)\, \mathrm{d}s + \varepsilon_n. \tag{3.2.7}$$

When $\boldsymbol{X}(t) = \sum_{m=1}^{d} \boldsymbol{\beta}_m f_m(t) + \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}(t)$ follows the location-scale model (3.2.5), and the static noise has the simplified form $\boldsymbol{\Sigma}_\varepsilon = \sigma^2 \cdot \boldsymbol{\Sigma}$, parametric inference can be conducted using the computationally efficient methods of Section 3.2.2. Explicit calculations for the fBM process with $\text{MSD}_Z(t) = t^\alpha$ are given in Appendix C.

Thus, the fBM + Savin-Doyle (fSD) model has three MSD parameters: $\boldsymbol{\varphi} = (\alpha, \tau, \sigma)$. Its maximum likelihood estimates of the subdiffusion parameters (α, D) are $\hat{\alpha}$ and $\hat{D} = (1/2k) \cdot \text{tr}(\hat{\boldsymbol{\Sigma}})$. While these estimates successfully correct for many types of high-frequency measurement errors, the fSD model has two important limitations. First, Figure 3.2(a) shows that the Savin-Doyle model has little ability to correct negatively biased MSDs at the shortest timescales. Indeed, the camera aperture time τ is typically at least an order of magnitude smaller than Δt, in which case the effect of the dynamic error in Figure 3.2(a) is extremely small, and insufficient to explain larger negative MSD biases as in Figure 3.1(a). Second, the Savin-Doyle model uses one parameter (τ) to depress the MSD, and a different parameter (σ) to inflate it. This leads to an identifiability issue which adversely affects the subdiffusion estimator, as we shall see in Section 3.4. Complementing the theoretically derived Savin-Doyle approach, we present a general high-frequency noise filtering framework in the following section.

3.3 Proposed Method

In order to formulate our proposed method of filtering the localization errors in single particle tracking experiments, we begin with the following definition of high frequency noise. Let us first focus on a one-dimensional zero-drift CSI process $X(t)$ with $E[X(t)] = 0$, and let $\boldsymbol{\mathcal{X}} = \{X_n : n \geqslant 0\}$ and $\boldsymbol{\mathcal{Y}} = \{Y_n : n \geqslant 0\}$ denote the true and recorded particle position process at times $t = n \cdot \Delta t$. Then we shall say that the observation process $\boldsymbol{\mathcal{Y}}$ contains only high frequency noise if the low-frequency second-order dynamics of the true and recorded particle positions are the same, namely

$$\lim_{n\to\infty} \frac{\text{MSD}_Y(n)}{\text{MSD}_X(n)} = 1. \tag{3.3.1}$$

Given the true position process $\boldsymbol{\mathcal{X}}$, our noise model sets the observed position process to be of autoregressive/moving-average ARMA(p, q) type:

$$Y_n = \sum_{i=1}^{p} \theta_i Y_{n-i} + \sum_{j=0}^{q} \rho_j X_{n-j}, \qquad n \geqslant r = \max\{p, q\}. \tag{3.3.2}$$

For $0 \leqslant n < r$, Y_n is defined via the stationary increment process $\Delta\boldsymbol{\mathcal{X}} = \{\Delta X_n : n \in \mathbb{Z}\}$. That is, with the usual parameter restrictions

$$\min_{\{z\in\mathbb{C}:|z|\leqslant 1\}} \left|1 - \textstyle\sum_{i=1}^{p} \theta_i z^i\right| > 0, \qquad \min_{\{z\in\mathbb{C}:|z|\leqslant 1\}} \left|\rho_0 - \textstyle\sum_{j=1}^{q} \rho_j z^j\right| > 0, \tag{3.3.3}$$

[e.g. Brockwell and Davis, 1991], the increment process $\Delta\boldsymbol{\mathcal{Y}} = \{\Delta Y_n : n \in \mathbb{Z}\}$ defined by

$$\Delta Y_n = \sum_{i=1}^{p} \theta_i \Delta Y_{n-i} + \sum_{j=0}^{q} \rho_j \Delta X_{n-j} \tag{3.3.4}$$

is a well-defined stationary process which can be causally derived from $\Delta\boldsymbol{\mathcal{X}}$, and vice-versa. Moreover, setting $Y_n = \sum_{i=0}^{n-1} \Delta Y_i$ one obtains the ARMA relation (3.3.2) on the position scale for $n \geqslant r$.

One may note in model (3.3.2) that $\boldsymbol{\rho} = (\rho_0, \ldots, \rho_q)$ and $\mathrm{var}(\Delta X_n)$ cannot be identified simultaneously. This issue is typically resolved in the time-series literature by imposing the restriction $\rho_0 = 1$. However, in order for the recorded positions to adhere to a high-frequency error model as defined by (3.3.1), a different restriction must be imposed:

Theorem 1. Let $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ denote the true and recorded position processes, with the latter defined by an ARMA(p, q) representation of the former as in (3.3.4). Then $\boldsymbol{\mathcal{Y}}$ is a high-frequency error model for $\boldsymbol{\mathcal{X}}$ as defined by (3.3.1) if and only if

$$\rho_0 = 1 - \sum_{i=1}^{p} \theta_i - \sum_{j=1}^{q} \rho_j.$$

The proof is given in Appendix D.3. Indeed, the following result (also proved in Appendix D.4) shows that the family of ARMA(p, q) noise models (3.3.2) is sufficient to

describe any high-frequency noise model to arbitrary accuracy:

Theorem 2. Let $\boldsymbol{\mathcal{Y}}$ be a stochastic process of recorded positions defined as a high-frequency noise model via (3.3.1). When $\boldsymbol{\mathcal{Y}}$ satisfies specific requirement (illustrated in Appendix D.4), for any $\epsilon > 0$, we may find an ARMA(p, q) noise model $\boldsymbol{\mathcal{Y}}^\star = \{Y_n^\star : n \geqslant 0\}$ satisfying (3.3.2) such that for all $n \geqslant 0$ we have

$$\left| \frac{\mathrm{MSD}_{Y^\star}(n)}{\mathrm{MSD}_Y(n)} - 1 \right| < \epsilon.$$

3.3.1 Efficient Computations for the Location-Scale Model

Let us now consider a k-dimensional position process $\boldsymbol{X}(t) = \sum_{j=1}^d \boldsymbol{\beta}_j f_j(t) + \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}(t)$ following the location-scale model (3.2.5). Then we may construct an ARMA(p, q) high-frequency model for the measured positions as follows. Starting from the drift-free stationary increment process $\Delta\tilde{\boldsymbol{\mathcal{X}}} = \{\Delta\tilde{\boldsymbol{X}}_n = \boldsymbol{\Sigma}^{1/2}\Delta\boldsymbol{Z}_n : n \in \mathbb{Z}\}$, define the increment process $\Delta\tilde{\boldsymbol{\mathcal{Y}}} = \{\Delta\boldsymbol{Y}_n : n \in \mathbb{Z}\}$ via

$$\Delta\tilde{\boldsymbol{Y}}_n = \sum_{i=1}^p \theta_i \Delta\tilde{\boldsymbol{Y}}_{n-i} + \sum_{j=0}^q \rho_j \Delta\tilde{\boldsymbol{X}}_{n-j}. \tag{3.3.5}$$

Then under parameter restrictions (3.3.3), $\Delta\tilde{\boldsymbol{\mathcal{Y}}}$ is a well-defined stationary process with $E[\Delta\tilde{\boldsymbol{Y}}_n] = \boldsymbol{0}$. In order to add a drift to the high-frequency noise model (3.3.5), let

$$\begin{aligned}
\Delta\boldsymbol{X}_n &= \begin{cases} \Delta\tilde{\boldsymbol{X}}_n, & n < 0, \\ \Delta\tilde{\boldsymbol{X}}_n + \sum_{m=1}^d \boldsymbol{\beta}_j \Delta f_{nj}, & n \geqslant 0, \end{cases} \\
\Delta\boldsymbol{Y}_n &= \begin{cases} \Delta\tilde{\boldsymbol{Y}}_n, & n < 0 \\ \sum_{i=1}^p \theta_i \Delta\boldsymbol{Y}_{n-i} + \sum_{j=0}^q \rho_j \Delta\boldsymbol{X}_{n-j}, & n \geqslant 0, \end{cases}
\end{aligned} \tag{3.3.6}$$

where $\Delta f_{nj} = f_j((n+1)\cdot\Delta t) - f_m(n\cdot\Delta t)$. Then for $n \geqslant 0$, $\boldsymbol{X}_n = \sum_{i=0}^{n-1} \Delta\boldsymbol{X}_i$ corresponds to discrete-time observations of $\boldsymbol{X}(t)$ from the location-scale model (3.2.5), and $\boldsymbol{Y}_n = \sum_{i=0}^{n-1} \Delta\boldsymbol{Y}_i$ satisfies the ARMA(p, q) relation (3.3.2). Moreover, the observed increments

$\Delta \boldsymbol{Y} = (\Delta \boldsymbol{Y}_0, \ldots, \Delta \boldsymbol{Y}_{N-1})$ follow a matrix-normal distribution

$$\Delta \boldsymbol{Y} \sim \text{MatNorm}(\boldsymbol{F}_{\boldsymbol{\varphi}} \boldsymbol{\beta}, \boldsymbol{V}_{\boldsymbol{\varphi}}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{F}_{\boldsymbol{\varphi}}$ is an $N \times k$ matrix with elements

$$F_{nm} = -\sum_{i=1}^{\min\{n,p\}} \theta_i F_{n-i,m} + \sum_{j=0}^{\min\{n,q\}} \rho_j \Delta f_{n-j,m},$$

and $\boldsymbol{V}_{\boldsymbol{\varphi}}$ is an $N \times N$ Toeplitz matrix with element (n, m) given by $V_{\boldsymbol{\varphi}}^{(n,m)} = \text{ACF}_{\Delta Y}(|n-m|)$. Thus, we may use the computationally efficient methods of Section 3.2.2 for parameter inference, given the autocorrelation function $\text{ACF}_{\Delta Y}(n)$ defined by (3.3.4). For pure moving-average processes ($p = 0$), this function is available in closed-form given an arbitrary true increment autocorrelation function $\text{ACF}_{\Delta Z}(n)$. For $p > 0$, an accurate and computationally efficient approximation is provided in Appendix D.2.

3.3.2 The Fractional MA(1) Noise Model

Perhaps the simplest ARMA(p, q) noise model is that with $p = 0$ and $q = 1$, i.e., the first-order moving-average MA(1) model given by

$$\boldsymbol{Y}_n = (1 - \rho)\boldsymbol{X}_n + \rho \boldsymbol{X}_{n-1}, \tag{3.3.7}$$

where $\rho < \frac{1}{2}$ is required to satisfy (3.3.1). The autocorrelation of the observed increments becomes

$$\text{ACF}_{\Delta \boldsymbol{Y}}(n) = \text{ACF}_{\Delta \boldsymbol{X}}(n) + (1 - \rho)\rho\big[\,\text{ACF}_{\Delta \boldsymbol{X}}(|n-1|) + \text{ACF}_{\Delta \boldsymbol{X}}(n+1) - 2\,\text{ACF}_{\Delta \boldsymbol{X}}(n)\big],$$

where $\text{ACF}_{\Delta \boldsymbol{X}}(n)$ is the autocorrelation of the true increment process. Of particular interest is when $\boldsymbol{X}(t)$ is fractional Brownian motion, for which we refer to the corresponding MA(1) noise model as fMA. The MSD of such a model is plotted in Figure 3.3(a) for a range of values $\rho \in [-1, \frac{1}{2})$. As with the fractional Savin-Doyle (fSD) model (3.2.7) $\rho > 0$ inflates the high-frequency correlations in the observation process, whereas $\rho < 0$ depresses them.

A similar MSD plot for the fSD model is given in Figure 3.3(b). While both high-frequency noise models can similarly inflate the MSD at short timescales, the fMA model has much higher capacity to depress it.

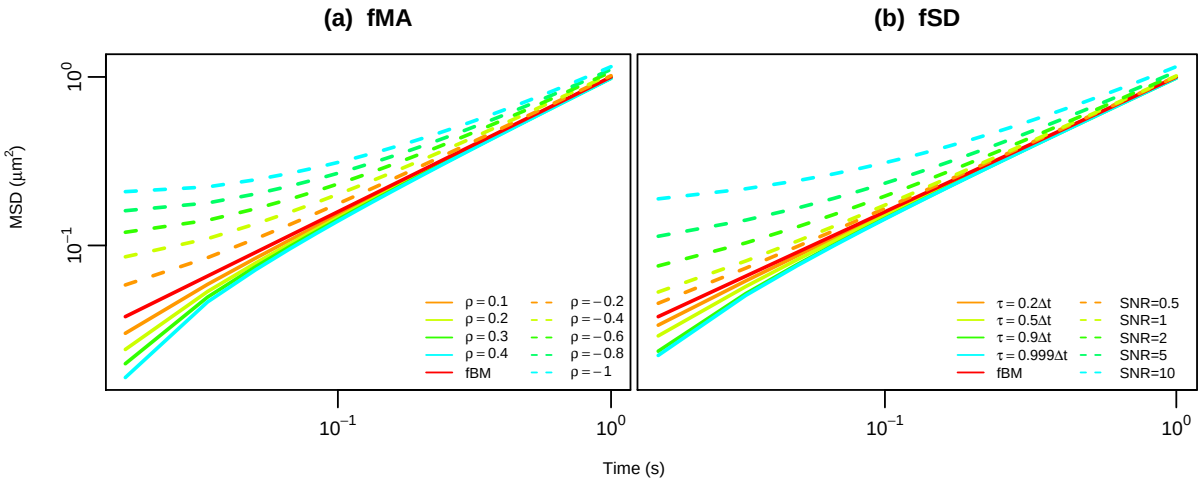


Figure 3.3: (a) MSD of the fMA model with $\alpha = 0.8$ and different values of ρ. (b) MSD of the fSD model with $\alpha = 0.8$ and different values of τ and signal-to-noise ratio $\text{SNR} = \text{var}(\Delta B^\alpha)/\sigma^2$.

In order to examine this difference more carefully, the following experiment is proposed. Suppose that observed increments $\Delta \boldsymbol{Y} = (\Delta \boldsymbol{Y}_0, \ldots, \Delta \boldsymbol{Y}_{N-1})$ are generated from a drift-free location-scale fSD model $\Delta \boldsymbol{Y} \sim p(\Delta \boldsymbol{Y} \mid \alpha, \boldsymbol{\Sigma}, \tau, \sigma)$. Then for fixed N and Δt, we may calculate the parameters of the (drift-free) fMA model $p(\Delta \boldsymbol{Y} \mid \alpha_\star, \boldsymbol{\Sigma}_\star, \rho)$ which minimize the Kullback-Liebler divergence from the true model,

$$\begin{aligned}(\hat{\alpha}_\star, \hat{\boldsymbol{\Sigma}}_\star, \hat{\rho}) &= \underset{(\alpha_\star, \boldsymbol{\Sigma}_\star, \rho)}{\arg\min}\, \text{KL}\big\{p(\Delta \boldsymbol{Y} \mid \alpha, \boldsymbol{\Sigma}, \tau, \sigma) \,\|\, p(\Delta \boldsymbol{Y} \mid \alpha_\star, \boldsymbol{\Sigma}_\star, \rho)\big\} \\ &= \underset{(\alpha_\star, \boldsymbol{\Sigma}_\star, \rho)}{\arg\min}\, \frac{1}{2} \times \left(\text{tr}(\boldsymbol{\Sigma}_\star^{-1}\boldsymbol{\Sigma})\text{tr}(\boldsymbol{V}_\star^{-1}\boldsymbol{V}) + \log\left(\frac{|\boldsymbol{\Sigma}_\star|^N |\boldsymbol{V}_\star|^k}{|\boldsymbol{\Sigma}|^N |\boldsymbol{V}|^k}\right) - Nk \right),\end{aligned}$$

where $\boldsymbol{V}$ and $\boldsymbol{V}_\star$ are $N \times N$ Toeplitz variance matrices with first row given by the auto-correlation function of the fSD and fMA models, respectively. Figure 3.4(a) displays the

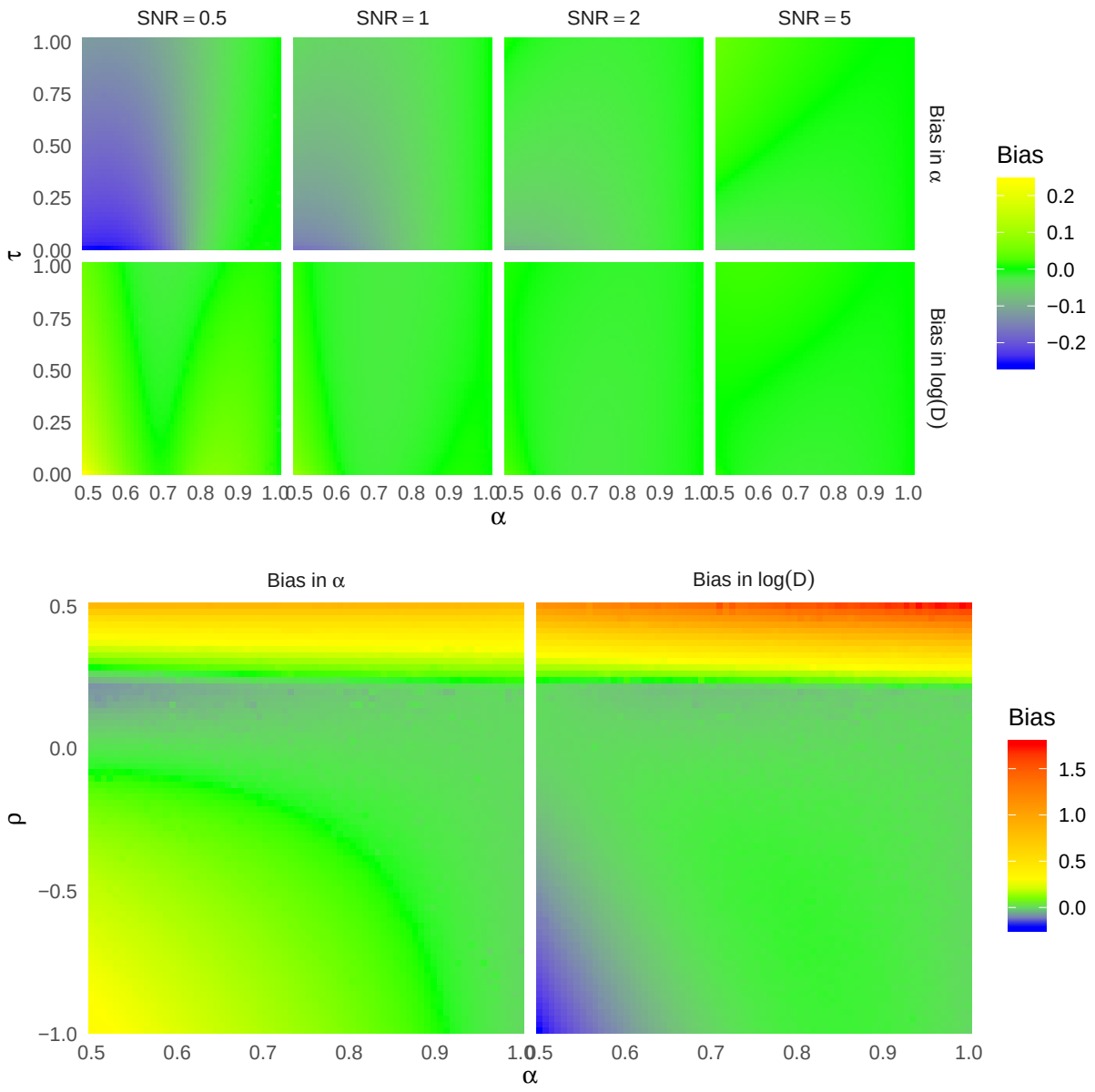


Figure 3.4: Model misspecification bias in α and D. (a) Best-fitting fMA model to true fSD models with different values of α, τ, and signal-to-noise ratio SNR $= \mathrm{var}(\Delta B_n^\alpha)/\sigma^2$. (b) Best-fitting fSD model to true fMA models with different values of α and ρ.

difference between true and best-fitting subdiffusion parameters $\hat{\alpha}_\star - \alpha$ and $\log \hat{D}_\star - \log D$, for $k = 2$, $\mathbf{\Sigma} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $N = 1800$, $\Delta t = 1/60$, and over a range of parameter values (α, τ, σ). Figure 3.4(b) does the same, but with the best-fitting fSD model to data generated from fMA. For all but very high static error σ (corresponding to low signal-to-noise ratio SNR $= \text{var}(\Delta X_n)/\sigma^2$), the fMA model can recover the true subdiffusion parameters (α, D) with little bias due to model misspecification. There is obviously more bias when fSD is used on data generated from fMA, particularly when $\rho > 0$ as suggested by Figure 3.3.

3.4 Simulation Study

In this section, we evaluate the performance of the proposed ARMA(p, q) high-frequency noise filters in various simulation settings. In each setting, we simulate $B = 500$ observed data trajectories $\boldsymbol{Y}^{(b)} = (\boldsymbol{Y}_0^{(b)}, \ldots, \boldsymbol{Y}_N^{(b)})$, $b = 1, \ldots, B$, each consisting of $N = 1800$ two-dimensional observations ($k = 2$) recorded at intervals of $\Delta t = 1/60\,\text{s}$.

3.4.1 Empirical Localization Error

Consider the following simulation setting designed to reflect the localization errors in our own experimental setup. Let $\boldsymbol{Y}_\text{v}$ denote the trajectory measurements for a particle undergoing ordinary diffusion in a viscous environment. Then we may estimate the MSD ratio

$$g(n) = \frac{\text{MSD}_{\tilde{\boldsymbol{Y}}_\text{v}}(n)}{\text{MSD}_{\boldsymbol{X}_\text{v}}(n)}, \tag{3.4.1}$$

where the MSD of the true position process is $\text{MSD}_{\boldsymbol{X}_\text{v}}(n) = 2Dt$ with D determined by the Stokes-Einstein relation (3.1.2), and the MSD of the drift-subtracted observation process $\tilde{\boldsymbol{Y}}_\text{v}$ can be accurately estimated by

$$\widehat{\text{MSD}}_{\tilde{\boldsymbol{Y}}_\text{v}}(n) = \frac{1}{M} \sum_{i=1}^{M} \widehat{\text{MSD}}_{\tilde{\boldsymbol{Y}}_\text{v}^{(i)}}(n),$$

where $\widehat{\mathrm{MSD}}_{\tilde{\boldsymbol{Y}}_{\mathrm{v}}^{(i)}}(n)$ is the empirical MSD (3.2.2) for each (drift-subtracted) particle trajectory $\tilde{\boldsymbol{Y}}_{\mathrm{v}}^{(1)}, \ldots, \tilde{\boldsymbol{Y}}_{\mathrm{v}}^{(M)}$ recorded in a given experiment (e.g., Figure 3.1(a)). We then suppose that the true trajectory is a drift-free fBM $\boldsymbol{X}(t) = \boldsymbol{\Sigma}^{1/2}\boldsymbol{B}^{\alpha}(t)$, and simulate the measured trajectories from

$$\boldsymbol{Y}^{(b)} \overset{\text{iid}}{\sim} \mathrm{MatNorm}\left(\boldsymbol{0}, \boldsymbol{V}, \boldsymbol{\Sigma}\right),$$

where $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ and the $(N+1) \times (N+1)$ variance matrix $\boldsymbol{V}$ is that of a CSI process with MSD given by

$$\mathrm{MSD}_{\boldsymbol{Y}}(n) = (\gamma \hat{g}(n) - \gamma + 1) \times \mathrm{MSD}_{\boldsymbol{X}}(n), \tag{3.4.2}$$

where $\hat{g}(n)$ is the estimated noise ratio (3.4.1) from a viscous experiment, and the noise factor $\gamma > 0$ can be used to suppress or amplify the empirical localization error with $\gamma < 1$ or $\gamma > 1$, respectively. Having constrained our estimator such that $\hat{g}(n) = 1$ for $n > N_0$, (3.4.2) is a high-frequency noise model as defined by (3.3.1). Figure 3.5 displays the observed MSD (3.4.2) for a true fBM trajectory with $\alpha = 0.6$, contaminated by empirical localization errors from two representative viscous experiments described in Table 3.3, illustrating the effects of high-frequency MSD suppression and amplification, respectively.

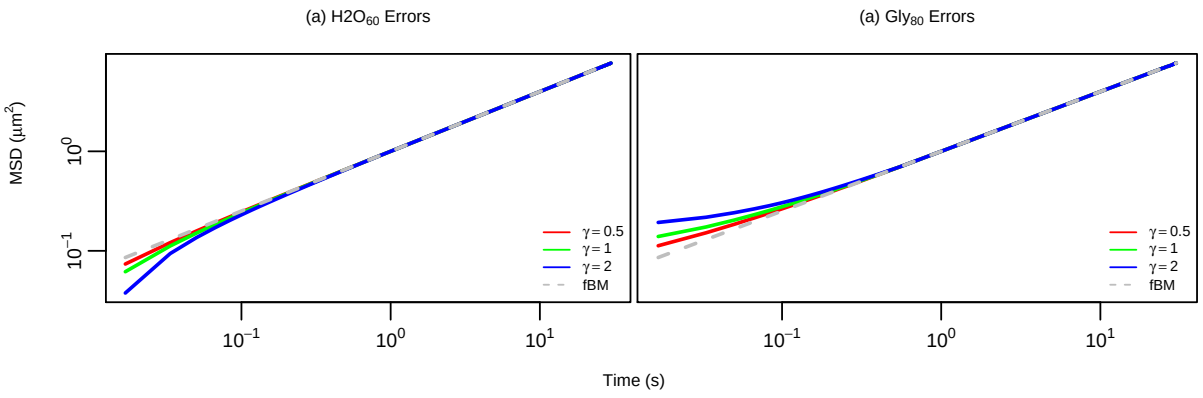


Figure 3.5: MSD of simulated observations with empirical localization error (3.4.2), where the true trajectory is an fBM process with $\alpha = 0.6$. (a) High-frequency MSD suppression as observed in $\mathrm{H2O}_{60}$ experiment (see Table 3.3). (b) High-frequency MSD amplification as observed in GLY_{60} experiment.

The following methods are used to estimate the subdiffusion parameters (α, D) for each set of simulated particle observations $\boldsymbol{Y}^{(b)}$, $b = 1, \dots B$:

1. **LS:** The semiparameteric least-squares estimator (3.2.4) applied to the drift-subtracted empirical MSD (3.2.2).

2. **fBM:** The MLE of an fBM-driven location-scale model with linear drift,

$$\boldsymbol{X}(t) = \boldsymbol{\mu} t + \boldsymbol{\Sigma}^{1/2} \boldsymbol{B}^{\alpha}(t), \tag{3.4.3}$$

   for which the model parameters are $(\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

3. **fSD:** The MLE of the Savin-Doyle error model (3.2.7) applied to (3.4.3), for which the model parameters are $(\alpha, \tau, \sigma, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

4. **fMA:** The MLE of the proposed MA(1) high-frequency noise filter (3.3.7) applied to (3.4.3), for which the model parameters are $(\alpha, \rho, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

5. **fMA2:** The MLE of the proposed MA(2) high-frequency noise filter

$$\boldsymbol{Y}_n = (1 - \rho_1 - \rho_2)\boldsymbol{X}_n + \rho_1 \boldsymbol{X}_{n-1} + \rho_2 \boldsymbol{X}_{n-2}$$

   applied to (3.4.3), for which the model parameters are $(\alpha, \rho_1, \rho_2, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

6. **fARMA:** The MLE of the proposed ARMA$(1,1)$ high-frequency noise filter

$$\boldsymbol{Y}_n = \theta \boldsymbol{Y}_{n-1} + (1 - \theta - \rho)\boldsymbol{X}_n + \rho \boldsymbol{X}_{n-1}$$

   applied to (3.4.3), for which the model parameters are $(\alpha, \theta, \rho, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Remark 1. The fSD exposure time parameter τ is typically known and therefore need not be estimated from the data. However, we have opted here to estimate it regardless, as this gives far greater ability to account for high-frequency MSD suppression (e.g., Figure 3.2(a)). We return to this point in Section 3.5.

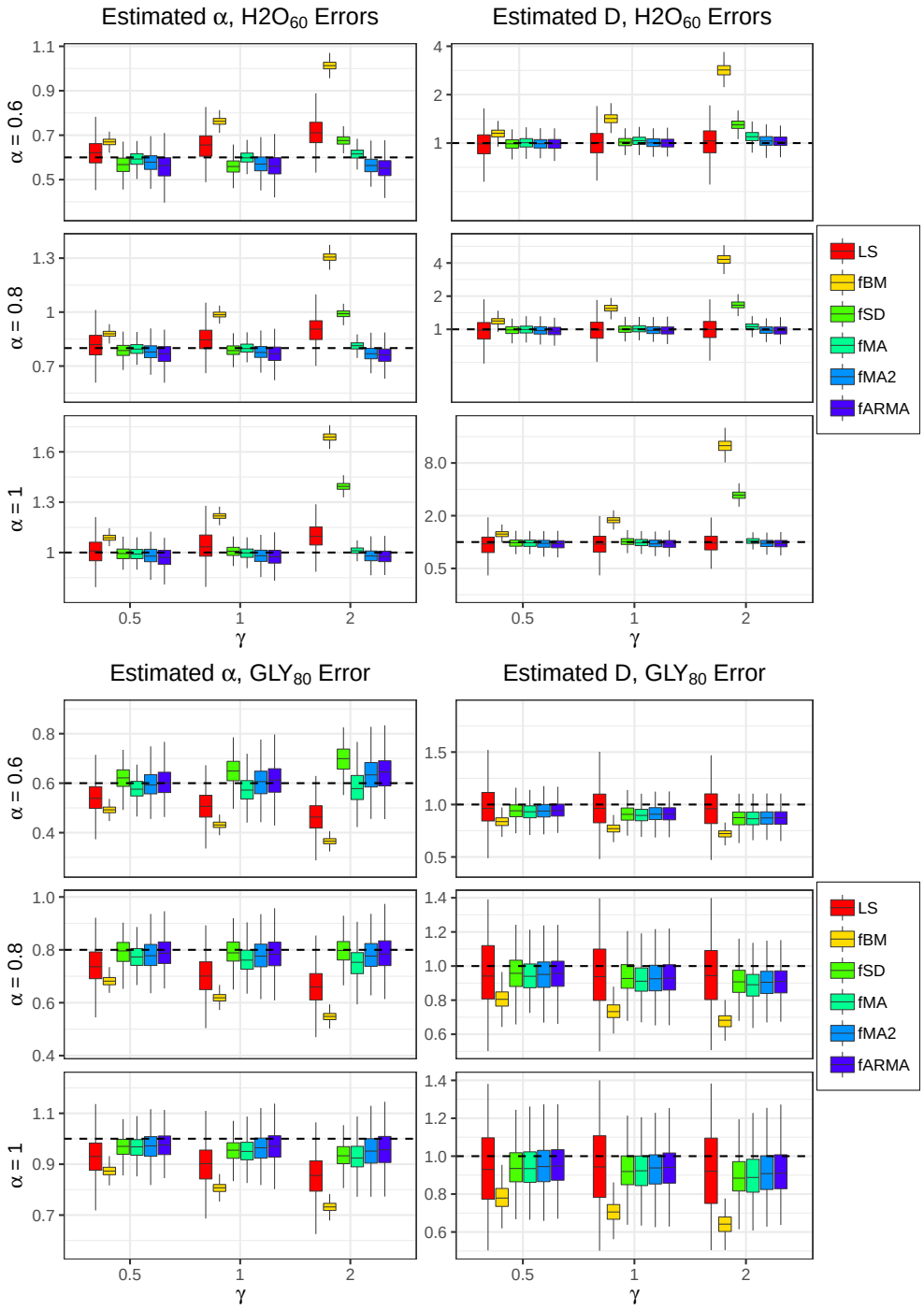


Figure 3.6: Estimates of (α, D) for true fBM trajectories with various types and degrees of empirical localization errors.

Table 3.1: Actual coverage by 95% confidence intervals with various types and degrees of empirical localization errors.

$P_{95}(\alpha)$		$H2O_{60}$ Errors			GLY_{80} Errors		
		$\gamma = 0.5$	$\gamma = 1$	$\gamma = 2$	$\gamma = 0.5$	$\gamma = 1$	$\gamma = 2$
$\alpha = 0.6$	fBM	5	0	0	0	0	0
	fSD	90	87	11	93	84	59
	fMA	96	96	90	91	88	88
	fMA2	91	91	84	94	95	94
	fARMA	92	93	87	89	93	93
$\alpha = 0.8$	fBM	4	0	0	0	0	0
	fSD	91	93	0	92	94	94
	fMA	93	94	93	87	84	81
	fMA2	93	91	87	92	91	93
	fARMA	92	91	88	89	90	93
$\alpha = 1$	fBM	1	0	0	0	0	0
	fSD	13	6	0	23	34	36
	fMA	95	94	93	87	81	70
	fMA2	92	92	94	90	88	84
	fARMA	91	92	92	87	86	85
$P_{95}(\log D)$		$H2O_{60}$ Errors			GLY_{80} Errors		
		$\gamma = 0.5$	$\gamma = 1$	$\gamma = 2$	$\gamma = 0.5$	$\gamma = 1$	$\gamma = 2$
$\alpha = 0.6$	fBM	57	1	0	20	1	0
	fSD	94	96	10	88	80	72
	fMA	96	95	88	86	73	85
	fMA2	94	95	95	86	79	66
	fARMA	94	95	95	87	79	65
$\alpha = 0.8$	fBM	48	0	0	18	2	0
	fSD	92	94	1	90	89	82
	fMA	95	94	94	89	82	76
	fMA2	93	94	94	89	86	83
	fARMA	91	93	93	89	88	84
$\alpha = 1$	fBM	42	0	0	16	1	0
	fSD	63	61	0	69	74	67
	fMA	95	94	95	90	88	80
	fMA2	92	92	94	91	90	85
	fARMA	90	91	93	91	89	85

The point estimates for (α, D) for true fBM trajectories with $\alpha \in \{.6, .8, 1\}$ and empirical error factor $\gamma \in \{.5, 1, 2\}$ are displayed in Figure 3.6. As expected, the semiparametric LS estimator is substantially more variable than any of the fully parametric estimators, and the error-unadjusted fBM estimator incurs considerable bias, even with the smallest noise factor $\gamma = 0.5$. The high-frequency estimators (fMA, fMA2, and fARMA) are fairly similar to each other, with the additional parameters of fMA2 and fARMA giving them slightly lower bias and higher variance. The high-frequency estimators are slightly more biased than fSD in the GLY_{80} simulation with $\alpha = 0.8$. In contrast, they are somewhat less biased than fSD for GLY_{80} with the stronger subdiffusive signal $\alpha = 0.6$, and considerably less so for H2O_{60} with the largest noise factor $\gamma = 2$.

Table 3.1 displays the true coverage of the 95% confidence intervals for each parametric estimator, calculated as

$$\text{P}_{95}(\psi) = \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\{\theta \in \hat{\psi}_b \pm 1.96\,\text{se}(\hat{\psi}_b)\},$$

where $\psi \in \{\alpha, \log D\}$, $\hat{\psi}_b$ is the MLE for dataset b, and $\text{se}(\hat{\psi}_b)$ is the square root of the corresponding diagonal element of the variance estimator $\widehat{\text{var}}(\hat{\boldsymbol{\theta}}_b) = -\left[\frac{\partial^2 \ell(\boldsymbol{Y}^{(b)}|\hat{\boldsymbol{\theta}}_b)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right]^{-1}$, where $\hat{\boldsymbol{\theta}}_b$ is the MLE of all model parameters. The true coverage of the fMA, fMA2, and fARMA confidence intervals is close to 95% when the bias is negligible and typically above 85%. This is also true for fSD, with the notable exception of either empirical error model and true $\alpha = 1$. Upon closer inspection, we found that the fSD model suffers from an identifiability issue in the diffusive (viscous) regime, wherein the MSD suppression by τ and amplification by σ achieve the same net effect over a range of values. This does not affect the estimate of (α, D), but significantly decreases the curvature of $\ell(\boldsymbol{Y} \mid \hat{\boldsymbol{\theta}})$, thus artifically inflating the observed Fisher information $\widehat{\text{var}}(\hat{\boldsymbol{\theta}}_b)^{-1}$.

Remark 2. Since the subdiffusion equation $\text{MSD}_{\boldsymbol{X}}(t) = 2Dt^\alpha$ dictates that D be measured in units of $\mu\text{m}^{(2)}\,\text{s}^{-\alpha}$, in order to compare estimates of D for different values of α as in Figure 3.6, we follow the convention of interpreting D as half the MSD at time $t = 1\,\text{s}$ [e.g., Lai et al., 2007, Wang et al., 2008], which for any α is measured uniformly in units of $\mu\text{m}^{(2)}$.

3.4.2 Modeling Transient Subdiffusion

In this section, we show how the proposed high-frequency filter can be used not only for measurement error correction, but also to estimate subdiffusion in models where the power-law relation $\text{MSD}_{\boldsymbol{X}}(t) \sim t^{\alpha}$ holds only for $t > t_{\min}$. For this purpose, here we shall generate particle trajectories from a so-called *Generalized Langevin Equation* (GLE), a physical model derived from the fundamental laws of thermodynamics for interacting-particle systems [e.g., Kubo, 1966, Zwanzig, 2001, Kou, 2008]. For a one-dimensional particle with negligible mass, the GLE for its trajectory $X(t)$ is a stochastic integro-differential equation of the form

$$\int_{-\infty}^{t} \phi(t-s)V(s)\,\mathrm{d}s = F(t), \tag{3.4.4}$$

where $V(t) = \frac{\mathrm{d}}{\mathrm{d}t}X(t)$ is the particle velocity, $\phi(t)$ is a memory kernel, and $F(t)$ is a stationary mean-zero Gaussian force process with $\text{ACF}_F(t) = k_B T \cdot \phi(t)$, where T is temperature and k_B is Boltzmann's constant. The memory of the process is modeled as a generalized Rouse kernel [McKinley et al., 2009]:

$$\phi(t) = \frac{\nu}{K}\sum_{k=1}^{K} \exp(-|t|/\tau_k), \quad \tau_k = \tau \cdot (K/k)^{\gamma}. \tag{3.4.5}$$

The sum-of-exponentials form of (3.4.5) is a longstanding linear model for viscoelastic relaxation [e.g., Soussou et al., 1970, Ferry, 1980, Mason and Weitz, 1995], whereas the specific parametrization of the relaxation modes τ_k has been shown for sufficiently large K to exhibit *transient* subdiffusion [McKinley et al., 2009],

$$\text{MSD}_X(t) = \begin{cases} 2D_{\text{eff}} \cdot t^{\alpha_{\text{eff}}} & t_{\min} < t < t_{\max} \\ C_{\min} \cdot t & t < t_{\min} \\ C_{\max} \cdot t & t > t_{\max}, \end{cases} \tag{3.4.6}$$

where the subdiffusive range parameters $(t_{\min}, t_{\max})$ and the effective subdiffusion parameters $(\alpha_{\text{eff}}, D_{\text{eff}})$ are implicit functions of K, γ, τ, and ν. Details of the parameter conversions

and the exact form of (3.4.6) are provided in Appendix E.

Figure 3.7 displays the MSD of various GLE processes with fixed $K = 300$, and $\{\gamma, \tau, \nu\}$ tuned to have $\alpha_{\text{eff}} = 0.63$, $D_{\text{eff}} = 0.58$, and values of $t_{\min}/\Delta t = \{5, 10, 20, 50, 100\}$. In all cases the value of $t_{\max}$ was several times larger than the experimental timeframe $N\Delta t = 30\,\text{s}$, such that the observable MSD could potentially be matched by the fBM-driven high-frequency models of Section 3.3. The trajectories for this experiment were simulated from

$$\boldsymbol{Y}^{(b)} \overset{\text{iid}}{\sim} \text{MatNorm}(\boldsymbol{0}, \boldsymbol{V}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ and $\boldsymbol{V}$ is the $(N+1) \times (N+1)$ variance matrix of the GLE process (3.4.4) with MSDs displayed in Figure 3.7.

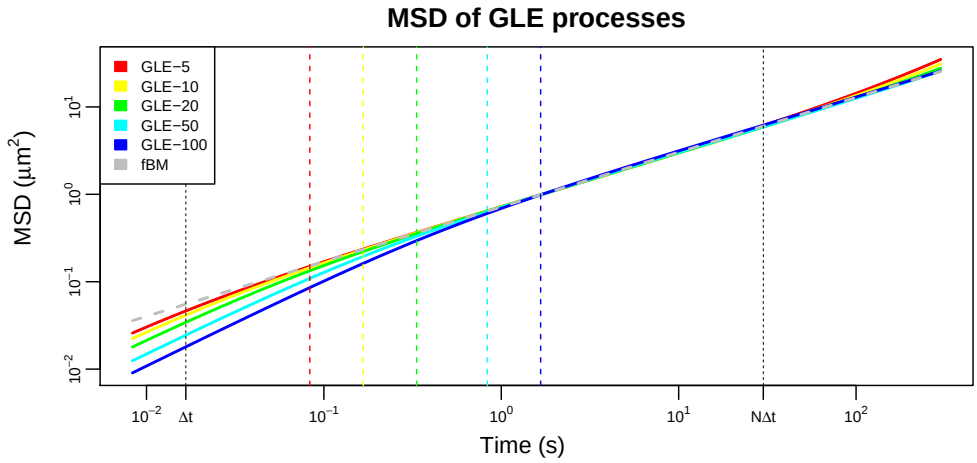


Figure 3.7: MSD of GLE processes with $\alpha_{\text{eff}} = 0.63$, $D_{\text{eff}} = 0.58$, and $t_{\min}/\Delta t = \{5, 10, 20, 50, 100\}$. The horizontal dashed lines indicated $t_{\min}$, and the diagonal dashed line corresponds to an fBM process with the same subdiffusive parameters $(\alpha_{\text{eff}}, D_{\text{eff}})$. The dotted vertical lines indicate the beginning and end of experiment, at $\Delta t = 1/60\,\text{s}$ and $N\Delta t = 30\,\text{s}$, respectively.

Figure 3.8 displays the parameter estimates of α_{eff} and D_{eff} for the six estimators described in Section 3.4.1, and Table 3.2 displays the true coverage probabilities of the corresponding 95% confidence intervals. As in Figure 3.6, the LS estimator has the highest

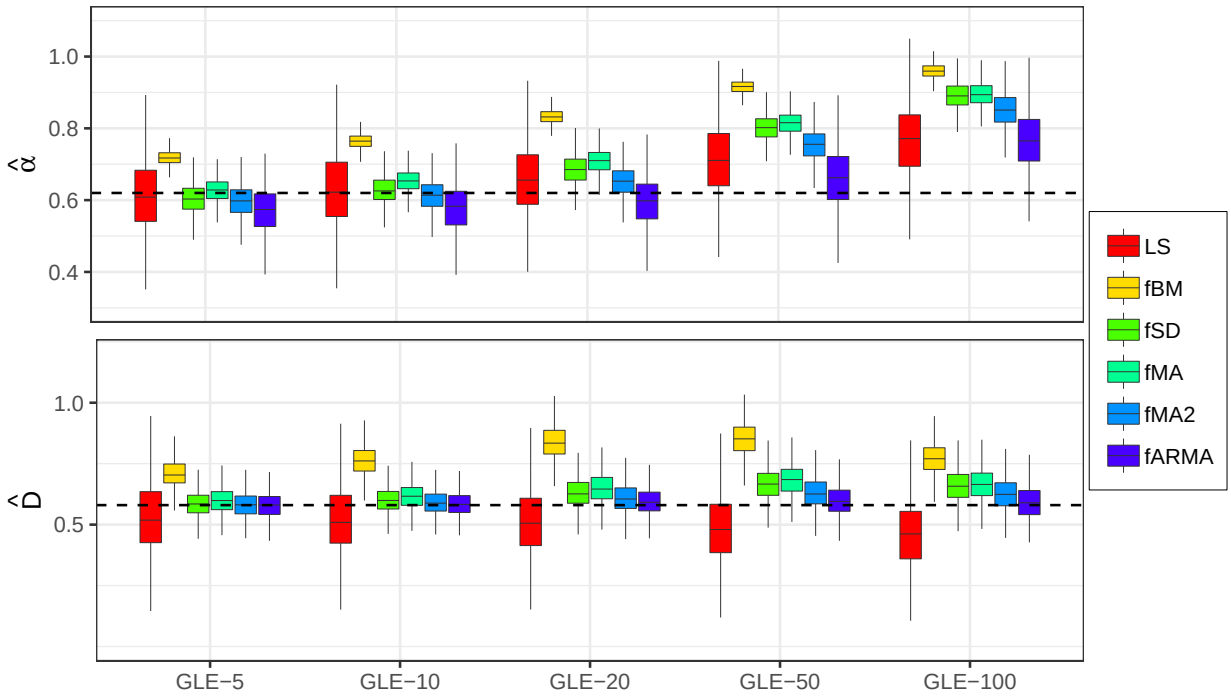


Figure 3.8: Estimates of α_{eff} and D_{eff} for simulated GLE trajectories with true parameters $\alpha_{\text{eff}} = 0.63$, $D_{\text{eff}} = 0.58$, $K = 300$, and $t_{\min}/\Delta t = \{5, 10, 20, 50, 100\}$.

Table 3.2: Actual coverage by 95% confidence intervals with different GLE processes.

$P_{95}(\alpha)$	GLE-5	GLE-10	GLE-20	GLE-50	GLE-100
fBM	0	0	0	0	0
fSD	96	96	64	0	0
fMA	95	84	25	0	0
fMA2	92	95	89	15	0
fARMA	92	92	95	85	53
$P_{95}(\log D)$	GLE-5	GLE-10	GLE-20	GLE-50	GLE-100
fBM	31	8	1	1	11
fSD	94	95	87	78	74
fMA	93	92	78	68	81
fMA2	94	95	93	93	92
fARMA	93	94	93	95	91

variance and fBM the largest bias. In this case however the fSD and fMA parameteric estimators exhibit considerable bias in estimating α, especially when $t_{\min} \gg \Delta t$. In contrast, the fARMA estimator displays good accuracy and reasonable coverage even when $t_{\min}$ is $50\times$ the interobservation time Δt.

3.5 Analysis of Experimental Data

We now investigate the performance of our high-frequency filters on a variety of real single-particle tracking experiments described in Table 3.3. For each experiment, Table 3.3 reports the interobservation time Δt, the number of observations per trajectory N, and the type of camera and particle tracking software. All tracked particles are inert polystyrene beads of diameter $d = 1\,\mu\text{m}$.

3.5.1 Viscous Fluids

The first six experiments are conducted in viscous fluids (water and glycerol), for which $\alpha = 1$ and the diffusivity constant D is derived from the Stokes-Einstein relation (3.1.2). For the six estimators described in Section 3.4.1, estimates of (α, D) and true coverage probabilities of the associated 95% confidence intervals are displayed in Figure 3.9 and Table 3.4, respectively. Both the fSD and proposed high-frequency estimators remove most of the bias of fBM without camera error correction. However, the fSD 95% confidence intervals suffer from severe under-coverage, due to the parameter identifiability issue noted in Section 3.4.1. Indeed, Table 3.5 shows that $\hat{\tau}$ is significantly larger than its true value τ, which is necessary in the H2O experiments to capture high-frequency MSD suppression. When τ is fixed at its true value, fSD estimation results are obviously biased, as in Figure 3.1.

3.5.2 Viscoelastic Fluids

The remaining 12 experiments from Table 3.3 are conducted in two kinds of viscoelastic media. The first consists of mucus harvested from primary human bronchial epithelial (HBE)

Table 3.3: Summary of experimental conditions for various single-particle tracking experiments. The different types of fluids are water (H2O), glycerol (GLY), mucus from human bronchial ephithelia cell cultures (HBE), and polyethilene oxide (PEO). The subscripts correspond to sampling frequency for H2O, percent concentration for GLY, and percent weight (wt%) for HBE and PEO. The two types of cameras are Flea3 USB 3.0 [Flea3: FLIR, 2019] and Panoptes [Pan: CISMM, 2019a]. The particle tracking software employed is either Video Spot Tracker [VS: CISMM, 2019b] or Net Tracker [Net: Newby et al., 2018].

Medium	Name	D	Δt (s)	N	Camera	Software
Viscous ($\alpha = 1$)	$H2O_{15}$	0.43	1/15	1800	Flea3	Net
	$H2O_{30}$	0.43	1/30	1800	Flea3	Net
	$H2O_{60}$	0.43	1/60	1800	Flea3	Net
	$H2O_{60b}$	0.43	1/60	1800	Flea3	VS
	GLY_{60}	0.09	1/60	1800	Flea3	VS
	GLY_{80}	0.022	1/60	1800	Flea3	VS
Viscoelastic (α unknown)	$HBE_{1.5}$	-	1/60	1800	Flea3	VS
	HBE_{2}	-	1/60	1800	Flea3	VS
	$HBE_{2.5}$	-	1/60	1800	Flea3	VS
	HBE_{3}	-	1/60	1800	Flea3	VS
	HBE_{4}	-	1/60	1800	Flea3	VS
	HBE_{5}	-	1/60	1800	Flea3	VS
	$PEO_{0.22}$	-	1/38.17	1145	Pan	VS
	$PEO_{0.45}$	-	1/38.17	1145	Pan	VS
	$PEO_{0.6}$	-	1/38.17	1145	Pan	VS
	$PEO_{0.75}$	-	1/38.17	1145	Pan	VS
	$PEO_{0.9}$	-	1/38.17	1145	Pan	VS
	$PEO_{1.22}$	-	1/38.17	1145	Pan	VS

Table 3.4: Actual coverage by 95% confidence intervals in viscous fluid study.

	$H2O_{15}$	$H2O_{30}$	$H2O_{60}$	$H2O_{60b}$	GLY_{60}	GLY_{80}
fBM	0	0	0	0	4	16
fSD	47	42	47	11	14	44
fMA	94	90	93	85	90	71
fMA2	95	91	92	87	91	75
fARMA	95	92	94	88	92	82

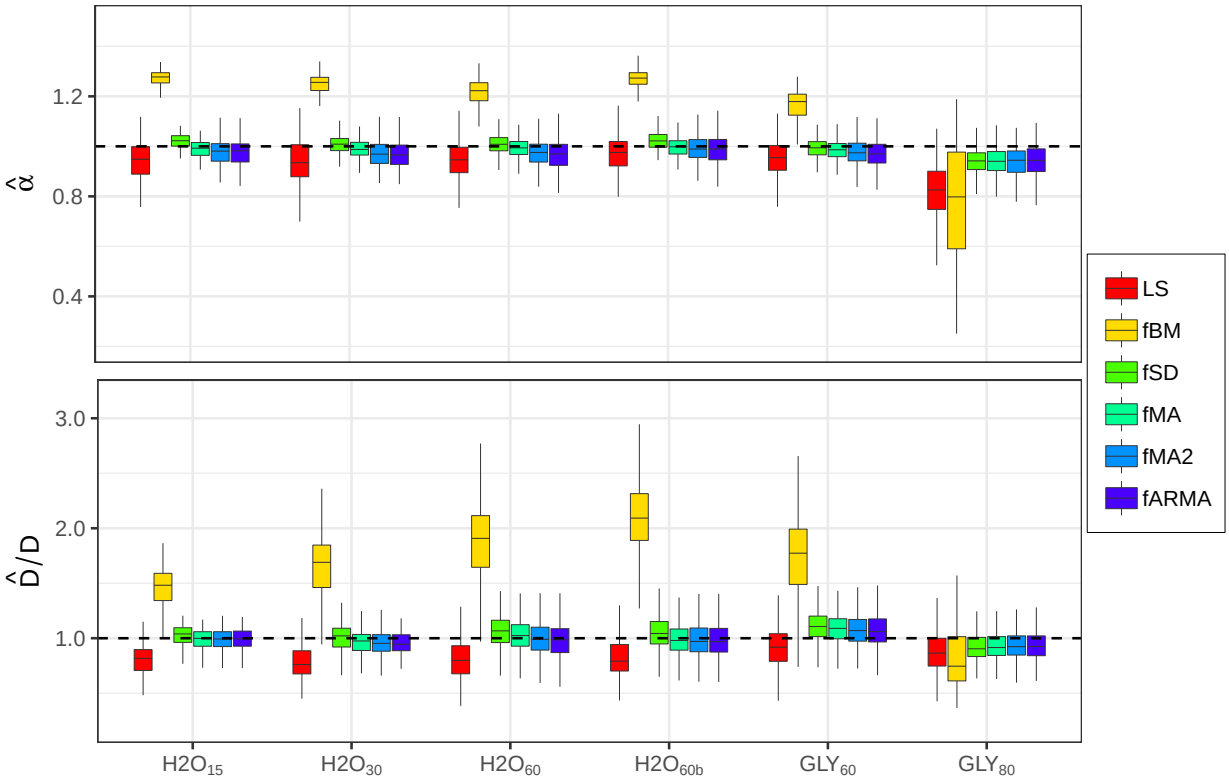


Figure 3.9: Estimated subdiffusive parameters (α, D), viscous fluids study.

Table 3.5: Ratio of true and estimated exposure time to interobservation time for fSD model.

	$H2O_{15}$	$H2O_{30}$	$H2O_{60}$	$H2O_{60b}$	GLY_{60}	GLY_{80}
True $\tau/\Delta t$	0.3	0.3	0.3	0.3	0.3	0.3
Estimated $\hat{\tau}/\Delta t$	0.93	0.91	0.89	0.91	0.85	0.54

cell cultures [Hill et al., 2014]. Washings from cultures were pooled and concentrated to desired weight percent solids (wt%). Higher concentrations of solids in lung mucus have been associated with disease states, so an accurate recovery of biophysical properties is critical in samples with volumes too small to directly measure wt% [Hill et al., 2014]. The second medium, polyethylene oxide (PEO), is a synthetic polyether compound with applications in diverse fields ranging from biomedicine to industrial manufacturing [Working et al., 1997]. The present data consists of trajectories in 5 megadalton (MDa) PEO at a range of wt% values. In all 12 viscoelastic experiments, subdiffusive motion $\alpha < 1$ is expected, but the true values of (α, D) are unknown.

Figure 3.10 displays the (α, D) estimates of the six subdiffusion estimators for the viscoelastic experiments. For viscoelastic fluids, the movement of particles is not predictable and the true value of subdiffusive parameters (α, D) remains unknown. In order to identify the most reliable model in extracting (α, D) for PEO and HBE experiments, we relate the accuracy in estimating subdiffusive parameters with the overall deviation of estimated MSDs from observed MSDs. For two particle trajectories $\boldsymbol{X}(t)$ and $\boldsymbol{Y}(t)$, we can quantify the "gap" between their MSDs using the following measurement:

$$d(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{T} \int_{t=0}^{T} |\log \frac{\text{MSD}_{\boldsymbol{X}}(t)}{\text{MSD}_{\boldsymbol{Y}}(t)}| \, \mathrm{d}t$$

As is demonstrated in Figure 3.11, $d(\boldsymbol{X}, \boldsymbol{Y})$ can be interpreted as the area of the gap between $\text{MSD}_{\boldsymbol{X}}$ and $\text{MSD}_{\boldsymbol{Y}}$ during $[0, T]$ in logarithmic scale.

To estimate the deviation between the MSD of experimental trajectories $\boldsymbol{X}_i$ and its estimated MSD using model M_j, we have

$$\hat{d}^{(i)}_{M_j} = \frac{1}{T} \sum_{n=1}^{T} |\log \frac{\widehat{\text{MSD}}_{\boldsymbol{X}_i}(n \cdot \Delta t)}{\widehat{\text{MSD}}^{(M_j)}_{\boldsymbol{X}_i}(n \cdot \Delta t)}|$$

where $\widehat{\text{MSD}}_{\boldsymbol{X}_i}$ is computed using Equation (3.1.1) and $\widehat{\text{MSD}}^{(M_j)}_{\boldsymbol{X}_i}$ is computed using Equation (D.1.1), $M_j \in \{\textbf{fBM}, \textbf{fSD}, \textbf{fMA}, \textbf{fMA2}, \textbf{fARMA}\}$. Because of the increasing bias in non-parametric estimated $\widehat{\text{MSD}}_{\boldsymbol{X}_i}(n \cdot \Delta t)$ as n increases [Mellnik et al., 2016], length of range T is selected to be 40% of total trajectory length N.

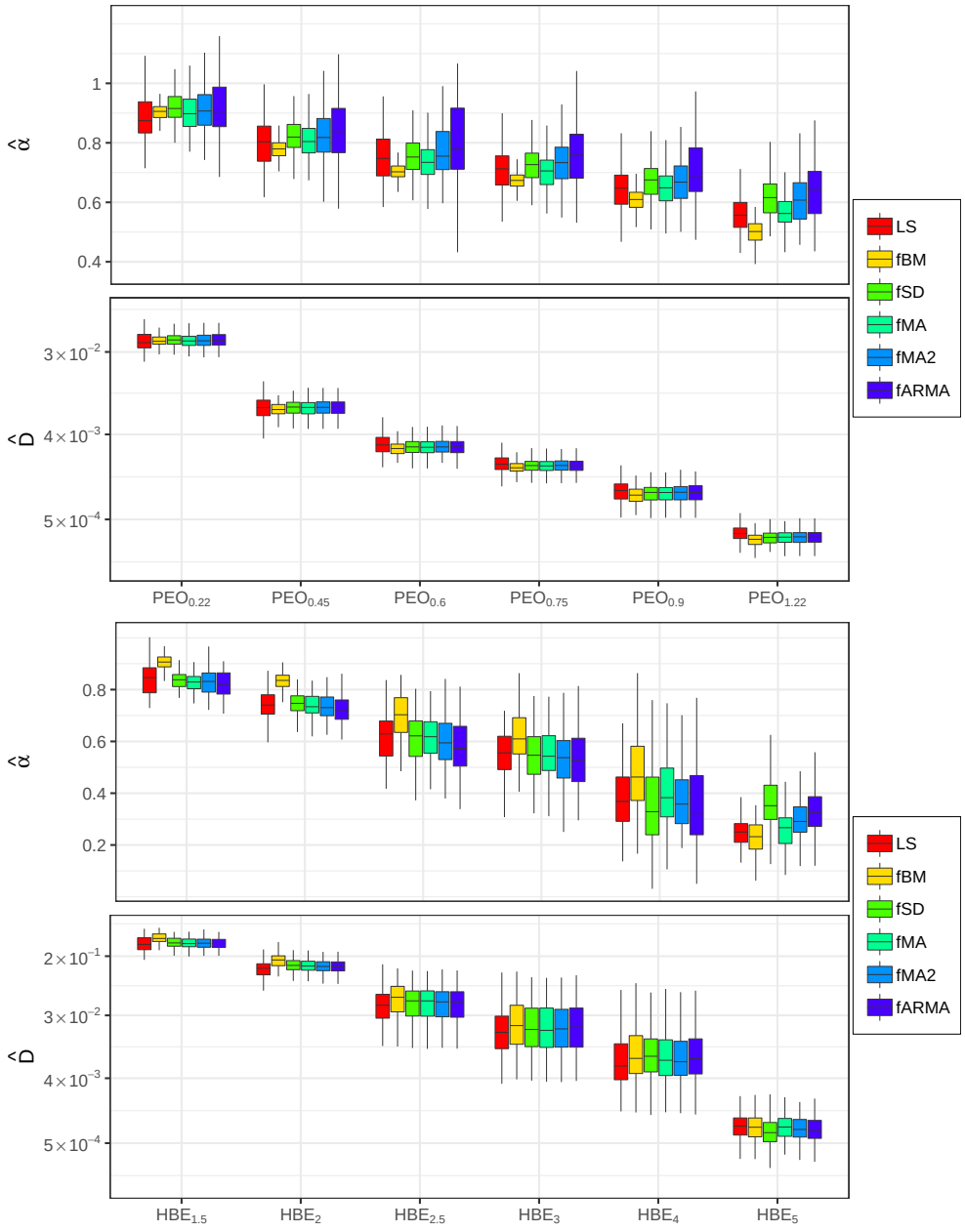


Figure 3.10: Estimated subdiffusive parameters (α, D), viscoelastic fluids study.

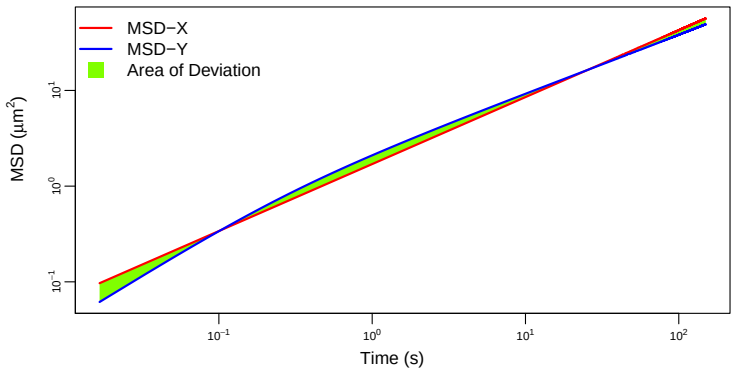


Figure 3.11: The area of the gap between two MSDs.

Table 3.6 demonstrates the average deviation $\hat{d}_{M_j} = \frac{1}{M}\sum_{i=1}^{M} \hat{d}_{M_j}^{(i)}$ for different methods, and shows that model fARMA provides best fitting for almost all experiments.

After having a closer look at the viscoelastic data, we find that the recorded trajectories are always multiples of a specific constant $c = 0.00703125$, which suggests that recorded coordinates of particles $\boldsymbol{Y}_i, i = 0, 1, \ldots, N$ are results of grid-fitting of true particle positions $\boldsymbol{X}_i$: $\boldsymbol{Y}_i = \lfloor \frac{\boldsymbol{X}_i}{c} \rceil \cdot c$, and suffer from the following round-off error

$$\boldsymbol{Y}_i = \boldsymbol{X}_i + \varepsilon_i, \qquad \varepsilon_i \overset{\text{iid}}{\sim} \text{Unif}(-c/2, c/2)$$

which is within the framework of the static error (3.2.6). Since the size of static error $\sigma^2 \approx \frac{c^2}{12}$ is constant across experiments while the variance of trajectory increments $\text{var}(\Delta \boldsymbol{X})$ (proportional to D in Figure 3.10) decreases as concentration increases, the overall signal-to-noise ratio $\text{SNR} = \frac{\text{var}(\Delta \boldsymbol{X})}{\sigma^2}$ is negatively-correlated with the medium concentration. Taking this into consideration, the fact that fSD outperforms fMA when concentration grows in Table 3.6 is consistent with the cost of model misspecification of fMA model demonstrated in Figure 3.4(a).

Generally speaking, fARMA has the best performance with respect to heavy experimental noises, but the "sensitivity" in AR parameter estimation sometimes translates to the increased variability in estimates of (α, D). On the contrary, fMA model is fastest (with minimal number of parameters) and has smallest variance (the least spread of whiskers in

Table 3.6: Measured deviation in the estimated MSD, viscoelastic fluid study. For each experiment, the estimator with the smallest deviation is highlighted in bold.

	$HBE_{1.5}$	HBE_2	$HBE_{2.5}$	HBE_3	HBE_4	HBE_5
fBM	0.160	0.230	0.200	0.190	0.210	0.110
fSD	0.097	0.065	0.085	0.087	0.070	0.080
fMA	0.094	0.071	0.091	0.092	0.076	0.089
fMA2	**0.088**	0.066	0.075	0.078	0.061	0.073
fARMA	0.093	**0.060**	**0.067**	**0.063**	**0.056**	**0.070**
	$PEO_{0.22}$	$PEO_{0.45}$	$PEO_{0.6}$	$PEO_{0.75}$	$PEO_{0.9}$	$PEO_{1.22}$
fBM	0.130	0.160	0.180	0.170	0.150	0.210
fSD	0.100	0.110	0.130	0.100	0.120	0.093
fMA	0.100	0.120	0.140	0.120	0.120	0.110
fMA2	0.098	**0.095**	0.100	0.092	0.097	0.087
fARMA	**0.088**	0.095	**0.094**	**0.084**	**0.086**	**0.081**

Figure 3.10), but can be more biased when heavy noises exist.

3.6 Discussion

In this chapter we present a ARMA(p, q) filter that theoretically accounts for all currently known high-frequency noise sources in particle tracking experiments. We also study its most simplified version: the fMA model, that can be conveniently modified and applied in real experiments. Under the framework of location-scale model, parameter estimation and further statistical inference can be efficiently obtained. In order to evaluate the performance of fMA model, rigorous comparison between fMA and fSD model is conducted, where fSD model is a parametric model that depends on the state-of-art theory for experimental noises. Comparison results show that even with only one parameter controlling the whole dynamic in noise filtering, fMA has comparable performance with the complicated fSD model for parameter estimation.

The validity of fMA model is further proven in simulation and the viscous fluid study, where the ground true about particle trajectories are given. It turns out that Savin and

Doyle's localization errors are not sufficient for the full spectrum of high-frequency noise sources, not only for its instability in diffusive environments ($\alpha = 1$) but also estimated exposure time τ is unrealistically high. In viscoelastic fluid experiments where the detailed mechanistic principles and precise underlying stochastic processes driving the "pure" particle motion remain unknown, we applied the fMA model to recorded positions to help understand the properties of particles in biological fluids.

There still exists many future work about proposed ARMA(p, q) filter. The determination of the order of autoregressive and moving-average terms remains unsolved. We also do not have the precise computation of the autocovariance of arbitrary ARMA(p, q) filter and use the numerical approximation instead. In addition, computation cost of maximum likelihood estimate grows exponentially as number of parameters increases, meaning that we cannot directly estimate an arbitrary order of ARMA(p, q) filter.

Chapter 4

Heterogeneity Metric for Particle Tracking

4.1 Introduction

Biological fluids, such as mucus, are considerably heterogeneous on microscopic to macroscopic length scales, due to their diverse molecular composition. Heterogeneity describes the lack of uniformity in the substance of such media and reflects the multiple-functionality of materials [Mellnik et al., 2014]. It also has a close relationship with other physical properties including viscoelasticity and may affect the behavior of particles that are diffusing in this medium [Wirtz, 2009]. For instance, the heterogeneous environment in mucus barriers of lung airways provides the biological material with the ability to regulate the diffusive dynamics of a wide range of particles [Lai et al., 2009] and tune viscoelastic moduli across a wide frequency spectrum [Matsui et al., 2005].

By analyzing the movements of particles within the same fluid, particle tracking microrheology provides unprecedented information about the heterogeneity of fluids [Valentine et al., 2001], where the resulting analysis hinges pivotally on the measurement of particles' MSD. In Figure 4.1 we demonstrate the observed MSD of particles diffusing in mucus from primary human bronchial epithelial (HBE) cell culture [Hill et al., 2014], and

compare it with a path-by-path simulation where the medium is assumed to be homogeneous.

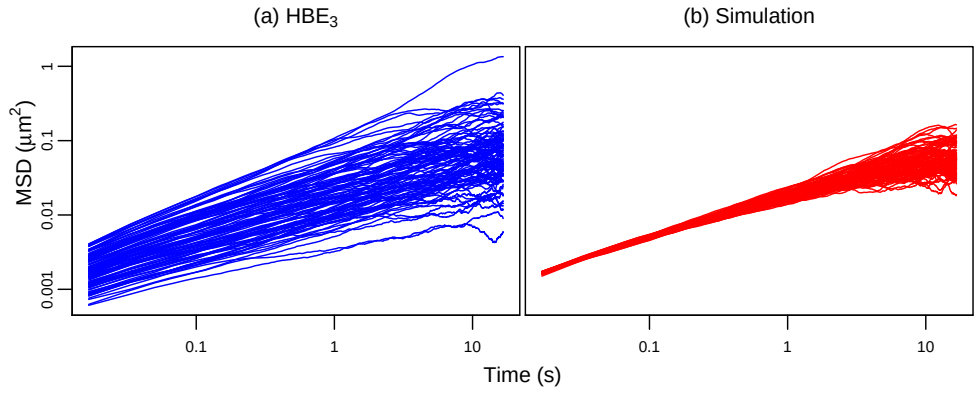


Figure 4.1: Estimated and simulated MSD for particles in HBE medium.
(a) Observed MSD computed using Equation (3.2.2).
(b) Simulated MSD, where trajectory coefficients are assumed to be identical for each path, and are estimated using the mean of parameters.

In addition to biological interests, heterogeneity of the fluid medium is also important for statistical reasons. In many applications, particle trajectories are analyzed by the ensemble average over paths [Tseng et al., 2002]. Such an approach is a natural protocol for scenarios where the arithmetic mean is a sufficient statistic, which does not necessarily hold for heterogeneous situations.

Considering the scientific and statistical importance of heterogeneity, it is valuable to have practical tools to detect and quantify material heterogeneity. To this end, several tools have been designed around the so-called van Hove correlation function [e.g., Yamamoto and Onuki, 1998], which is the distribution of increments

$$\Delta \boldsymbol{Y}_{i,n} = \boldsymbol{Y}_i((n+1)\Delta t) - \boldsymbol{Y}_i(n\Delta t)$$

over particles $i = 1, ..., M$ in a given medium for a given interobservation time Δt, where $\boldsymbol{Y}(t)$ is the process of particle trajectories. If $\boldsymbol{Y}_i(t)$ is a CSI Gaussian process and the medium is homogeneous, then the van Hove correlation function is Gaussian. Examples of

related heterogeneity metrics include the non-Gaussian parameter NG_τ [Rahman, 1964], the excess kurtosis of the van Hove function [Houghton et al., 2008] and application of "bin partitions" of compliance values $\Gamma(\tau)$ [Tseng et al., 2002]. In additional cases, heterogeneity is related to the variability of particle trajectory parameters. For instance, the scatter in the observed diffusion coefficients D is used as a measure of the heterogeneity of a cell membrane [Saxton, 1997], and an F-statistic that compares the standard deviation of individual particle increments relative to a chosen particle [Valentine et al., 2001].

In this chapter, we quantify the heterogeneity of the experimental media through the dispersion of estimated particle trajectory parameters, especially the subdiffusion parameters (α, D). Existing methods following this approach compare the variability of parameters against some baselines [Saxton, 1997, Valentine et al., 2001], where accuracy is greatly influenced by the selection of benchmarks. On the contrary, our proposed metric, which is constructed from the coefficient of variation (CV) of estimated trajectory parameters, comes with a computationally efficient estimator, which greatly stabilizes the whole mechanism. In addition, we propose a parametric bootstrap method to statistically test against the hypothesis that the fluid is homogeneous. This homogeneity test is shown to be consistent with a likelihood ratio test for homogeneity in further simulation studies and analyses of experimental data, where the proposed metric successfully quantifies the heterogeneity of fluids in different experiments.

The remainder of this chapter is organized as follows. In Section 4.2 we present a hierarchical model to describe the heterogeneity phenomena in particle tracking experiments, and propose a metric based on this model. Then we provide a computationally efficient estimator for the metric and present a homogeneity test based on the proposed metric, with a likelihood ratio test that serves a similar purpose. In Section 4.3 we categorize "heterogeneity" into two kinds: large variance and clusters, and propose an EM algorithm for identifying different clusters. In Section 4.4 we numerically examine the validity of the proposed metric and tests. In Section 4.5 we apply the heterogeneity metric to particle tracking data. Concluding remarks are offered in Section 4.6.

4.2 Our Approach

4.2.1 Hierarchical Model Structure

For the M different particle trajectories $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_M$ observed in the same fluid, the hierarchical model on all M datasets is

$$\begin{aligned} \boldsymbol{Y}_i \mid \boldsymbol{\theta}_i &\overset{\text{ind}}{\sim} f(\boldsymbol{Y}_i \mid \boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i &\overset{\text{iid}}{\sim} q(\boldsymbol{\theta} \mid \boldsymbol{\eta}), \end{aligned} \tag{4.2.1}$$

where $f(\boldsymbol{Y}_i \mid \boldsymbol{\theta}_i)$ is the density function of the particle trajectory $\boldsymbol{Y}_i$, $\boldsymbol{\theta}_i$ is the vector of particle trajectory parameters, and $\boldsymbol{\eta}$ are the hyper-parameters of the hierarchical model. Such a model naturally induces the between-path particle heterogeneity.

Bayesian inference for this hierarchical model requires a prior on $\boldsymbol{\eta} \sim \pi(\boldsymbol{\eta})$ and produces a posterior distribution

$$p(\boldsymbol{\eta}, \boldsymbol{\theta} \mid \mathscr{Y}) = \pi(\boldsymbol{\eta}) \cdot \prod_{i=1}^{M} f(\boldsymbol{Y}_i \mid \boldsymbol{\theta}_i) q(\boldsymbol{\theta}_i \mid \boldsymbol{\eta}),$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M)$ and $\mathscr{Y} = (\boldsymbol{Y}_1, ..., \boldsymbol{Y}_M)$.

While this construction is conceptually appealing, exact parameter inference must typically be conducted by Markov chain Monte Carlo (MCMC) sampling of the joint posterior distribution, which requires numerous evaluations of the likelihood functions $f(\boldsymbol{Y}_i|\boldsymbol{\theta}_i)$. Here we model $\boldsymbol{Y}_i(t)$ using the location-scale model (3.2.5) of Chapter 3. Thus, even using the superfast inference algorithms of Chapter 2, each step of an exact MCMC algorithm scales as $\mathcal{O}(MN \log^2 N)$, which quickly becomes prohibitively expensive when M is moderate to large. As an alternative, we conduct approximate Bayesian inference using the following approach of Lysy et al. [2016]

The approximation stems from the fact that the path-wise likelihood $L(\boldsymbol{\theta}_i|\boldsymbol{Y}_i)$, which is proportional to the density function

$$L(\boldsymbol{\theta}_i|\boldsymbol{Y}_i) \quad \propto \quad f(\boldsymbol{Y}_i|\boldsymbol{\theta}_i),$$

is asymptotically equivalent to

$$\phi(\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_i \mid \boldsymbol{F}_i^{-1}),$$

where $\widehat{\boldsymbol{\theta}}_i$ is the MLE estimate of $\boldsymbol{\theta}_i$, $\boldsymbol{F}_i$ is the observed Fisher information, and $\phi(\boldsymbol{z} \mid \boldsymbol{\Sigma})$ is the PDF of a multivariate normal with mean zero and variance $\boldsymbol{\Sigma}$ evaluated at $\boldsymbol{z}$. Thus by setting

$$\boldsymbol{\theta}_i \overset{\text{iid}}{\sim} q(\boldsymbol{\theta}|\boldsymbol{\eta}) \qquad \equiv \qquad \boldsymbol{\theta}_i \overset{\text{iid}}{\sim} N(\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0), \tag{4.2.2}$$

the approximate distribution on $\boldsymbol{\eta} = (\boldsymbol{\lambda}_0, \boldsymbol{\Omega}_0)$ and $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M)$ is that of the normal-normal hierarchical model

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_i &\overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{\theta}_i, \boldsymbol{F}_i^{-1}) \\ \boldsymbol{\theta}_i &\overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0). \end{aligned} \tag{4.2.3}$$

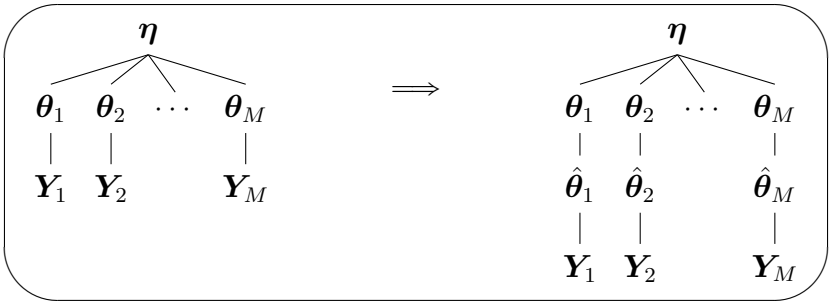


Figure 4.2: The approximated normal-normal hierarchical model.

This approximation procedure (illustrated in Figure 4.2) converts the original hierarchical model (4.2.1) into a highly tractable one. For one, calculating the MLEs and Fisher informations $\widehat{\boldsymbol{\theta}}_i$ and $\boldsymbol{F}_i$ for each particle can be done in parallel. Moreover, for the scale-invariant hyper-parameter prior

$$\pi(\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0) \quad \propto \quad |\boldsymbol{\Omega}_0|^{-(\omega+L+1)/2},$$

where L is the dimension of $\boldsymbol{\theta}$, and ω is the degree of freedom of the Inverse-Wishart prior for $\boldsymbol{\Omega}_0$, the approximate posterior (4.2.3) has precisely the form of a multivariate normal distribution and can be analyzed in a Bayesian approach using a Gibbs sampler [Lysy

et al., 2016].

4.2.2 Single-Parameter Heterogeneity Metric

In order to quantify the heterogeneity from the distribution of one parameter $\tau \in \boldsymbol{\theta}$, we propose to defining a heterogeneity metric B as the fraction of the variance of the MLE estimate $\hat{\tau}$ that is "unexplained" by the underlying variance of the parameter τ:

$$B(\tau) = 1 - \frac{\text{var}(E[\hat{\tau}_i|\boldsymbol{\theta}_i])}{\text{var}(\hat{\tau}_i)} = \frac{E[\text{var}(\hat{\tau}_i|\boldsymbol{\theta}_i)]}{\text{var}(\hat{\tau}_i)}.$$

If $\boldsymbol{\theta}_i$ is the same for all i, we have that $\text{var}(\hat{\tau}_i \mid \boldsymbol{\theta}_i) = \text{var}(\hat{\tau}_i)$, a number equal to its expectation, meaning that $B = 1$. Otherwise for heterogeneous situations we have that $B < 1$. Without loss of generality, we assume that the parameter τ is the first element of $\boldsymbol{\theta} = \{\theta^{(1)}, \ldots, \theta^{(L)}\}$, i.e. $\tau = \theta^{(1)}$. This metric can also be interpreted as the shrinkage factor for the multilevel model (4.2.3) [Morris et al., 2012].

In the following, we are going to propose an estimator for B. The estimation step starts from a normal approximation of the distribution of the MLE estimate $\hat{\tau}_i$ that

$$\hat{\tau}_i \sim \mathcal{N}(\tau_i, \sigma_i^2),$$

where σ_i^2 is the first diagonal element of the inverse Fisher information $\boldsymbol{\Sigma}_i = \boldsymbol{F}_i^{-1}$. This approximation stems from the hierarchical model $\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_i \overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{\theta}_i, \boldsymbol{F}_i^{-1})$, and reveals that

$$\text{var}(\hat{\tau}_i \mid \boldsymbol{\theta}_i) = \sigma_i^2.$$

As a result, the expected variance of $\hat{\tau}_i \mid \boldsymbol{\theta}_i$ can be approximated by the average variance

$$E[\text{var}(\hat{\tau}_i|\boldsymbol{\theta}_i)] \approx \frac{1}{M}\sum_{i=1}^{M} \text{var}(\hat{\tau}_i \mid \boldsymbol{\theta}_i) = \frac{1}{M}\sum_{i=1}^{M} \sigma_i^2.$$

In general the variance σ_i^2 is incalculable. However, if the MLE $\widehat{\boldsymbol{\theta}}_i$ is computed using the

location-scale model for particle trajectories (3.2.5)

$$\Delta \boldsymbol{Y}_i \sim \text{MatNorm}(\boldsymbol{G}\boldsymbol{\beta}, \boldsymbol{V}_{\boldsymbol{\varphi}}, \boldsymbol{\Sigma}), \tag{4.2.4}$$

where $\boldsymbol{G}$ is the drift process for particle trajectory increments $\Delta \boldsymbol{Y}_i$ and $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\varphi}, \boldsymbol{\Sigma}\}$, we can compute the expected Fisher information efficiently and analytically using Equation (2.3.6)

$$\boldsymbol{F}_i = E\left[-\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i) \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i} \right],$$

where $\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)$ is the log-likelihood of the location-scale model (4.2.4). The detailed computation of $\boldsymbol{F}_i$ is described in Appendix F.

Because of the lack of the true value $\boldsymbol{\theta}_i$, we can plug in the MLE estimate $\widehat{\boldsymbol{\theta}}_i$ and estimate the inverse Fisher information $\boldsymbol{F}^{-1}$ using

$$\widehat{\boldsymbol{\Sigma}}_i = E\left[-\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i) \Big|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}_i} \right]^{-1}. \tag{4.2.5}$$

The first diagonal element of $\widehat{\boldsymbol{\Sigma}}_i$ is denoted as $\hat{\sigma}_i^2$, which is an estimate for $\sigma_i^2 = \text{var}(\hat{\tau}_i \mid \boldsymbol{\theta}_i)$.

In addition, some particles will move far enough in the direction perpendicular to the focal plane so as to no longer be detectable, therefore their trajectories have different lengths and are shorter than the experimental duration N. However, the current estimate of the numerator does not account for this fact. Considering that the Fisher information in trajectory $\boldsymbol{Y}_i$ is proportional to its length N_i, we introduce a weight $w_i = \frac{N_i}{N}$ to resolve the bias induced by particles randomly exiting the camera focal plane, and have the following weighted average to estimate the expected conditional variance

$$E[\text{var}(\hat{\tau}_i | \boldsymbol{\theta}_i)] \approx \frac{1}{M} \sum_{i=1}^{M} w_i \cdot \hat{\sigma}_i^2. \tag{4.2.6}$$

The denominator $\text{var}(\hat{\tau})$ is estimated using the traditional method, where the sample variance $s^2(\hat{\tau}) = \frac{1}{M-1} \sum_{i=1}^{M} (\hat{\tau}_i - \frac{1}{M} \sum_{j=1}^{M} \hat{\tau}_j)^2$ is applied. In conclusion, for a single parameter

τ, the estimator for its heterogeneity metric $B(\tau)$ has the following form

$$\hat{B}(\tau) = \frac{\sum_{i=1}^{M} w_i \cdot \hat{\sigma}_i^2}{M \cdot s^2(\hat{\tau})}. \tag{4.2.7}$$

4.2.3 Multiple-Parameter Heterogeneity Metric

When multiple parameters are included to measure the heterogeneity of the fluid, both $E[\text{var}(\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_i)]$ and $\text{var}(\widehat{\boldsymbol{\theta}}_i)$ are matrices instead of numbers. To define a multi-parameter heterogeneity metric that still reflects the fraction of the variance of the MLE estimated $\widehat{\boldsymbol{\theta}}_i$ that is "unexplained" by the covariance of the parameter $\boldsymbol{\theta}$, we use the matrix trace to quantify the variance in $\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_i$, i.e. the multi-parameter metric is defined as

$$B(\boldsymbol{\theta}) = \frac{\text{tr}\{E[\text{var}(\widehat{\boldsymbol{\theta}}_i|\boldsymbol{\theta}_i)]\}}{\text{tr}\{\text{var}(\widehat{\boldsymbol{\theta}}_i)\}}.$$

The estimation of $B(\boldsymbol{\theta})$ starts from the multi-variate normal approximation of the distribution of $\widehat{\boldsymbol{\theta}}_i$ in the normal-normal hierarchical model (4.2.3), where

$$\widehat{\boldsymbol{\theta}}_i \sim \mathcal{N}(\boldsymbol{\theta}_i, \boldsymbol{\Sigma}_i), \quad \boldsymbol{\Sigma}_i = E\left[-\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \ell(\boldsymbol{\theta} \mid \mathbf{Y}_i)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i} \right]^{-1}. \tag{4.2.8}$$

Following the same logic in (4.2.6), we estimate the expected covariance of $\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_i$ with

$$E[\text{var}(\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_i)] \approx \frac{1}{M} \sum_{i=1}^{M} w_i \widehat{\boldsymbol{\Sigma}}_i,$$

where $\widehat{\boldsymbol{\Sigma}}_i$ is defined in (4.2.5). In addition, the denominator $\text{var}(\widehat{\boldsymbol{\theta}}_i)$ can be computed following the traditional method

$$\text{var}(\widehat{\boldsymbol{\theta}}_i) = \frac{1}{M-1} \sum_{i=1}^{M} (\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_0)(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_0)',$$

where $\widehat{\boldsymbol{\theta}}_0 = \frac{1}{M}\sum_{i=1}^{M}\widehat{\boldsymbol{\theta}}_i$. In conclusion, the multi-parameter heterogeneity metric $\hat{B}(\boldsymbol{\theta})$ can be estimated using

$$\hat{B}(\boldsymbol{\theta}) = \frac{\mathrm{tr}\{\frac{1}{M}\sum_{i=1}^{M} w_i \widehat{\boldsymbol{\Sigma}}_i\}}{\mathrm{tr}\{\mathrm{var}(\widehat{\boldsymbol{\theta}}_i)\}}. \tag{4.2.9}$$

In addition to the point estimate of the single-parameter metric (4.2.7) and multi-parameter metric (4.2.9), we provide a bootstrap sampling approach to approximate the unconditional distribution of $\hat{B}$:

1. Sample M parameters from original MLE estimates $\{\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_M\}$ with replacement and obtain a new dataset $\{\widehat{\boldsymbol{\theta}}^{\star}_1, \ldots, \widehat{\boldsymbol{\theta}}^{\star}_M\}$.

2. For each sample, compute its heterogeneity metric $\hat{B}(\boldsymbol{\theta}^{\star})$ using Equation (4.2.9). For the single parameter metric, we extract $\hat{\tau}^{\star}_i$ from $\widehat{\boldsymbol{\theta}}^{\star}_i$ and compute the metric $\hat{B}(\tau^{\star})$ using Equation (4.2.7).

3. Repeat **step (1) - (2)** K times to obtain a size K sample of $\hat{B} : \{\hat{B}_1, \cdots, \hat{B}_K\}$.

With the non-parametric bootstrap sample $\{\hat{B}_1, \cdots, \hat{B}_K\}$, we can obtain the numerical approximation of the variance of $\hat{B}$ using the sample variance $s^2(\hat{B})$ and further construct the its 95% confidence interval with the empirical quantiles.

4.2.4 Hypothesis Test

In addition to the estimator for the heterogeneity metric $\hat{B}(\boldsymbol{\theta})$, we propose a test for the null hypothesis of homogeneity, which has two equivalent representations

$$\begin{aligned} &H_0 : B(\boldsymbol{\theta}) = 1, \qquad H_A : B(\boldsymbol{\theta}) < 1 \\ &H_0 : \boldsymbol{\Omega}_0 = \mathbf{0}, \qquad H_A : \boldsymbol{\Omega}_0 \text{ is a positive definite matrix,} \end{aligned}$$

where $\boldsymbol{\Omega}_0$ is the covariance of $\boldsymbol{\theta}_i$ in the prior (4.2.2). Under H_0, $\widehat{\boldsymbol{\theta}}_i$ approximately follows a multivariate normal distribution

$$\widehat{\boldsymbol{\theta}}_i \overset{\text{ind}}{\sim} \mathcal{N}\left(\boldsymbol{\theta}_0, \frac{1}{w_i} \cdot \boldsymbol{\Sigma}_0\right), \tag{4.2.10}$$

where $\boldsymbol{\theta}_0 \equiv \boldsymbol{\theta}_i$ is the true value of model parameters, $\boldsymbol{\Sigma}_0 = E\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}\right]^{-1}$ is the inverse Fisher information at $\boldsymbol{\theta}_0$ and the weight $\frac{1}{w_i}$ is used to adjust the size of the variance according to particle trajectory lengths N_i. To estimate the value of $\boldsymbol{\theta}_0$ under H_0, we maximize the log-likelihood of (4.2.10)

$$\ell(\boldsymbol{\theta}_0 \mid \widehat{\boldsymbol{\theta}}_i) = \sum_{i=1}^{M}\{-\frac{1}{2}w_i(\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_0)'\boldsymbol{\Sigma}_0^{-1}(\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_0) - \frac{1}{2}\log|\boldsymbol{\Sigma}_0|\}$$

where the matrix

$$\boldsymbol{\Sigma}_0 = E\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}\right]^{-1}$$

is a function of $\boldsymbol{\theta}_0$.

With the MLE estimate $\widehat{\boldsymbol{\theta}}_0 = \arg\max_{\boldsymbol{\theta}_0}\ell(\boldsymbol{\theta}_0 \mid \widehat{\boldsymbol{\theta}}_i)$ and $\widehat{\boldsymbol{\Sigma}}_0 = E\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)\Big|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}_0}\right]^{-1}$, we propose a parametric bootstrap procedure to estimate the distribution of $\hat{B}(\boldsymbol{\theta})$ under H_0:

1. Sample $\widehat{\boldsymbol{\theta}}_i^\star$ from the multivariate normal distribution

   $$\widehat{\boldsymbol{\theta}}_i^\star \sim \mathcal{N}(\widehat{\boldsymbol{\theta}}_0, \widehat{\boldsymbol{\Sigma}}_0).$$

   For each simulated $\widehat{\boldsymbol{\theta}}_i^\star$, pair it with a covariance matrix $\widehat{\boldsymbol{\Sigma}}_i^\star$, where

   $$\widehat{\boldsymbol{\Sigma}}_i^\star = E\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)\Big|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}_i^\star}\right]^{-1}$$

   and obtain $\left(\widehat{\boldsymbol{\theta}}_i^\star, \frac{1}{w_i}\widehat{\boldsymbol{\Sigma}}_i^\star\right)$.

2. Iterate **step (1)** from $i = 1, \ldots, M$, where M is the number of trajectories in the experiment. Then we have a size M sample of $\left(\widehat{\boldsymbol{\theta}}_i^\star, \frac{1}{w_i}\widehat{\boldsymbol{\Sigma}}_i^\star\right)$ under H_0.

3. With the sample $\{\left(\widehat{\boldsymbol{\theta}}_i^\star, \frac{1}{w_i}\widehat{\boldsymbol{\Sigma}}_i^\star\right)\}_{i=1}^M$ we can compute the corresponding heterogeneity

metric $\hat{B}^{\star}(\boldsymbol{\theta})$ from simulated data set $\{\left(\widehat{\boldsymbol{\theta}}_j^{\star}, \widehat{\boldsymbol{\Sigma}}_j^{\star}\right)\}_{j=1}^{M}$ using Equation (4.2.9). If we wish to test the homogeneity of a single parameter τ, we can extract $\left(\hat{\tau}_i^{\star}, \frac{1}{w_i}\hat{\sigma}_i^{2\star}\right)$ from the parametric samples where $\hat{\tau}_i^{\star}$ is the first element of $\widehat{\boldsymbol{\theta}}_i^{\star}$ and $\hat{\sigma}_i^{2\star}$ is the first diagonal element of $\widehat{\boldsymbol{\Sigma}}_i^{\star}$, then compute the metric $\hat{B}^{\star}(\tau)$ using Equation (4.2.7).

4. Repeat **step (3)** K times to obtain the size K simulation of $\hat{B}^{\star}$ under null hypothesis H_0.

Based on the samples $\{\hat{B}_1^{\star}, \hat{B}_2^{\star}, \ldots, \hat{B}_K^{\star}\}$ we can obtain the empirical distribution of metric B under H_0

$$\hat{\pi}_B(t) = \frac{1}{M}\sum_{i=1}^{M} \mathrm{I}_{\hat{B}_i^{\star} \leqslant t},$$

and construct the hypothesis test based on $\pi_B(t)$. We will reject the null hypothesis when the original metric $\hat{B} < q_{\pi}(a)$ for significance level a, where $q_{\pi}(a)$ is the a quantile of empirical distribution $\hat{\pi}_B(t)$.

4.2.5 Likelihood Ratio Test for Homogeneity

In addition to the hypothesis test based on the proposed metric, we also introduce a likelihood ratio test for homogeneity, where the null hypothesis is

$$H_0 : \boldsymbol{\Omega}_0 = \mathbf{0} \qquad H_A : \boldsymbol{\Omega}_0 \text{ is a positive definite matrix.}$$

Within the framework of the previously proposed distribution of $\widehat{\boldsymbol{\theta}}_i$ (4.2.8) and the multivariate normal prior (4.2.2), we have the distribution of $\widehat{\boldsymbol{\theta}}_i$ for given parameters $\{\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0\}$

$$\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\theta}_0, \boldsymbol{\Omega} \sim \mathcal{N}(\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_i + \boldsymbol{\Omega}_0). \tag{4.2.11}$$

To test the null hypothesis, we can construct a likelihood ratio test by maximizing

following two likelihoods:

$$\ell_0(\boldsymbol{\theta}_0 \mid \widehat{\boldsymbol{\theta}}_i) = -\frac{1}{2}\sum_{i=1}^{M}\{\log|\widehat{\boldsymbol{\Sigma}}_i| + (\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_0)'\widehat{\boldsymbol{\Sigma}}_i^{-1}(\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_0)\}$$
$$\ell_A(\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0 \mid \widehat{\boldsymbol{\theta}}_i) = -\frac{1}{2}\sum_{i=1}^{M}\{\log|\widehat{\boldsymbol{\Sigma}}_i + \boldsymbol{\Omega}_0| + (\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_0)'[\widehat{\boldsymbol{\Sigma}}_i + \boldsymbol{\Omega}_0]^{-1}(\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_0)\},$$

where $\ell_0(\boldsymbol{\theta}_0 \mid \widehat{\boldsymbol{\theta}}_i)$ is the likelihood of model (4.2.11) under H_0 and $\ell_A(\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0 \mid \widehat{\boldsymbol{\theta}}_i)$ is the likelihood of model (4.2.11) under $H_0 \cup H_A$. Their MLE estimates are respectively

$$\widehat{\boldsymbol{\theta}}_0^{(0)} = \arg\max_{\boldsymbol{\theta}_0} \ell_0(\boldsymbol{\theta}_0 \mid \widehat{\boldsymbol{\theta}}_i)$$
$$(\widehat{\boldsymbol{\theta}}_0^{(A)}, \widehat{\boldsymbol{\Omega}}_0^{(A)}) = \arg\max_{\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_0} \ell_A(\boldsymbol{\theta}_0, \boldsymbol{\Omega}_0 \mid \widehat{\boldsymbol{\theta}}_i),$$

which can be obtained through numerical optimization in particle tracking experiments.

As the number of particle trajectories M goes to infinity, we have that

$$t(\widehat{\boldsymbol{\theta}}_i) = -2\left(\ell_0(\widehat{\boldsymbol{\theta}}_0^{(0)} \mid \widehat{\boldsymbol{\theta}}_i) - \ell_A(\widehat{\boldsymbol{\theta}}_0^{(A)}, \widehat{\boldsymbol{\Omega}}_0^{(A)} \mid \widehat{\boldsymbol{\theta}}_i)\right) \quad \xrightarrow{\text{d}} \quad \chi^2_{L(L-1)/2}.$$

where $L(L-1)/2$ is the difference between the number of parameters in ℓ_0 and ℓ_A

In the likelihood ratio test, H_0 will be rejected when the test statistic $t(\widehat{\boldsymbol{\theta}}_i) > q_{\chi^2_{L(L-1)/2}}(a)$ for significance level a, where $q_{\chi^2_{L(L-1)/2}}(a)$ is the a-th quantile of $\chi^2_{L(L-1)/2}$ distribution.

4.3 Heterogeneity: Large Variance and Clusters

The definition of heterogeneity refers to the "over-dispense" in the distribution of particle trajectories. However the over-dispersion caused by a large variance and the over-dispersion caused by the presence of multiple clusters have distinct interpretations in particle tracking experiments. The former case is the indication of ergodicity breaking, while the latter case is more likely to be caused by unaccounted experimental or instrumental errors.

When multiple clusters exist in the particle tracking data, the whole experimental

results will definitely look heterogeneous and have a low value of the heterogeneity metric. However, the trajectories within the same cluster can still be homogeneous. In order to examine the homogeneity of trajectories within a cluster, we need an algorithm to identify different clusters in particle tracking data and label their members. In the following we introduce a mixture model with K components for M estimated parameters $\widehat{\boldsymbol{\theta}}_i$, $i = 1, \ldots, M$

$$
\begin{aligned}
z_i &\sim \text{Multinomial}(\boldsymbol{\pi}) \\
\widehat{\boldsymbol{\theta}}_i \mid z_i = k &\sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Phi}_k),
\end{aligned}
$$

where $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$ is a vector of length K and its elements $\pi_k = P(z_i = k)$ are the weights of different clusters which sum up to 1. The members in the k-th cluster are assumed to follow a multivariate normal distribution with mean $\boldsymbol{\mu}_k$ and variance $\boldsymbol{\Phi}_k$. The parameters of the Gaussian mixture model are not estimated by maximizing the log-likelihood

$$
\ell(\boldsymbol{\pi}, \boldsymbol{\mu}_k, \boldsymbol{\Phi}_k \mid \widehat{\boldsymbol{\theta}}_i) = \sum_{i=1}^{M} \left(\log \sum_{k=1}^{K} \pi_k p_k(\widehat{\boldsymbol{\theta}}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Phi}_k) \right),
$$

because of its unbounded likelihood function, infinite Fisher information and other undesirable properties [Hartigan, 1985, Chen et al., 2009]. An EM algorithm of the following updating steps can be applied for a consistent parameter estimation of the multivariate Gaussian mixture model.

For the $(n-1)$th guess of parameters $\Theta^{(n-1)} = \{\boldsymbol{\pi}^{(n-1)}, \boldsymbol{\mu}_k^{(n-1)}, \boldsymbol{\Phi}_k^{(n-1)}\}$, $k = 1, \ldots, K$, we first compute the "membership weight" for data $\widehat{\boldsymbol{\theta}}_i$ in cluster k

$$
w_{i,k}^{(n)} = p(z_i = k \mid \widehat{\boldsymbol{\theta}}_i, \Theta^{(n-1)}) = \frac{\pi_k^{(n-1)} \cdot p_k(\widehat{\boldsymbol{\theta}}_i \mid z_i = k, \boldsymbol{\mu}_k^{(n-1)}, \boldsymbol{\Phi}_k^{(n-1)})}{\sum_{j=1}^{K} \pi_j^{(n-1)} \cdot p_j(\widehat{\boldsymbol{\theta}}_i \mid z_i = j, \boldsymbol{\mu}_j^{(n-1)}, \boldsymbol{\Phi}_j^{(n-1)})},
$$

where p_k is the density of a D-dimensional multivariate Gaussian

$$
p_k(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Phi}_k) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Phi}_k|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)' \boldsymbol{\Phi}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)\}.
$$

Let $M_k^{(n)} = \sum_{i=1}^{M} w_{ik}^{(n)}$ be the sum of the membership weights for the k-th component,

we update $\Theta^{(n)}$ with

$$\begin{aligned}
\pi_k^{(n)} &= \frac{{M_k}^{(n)}}{M} \\
\boldsymbol{\mu}_k^{(n)} &= \frac{1}{M_k^{(n)}} \sum_{i=1}^{M} w_{i,k}^{(n)} \cdot \widehat{\boldsymbol{\theta}}_i \\
\boldsymbol{\Sigma}_k^{(n)} &= \frac{1}{M_k^{(n)}} \sum_{i=1}^{M} w_{i,k}^{(n)} \cdot (\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\mu}_k^{(n)})(\widehat{\boldsymbol{\theta}}_i - \boldsymbol{\mu}_k^{(n)})'.
\end{aligned}$$

The iteration will stop when the improvement in the likelihood $\varepsilon_n = \ell(\boldsymbol{\pi}^{(n)}, \boldsymbol{\mu}_k^{(n)}, \boldsymbol{\Phi}_k^{(n)} \mid \widehat{\boldsymbol{\theta}}_i) - \ell(\boldsymbol{\pi}^{(n-1)}, \boldsymbol{\mu}_k^{(n-1)}, \boldsymbol{\Phi}_k^{(n-1)} \mid \widehat{\boldsymbol{\theta}}_i)$ is too small, and the algorithm finally returns the estimated parameters $\hat{\Theta} = \{\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\}$. We can further identify the members of different clusters by checking their membership weight, where data $\widehat{\boldsymbol{\theta}}_i$ belongs to the cluster k such that $w_{i,k}$ is the largest.

4.4 Simulation Study

In this section, we evaluate the performance of the heterogeneity metric B in various simulation settings. In each setting, we simulate $M = 500$ observed data trajectories $\boldsymbol{Y}^{(m)} = [\boldsymbol{Y}_1^{(m)}, \ldots, \boldsymbol{Y}_N^{(m)}]$, $m = 1, \ldots, M$, each consisting of $N = 1800$ two-dimensional observations recorded at time intervals $\Delta t = 1/60$.

To better fit the assumption of a multivariate Gaussian prior (4.2.2), model parameters are generated and estimated in transformed scales such that their distribution is more "Gaussian". For instance the diffusivity $D > 0$ is transformed into the log scale $\lambda = \log D$ and the subdiffusive parameter $\alpha \in (0, 2)$ is transformed into a boundless form $\gamma = \log \frac{\alpha}{2-\alpha}$.

Since the sub-linear power law of MSD

$$\text{MSD}(t) \sim 2D \cdot t^{\alpha}$$

is determined by the subdiffusive parameters (α, D), we can investigate the heterogeneity of fluids by examining the heterogeneity metric on (α, D) . In the following analyses of

simulated and experimental data, we look into the heterogeneity metrics $B(\alpha)$, $B(D)$ and $B(\alpha, D)$ to study the features of the proposed metric and properties of the media.

4.4.1 fBM Simulation

Consider the following simulation setting designed to reveal how the metric B can quantify the heterogeneity, where various levels of dispersion of parameters are manually included in the simulated fBM processes. The particle trajectories are simulated from

$$\boldsymbol{Y}^{(m)} \overset{\text{iid}}{\sim} \text{MatNorm}(\boldsymbol{0}, \boldsymbol{V}_\star, \boldsymbol{\Psi}_\star), \tag{4.4.1}$$

where $\boldsymbol{\Psi}_\star = \left[\begin{smallmatrix} D_\star & 0 \\ 0 & D_\star \end{smallmatrix}\right]$ is a diagonal matrix and the size $N \times N$ covariance matrix $\boldsymbol{V}_\star$ is that of a length-N fBM process with coefficient $\alpha_\star$. The subdiffusive coefficients $(\alpha_\star, D_\star)$ are simulated independently from normal distributions

$$\begin{aligned} \alpha_\star &\overset{\text{iid}}{\sim} \mathcal{N}(\alpha_0, d \cdot \sigma^2_{\alpha_0}) \\ D_\star &\overset{\text{iid}}{\sim} \mathcal{N}(D_0, d \cdot \sigma^2_{D_0}), \end{aligned}$$

where $\alpha_0 = 0.8$, $D_0 = 1$, $\sigma^2_{\alpha_0}$ and $\sigma^2_{D_0}$ are the first and second diagonal elements of the inverse of observed Fisher matrix

$$\left[- \frac{\partial^2}{\partial \boldsymbol{a} \partial \boldsymbol{a}'} \ell(\boldsymbol{a} \mid \boldsymbol{Y}^{(m)}) \Big|_{\boldsymbol{a}=\boldsymbol{a}_0} \right]^{-1}$$

where $\boldsymbol{a} = (\alpha, D)$, $\boldsymbol{a}_0 = (\alpha_0, D_0)$ and $\ell(\boldsymbol{a} \mid \boldsymbol{Y}^{(m)})$ is the likelihood of (4.4.1). The heterogeneity factor $d \geqslant 0$ can be used to control the degree of heterogeneity in simulated coefficients $(\alpha_\star, D_\star)$. In this section we generate 6 groups of particle trajectories $\boldsymbol{Y}^{(m)}$ with different levels of $d = \{0, 1/100, 1/10, 1/2, 1/5, 1\}$, and estimate the parameters using the **fBM** method described in (3.4.3) in the transformed scale. Figure 4.3 displays the estimation from the generated data sets, where the dispersion in estimated parameters grows as d increases.

In Table 4.1 we demonstrate the estimated metric B for α, D and (α, D) together.

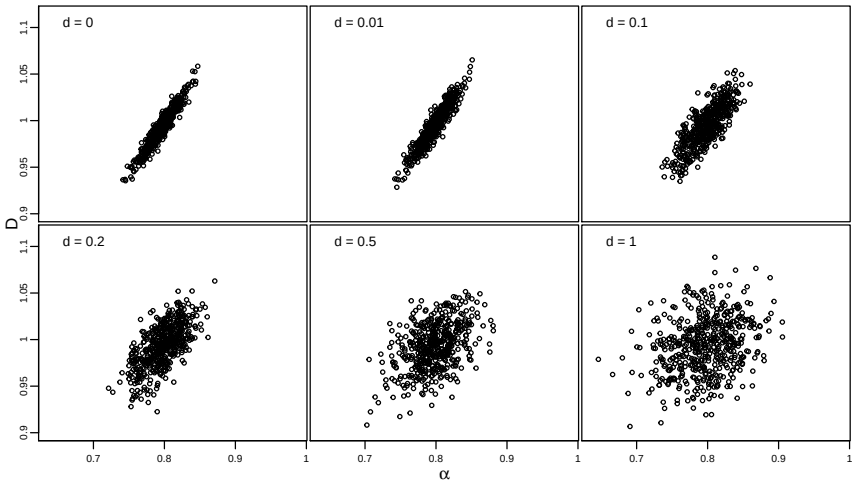


Figure 4.3: Estimated subdiffusive parameters (α, D), simulated fBM process.

For each estimated metric we test them against the heterogeneity using the parametric Bootstrap method. From the metric and its hypothesis test, we find that metric B remains a high value for the "absolute homogeneous" ($d = 0$) case and "almost homogeneous" ($d = 0.01$) case. When the variety in true parameters (α, D) is relatively high ($d \geqslant 0.1$), the heterogeneity metric will drop below 0.9 and tested to be heterogeneous. In addition, we run the likelihood ratio test on (α, D) to examine their homogeneity, and the result is consistent with the heterogeneity metric.

Table 4.1: Estimated heterogeneity metric, fBM simulation.
For each experiment, the metric that passes the homogeneity test is marked with *.

d	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
0	$1^\star$	$1^\star$	$1^\star$	0.91
0.01	$0.99^\star$	$0.95^\star$	$0.96^\star$	0.39
0.1	0.85	0.77	0.79	0
0.2	0.73	0.51	0.55	0
0.5	0.62	0.28	0.32	0
1	0.53	0.15	0.18	0

4.4.2 fBM Simulation, fixed α

For simulated particle trajectories using fBM model (4.4.1), even when the subdiffusive parameters (α, D) are independently generated, their MLE estimate $(\hat{\alpha}, \hat{D})$ are still correlated through the Fisher information under location-scale framework. Since $\text{cov}(\hat{\alpha}, \hat{D}) \neq 0$, the heterogeneity in D will not only influence its MLE $\hat{D}$, but also affect the distribution of $\hat{\alpha}$. In order to examine whether the heterogeneity in one parameter "infects" the estimates of other homogeneous parameters, we consider the following simulation procedure where fBM processes with the same α but different D are simulated. The particle trajectories are simulated from

$$\boldsymbol{Y}^{(m)} \overset{\text{iid}}{\sim} \text{MatNorm}(\boldsymbol{0}, \boldsymbol{V}_0, \boldsymbol{\Psi}_\star), \tag{4.4.2}$$

where $\boldsymbol{\Psi}_\star = \left[\begin{smallmatrix} D_\star & 0 \\ 0 & D_\star \end{smallmatrix}\right]$ is a diagonal matrix and the size $N \times N$ covariance matrix $\boldsymbol{V}_0$ is that of an fBM process with coefficient $\alpha_0 = 0.8$. The diffusivity $D_\star$ is simulated from a normal distribution

$$D_\star \overset{\text{iid}}{\sim} \mathcal{N}(D_0, d \cdot \sigma^2_{D_0}),$$

where $D_0 = 1$ and $\sigma^2_{D_0}$ is the second diagonal element of the inverse of observed Fisher matrix

$$\left[-\frac{\partial^2}{\partial \boldsymbol{a} \partial \boldsymbol{a}'} \ell(\boldsymbol{a} \mid \boldsymbol{Y}^{(m)}) \Big|_{\boldsymbol{a}=\boldsymbol{a}_0} \right]^{-1}$$

where $\boldsymbol{a} = (\alpha, D)$, $\boldsymbol{a}_0 = (\alpha_0, D_0)$ and $\ell(\boldsymbol{a} \mid \boldsymbol{Y}^{(m)})$ is the likelihood of (4.4.2). The heterogeneity factor has values $d = \{0, 1/100, 1/10, 1/2, 1/5, 1\}$ in different simulation scenarios. The particle trajectories are estimated using the same **fBM** method in transformed scales. Figure 4.4 displays the estimated (α, D) from the generated data sets.

In Table 4.2 we demonstrate the estimated metric for α, D and (α, D), together with their test result against heterogeneity using the parametric bootstrap method. We find that in all scenarios $B(\alpha)$ is very high and passes the homogeneity test, while for $B(D)$ it is essentially equal to one for the $d = 0$ and $d = 0.01$ cases, which is similar to the result in the previous simulation. In addition, we find that $B(\alpha, D)$ is closer to the minimal value among $\{B(\alpha), B(D)\}$, suggesting that when more than one parameter is applied to

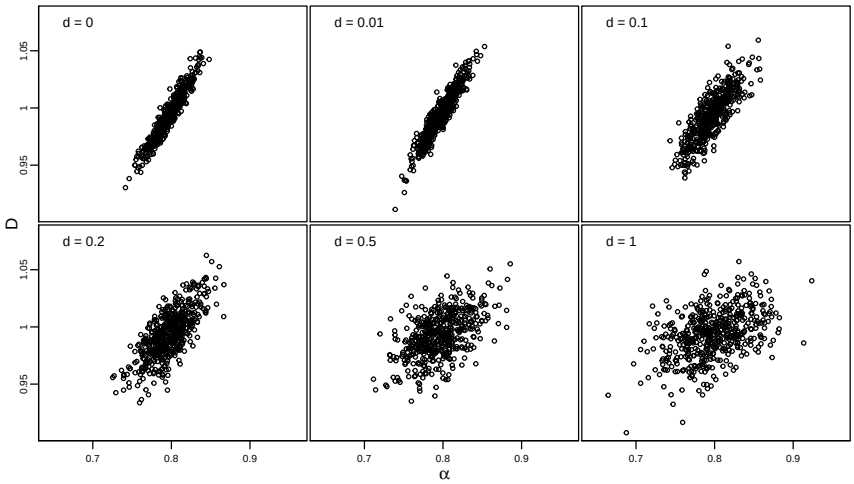


Figure 4.4: Estimated subdiffusive parameters (α, D), simulated fBM processes with fixed α.

measure the heterogeneity, the overall performance is dominated by the most heterogeneous parameter. From this simulation we learn that the heterogeneity metric of different parameters are almost independent, and homogeneous parameters remain homogeneous, despite the heterogeneity in other dependent parameters.

Table 4.2: Estimated heterogeneity metric, fBM simulation with fixed α.
For each experiment, the metric that passes the homogeneity test is marked with $\star$.

d	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
0	$1^\star$	$0.99^\star$	$1^\star$	0.89
0.01	$0.93^\star$	$0.92^\star$	$0.92^\star$	0.44
0.1	$0.93^\star$	0.68	0.71	0
0.2	$0.95^\star$	0.55	0.60	0
0.5	$1^\star$	0.32	0.39	0
1	$0.97^\star$	0.25	0.30	0

4.4.3 Model Misspecification Simulation

In this section, we show how the misspecified model will jeopardize the proposed metric and influence the power of the homogeneity test. For this purpose, here we generate particle trajectories using the **fMA** model proposed in Section 3.3.2, where the particle trajectories are simulated from

$$\boldsymbol{Y}^{(m)} \overset{\text{iid}}{\sim} \text{MatNorm}(\boldsymbol{0}, \boldsymbol{V}_\star, \boldsymbol{\Psi}_0), \tag{4.4.3}$$

where $\boldsymbol{\Psi}_0 = \left[\begin{smallmatrix} D_0 & 0 \\ 0 & D_0 \end{smallmatrix}\right]$, $D_0 = 1$ is a diagonal matrix and the size $N \times N$ covariance matrix $\boldsymbol{V}_\star$ is that of an fMA model (3.3.7) with coefficients $\alpha_0 = 0.8$ and $\rho_\star$ generated from a normal distribution

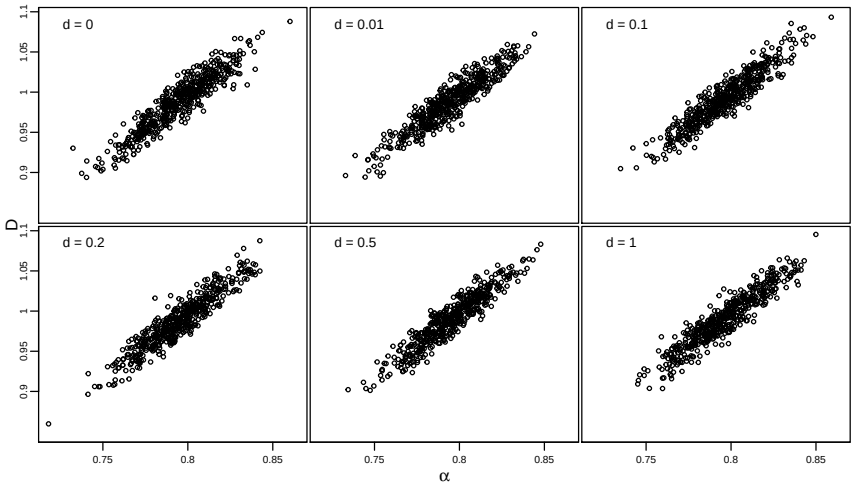
$$\rho_\star \overset{\text{iid}}{\sim} \mathcal{N}(\rho_0, d \cdot \sigma^2_{\rho_0}),$$

where $\rho_0 = 0.2$, and $\sigma^2_{\rho_0}$ is the second diagonal element of the inverse Fisher matrix

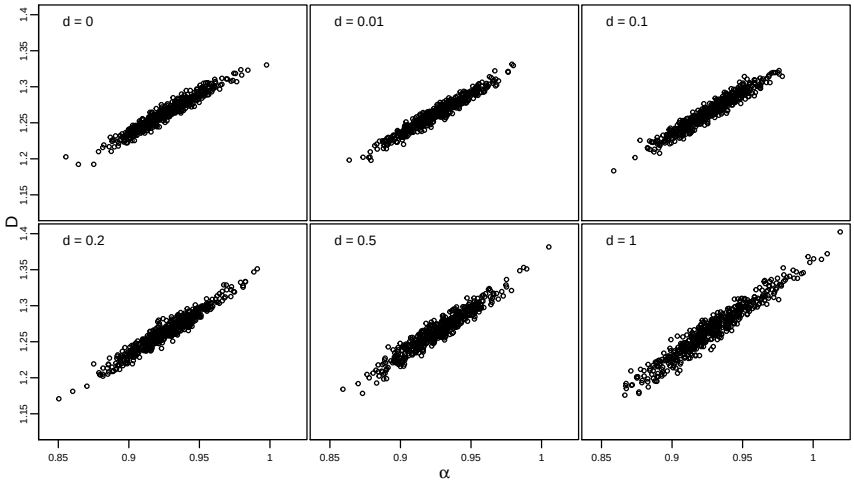
$$E\left[-\frac{\partial^2}{\partial \boldsymbol{a} \partial \boldsymbol{a}'} \ell(\boldsymbol{a} \mid \boldsymbol{Y}^{(m)}) \Big|_{\boldsymbol{a}=\boldsymbol{a}_0} \right]^{-1}$$

where $\boldsymbol{a} = (\alpha, \rho, D)$ $\boldsymbol{a}_0 = (\alpha_0, \rho_0, D_0)$ and $\ell(\boldsymbol{a} \mid \boldsymbol{Y}^{(m)})$ is the likelihood of (4.4.3). The generated $\boldsymbol{Y}^{(m)}$ simulates the movement of a particle that is affected by a certain degree of high-frequency noises.

In this section we generate 6 groups of particle trajectories $\boldsymbol{Y}^{(m)}$ with identical subdiffusive parameters (α_0, D_0) and different levels of heterogeneity in high-frequency noises by putting $d = \{0, 1/100, 1/10, 1/2, 1/5, 1\}$. To investigate the aftermath of applying wrong models in particle tracking experiments and the further influence on heterogeneity metric, we estimate the subdiffusive parameters (α, D) using **fBM** and **fMA** separately. Figure 4.5(a) shows the estimated (α, D) using the **fMA** method, where estimations for all cases look almost identical. Figure 4.5(b) displays the estimated (α, D) using the **fBM** method, where estimations are systematically biased because of the unaccounted high-frequency noises. The increasing dispersion in the estimation implies that the heterogeneity in parameters will contaminate the homogeneous parameters (α, D) when the wrong parametric model is applied.



(4.5(a)) Estimated subdiffusive parameters in transformed scale (λ, γ) where the correct fMA model is applied.



(4.5(b)) Estimated subdiffusive parameters in transformed scale (λ, γ) where fBM model is incorrectly applied.

In Table 4.3 we show the heterogeneity metric of α, D and (α, D) for **fMA** and **fBM** estimates. Both subdiffusive parameters from the **fMA** model come with a high value in metric B, and are tested to be homogeneous for all d value. This finding is consistent with the conclusion in the previous simulation study that homogeneity of parameters α, D will not be affected by the heterogeneous parameter ρ. As for the **fBM** cases, the heterogeneity metrics $B(\alpha), B(D), B(\alpha, D)$ are obviously inflated and greater than 1. The metric B are defined to be a value between 0 and 1, but its estimate is not necessarily bounded by 1, especially when incorrect models are applied. Despite the high metric value, none of them passes the homogeneity test, even for the homogeneous ($d = 0$) case and "almost-homogeneous" ($d = 1/100$) case. This finding suggests that our metric and homogeneity test will completely fail when wrong models are used. Considering the estimator of heterogeneity metric (4.2.7) (4.2.9) where the Fisher information is involved, the poor performance of the heterogeneity metric under model misspecification is understandable, because the Fisher information can change considerably for different models.

Table 4.3: Estimated heterogeneity metric, model misspecification simulation.
For each experiment, the metric that passes the homogeneity test is marked with *.

d	**fMA**				**fBM**			
	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
0	$1^\star$	$0.97^\star$	$0.98^\star$	0.87	1.2	1.1	1.1	0.77
0.01	$0.98^\star$	$0.95^\star$	$0.95^\star$	0.67	1.1	1.1	1.1	0.29
0.1	$0.96^\star$	$0.97^\star$	$0.96^\star$	0.43	1	1.1	1.1	0
0.2	$0.99^\star$	$0.93^\star$	$0.94^\star$	0.32	0.84	0.97	0.95	0
0.5	$0.94^\star$	$0.96^\star$	$0.95^\star$	0.40	0.74	0.87	0.82	0
1	$0.97^\star$	$0.93^\star$	$0.94^\star$	0.36	0.44	0.63	0.58	0

4.5 Experimental Study

We now investigate the performance of our heterogeneity metric on a variety of real single-particle tracking experiments described in Table 4.4. For each experiment, Table 4.4 reports

the theoretical diffusivity D, the interobservation time Δt, the number of observations per trajectory N, the type of camera and particle tracking software. All tracked particles are inert polystyrene beads of diameter $d = 1\,\mu\text{m}$.

Table 4.4: Summary of experimental conditions for various single-particle tracking experiments. The different types of fluids are water (H2O), glycerol (GLY), polyethilene oxide (PEO) and mucus from human bronchial ephithelia cell cultures (HBE). The subscripts correspond to sampling frequency for H2O, percent concentration for GLY, and percent weight (wt%) for HBE and PEO. The two types of cameras are Flea3 USB 3.0 [Flea3: FLIR, 2019] and Panoptes [Pan: CISMM, 2019a]. The particle tracking software employed is either Video Spot Tracker [VS: CISMM, 2019b] or Net Tracker [Net: Newby et al., 2018]

Medium	Name	D	Δt (s)	N	Camera	Software
Homogeneous	$H2O_{15}$	0.43	1/15	1800	Flea3	Net
	$H2O_{30}$	0.43	1/30	1800	Flea3	Net
	$H2O_{60}$	0.43	1/60	1800	Flea3	Net
	$H2O_{60b}$	0.43	1/60	1800	Flea3	VS
	GLY_{60}	0.09	1/60	1800	Flea3	VS
	GLY_{80}	0.022	1/60	1800	Flea3	VS
	$PEO_{0.22}$	-	1/38.17	1145	Flea3	VS
	$PEO_{0.3}$	-	1/38.17	1145	Flea3	VS
	$PEO_{0.45}$	-	1/38.17	1145	Flea3	VS
	$PEO_{0.6}$	-	1/38.17	1145	Flea3	VS
	$PEO_{0.75}$	-	1/38.17	1145	Flea3	VS
	$PEO_{0.9}$	-	1/38.17	1145	Flea3	VS
	$PEO_{1.22}$	-	1/38.17	1145	Flea3	VS
Heterogeneous	$HBE_{1.5}$	-	1/60	1800	Pan	VS
	HBE_{2}	-	1/60	1800	Pan	VS
	$HBE_{2.5}$	-	1/60	1800	Pan	VS
	HBE_{3}	-	1/60	1800	Pan	VS
	HBE_{4}	-	1/60	1800	Pan	VS
	HBE_{5}	-	1/60	1800	Pan	VS

4.5.1 Particle Trajectories in Water

The first three experiments are conducted in water, for which the medium is theoretically homogeneous. However the scatter plot of estimated (α, D) in Figure 4.6 suggests that there exists more than one cluster, where the estimated parameters in "majority cluster" is unbiased and the members of "minority cluster" significantly underestimate the diffusivity D. With the EM algorithm in Section 4.3, we successfully separate the clusters and mark their members with different colors in Figure 4.6. Due to the lack of some critical information including the signal-to-noise ratio (SNR), we fail to find a valid explanation for these universal existence of the low-diffusivity cluster in Net-Tracker data. By marking the trajectories in the minor cluster as "polluted by unknown noises" and excluding them from further analyses, we can investigate the degree of heterogeneity in water data by computing the metric B based on the majority cluster trajectories, which is demonstrated in Table 4.5.

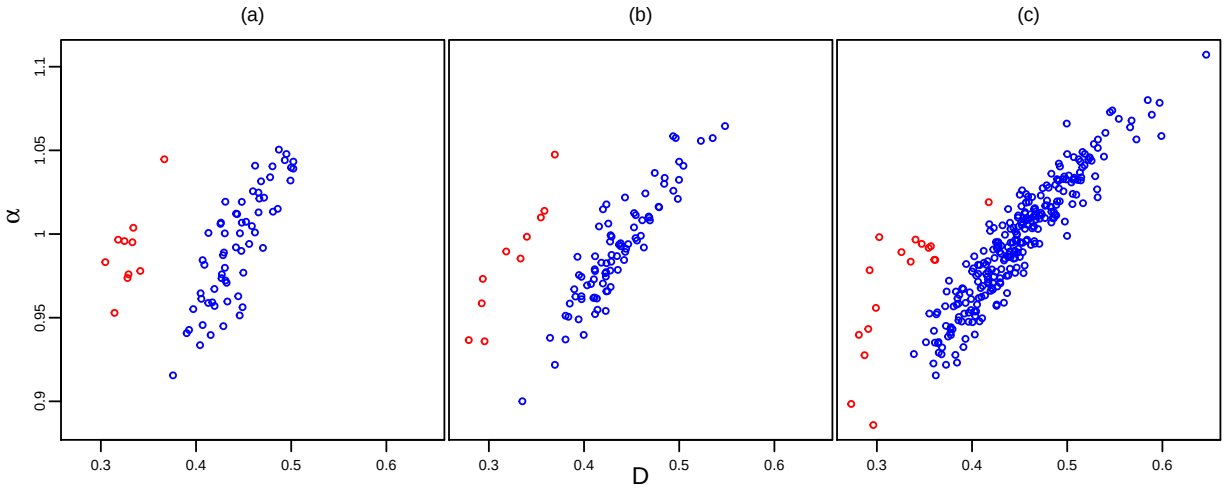


Figure 4.6: Estimated subdiffusive parameters (α, D), water data.
(a) $H2O_{15}$ medium (b) $H2O_{30}$ medium (c) $H2O_{60}$ medium

The heterogeneity metrics for Net-Tracker data are all of high value, and all pass the homogeneity test. From the metric we can see that $H2O_{60}$ data is less homogeneous than

the lower-frequency cases because of some outliers in the estimated (α, D) in $\mathrm{H2O}_{60}$ data.

Table 4.5: Estimated heterogeneity metric, particles in water.
For each experiment, the metric that passes the homogeneity test is marked with *

	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
$\mathrm{H2O}_{15}$	$0.98^\star$	$0.97^\star$	$0.97^\star$	0.87
$\mathrm{H2O}_{30}$	$1^\star$	$0.96^\star$	$0.97^\star$	0.75
$\mathrm{H2O}_{60}$	$0.92^\star$	$0.94^\star$	$0.93^\star$	0.67

4.5.2 Particle Trajectories in Viscous Media

The following ten experiments in Table 4.4 are tracked with the VS algorithm, and they are conducted in three homogeneous media. The first three data ($\mathrm{H2O}_{60b}$, GLY_{60} and GLY_{80}) record particle trajectories in water and glycerol solutions of different concentrations, with their estimated subdiffusive parameters (α, D) demonstrated in Figure 4.7, where the dispersion in estimation is most obvious for GLY_{80} data.

The computed metric B in Table 4.6 is consistent with the information in Figure 4.7, that $\mathrm{H2O}_{60b}$ and GLY_{60} data are more homogeneous than GLY_{80}. For $\mathrm{H2O}_{60b}$ and GLY_{60} their $B(\alpha)$ is of high value, while the $B(D)$ is too low to pass the homogeneity test. And their $B(\alpha, D)$ is dominated by the heterogeneity in D and therefore rejects the homogeneous hypothesis. For GLY_{80} data, both estimated subdiffusive parameters $(\hat{\alpha}, \hat{D})$ contain huge amount of dispersion and thus have a heterogeneity metric lower than 0.7. Considering that the medium of the experiment (glycerol solution) is theoretically homogeneous, there must exist some instrumental or experimental errors for the unexpected heterogeneity.

PEO is a synthetic polymer solution which should also be homogeneous. In Figure 4.8 we demonstrate the estimated (α, D) from the fMA model and color them separately according to the weight percentage of solution, where the dispersion in estimated (α, D) is almost consistent for all data concentrations. In Table 4.7 we show the heterogeneity metric for PEO experiments, where data of all concentrations are tested to be heterogeneous.

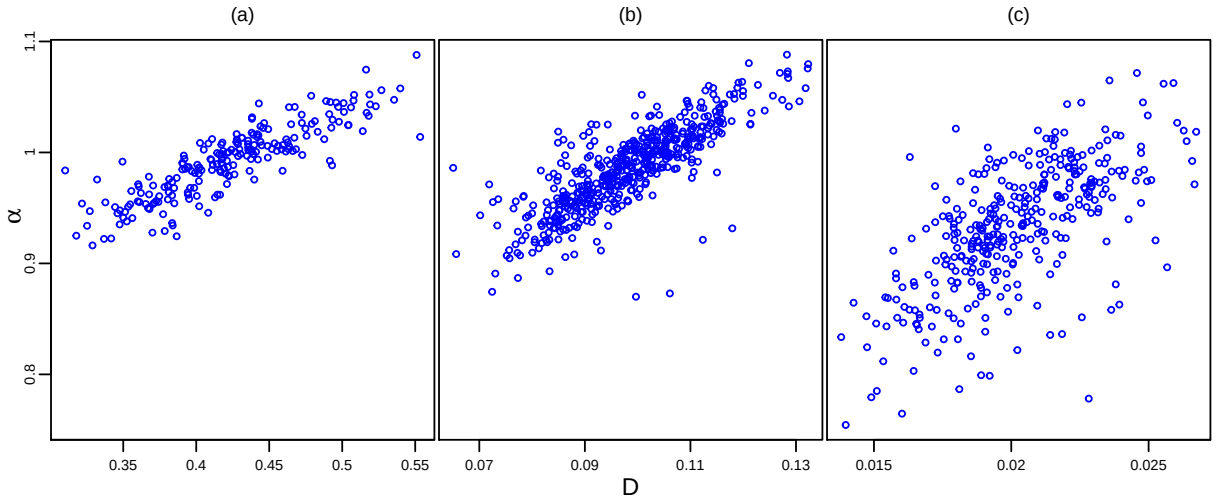


Figure 4.7: Estimated subdiffusive parameters (α, D), VS-Trackers water and glycerol solution data.
(a) $\mathrm{H2O}_{60b}$ medium (b) GLY_{60} medium (c) GLY_{80} medium

Table 4.6: Estimated heterogeneity metric, particles in water and glycerol solution.
For each experiment, the metric that passes the homogeneity test is marked with *

	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
$\mathrm{H2O}_{60b}$	$1^\star$	0.80	0.85	0.02
GLY_{60}	$0.91^\star$	0.85	0.86	0.01
GLY_{80}	0.56	0.62	0.60	0

The heterogeneity metric for different concentrations are similar, except for the $\text{PEO}_{0.9}$ case where some outliers exist. In addition, we find that the diffusivity D is much more heterogeneous than α.

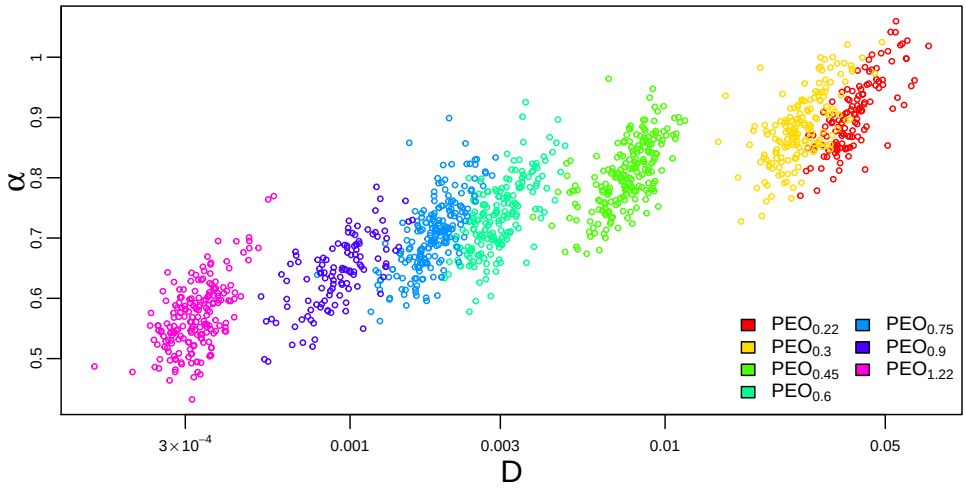


Figure 4.8: Estimated subdiffusive parameters (α, D), PEO data.

Table 4.7: Estimated heterogeneity metric, particles in PEO media.
For each experiment, the metric that passes the homogeneity test is marked with *

	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
$\text{PEO}_{0.22}$	0.71	0.39	0.49	0
$\text{PEO}_{0.3}$	0.81	0.35	0.47	0
$\text{PEO}_{0.45}$	0.71	0.29	0.41	0
$\text{PEO}_{0.6}$	0.71	0.29	0.42	0
$\text{PEO}_{0.75}$	0.68	0.29	0.42	0
$\text{PEO}_{0.9}$	0.75	0.34	0.48	0
$\text{PEO}_{1.22}$	0.82	0.23	0.44	0

Generally speaking, for all the homogeneous experiments with particle trajectories tracked by the VS algorithm, the experimental data are somehow heterogeneous, and

the PEO media turn out to be more heterogeneous than water or glycerol solution. The universal existence of impurities and errors in experimental procedures can be the reason for the unexpected heterogeneity in all media. However, considering that all these experiments, together with the water data tracked by Net Tracker in the previous section that turns out to be homogeneous, are conducted in similar environments (from the same laboratory) and are recorded with the same camera (Flea3), we here make a conjecture about the performance of the VS-tracker algorithm, that this algorithm may introduce extra heterogeneity in observed particle trajectories, especially in diffusivity D.

4.5.3 Particle Trajectories in Biological Fluids

The HBE data contains the trajectories of particles diffusing in the mucus harvested from primary human bronchial epithelial cell cultures [Hill et al., 2014]. The media contains washings from cultures which are concentrated to desired weight percent solids (wt%). As biological fluids, HBE environment is expected to be heterogeneous. In Figure 4.9 we demonstrate the estimated subdiffusive parameters (α, D) in transformed scale, where their distribution and dispersion vary significantly across the concentrations of media.

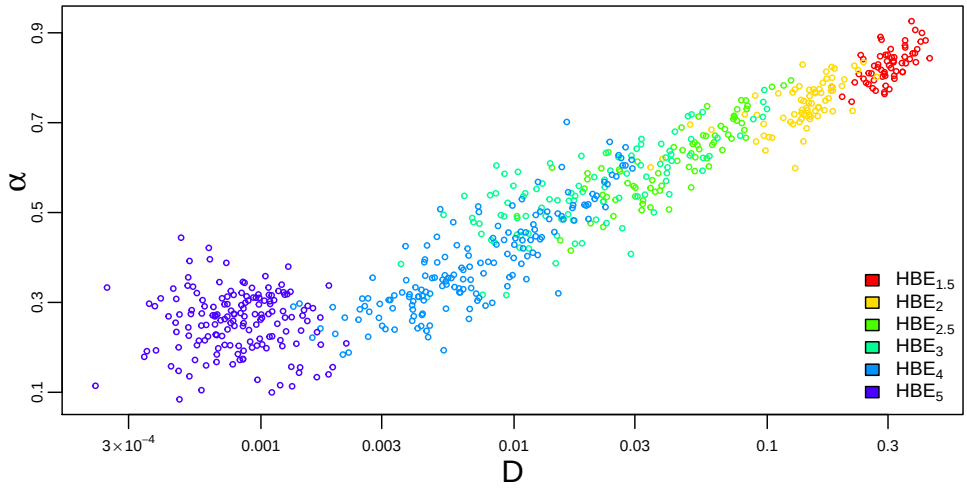


Figure 4.9: Estimated subdiffusive parameters (α, D), HBE data.

In Table 4.8 we show the heterogeneity metric of HBE experiment results. For the 1.5 wt% case where the concentration of solids in lung mucus is smallest, its α has a metric over 0.9 and is tested to be homogeneous. In addition, its diffusivity D also comes with the highest $B(D)$. As the weight percent solids increases from 2 wt% to 5 wt%, the heterogeneity metric drops rapidly to a very small value. Such a level of heterogeneity metric almost implies that the estimated parameters are evenly distributed around some areas, instead of clustering around a certain point. Since lower concentrations of solids in lung mucus have been associated with health states and high concentrations are related to disease conditions, the negative relation between the concentration and the heterogeneity metric indicates that healthy people's HBE mucus is more homogeneous than that of ill people.

Table 4.8: Estimated heterogeneity metric, particles in HBE media.
For each experiment, the metric that passes the homogeneity test is marked with *

	$B(\alpha)$	$B(D)$	$B(\alpha, D)$	p_{LRT}
$\text{HBE}_{1.5}$	$0.91^\star$	0.30	0.40	0
HBE_2	0.43	0.06	0.10	0
$\text{HBE}_{2.5}$	0.15	0.02	0.04	0
HBE_3	0.13	0.01	0.05	0
HBE_4	0.14	0.01	0.04	0
HBE_5	0.10	0.01	0.02	0

4.6 Discussion

In this chapter we present a metric B that quantifies the heterogeneity of the media in particle tracking experiments. This metric B is based on a hierarchical model that naturally induces the between-path particle heterogeneity. We also propose an estimator for particle trajectory parameters estimated using the location-scale model. In addition, we present a hypothesis test based on the metric B for the null hypothesis of homogeneity, together with a likelihood ratio test that serves a similar purpose. With extensive numerical simulations,

we can see the relation between the estimated metric and the degree of heterogeneity. Empirically a homogeneous parameter will have a metric over 0.9, and parameters with lower metric are most likely to be heterogeneous. In the analyses of particles diffusing in the mucus from human lungs, we find a possible relation between the heterogeneity of mucus and their disease states, which deserves further investigation. In addition, we categorize the general idea of "heterogeneity" into two kinds: large variance and multiple clusters. With an EM algorithm, we figure out that the original Net Tracker data looks heterogeneous because of the existence of multiple clusters.

There still exist some limitations in the proposed metric. For ML estimates, the numerator of the metric B is the weighted average of the inverse Fisher information $\boldsymbol{F}_i^{-1}$, which is proportional to $\frac{1}{N}$ where N is the maximum length of particle trajectories. As N goes to infinity, the Fisher information also goes to infinity and the heterogeneity metric B will inevitably converge to 0 while the actual degree of heterogeneity in the experimental medium remains unchanged. To better understand this problem, we here modify the simulation study in Section 4.4.1 by generating particle trajectories of different lengths and estimating their metric B. In Figure 4.10 we demonstrate the change of estimated B as the experimental time increases from $1s$ ($N = 60$) to $1000s$ ($N = 6 \times 10^4$). One possible solution is to determine a fixed trajectory length, say $N^\star$, and adjust the Fisher information $\boldsymbol{F}_i$ with factor $\frac{N^\star}{N}$ such that the estimated metric B scales similarly for the same medium.

As is revealed in the model misspecification simulations, the proposed metric is sensitive to the applied model, and will return untrustworthy results when incorrect models are used to estimate subdiffusive parameters (α, D). As a result we must ensure the correctness of particle trajectory models before using the metric to quantify the heterogeneity in the fluids. In addition, the model that returns unbiased estimates but incorrect Fisher information (for example the fSD model (3.2.7) when α is around 1) can be unsuitable for this heterogeneity metric.

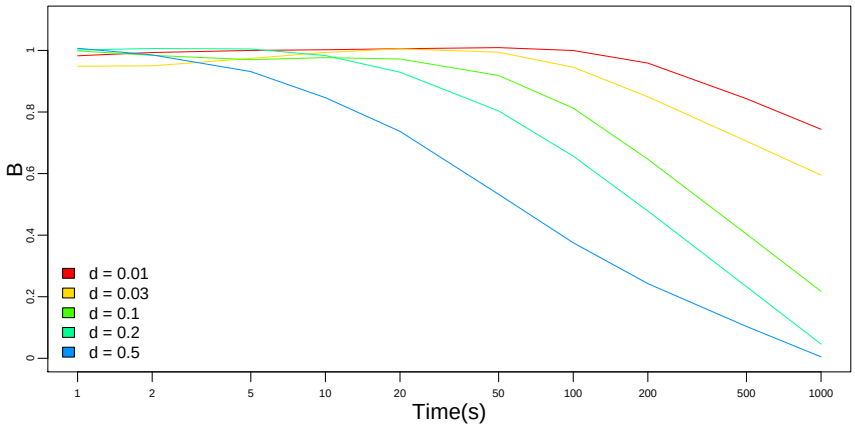


Figure 4.10: Estimated heterogeneity metric B for different particle trajectory lengths, simulated fBM processes.
For different particle trajectory lengths and different d, we simulated $M = 1000$ trajectories from model (4.4.1) and compute the corresponding metric B.

Chapter 5

Future Work

This thesis provides a superfast toolkit for the analyses of passive particle tracking data, which consists of three components, namely the superfast inference for stationary Gaussian processes, the parametric method that filters out the high-frequency noises in experimental data and the heterogeneity metric based on unbiased parametric estimates. This chapter discusses some future work on each component.

5.1 Future Work of Chapter 2

In Chapter 2 we provide the superfast inference for stationary Gaussian processes, where the term "stationary" only applies to the 1-dimensional time series. In order to generalize the proposed method to multi-dimensional cases we actually define a time series with properties "stationary" and "separable" (2.3.1). In a more general setup for multi-dimensional time series, they have an isotropic correlation function and a covariance matrix in forms of block Toeplitz with Toeplitz blocks, which is much more complex than the Toeplitz matrix we look into in this chapter. If we are able extend the superfast algorithm to isotropic Gaussian processes, we will greatly boost the computational efficiency for relative analyses, including the study of spatial lattice data [Stroud et al., 2016, Guinness and Fuentes, 2017].

In addition, for a one-dimensional mean-zero time series $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V})$, its residuals can be generated through the inverse of the Cholesky decomposition of its Toeplitz covariance matrix $\boldsymbol{V}$

$$\boldsymbol{z} = \boldsymbol{L}^{-1}\boldsymbol{x}$$

where $\boldsymbol{L}$ is the Cholesky decomposition of $\boldsymbol{V} = \boldsymbol{L}'\boldsymbol{L}$. Currently this step does not follow immediately from Ammar-Gragg's superfast algorithm, because we need the inverse of the Cholesky factor $\boldsymbol{L}^{-1}$ to compute the residuals $\boldsymbol{z}$ while the generalized Schur algorithm only provides the inverse of the Toeplitz matrix $\boldsymbol{V}^{-1}$. So far we are using the $\mathcal{O}(N^2)$ Levinson's algorithm for the inverse Cholesky decomposition, and a superfast algorithm that achieves this purpose will greatly improve the computational efficiency of relative statistical inferences.

5.2 Future Work of Chapter 3

In Chapter 3 we propose an ARMA filter to correct the high-frequency noises in particle tracking data. For given filter order p and q, the model coefficients of the ARMA(p, q) filter can be estimated efficiently. However, when multiple combinations of filter orders (p, q) are applied, it is still unclear which model has the best performance in estimating subdiffusive parameters (α, D). The common methods in model comparison are usually based on the likelihood, which places too much weight on the high-frequency domain and overrates the models that are distorted by high-frequency errors and departs from empirical observation in the long term. So far we tried a non-parametric measurement that quantifies the overall departure of model-estimated MSD from empirical observations in Section 3.5.2, while a parametric comparison method that serves a similar purpose will be of great help.

For a given time series where an ARMA(p, q) model applies, the order of the ARMA(p, q) model can be determined conveniently by checking the ACF and PACF [McLeod and Li, 1983]. For a given particle trajectory, we do not have a reliable procedure to determine the order of autoregressive and moving-average terms. One possible solution is related to the "model comparison" problem mentioned previously. With the model comparison method provided, we can at least try some different ARMA filters and use the best. In

conclusion, a working diagnosis procedure will greatly improve the validity of the proposed ARMA(p, q) filter.

5.3 Future Work of Chapter 4

In Chapter 4 we propose a simple metric that quantifies the degree of heterogeneity in experimental fluids. This metric basically measures the fraction of the variance that is unexplained by the Fisher information. As is described in Section 4.6, such metric will inevitable diminish as experimental duration increases. To avoid this, the coefficient of variation (CV) can be applied instead, which is the ratio between standard deviation and mean of estimated parameters. Although CV is conceptually appealing, it is not applicable for mean-zero coefficients (e.g. drift parameters in some experiments).

Consider the sub-linear power law of MSD

$$\text{MSD}(t \mid \alpha, D) = \tau(t) = 2D \cdot t^{\alpha}$$

where the MSDs $\tau(t)$ is a function of time t. We may characterize the heterogeneity of the experimental medium via the CV of $\tau(t)$. For a fixed time t, we have that $\tau(t)$ follows a log normal distribution where the mean and variance terms are all functions of (α, D). It is unclear how to construct a homogeneity test with null hypothesis $H_0 : \text{CV}(\tau) = 0$, while one possible solution is the parametric Bootstrap sampling similar to the one applied in the homogeneity test for multiple parameters $B(\boldsymbol{\theta})$.

In addition, the estimator of the metric B involves the variance of estimated parameters. However variance itself is quite sensitive to the existence of outliers, and an "unexpected" estimation can totally destroy the heterogeneity result of an experiment. As a result, a more robust version of a heterogeneity metric will be of great importance.

References

Hirotugu Akaike. Block toeplitz matrix inversion. *SIAM Journal on Applied Mathematics*, 24(2):234–241, 1973.

François Amblard, Anthony C Maggs, Bernard Yurke, Andrew N Pargellis, and Stanislas Leibler. Subdiffusion and anomalous local viscoelasticity in actin networks. *Physical review letters*, 77(21):4470, 1996.

Gregory S Ammar. Classical foundations of algorithms for solving positive definite toeplitz equations. *Calcolo*, 33(1-2):99–113, 1996.

Gregory S Ammar and William B Gragg. The generalized schur algorithm for the superfast solution of toeplitz systems. In *Rational approximation and its applications in mathematics and physics*, pages 315–330. Springer, 1987.

Gregory S Ammar and William B Gragg. Superfast solution of real positive definite toeplitz systems. *SIAM Journal on Matrix Analysis and Applications*, 9(1):61–76, 1988.

Gregory S Ammar and William B Gragg. Numerical experience with a superfast real toeplitz solver. *Linear Algebra and its Applications*, 121:185–206, 1989.

Cedric Archambeau, Dan Cornford, Manfred Opper, and John Shawe-Taylor. Gaussian process approximations of stochastic differential equations. *Journal of machine learning research*, 1:1–16, 2007.

Trevor T Ashley and Sean B Andersson. Method for simultaneous localization and parameter estimation in particle tracking experiments. *Physical Review E*, 92(5):052707, 2015.

Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000.

Sudipto Banerjee, Bradley P Carlin, and Alan E Gelfand. *Hierarchical modeling and analysis for spatial data*. Crc Press, 2014.

Erwin H Bareiss. Numerical solution of linear equations with toeplitz and vector toeplitz matrices. *Numerische Mathematik*, 13(5):404–424, 1969.

Andrew J Berglund. Statistics of camera-based single-particle tracking. *Physical Review E*, 82(1):011917, 2010.

Simeon M Berman. Gaussian processes with stationary increments: Local times and sample function properties. *The Annals of Mathematical Statistics*, pages 1260–1272, 1970.

Robert R Bitmead and Brian DO Anderson. Asymptotically fast solution of toeplitz and related systems of linear equations. *Linear Algebra and its Applications*, 34:103–116, 1980.

Adam W Bojanczyk, Richard P Brent, Frank R De Hoog, and Douglas R Sweet. On the stability of the bareiss and related toeplitz factorization algorithms. *SIAM Journal on Matrix Analysis and Applications*, 16(1):40–57, 1995.

AW Bojanczyk, RP Brent, and FR De Hoog. Stability analysis of fast toeplitz linear system solvers. Technical report, Report CMA-MR17-91, Centre for Mathematical Analysis, ANU, 1991.

George EP Box. Robustness in the strategy of scientific model building. In *Robustness in statistics*, pages 201–236. Elsevier, 1979.

F Jay Breidt, Nuno Crato, and Pedro De Lima. The detection and estimation of long memory in stochastic volatility. *Journal of econometrics*, 83(1):325–348, 1998.

Richard P Brent, Fred G Gustavson, and David YY Yun. Fast solution of toeplitz systems of equations and computation of padé approximants. *Journal of Algorithms*, 1(3):259–295, 1980.

Peter J Brockwell and Richard A Davis. *Time Series: Theory and Methods*. Springer-Verlag, New York, 1991.

Peter J Brockwell, Richard A Davis, and Matthew V Calder. *Introduction to time series and forecasting*, volume 2. Springer, 2002.

I Bronstein, Y Israel, E Kepten, S Mai, Yaron Shav-Tal, E Barkai, and Y Garini. Transient anomalous diffusion of telomeres in the nucleus of mammalian cells. *Physical review letters*, 103(1):018102, 2009.

Roelof K Brouwer. *Particular methods for solving particular systems of linear equations*. University of Waterloo., 1971.

James R Bunch. Stability of methods for solving toeplitz systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 6(2):349–364, 1985.

Stanislav Burov, Patrick Figliozzi, Binhua Lin, Stuart A Rice, Norbert F Scherer, and Aaron R Dinner. Single-pixel interior filling function approach for detecting and correcting errors in particle tracking. *Proceedings of the National Academy of Sciences*, 114(2): 221–226, 2017.

M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems*, pages 1881–1888, 2009.

Ben Calderhead, Mark Girolami, and Neil D Lawrence. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2009.

Christopher P Calderon. Motion blur filtering: A statistical approach for extracting confinement forces and diffusivity from a single blurred trajectory. *Physical Review E*, 93 (5):053303, 2016.

Shiv Chandrasekaran, Ming Gu, X Sun, J Xia, and J Zhu. A superfast algorithm for toeplitz systems of linear equations. *SIAM Journal on Matrix Analysis and Applications*, 29(4): 1247–1266, 2007.

Shivkumar Chandrasekaran and Ali H Sayed. Stabilizing the generalized schur algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):950–983, 1996.

Shivkumar Chandrasekaran and Ali H Sayed. A fast stable solver for nonsymmetric toeplitz and quasi-toeplitz systems of linear equations. *SIAM Journal on Matrix Analysis and Applications*, 19(1):107–139, 1998.

Michael K Cheezum, William F Walker, and William H Guilford. Quantitative comparison of algorithms for tracking single fluorescent particles. *Biophysical journal*, 81(4):2378–2388, 2001.

Jiahua Chen, Pengfei Li, et al. Hypothesis test for normal mixture models: The em approach. *The Annals of Statistics*, 37(5A):2523–2542, 2009.

Willa W Chen, Clifford M Hurvich, and Yi Lu. On the correlation matrix of the discrete fourier transform and the fast solution of large toeplitz systems for long-memory time series. *Journal of the American Statistical Association*, 101(474):812–822, 2006.

Li-Chiun Cheng, Lilian C Hsiao, and Patrick S Doyle. Multiple particle tracking study of thermally-gelling nanoemulsions. *Soft matter*, 13(37):6606–6619, 2017.

Nicolas Chenouard, Ihor Smal, Fabrice de Chaumont, Martin Maška, Ivo F Sbalzarini, Yuanhao Gong, Janick Cardinale, Craig Carthel, Stefano Coraluppi, Mark Winter, Andrew R Cohen, William J Godinez, Karl Rohr, Yannis Kalaidzidis, Liang Liang, James Duncan, Hongying Shen, Yingke Xu, Klas E G Magnusson, Joakim Jaldén, Helen M Blau, Perrine Paul-Gilloteaux, Philippe Roudot, Charles Kervrann, François Waharte, Jean-Yves Tinevez, Spencer L Shorte, Joost Willemse, Katherine Celler, Gilles P van Wezel, Han-Wei Dan, Yuh-Show Tsai, Carlos Ortiz de Solórzano, Jean-Christophe Olivo-Marin, and Erik Meijering. Objective comparison of particle tracking methods. *Nature methods*, 11(3):281, 2014.

CISMM. Camera panoptes. http://cismm.web.unc.edu/core-projects/force-microscopy/high-throughput-microscopy, 2019a. [Online; Accessed: 2019-06-13].

CISMM. Video spot tracker. http://cismm.web.unc.edu/resources/software-manuals/video-spot-tracker-manual, 2019b. [Online; Accessed: 2019-06-13].

Nicolas Courtois, Louis Goubin, Willi Meier, and Jean-Daniel Tacier. Solving underdefined systems of multivariate quadratic equations. In *International Workshop on Public Key Cryptography*, pages 211–227. Springer, 2002.

John C Crocker and David G Grier. Methods of digital video microscopy for colloidal studies. *Journal of colloid and interface science*, 179(1):298–310, 1996.

George Cybenko. The numerical stability of the levinson-durbin algorithm for toeplitz systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 1(3):303–319, 1980.

Frank de Hoog. A new algorithm for solving toeplitz systems of equations. *Linear Algebra and its Applications*, 88:123–138, 1987.

Frank Robert De Hoog. *On the solution of Toeplitz systems of equations*. Australian National University, Centre for Mathematical Analysis, 1984.

Philippe Delsarte and Y Genin. Spectral properties of finite toeplitz matrices. In *Mathematical theory of networks and systems*, pages 194–213. Springer, 1984.

Rohit Deo, Clifford Hurvich, and Yi Lu. Forecasting realized volatility using a long-memory stochastic volatility model: estimation, prediction and seasonal adjustment. *Journal of Econometrics*, 131(1):29–58, 2006.

Hendrik Deschout, Francesca Cella Zanacchi, Michael Mlodzianoski, Alberto Diaspro, Joerg Bewersdorf, Samuel T Hess, and Kevin Braeckmans. Precisely and accurately localizing single emitters in fluorescence microscopy. *Nature methods*, 11(3):253, 2014.

J. Durbin. The fitting of time series models. *Review of the International Statistical Institute*, 28:233 – 243, 1960.

John T Edward. Molecular volumes and the stokes-einstein equation. *Journal of Chemical Education*, 47(4):261, 1970.

Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956.

Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 201–208. ACM, 2005.

Matthias Ernst, Thomas John, Marco Guenther, Christian Wagner, Ulrich F Schaefer, and Claus-Michael Lehr. A model for the transient subdiffusive behavior of particles in mucus. *Biophysical Journal*, 112(1):172–179, 2017.

John D Ferry. *Viscoelastic properties of polymers*. John Wiley & Sons, 1980.

FLIR. Camera flear usb3.0. https://www.ptgrey.com/products/flea3-usb3, 2019. [Online; Accessed: 2019-06-13].

Erika J Fong, Yasha Sharma, Brian Fallica, Dylan B Tierney, Sarah M Fortune, and Muhammad H Zaman. Decoupling directed and passive motion in dynamic systems: particle tracking microrheology of sputum. *Annals of biomedical engineering*, 41(4): 837–846, 2013.

David A Freedman. On the so-called "huber sandwich estimator" and "robust standard errors". *The American Statistician*, 60(4):299–302, 2006.

Roland W Freund. Look-ahead schur-type algorithm for solving general toeplitz systems. In *SPIE's 1994 International Symposium on Optics, Imaging, and Instrumentation*, pages 482–493. International Society for Optics and Photonics, 1994.

Matteo Frigo and Steven G Johnson. The design and implementation of fftw3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

Naama Gal, Diana Lechtman-Goldstein, and Daphne Weihs. Particle tracking in living cells: A review of the mean square displacement method and beyond. *Rheologica Acta*, 52(5):425–443, 2013.

Ben NG Giepmans, Stephen R Adams, Mark H Ellisman, and Roger Y Tsien. The fluorescent toolbox for assessing protein location and function. *science*, 312(5771):217–224, 2006.

I Gohberg and A Semencul. On the inversion of finite toeplitz matrices and their continuous analogs. *Mat. issled*, 2:201–233, 1972.

Clive WJ Granger and Roselyne Joyeux. An introduction to long-memory time series models and fractional differencing. *Journal of time series analysis*, 1(1):15–29, 1980.

Joseph Guinness and Montserrat Fuentes. Circulant embedding of approximate covariances for inference from gaussian data on large lattices. *Journal of computational and Graphical Statistics*, 26(1):88–97, 2017.

JA Hartigan. A failure of likelihood asymptotics for normal mixtures. In *Proc. Barkeley Conference in Honor of J. Neyman and J. Kiefer*, volume 2, pages 807–810, 1985.

Andrew C Harvey. Long memory in stochastic volatility. In *Forecasting volatility in the financial markets*, pages 307–320. Elsevier, 2002.

Nicholas J Higham. *Accuracy and stability of numerical algorithms*. Siam, 2002.

David B Hill, Paula A Vasquez, John Mellnik, Scott A McKinley, Aaron Vose, Frank Mu, Ashley G Henderson, Scott H Donaldson, Neil E Alexis, Richard C Boucher, et al. A biophysical basis for mucus solids concentration as a candidate biomarker for airways disease. *PloS one*, 9(2):e87681, 2014.

Felix Höfling and Thomas Franosch. Anomalous transport in the crowded world of biological cells. *Reports on Progress in Physics*, 76(4):046602, 2013.

Jonathan RM Hosking. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

HA Houghton, IA Hasnain, and AM Donald. Particle tracking to reveal gelation of hectorite dispersions. *The European Physical Journal E*, 25(2):119–127, 2008.

M Jankowski and H Woźniakowski. Iterative refinement implies numerical stability. *BIT Numerical Mathematics*, 17(3):303–311, 1977.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Thomas Kailath and Ali H Sayed. *Fast reliable algorithms for matrices with structure*. Siam, 1999a.

Thomas Kailath and Ali H Sayed. *Fast reliable algorithms for matrices with structure*. SIAM, 1999b.

Thomas Kailath, Sun-Yuan Kung, and Martin Morf. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications*, 68(2):395–407, 1979.

Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnovic. Power law and exponential decay of intercontact times between mobile devices. *IEEE Transactions on Mobile Computing*, 9(10):1377–1390, 2010.

Elena F Koslover, Caleb K Chan, and Julie A Theriot. Disentangling random motion and flow in a complex medium. *Biophysical journal*, 110(3):700–709, 2016.

Samuel C Kou. Stochastic modeling in nanoscale biophysics: subdiffusion within proteins. *The Annals of Applied Statistics*, 2(2):501–535, 2008.

Anne Kowalczyk, Claude Oelschlaeger, and Norbert Willenbacher. Tracking errors in 2d multiple particle tracking microrheology. *Measurement Science and Technology*, 26(1): 015302, 2014.

Peter Kravanja and Marc Van Barel. Coupled vandermonde matrices and the superfast computation of toeplitz determinants. *Numerical Algorithms*, 24(1-2):99–116, 2000.

R. Kubo. The fluctutation-dissipation theorem. *Reports on Progress in Physics*, 29:255–284, 1966.

Samuel K Lai, D Elizabeth O'Hanlon, Suzanne Harrold, Stan T Man, Ying-Ying Wang, Richard Cone, and Justin Hanes. Rapid transport of large polymeric nanoparticles in fresh undiluted human mucus. *Proceedings of the National Academy of Sciences*, 104(5): 1482–1487, 2007.

Samuel K Lai, Ying-Ying Wang, and Justin Hanes. Mucus-penetrating nanoparticles for drug and gene delivery to mucosal tissues. *Advanced drug delivery reviews*, 61(2):158–171, 2009.

Sang-Hyuk Lee, Yohai Roichman, Gi-Ra Yi, Shin-Hyun Kim, Seung-Man Yang, Alfons Van Blaaderen, Peter Van Oostrum, and David G Grier. Characterizing and tracking single colloidal particles with video holographic microscopy. *Optics Express*, 15(26): 18275–18282, 2007.

Evgeny Levi and Radu V Craiu. Bayesian inference for conditional copulas using gaussian process single index models. *Computational Statistics & Data Analysis*, 122:115–134, 2018.

N. Levinson. The Wiener RMS error criterion in filter design and prediction. *Journal Of Mathematical Physics*, 25:261 – 278, 1947.

Norman Levinson. The wiener (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 25(1):261–278, 1946.

Bruce G Lindsay. Composite likelihood methods. *Contemporary Mathematics*, 80(1):221–239, 1988.

Yun Ling and Martin Lysy. *SuperGauss: Superfast Likelihood Inference for Stationary Gaussian Time Series*, 2017. URL https://CRAN.R-project.org/package=SuperGauss. R package version 1.0.

Martin Lysy, Natesh S Pillai, David B Hill, M Gregory Forest, John WR Mellnik, Paula A Vasquez, and Scott A McKinley. Model comparison and assessment for single particle

tracking in biological fluids. *Journal of the American Statistical Association*, 111(516): 1413–1426, 2016.

D Makarov. The lower bound for the number of multiplications for algorithms for calculating the product of hankel matrices. *USSR Comput. Math. and Math. Phys*, 17:195–199, 1978.

Douglas S Martin, Martin B Forstner, and Josef A Käs. Apparent subdiffusion inherent to single particle tracking. *Biophysical journal*, 83(4):2109–2117, 2002.

TG Mason, K Ganesan, JH Van Zanten, D Wirtz, and SC Kuo. Particle tracking microrheology of complex fluids. *Physical Review Letters*, 79(17):3282, 1997.

Thomas G Mason and DA Weitz. Optical measurements of frequency-dependent linear viscoelastic moduli of complex fluids. *Physical review letters*, 74(7):1250, 1995.

Hirotoshi Matsui, Margrith W Verghese, Mehmet Kesimer, Ute E Schwab, Scott H Randell, John K Sheehan, Barbara R Grubb, and Richard C Boucher. Reduced three-dimensional motility in dehydrated airway mucus prevents neutrophil capture and killing bacteria on airway epithelial surfaces. *The journal of immunology*, 175(2):1090–1099, 2005.

Scott A McKinley, Lingxing Yao, and M Gregory Forest. Transient anomalous diffusion of tracer particles in soft matter. *Journal of Rheology*, 53(6):1487–1506, 2009.

Allan I McLeod and William K Li. Diagnostic checking arma time series models using squared-residual autocorrelations. *Journal of Time Series Analysis*, 4(4):269–273, 1983.

Erik Meijering, Oleh Dzyubachyk, and Ihor Smal. Methods for cell and particle tracking. In *Methods in enzymology*, volume 504, pages 183–200. Elsevier, 2012.

John Mellnik, Paula A Vasquez, Scott A McKinley, Jacob Witten, David B Hill, and M Gregory Forest. Micro-heterogeneity metrics for diffusion in soft matter. *Soft matter*, 10(39):7781–7796, 2014.

John WR Mellnik, Martin Lysy, Paula A Vasquez, Natesh S Pillai, David B Hill, Jeremy Cribb, Scott A McKinley, and M Gregory Forest. Maximum likelihood estimation for

single particle, passive microrheology data with drift. *Journal of Rheology*, 60(3):379–392, 2016.

Ralf Metzler, Jae-Hyung Jeon, Andrey G Cherstvy, and Eli Barkai. Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking. *Physical Chemistry Chemical Physics*, 16(44):24128–24164, 2014.

Xavier Michalet. Mean square displacement analysis of single-particle trajectories with localization error: Brownian motion in an isotropic medium. *Physical Review E*, 82(4): 041914, 2010.

Xavier Michalet and Andrew J Berglund. Optimal diffusion coefficient estimation in single-particle tracking. *Physical Review E*, 85(6):061916, 2012.

WE Moerner. A dozen years of single-molecule spectroscopy in physics, chemistry, and biophysics, 2002.

Pedro A Morettin. The levinson algorithm and its applications in time series analysis. *International statistical review/revue internationale de statistique*, pages 83–92, 1984.

Martin Morf. Doubling algorithms for toeplitz and related equations. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80.*, volume 5, pages 954–959. IEEE, 1980.

Carl N Morris, Martin Lysy, et al. Shrinkage estimation in multilevel normal models. *Statistical Science*, 27(1):115–134, 2012.

Kim I Mortensen, L Stirling Churchman, James A Spudich, and Henrik Flyvbjerg. Optimized localization analysis for single-molecule tracking and super-resolution microscopy. *Nature methods*, 7(5):377, 2010.

Bruce R Musicus. *Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices*. Citeseer, 1988.

Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

Jay M Newby, Alison M Schaefer, Phoebe T Lee, M Gregory Forest, and Samuel K Lai. Deep neural networks automate detection for tracking of submicron scale particles in 2d and 3d. *arXiv preprint arXiv:1704.03009*, 2017.

Jay M Newby, Alison M Schaefer, Phoebe T Lee, M Gregory Forest, and Samuel K Lai. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2d and 3d. *Proceedings of the National Academy of Sciences*, 115(36):9026–9031, 2018.

Marcello Pagano. An algorithm for fitting autoregressive schemes. *Applied statistics*, pages 274–281, 1972.

Hong Qian, Michael P Sheetz, and Elliot L Elson. Single particle tracking. analysis of diffusion and flow in two-dimensional systems. *Biophysical journal*, 60(4):910–921, 1991.

A Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A): A405, 1964.

Christopher J Rowlands and Peter TC So. On the correction of errors in some multiple particle tracking experiments. *Applied physics letters*, 102(2):021913, 2013.

PG Saffman. Brownian motion in thin sheets of viscous fluid. *Journal of Fluid Mechanics*, 73(4):593–602, 1976.

Simo Särkkä, Jouni Hartikainen, Lennart Svensson, and Fredrik Sandblom. Gaussian process quadratures in nonlinear sigma-point filtering and smoothing. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2014.

Thierry Savin and Patrick S Doyle. Static and dynamic errors in particle tracking microrheology. *Biophysical journal*, 88(1):623–638, 2005.

Michael J Saxton. Single-particle tracking: the distribution of diffusion coefficients. *Biophysical journal*, 72(4):1744–1753, 1997.

Michael J Saxton and Ken Jacobson. Single-particle tracking: applications to membrane dynamics. *Annual review of biophysics and biomolecular structure*, 26(1):373–399, 1997.

Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004.

Harlan Sexton. Analysis of an algorithm of bitmead and anderson for the inversion of toeplitz systems. Technical report, DTIC Document, 1982.

Hao Shen, Lawrence J Tauzin, Rashad Baiyasi, Wenxiao Wang, Nicholas Moringo, Bo Shuang, and Christy F Landes. Single particle tracking: from theory to biophysical applications. *Chemical Reviews*, 117(11):7331–7376, 2017.

Grzegorz Sikora, Marek Teuerle, Agnieszka Wyłomańska, and Denis Grebenkov. Statistical properties of the anomalous scaling exponent estimator based on time-averaged mean-square displacement. *Physical Review E*, 96(2):022132, 2017.

Alex J Smola and Peter L Bartlett. Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625, 2001.

JE Soussou, F Moavenzadeh, and MH Gradowczyk. Application of prony series to linear viscoelasticity. *Transactions of the Society of Rheology*, 14(4):573–584, 1970.

Richard B Stein. A theoretical analysis of neuronal variability. *Biophysical Journal*, 5(2): 173–194, 1965.

Gilbert Stengle. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1974.

David J Stephens and Victoria J Allan. Light microscopy techniques for live cell imaging. *Science*, 300(5616):82–86, 2003.

Gilbert W Stewart. *Introduction to matrix computations*. Elsevier, 1973.

Michael Stewart. A superfast toeplitz solver with improved numerical stability. *SIAM journal on matrix analysis and applications*, 25(3):669–693, 2003.

Michael Stewart and Paul Van Dooren. Stability issues in the factorization of structured matrices. *SIAM Journal on Matrix Analysis and Applications*, 18(1):104–118, 1997.

Petre Stoica and Randolph L Moses. *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

Jonathan R Stroud, Michael L Stein, and Shaun Lysen. Bayesian and maximum likelihood estimation for gaussian processes on an incomplete lattice. *Journal of Computational and Graphical Statistics*, (just-accepted), 2016.

Junghae Suh, Michelle Dawson, and Justin Hanes. Real-time multiple-particle tracking: applications to drug and gene delivery. *Advanced drug delivery reviews*, 57(1):63–78, 2005.

Douglas Robin Sweet. *Numerical methods for Toeplitz matrices*. University of Adelaide, Department of Computer Science, University of Adelaide, 1982.

Jedrzej Szymanski and Matthias Weiss. Elucidating the origin of anomalous diffusion in crowded fluids. *Physical review letters*, 103(3):038102, 2009.

William F Trench. An algorithm for the inversion of finite toeplitz matrices. *Journal of the Society for Industrial and Applied Mathematics*, 12(3):515–522, 1964.

Yiider Tseng, Thomas P Kole, Ssu-Hsien J Lee, and Denis Wirtz. Local dynamics and viscoelastic properties of cell biological systems. *Current opinion in colloid & interface science*, 7(3):210–217, 2002.

Megan T Valentine, Peter D Kaplan, D Thota, John C Crocker, Thomas Gisler, Robert K Prud'homme, M Beck, and David A Weitz. Investigating the microenvironments of inhomogeneous soft materials with multiple particle tracking. *Physical Review E*, 64(6): 061506, 2001.

Marc Van Barel, Georg Heinig, and Peter Kravanja. A stabilized superfast solver for nonsymmetric toeplitz systems. *SIAM Journal on Matrix Analysis and Applications*, 23 (2):494–510, 2001.

Hilde M van der Schaar, Michael J Rust, Chen Chen, Heidi van der Ende-Metselaar, Jan Wilschut, Xiaowei Zhuang, and Jolanda M Smit. Dissecting the cell entry pathway of dengue virus by single-particle tracking in living cells. *PLoS pathogens*, 4(12):e1000244, 2008.

Cristiano Varin and Paolo Vidoni. A note on composite likelihood inference and model selection. *Biometrika*, 92(3):519–528, 2005.

Cristiano Varin, Nancy Reid, and David Firth. An overview of composite likelihood methods. *Statistica Sinica*, pages 5–42, 2011.

Christian L Vestergaard, Paul C Blainey, and Henrik Flyvbjerg. Optimal estimation of diffusion coefficients from single-particle trajectories. *Physical Review E*, 89(2):022726, 2014.

Ying-Ying Wang, Samuel K Lai, Jung Soo Suk, Amanda Pace, Richard Cone, and Justin Hanes. Addressing the peg mucoadhesivity paradox to engineer nanoparticles that "slip" through the human mucus barrier. *Angewandte Chemie International Edition*, 47(50): 9726–9729, 2008.

Stephanie C Weber, Michael A Thompson, WE Moerner, Andrew J Spakowitz, and Julie A Theriot. Analytical tools to distinguish the effects of localization error, confinement, and medium elasticity on the velocity autocorrelation function. *Biophysical journal*, 102(11): 2443–2450, 2012.

Daphne Weihs, Michael A Teitell, and Thomas G Mason. Simulations of complex particle transport in heterogeneous active liquids. *Microfluidics and Nanofluidics*, 3(2):227–237, 2007.

Matthias Weiss. Single-particle tracking data reveal anticorrelated fractional brownian motion in crowded fluids. *Physical Review E*, 88(1):010101, 2013.

Matthias Weiss, Markus Elsner, Fredrik Kartberg, and Tommy Nilsson. Anomalous subdiffusion is a measure for cytoplasmic crowding in living cells. *Biophysical journal*, 87(5): 3518–3524, 2004.

Peter Whittle. On stationary processes in the plane. *Biometrika*, pages 434–449, 1954.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Kim A Winick. Cramer-Rao lower bounds on the performance of charge-coupled-device optical position estimators. *Journal of the Optical Society of America A*, 3(11):1809–1815, 1986.

Denis Wirtz. Particle-tracking microrheology of living cells: principles and applications. *Annual review of biophysics*, 38:301–326, 2009.

IY Wong, ML Gardel, DR Reichman, Eric R Weeks, MT Valentine, AR Bausch, and David A Weitz. Anomalous diffusion probes microstructure dynamics of entangled f-actin networks. *Physical review letters*, 92(17):178101, 2004.

Peter K Working, Mary S Newman, Judy Johnson, and Joel B Cornacoff. Safety of poly (ethylene glycol) and poly (ethylene glycol) derivatives. ACS Publications, 1997.

Yuanzhe Xi, Jianlin Xia, Stephen Cauley, and Venkataramanan Balakrishnan. Superfast and stable structured solvers for toeplitz least squares via randomized sampling. *SIAM Journal on Matrix Analysis and Applications*, 35(1):44–72, 2014.

Jianlin Xia, Yuanzhe Xi, and Ming Gu. A superfast structured solver for toeplitz linear systems via randomized sampling. *SIAM Journal on Matrix Analysis and Applications*, 33(3):837–858, 2012.

Andrew E Yagle and Bernard C Levy. The schur algorithm and its applications. *Acta Applicandae Mathematica*, 3(3):255–284, 1985.

Ryoichi Yamamoto and Akira Onuki. Heterogeneous diffusion in highly supercooled liquids. *Physical review letters*, 81(22):4915, 1998.

Shalhav Zohar. Toeplitz matrix inversion: The algorithm of wf trench. *Journal of the ACM (JACM)*, 16(4):592–601, 1969.

R. Zwanzig. *Nonequilibrium Statistical Mechanics*. New York: Oxford University Press, 2001.

APPENDICES

Appendix A

Multiplication of Toeplitz Matrix and Vector

Assuming that $\boldsymbol{V}$ is a Toeplitz matrix of the following form

$$\boldsymbol{V} = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_N \\ \gamma_{-2} & \gamma_1 & \gamma_2 & \cdots & \gamma_{N-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \gamma_{-N} & \gamma_{1-N} & \gamma_{2-N} & \cdots & \gamma_1 \end{bmatrix},$$

and $\boldsymbol{x} = [x_1, x_2, \cdots, x_N]$ is a length-N vector. The normal computation of $\boldsymbol{V} \times \boldsymbol{x}$ requires $\mathcal{O}(N^2)$ steps, but with the Fast Fourier Transformation (FFT) [Kailath and Sayed, 1999b] we can achieve this in $\mathcal{O}(N \log N)$ steps.

We first generate a circulant embedding form of the Toeplitz matrix $\boldsymbol{V}$

$$\boldsymbol{V}_0 = \left[\begin{array}{ccccc|ccccc} \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_N & 0 & \gamma_{-N} & \gamma_{1-N} & \cdots & \gamma_{-2} \\ \gamma_{-2} & \gamma_1 & \gamma_2 & \cdots & \gamma_{N-1} & \gamma_N & 0 & \gamma_{-N} & \cdots & \gamma_{-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \cdots & \ddots \\ \gamma_{-N} & \gamma_{1-N} & \gamma_{2-N} & \cdots & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \cdots & 0 \\ \hline 0 & \gamma_{-N} & \gamma_{1-N} & \cdots & \gamma_{-2} & \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_N \\ \gamma_N & 0 & \gamma_{-N} & \cdots & \gamma_{-3} & \gamma_{-2} & \gamma_1 & \gamma_2 & \cdots & \gamma_{N-1} \\ \vdots & \vdots & \vdots & \cdots & \ddots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \gamma_2 & \gamma_3 & \gamma_4 & \cdots & 0 & \gamma_{-N} & \gamma_{1-N} & \gamma_{2-N} & \cdots & \gamma_1 \end{array}\right] = \left[\begin{array}{c|c} \boldsymbol{V} & \boldsymbol{M} \\ \hline \boldsymbol{M} & \boldsymbol{V} \end{array}\right],$$

and set $\boldsymbol{x}_0 = [\boldsymbol{x}, \boldsymbol{0}]'$ such that $\boldsymbol{x}_0$ is a length $2N$ vector. Then we have that

$$\boldsymbol{V}_0 \times \boldsymbol{x}_0 = [\boldsymbol{V} \times \boldsymbol{x}, \boldsymbol{M} \times \boldsymbol{x}]',$$

where the first half is exactly what we want. Since circulant matrix can be diagonalized by the Fourier matrix

$$\boldsymbol{V}_0 = \boldsymbol{F}_N^* \times \text{diag}(\boldsymbol{F}_N \times \boldsymbol{\gamma}) \times \boldsymbol{F}_N,$$

where $\boldsymbol{F}_N$ is a size N Fourier matrix and $\boldsymbol{F}_N^*$ is its conmugate transpose, $\boldsymbol{\gamma}$ is the first column of matrix $\boldsymbol{V}_0$ and the result of $\boldsymbol{F}_N \times \boldsymbol{\gamma}$ equals $\boldsymbol{\mathcal{F}\gamma}$, the fast Fourier transformation on $\boldsymbol{\gamma}$. Here we present a $\mathcal{O}(N \log N)$ complexity procedure in computing $\boldsymbol{y}_0 = \boldsymbol{V}_0 \times \boldsymbol{x}_0$

- compute $\boldsymbol{f} = \boldsymbol{\mathcal{F}\gamma}$,
- compute $\boldsymbol{g} = \boldsymbol{\mathcal{F}x}_0$,
- compute $\boldsymbol{h} = \boldsymbol{f} \cdot \boldsymbol{g}$ (element-wise product for vectors),
- compute $\boldsymbol{y}_0 = \boldsymbol{\mathcal{F}}^{-1}\boldsymbol{h}$.

By extracting the first N elements of $\boldsymbol{y}_0$ we can obtain the result of $\boldsymbol{V} \times \boldsymbol{x}$.

Appendix B

Superfast Computation of the Gradient and Hessian matrix of Profile Likelihoood

For the profile likelihood

$$\ell_{\text{prof}}(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{G}) = -\frac{Nd}{2}\log(2\pi) - \frac{d}{2}\log|\boldsymbol{V}_{\boldsymbol{\theta}}| - \frac{N}{2}\log|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}| - \frac{N}{2},$$

where

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} = (\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G})^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X}, \quad \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} = \frac{1}{N}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}),$$

its first derivative with respect to parameter $\theta_i \in \boldsymbol{\theta}$ is

$$\frac{\partial}{\partial\theta_i}\ell_{\text{prof}}(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{G}) = -\frac{d}{2}\text{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\} - \frac{N}{2}\text{tr}\{\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_i\}$$

where

$$\widehat{\boldsymbol{\Sigma}}_i = \frac{\partial\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}}{\partial\theta_i} = -\widehat{\boldsymbol{\beta}}_i'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}) - (\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\widehat{\boldsymbol{\beta}}_i -$$
$$(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}),$$

and

$$\widehat{\boldsymbol{\beta}}_i = \frac{\partial \widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}}{\partial \theta_i} = \left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} - \left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X}.$$

Since the computation of $\widehat{\boldsymbol{\Sigma}}_i$ and $\widehat{\boldsymbol{\beta}}_i$ is $\mathcal{O}(N \log N)$, and $\widehat{\boldsymbol{\Sigma}}_i$ is a size $d \times d$ matrix whose computation of $\mathrm{tr}\{\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_i\}$ is $\mathcal{O}(d^3)$, the total computation of $\frac{\partial}{\partial \theta_i}\ell_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{G})$ is superfast.

For the second derivative of the profile likelihood (2.3.8) with respect to $\theta_i, \theta_m \in \boldsymbol{\theta}$, we have

$$\frac{\partial^2}{\partial \theta_i \partial \theta_m}\ell_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{G}) = -\frac{d}{2}\mathrm{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_{im} - \boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\} - \frac{N}{2}\mathrm{tr}\{\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_{im} - \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_m\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_i\},$$

where

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}_{im} = \frac{\partial^2 \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}}{\partial \theta_i \partial \theta_m} = &- \widehat{\boldsymbol{\beta}}_{im}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}) + 2 \cdot \widehat{\boldsymbol{\beta}}_i'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}) + \widehat{\boldsymbol{\beta}}_i'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\widehat{\boldsymbol{\beta}}_m + \\ &\widehat{\boldsymbol{\beta}}_m'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\widehat{\boldsymbol{\beta}}_i + 2 \cdot (\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\widehat{\boldsymbol{\beta}}_i - (\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\widehat{\boldsymbol{\beta}}_{im} + \\ &(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}) - \\ &(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_{im}\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}) + \\ &(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}})'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}(\boldsymbol{X} - \boldsymbol{G}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}),\end{aligned}$$

and

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{im} = \frac{\partial^2 \widehat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}}{\partial\theta_i \partial\theta_m} = &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} - \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} + \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_{im}\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} - \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} + \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} - \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} + \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} - \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} + \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_{im}\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X} + \\ &\left(\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{G}\right)^{-1}\boldsymbol{G}'\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{X}.\end{aligned}$$

Despite the complex expression of $\widehat{\boldsymbol{\beta}}_{im}$ and $\widehat{\boldsymbol{\Sigma}}_{im}$, their computation is still $\mathcal{O}(N \log N)$. Since part $\mathrm{tr}\{\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_{im} - \boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\theta}}^{-1}\boldsymbol{V}_i\}$ is proved to be superfast and $\mathrm{tr}\{\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_{im} - \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_m\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}\widehat{\boldsymbol{\Sigma}}_i\}$ only involves some $d \times d$ matrices, the total computation of the Hessian term $\frac{\partial^2}{\partial\theta_i\partial\theta_m}\ell_{\mathrm{prof}}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{G})$ is also superfast.

Appendix C

Inference for the fSD Model

The k-dimensional fSD model (3.2.7) takes the form

$$\begin{aligned} \boldsymbol{X}(t) &= \sum_{j=1}^{d} \boldsymbol{\beta}_j f_j(t) + \boldsymbol{\Sigma}^{1/2} \boldsymbol{Z}(t), \\ \boldsymbol{Y}_n &= \frac{1}{\tau} \int_0^\tau \boldsymbol{X}(t_n - s)\, \mathrm{d}s + \varepsilon_n, \end{aligned} \tag{C.0.1}$$

where $t_n = n \cdot \Delta t$, $\boldsymbol{Z}(t) = \big(Z_1(t), \ldots, Z_k(t)\big)$ with $Z_i(t) \overset{\text{iid}}{\sim} B_\alpha(t)$, and $\varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2 \cdot \boldsymbol{\Sigma})$ are independent of $\boldsymbol{Z}(t)$. Letting $\Delta \boldsymbol{Y}_n = \boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n$, we can rewrite (C.0.1) to obtain

$$\Delta \boldsymbol{Y}_n = \sum_{j=1}^{d} \boldsymbol{\beta}_j \Delta f^\star_{nj} + \boldsymbol{\Sigma}^{1/2} (\Delta \boldsymbol{Z}^\star_n - \Delta \boldsymbol{\eta}_n),$$

where

$$f^\star_{nj} = \frac{1}{\tau} \int_0^\tau f_j(t_n - s)\, \mathrm{d}s, \qquad Z^\star_{ni} = \frac{1}{\tau} \int_0^\tau Z_i(t_n - s)\, \mathrm{d}s,$$

and $\boldsymbol{\eta}_n = \boldsymbol{\Sigma}^{-1/2} \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_d)$. Thus we have $f^\star_{nj} = \frac{1}{\tau} \int_0^\tau f_j(t_n - s)\, \mathrm{d}s$, $\boldsymbol{Z}^\star_n = (Z^\star_{n1}, \ldots Z^\star_{nk})$ with $Z^\star_{ni} = \frac{1}{\tau} \int_0^\tau Z_i(t_n - s)\, \mathrm{d}s$, and $\boldsymbol{\eta}_n = \boldsymbol{\Sigma}^{-1/2} \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_d)$. Thus, we have

$$\Delta \boldsymbol{Y}_{N \times k} \sim \text{MatNorm}(\boldsymbol{F}\boldsymbol{\beta}, \boldsymbol{V}_{\boldsymbol{\varphi}}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{F}_{N\times d}$ has elements $\boldsymbol{F}_{nj} = \Delta f^{\star}_{nj}$, $\boldsymbol{V}_{\boldsymbol{\varphi}}$ is a variance matrix parametrized by $\boldsymbol{\varphi} = (\alpha, \tau, \sigma)$ with elements

$$\begin{aligned}\boldsymbol{V}_{\boldsymbol{\varphi}}^{(n,m)} &= \operatorname{cov}(\Delta Z^{\star}_{ni} + \Delta\eta_{ni}, \Delta Z^{\star}_{mi} + \Delta\eta_{mi}) \\ &= \operatorname{cov}(\Delta Z^{\star}_{ni}, \Delta Z^{\star}_{mi}) + \operatorname{cov}(\Delta\eta_{ni}, \Delta\eta_{mi}).\end{aligned}$$

To finish the calculations, without loss of generality we may focus on the one-dimensional case $Z_i(t) = Z(t) = B_{\alpha}(t)$ and $\eta_{in} = \eta_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$. Thus we have

$$\begin{aligned}\operatorname{cov}(Z^{\star}_n, Z^{\star}_m) &= E[Z^{\star}_n Z^{\star}_m] \\ &= \frac{1}{\tau^2} E\left[\int_0^{\tau} Z(t_n - s)\,\mathrm{d}s \cdot \int_0^{\tau} Z(t_m - u)\,\mathrm{d}u\right] \\ &= \frac{1}{\tau^2} E\left[\int_0^{\tau}\int_0^{\tau} Z(t_n - s)Z(t_m - u)\,\mathrm{d}s\,\mathrm{d}u\right] \\ &= \frac{1}{\tau^2}\int_0^{\tau}\int_0^{\tau} E\left[Z(t_n - s)Z(t_m - u)\right]\mathrm{d}s\,\mathrm{d}u,\end{aligned}$$

where the last line is obtained from the Fubini-Tonelli theorem, since by Cauchy-Schwarz we have

$$\begin{aligned}\int_0^{\tau}\int_0^{\tau} E\Big[|Z(t_n - s)Z(t_m - u)|\Big]\,\mathrm{d}s\,\mathrm{d}u &\leqslant \sqrt{\int_0^{\tau} E[Z(t_n - s)^2]\,\mathrm{d}s \cdot \int_0^{\tau} E[Z(t_m - u)^2]\,\mathrm{d}u} \\ &= \sqrt{\int_0^{\tau} \operatorname{MSD}_Z(t_n - s)\,\mathrm{d}s \cdot \int_0^{\tau} \operatorname{MSD}_Z(t_m - u)\,\mathrm{d}u},\end{aligned}$$

and the right-hand side is finite as long as $\operatorname{MSD}_Z(t)$ is continuous for $t \geqslant 0$. Thus, for the fBM process $Z(t) = B_{\alpha}(t)$ we have

$$\begin{aligned}\operatorname{cov}(Z^{\star}_n, Z^{\star}_m) &= \frac{1}{2\tau^2}\int_0^{\tau}\int_0^{\tau} (t_n - s)^{\alpha} + (t_m - u)^{\alpha} - |(t_n - t_m) - (s - u)|^{\alpha}\,\mathrm{d}s\,\mathrm{d}u \\ &= h_{\tau}(t_n) + h_{\tau}(t_m) - g_{\tau}(t_n - t_m),\end{aligned}$$

where

$$g_\tau(t) = \frac{|t+\tau|^{\alpha+2} + |t-\tau|^{\alpha+2} - 2|t|^{\alpha+2}}{2\tau^2(\alpha+1)(\alpha+2)}, \qquad h_\tau(t) = \frac{(t-\tau)^\alpha - t^\alpha}{2\tau(\alpha+1)}.$$

Finally, since for any increment process ΔX_n we have

$$\operatorname{cov}(\Delta X_n, \Delta X_m) = E\left[X_{n+1}X_{m+1}\right] - E\left[X_{n+1}X_m\right] - E\left[X_n X_{m+1}\right] + E\left[X_n X_m\right],$$

we may calculate that

$$\text{ACF}_{\Delta Z^\star}(n) = \operatorname{cov}(\Delta Z^\star_n, \Delta Z^\star_{m+n}) = g_\tau(|n+1|\Delta t) + g_\tau(|n-1|\Delta t) - 2g_\tau(|n|\Delta t).$$

Similarly, we obtain

$$\text{ACF}_{\Delta\eta}(n) = \sigma^2 \times \big\{2 \cdot \mathbb{1}(n=0) - \mathbb{1}(n=1)\big\},$$

such that $\boldsymbol{V}_{\boldsymbol{\varphi}}$ is a Toeplitz matrix with elements

$$\boldsymbol{V}_{\boldsymbol{\varphi}}^{(n,m)} = \text{ACF}_{\Delta Z^\star}(n-m) + \text{ACF}_{\Delta\eta}(n-m).$$

Appendix D

Calculations for ARMA Noise Models

D.1 Relationship Between ACF and MSD

Let $X(t)$ be a one-dimensional CSI process with evenly-spaced observations $X_n = X(n\Delta t)$, such that

$$\text{MSD}_X(n) = E[(X_n - X_0)^2].$$

If $\Delta X_n = X_{n+1} - X_n$ is the corresponding increment process, then we have

$$\text{ACF}_{\Delta X}(n) = E[X_{n+1}X_1] + E[X_nX_0] - E[X_{n+1}X_0] - E[X_nX_1].$$

Combined with the fact that

$$\text{MSD}_X(n) = E[X_n^2] + E[X_0^2] - 2E[X_nX_0],$$

we find that

$$\text{ACF}_{\Delta X}(n) = \frac{1}{2}\{\text{MSD}_X(|n-1|) + \text{MSD}_X(|n+1|) - 2\,\text{MSD}_X(|n|)\}.$$

Conversely, we have

$$\mathrm{MSD}_X(n) = \mathrm{MSD}_X(n-1) + \mathrm{ACF}_{\Delta X}(0) + 2\sum_{h=1}^{n-1} \mathrm{ACF}_{\Delta X}(h),$$

such that

$$\mathrm{MSD}_X(n) = (n+1)\,\mathrm{ACF}_{\Delta X}(0) + 2\sum_{h=1}^{n}(n+1-h)\,\mathrm{ACF}_{\Delta X}(h). \tag{D.1.1}$$

D.2 Autocorrelation Function of the $\mathrm{ARMA}(p,q)$ Filter

Consider a one-dimensional stationary increments process determined by the $\mathrm{ARMA}(p,q)$ filter (3.3.6),

$$\Delta Y_n = \sum_{i=1}^{p} \theta_i \Delta Y_{n-i} + \sum_{j=0}^{q} \rho_j \Delta X_{n-j},$$

for which the driving process ΔX_n is assumed to have mean zero. In the following subsections we shall calculate the autocorrelation function $\mathrm{ACF}_{\Delta Y}(n)$ as a function of $\mathrm{ACF}_{\Delta X}(n) = \mathrm{cov}(\Delta X_m, \Delta X_{m+n})$.

D.2.1 Autocorrelation of the $\mathrm{MA}(q)$ Filter

For a purely moving-average process

$$\Delta Y_n = \sum_{i=0}^{q} \rho_i \Delta X_{n-i},$$

we have

$$\mathrm{ACF}_{\Delta Y}(n) = \sum_{i=0}^{q}\sum_{j=0}^{q} \rho_i \rho_j\, \mathrm{ACF}_{\Delta X}(n+i-j). \tag{D.2.1}$$

This can be computed efficiently for all values of $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_{N-1})$, $\gamma_n = \mathrm{ACF}_{\Delta Y}(n)$, using the following method. Let $\eta_n = \mathrm{ACF}_{\Delta X}(n)$, $\mathbf{0}_N$ denote the vector of N zeros, and for vectors $\boldsymbol{a} = (a_1, \ldots, a_N)$ and $\boldsymbol{b} = (a_1, b_2, \ldots, b_M)$, let $\mathrm{Toep}(\boldsymbol{a}, \boldsymbol{b})$ denote the $M \times N$

Toeplitz matrix with first row being $\boldsymbol{a}$ and first column $\boldsymbol{b}$:

$$\text{Toep}(\boldsymbol{a}, \boldsymbol{b}) = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & \cdots & a_N \\ b_2 & a_1 & a_2 & \ddots & & \vdots \\ b_3 & b_2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_2 & a_3 \\ \vdots & & \ddots & b_2 & a_1 & a_2 \\ b_M & \cdots & \cdots & b_3 & b_2 & a_1 \end{bmatrix}.$$

Then $\boldsymbol{\gamma}$ can be computed by the matrix multiplication

$$\boldsymbol{\gamma} = \text{Toep}(\boldsymbol{\rho}_1, \boldsymbol{\rho}_2) \cdot \text{Toep}(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) \cdot \boldsymbol{\rho}_0,$$

where

$$\begin{aligned} &\boldsymbol{\eta}_1 = (\eta_0, \ldots, \eta_q), && \boldsymbol{\eta}_2 = (\eta_0, \ldots, \eta_{N+q}), \\ &\boldsymbol{\rho}_0 = (\rho_0, \ldots, \rho_q), && \boldsymbol{\rho}_1 = (\boldsymbol{\rho}_0, \mathbf{0}_{N+1}), && \boldsymbol{\rho}_2 = (\rho_0, \mathbf{0}_{N-1}). \end{aligned}$$

Moreover, Toeplitz matrix-vector multiplication can be computed efficiently using the fast Fourier transform (FFT) [e.g., Kailath and Sayed, 1999a]. That is, let $\boldsymbol{\mathcal{F}}$ denote FFT the matrix of the appropriate dimension. In order to compute $\boldsymbol{\gamma}$, we perform the following steps:

1. Let $\boldsymbol{v}_3 = \boldsymbol{\mathcal{F}}^{-1}(\boldsymbol{\mathcal{F}}\boldsymbol{v}_1 \odot \boldsymbol{\mathcal{F}}\boldsymbol{v}_2)$, where $\boldsymbol{v}_1 = (\boldsymbol{\eta}_2, 0, \eta_q, \ldots, \eta_1)$, $\boldsymbol{v}_2 = (\boldsymbol{\rho}_0, \mathbf{0}_{N+q+1})$, and $\odot$ denotes the elementwise product between vectors.
2. Let $\boldsymbol{v}_4$ denote the first $N + q + 1$ elements of $\boldsymbol{v}_3$.
3. Let $\boldsymbol{v}_7 = \boldsymbol{\mathcal{F}}^{-1}(\boldsymbol{\mathcal{F}}\boldsymbol{v}_5 \odot \boldsymbol{\mathcal{F}}\boldsymbol{v}_6)$, where $\boldsymbol{v}_5 = (\rho_0, \mathbf{0}_{2N}, \rho_q, \ldots, \rho_1)$ and $\boldsymbol{v}_6 = (\boldsymbol{v}_4, \mathbf{0}_N)$.
4. $\boldsymbol{\gamma}$ is given by the first N elements of $\boldsymbol{v}_7$.

D.2.2 Autocorrelation of the AR(p) Filter

For a purely autoregressive process

$$\Delta Y_n = \sum_{i=1}^{p} \theta_i \Delta Y_{n-i} + \Delta X_n,$$

the autocorrelation $\text{ACF}_{\Delta Y}(n)$ involves an infinite summation which generally cannot be simplified further. Instead, we approximate the AR(p) filter with an MA(q) filter and use the result of Section D.2.1. To do this, we rewrite ΔY_n in terms of the lag operator B, such that

$$\Delta Y_n = \theta(B)\Delta Y_n + \Delta X_n,$$

where $\theta(x) = \theta_1 x + \cdots + \theta_p x^p$, and $B^k \Delta Y_n = \Delta Y_{n-k}$. Rearranging terms and expanding into a power series, we find that

$$\begin{aligned} \Delta Y_n &= \left[1 - \theta(B)\right]^{-1} \Delta X_n \\ &= \left[1 + \textstyle\sum_{i=1}^{\infty} [\theta(B)]^i\right] \Delta X_n = \left[1 + \textstyle\sum_{i=1}^{\infty} \rho_i B^i\right] \Delta X_n, \end{aligned}$$

such that ΔY_n may be expressed as an MA(∞) series. Truncating to order q, the true autocorrelation $\text{ACF}_{\Delta Y}(n)$ is approximated by the autocorrelation (D.2.1) of the corresponding MA(q) process $\Delta Y_n \approx \sum_{i=1}^{q} \rho_i \Delta X_{n-i}$. The following lemma can be used to efficiently calculate the coefficients ρ_i.

Lemma 3. Consider a polynomial $g(x) = \sum_{k=0}^{p} a_k x^k$ and its n-th power, $G(x) = [g(x)]^n = \sum_{k=0}^{m} b_k^{(n)} x^k$, where $m = n \cdot p$. Then we have

$$\left[\tfrac{\mathrm{d}}{\mathrm{d}x} G(x)\right] g(x) = n \left[\tfrac{\mathrm{d}}{\mathrm{d}x} g(x)\right] G(x).$$

As a result, when $a_0 \neq 0$ we can derive the coefficients of $G(x)$ recursively, with $b_0^{(n)} = a_0^n$ and

$$b_k^{(n)} = \frac{1}{k a_0} \times \left[nk b_0^{(n)} a_k + \sum_{i=1}^{k-1} (k-i)(n b_i^{(n)} a_{k-i} - a_i b_{k-i}^{(n)}) \right]. \tag{D.2.2}$$

Using Lemma 3 with $g(x) = \theta(x)/x = \theta_1 + \cdots + \theta_p x^{p-1}$, we find that $\rho_i = \sum_{j=1}^{i} b_{i-j}^{(j)}$,

where $b_{i-j}^{(j)}$ is given by (D.2.2) for $i-j \leqslant j \cdot p$, and $b_{i-j}^{(j)} = 0$ otherwise. In the simulations and data analyses of sections 3.4 and 3.5, we approximate all AR(p) filters by MA(50) filters. Numerical experiments indicate that changing the order to MA(500) does not change the approximated autocorrelations by more than 10^{-14}.

D.2.3 Autocorrelation of the ARMA(p, q) Filter

For the general ARMA(p, q) filter, we obtain the autocorrelation in two steps:

1. Let $\Delta Z_n = \sum_{j=0}^{q} \rho_j \Delta X_{n-j}$, and calculate the autocorrelation of this MA(q) process using (D.2.1).

2. Now we rewrite the original ARMA(p, q) process as

$$\Delta Y_n = \sum_{i=1}^{p} \theta_i \Delta Y_{n-i} + \Delta Z_n,$$

   and we may approximate the autocorrelation of this AR(p) process by applying the technique of Appendix D.2.2 to $\text{ACF}_{\Delta Z}(n)$ obtained in Step 1.

D.3 Proof of Theorem 1

In order to parametrize the ARMA(p, q) filter such that it satisfies the high-frequency error hypothesis (3.3.1), we begin by studying the relation between the MSD of a discrete-time univariate CSI process $\{X_n : n \geqslant 0\}$, and the power spectral density (PSD) of its stationary increment process, $\Delta X_n = X_{n+1} - X_n$.

For a stationary time series $\{\Delta X_n : n \in \mathbb{Z}\}$ which is purely non-deterministic in the sense of the Wold decomposition [e.g., Brockwell and Davis, 1991], the PSD $S_{\Delta X}(\omega)$ is defined as the unique nonnegative symmetric integrable function for which the autocorrelation of ΔX_n is given by

$$\text{ACF}_{\Delta X}(n) = \int_{-\pi}^{\pi} e^{-in\omega} S_{\Delta X}(\omega) \, \mathrm{d}\omega. \tag{D.3.1}$$

In order to prove Theorem 1 we begin by proving the following lemma:

Lemma 4. For two CSI process X and Y with corresponding increment processes ΔX and ΔY, if $S_{\Delta Y}(\omega)$ is positive in a neighborhood of $\omega = 0$, and the PSD ratio satisfies

$$\lim_{\omega \to 0} \frac{S_{\Delta X}(\omega)}{S_{\Delta Y}(\omega)} = 1,$$

then X and Y satisfy the high-frequency error definition (3.3.1), namely

$$\lim_{n \to \infty} \frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)} = 1.$$

Proof. Using (D.1.1) and (D.3.1) we can relate $\mathrm{MSD}_X(n)$ to $S_{\Delta X}(\omega)$, such that

$$\mathrm{MSD}_X(n+1) - \mathrm{MSD}_X(n) = \int_{-\pi}^{\pi} \sum_{j=-n}^{n} e^{-ij\omega} S_{\Delta X}(\omega)\, \mathrm{d}\omega = \int_{-\pi}^{\pi} D_n(\omega) S_{\Delta X}(\omega)\, \mathrm{d}\omega,$$

where $D_n(\omega) = \sum_{j=-n}^{n} e^{-ij\omega}$ is the n-th order Dirichlet kernel. Thus we have

$$\mathrm{MSD}_X(n) = \int_{-\pi}^{\pi} \sum_{k=0}^{n-1} D_k(\omega) S_{\Delta X}(\omega)\, \mathrm{d}\omega = n \int_{-\pi}^{\pi} F_n(\omega) S_{\Delta X}(\omega)\, \mathrm{d}\omega, \tag{D.3.2}$$

where $F_n(\omega) = \frac{1}{n} \sum_{k=0}^{n-1} D_k(\omega)$ is the n-th order Fejér kernel. Since $F_n(\omega)$ is symmetric about 0, we may rewrite $\mathrm{MSD}_X(n)$ as a convolution integral

$$\mathrm{MSD}_X(n) = n2\pi \times \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{\Delta X}(\omega) F_n(-\omega)\, \mathrm{d}\omega = n2\pi \times \{S_{\Delta X} * F_n\}(0).$$

By the Fejér kernel's summability property, we have

$$\{S_{\Delta X} * F_n\}(\omega) \to S_{\Delta X}(\omega) \quad \text{a.e.},$$
$$\{S_{\Delta Y} * F_n\}(\omega) \to S_{\Delta Y}(\omega) \quad \text{a.e.}.$$

Since $S_{\Delta Y}(\omega) > 0$ in a neighborhood of $\omega = 0$, we may thus find $\varepsilon > 0$ such that both $\{S_{\Delta Y} * F_n\}(0) \to S_{\Delta Y}(0) > 0$ and $\{S_{\Delta Y} * F_n\}(\omega_0) \to S_{\Delta Y}(\omega_0) > 0$ for $|\omega_0| < \varepsilon$. Given this,

we can express the MSD ratio as

$$\begin{aligned}\frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)} &= \frac{\{S_{\Delta X} * F_n\}(0)}{\{S_{\Delta Y} * F_n\}(0)} \\ &= \frac{\{S_{\Delta X} * F_n\}(0)}{\{S_{\Delta Y} * F_n\}(0)} - \frac{\{S_{\Delta X} * F_n\}(\omega_0)}{\{S_{\Delta Y} * F_n\}(\omega_0)} \\ &\quad + \frac{\{S_{\Delta X} * F_n\}(\omega_0)}{\{S_{\Delta Y} * F_n\}(\omega_0)} - \frac{S_{\Delta X}(\omega_0)}{S_{\Delta Y}(\omega_0)} + \frac{S_{\Delta X}(\omega_0)}{S_{\Delta Y}(\omega_0)}.\end{aligned}$$

Since $S_{\Delta X}(\omega)$ and $F_n(\omega)$ are both integrable and $\int F_n(\omega)d\omega = 1$, the convolution $\{S_{\Delta X} * F_n\}(\omega)$ is a uniformly continuous function. The same argument applies to $\{S_{\Delta Y} * F_n\}(\omega)$. Since $f_n(\omega) = \frac{\{S_{\Delta X} * F_n\}(\omega)}{\{S_{\Delta Y} * F_n\}(\omega)}$ is a ratio between two continuous functions, it is also a continuous function, which means that we can find $\omega_1 > 0$ such that for $|\omega| < \omega_1$ we have $|f_n(0) - f_n(\omega)| < \frac{\varepsilon}{3}$. Moreover, by Fejér summability we have

$$f_n(\omega) = \frac{\{S_{\Delta X} * F_n\}(\omega)}{\{S_{\Delta Y} * F_n\}(\omega)} \to \frac{S_{\Delta X}(\omega)}{S_{\Delta Y}(\omega)} = f(\omega) \quad \text{a.e.},$$

such that we may find N_1 such that $|f_n(\omega) - f(\omega)| < \frac{\varepsilon}{3}$ uniformly in ω for $n > N_1$. Thus, if

$$\lim_{\omega \to 0} \frac{S_{\Delta X}(\omega)}{S_{\Delta Y}(\omega)} = 1,$$

we may find $\omega_2 > 0$ such that $|f(\omega) - 1| < \frac{\varepsilon}{3}$ for $|\omega| < \omega_2$, and thus for $n > N_1$ and any ω such that $|\omega| < \min\{\omega_1, \omega_2\}$, we have

$$\begin{aligned}\left|\frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)} - 1\right| &\leqslant |f_n(0) - f_n(\omega)| + |f_n(\omega) - f(\omega)| + |f(\omega) - 1| \\ &\leqslant \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} = \varepsilon,\end{aligned}$$

such that

$$\lim_{n \to \infty} \frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)} = 1.$$

□

To complete the proof of Theorem 1, we apply Lemma 4 to the CSI process X_n and its

ARMA(p,q) filter Y_n as defined by (3.3.2). That is, for the increment processes ΔX_n and $\Delta Y_n = \sum_{i=1}^{p} \theta_i \Delta Y_{n-i} + \sum_{j=0}^{q} \rho_j \Delta X_{n-i}$,

$$\lim_{\omega \to 0} \frac{S_{\Delta X}(\omega)}{S_{\Delta Y}(\omega)} = \lim_{\omega \to 0} \frac{|1 - \sum_{k=1}^{p} \theta_k \cdot e^{-ik\omega}|^2}{|\sum_{j=0}^{q} \rho_j e^{-ij\omega}|^2} = \left| \frac{1 - \sum_{i=1}^{p} \theta_i}{\sum_{j=0}^{q} \rho_j} \right|^2 .$$

Thus by setting $\rho_0 = 1 - \sum_{i=1}^{p} \theta_i - \sum_{j=1}^{q} \rho_j$, we have

$$\lim_{\omega \to 0} \frac{S_{\Delta X}(\omega)}{S_{\Delta Y}(\omega)} = \left(\frac{1 - \sum_{i=1}^{p} \theta_i}{\sum_{j=0}^{q} \rho_j} \right)^2 = 1 \quad \Longrightarrow \quad \lim_{n \to \infty} \frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)} = 1,$$

which completes the proof of Theorem 1.

D.4 Proof of Theorem 2

The complete statement of Theorem 2 is as follows.

Let $\boldsymbol{\mathcal{X}} = \{X_n : n \geqslant 0\}$ denote the true positions of a CSI process, for which $\boldsymbol{\mathcal{Y}} = \{Y_n : n \geqslant 0\}$ is the measurement process satisfying the high-frequency error definition (3.3.1). For the corresponding increment processes $\Delta\boldsymbol{\mathcal{X}} = \{\Delta X_n : n \in \mathbb{Z}\}$ and $\Delta\boldsymbol{\mathcal{Y}} = \{\Delta Y_n : n \in \mathbb{Z}\}$, suppose the PSD ratio

$$g(\omega) = \frac{S_{\Delta Y}(\omega)}{S_{\Delta X}(\omega)}$$

is continuous on the interval $\omega \in [-\pi, \pi]$. Then there exists an ARMA(p,q) noise model $\boldsymbol{\mathcal{Y}}^\star = \{Y_n^\star : n \geqslant 0\}$ satisfying (3.3.2) such that for all $n \geqslant 0$ we have

$$\left| \frac{\mathrm{MSD}_{Y^\star}(n)}{\mathrm{MSD}_Y(n)} - 1 \right| < \epsilon. \tag{D.4.1}$$

.

Proof. In order to show that there exits an ARMA(p, q) process

$$Y_n^\star = \sum_{i=1}^{p} \theta_i Y_{n-i}^\star + \sum_{j=0}^{q} \rho_j X_{n-j},$$

satisfying (D.4.1), we use (D.3.2) to write

$$\begin{aligned}
\left| \frac{\mathrm{MSD}_{Y^\star}(n)}{\mathrm{MSD}_Y(n)} - 1 \right| &= \frac{|\,\mathrm{MSD}_{Y^\star}(n) - \mathrm{MSD}_Y(n)|}{\mathrm{MSD}_Y(n)} \\
&\leqslant \frac{\int_{-\pi}^{\pi} F_n(\omega) \cdot |S_{\Delta Y^\star}(\omega) - S_{\Delta Y}(\omega)|\,\mathrm{d}\omega}{\int_{-\pi}^{\pi} F_n(\omega) S_{\Delta Y}(\omega)\,\mathrm{d}\omega} \\
&= \frac{\int_{-\pi}^{\pi} |r(\omega) - g(\omega)| \cdot F_n(\omega) S_{\Delta X}(\omega)\,\mathrm{d}\omega}{\int_{-\pi}^{\pi} F_n(\omega) S_{\Delta Y}(\omega)\,\mathrm{d}\omega},
\end{aligned}$$

where $g(\omega) = S_{\Delta Y}(\omega)/S_{\Delta X}(\omega)$ and

$$r(\omega) = \frac{S_{\Delta Y^\star}(\omega)}{S_{\Delta X}(\omega)} = \left| \frac{\sum_{j=0}^{q} \rho_j e^{-ij\omega}}{1 - \sum_{k=1}^{p} \theta_k \cdot e^{-ik\omega}} \right|^2.$$

Because $g(\omega)$ is a ratio of nonnegative symmetric functions, it is also nonnegative symmetric, and since it is continuous, it satisfies the definition of a continuous PSD. Therefore, by Corollary 4.4.1 of Brockwell and Davis [1991], we can find a stationary MA(q) process

$$Z_n = \sum_{j=0}^{q} \rho_j \eta_{n-j}, \qquad \eta_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$$

satisfying parameter restrictions (3.3.3), such that if $S_Z(\omega) = |\sum_{j=0}^{q} \rho_j e^{-ij\omega}|^2$ is the PSD of this process,

$$|S_Z(\omega) - g(\omega)| < \varepsilon_0 \text{ for } \omega \in [-\pi, \pi].$$

Therefore, let $\Delta Y_n^\star = \sum_{j=0}^{q} \rho_j \Delta X_n$, such that $r(\omega) = S_{\Delta Y^\star}(\omega)/S_{\Delta X}(\omega) = S_Z(\omega) =$

$|\sum_{j=0}^{q} \rho_j e^{-ij\omega}|^2$. Then we have

$$\left|\frac{\mathrm{MSD}_{Y^\star}(n)}{\mathrm{MSD}_Y(n)} - 1\right| \leqslant \frac{\int_{-\pi}^{\pi} |r(\omega) - g(\omega)| \cdot F_n(\omega) S_{\Delta X}(\omega)\,\mathrm{d}\omega}{\int_{-\pi}^{\pi} F_n(\omega) S_{\Delta Y}(\omega)\,\mathrm{d}\omega}$$
$$\leqslant \varepsilon_0 \cdot \frac{\int_{-\pi}^{\pi} F_n(\omega) S_{\Delta X}(\omega)\,\mathrm{d}\omega}{\int_{-\pi}^{\pi} F_n(\omega) S_{\Delta Y}(\omega)\,\mathrm{d}\omega} = \varepsilon_0 \cdot \frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)}.$$

Since $\lim_{n\to\infty} \mathrm{MSD}_X(n)/\mathrm{MSD}_Y(n)$ exists, there exists $L > 0$ such that for every n we have

$$0 \leqslant \frac{\mathrm{MSD}_X(n)}{\mathrm{MSD}_Y(n)} \leqslant L.$$

Thus by letting $\varepsilon_0 = \varepsilon/L$, for every n we have

$$\left|\frac{\mathrm{MSD}_{Y^\star}(n)}{\mathrm{MSD}_Y(n)} - 1\right| < \varepsilon.$$

□

Appendix E

Calculations for the GLE Process

For the GLE process $X(t)$ defined by (3.4.4) with sum-of-exponentials memory kernel

$$\phi(t) = \frac{\nu}{K} \sum_{k=1}^{K} \exp(-|t|\alpha_k),$$

McKinley et al. [2009] derive its MSD to be

$$\text{MSD}_X(t) = \frac{2k_BT}{\nu/K} \left(C_0^2 t + \sum_{j=1}^{K-1} \frac{C_j^2}{r_j}(1 - e^{-r_j t}) \right),$$

where $r_1, \ldots, r_{K-1}$ are the roots of $q(y) = \prod_{k=1}^{K}(y - \alpha_k)$, and

$$C_0 = \left(\sum_{k=1}^{K} \frac{1}{\alpha_k} \right)^{1/2}, \qquad C_j = \frac{1}{r_j} \times \frac{\sqrt{\sum_{k=1}^{K} \frac{1}{(1-r_j\alpha_k)^2}}}{(\sum_{k=1}^{K} \frac{\alpha_k}{1-r_j\alpha_k})^2 - \sum_{k=1}^{K} \frac{\alpha_k}{1-r_j\alpha_k}}.$$

For the particular case of the Rouse memory kernel

$$\phi(t) = \frac{\nu}{K} \sum_{k=1}^{K} \exp(-|t|/\tau_k), \quad \tau_k = \tau \cdot (K/k)^{\gamma},$$

McKinley et al. [2009] show that for sufficiently large K, the MSD exhibits (anomalous) transient subdiffusion,

$$\mathrm{MSD}_X(t) = \begin{cases} 2 \cdot D_{\text{eff}} \cdot t^{\alpha_{\text{eff}}} & t_{\min} < t < t_{\max} \\ 2 \cdot D_{\min} \cdot t & t < t_{\min} \\ 2 \cdot D_{\max} \cdot t & t > t_{\max}. \end{cases}$$

This is illustrated in Figure E.1 with $K = 300$ and GLE parameters $\gamma = 1.67$, $\tau = 0.01$, $\nu = 1$. Figure E.1 also displays the subdiffusion timescale $(t_{\min}, t_{\max})$ along with the power

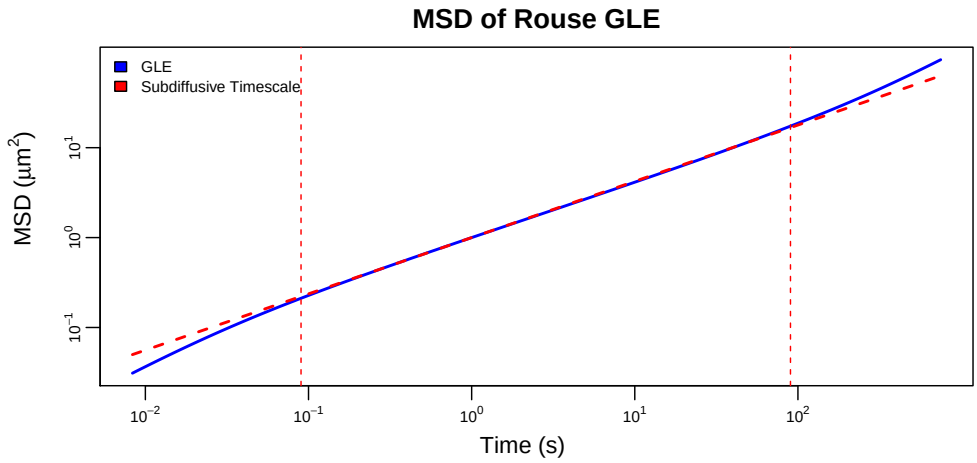


Figure E.1: MSD of a Rouse GLE with $K = 300$ and $\gamma = 1.67$, $\tau = 0.01$, $\nu = 1$ (solid blue line). Also displayed is the subdiffusion timescale $(t_{\min}, t_{\max})$ along with the power law $\mathrm{MSD}_X(t) = 2D_{\text{eff}} \cdot t^{\alpha_{\text{eff}}}$ on that range (red dotted lines).

law $\mathrm{MSD}_X(t) = 2D_{\text{eff}} \cdot t^{\alpha_{\text{eff}}}$ on that range. The values of $(t_{\min}, t_{\max}, \alpha_{\text{eff}}, D_{\text{eff}})$ are determined from the GLE parameters K and $\boldsymbol{\varphi} = (\gamma, \tau, \nu)$ via the following method.

1. Calculate $x_n = \log(t_n)$ and $y_n = \log \mathrm{MSD}_X(t_n \mid \boldsymbol{\varphi}, K)$ on a range of time points $t_0, \ldots, t_N$. These should be picked on a fine grid such that $t_0 \ll t_{\min}$ and $t_N \gg t_{\max}$.
2. Let $\boldsymbol{\Upsilon} = (t_{\min}, t_{\max})$, and let $I_{\boldsymbol{\Upsilon}} = \{n : t_{\min} < t_n < t_{\max}\}$. Then for any $\boldsymbol{\Upsilon}$ we

calculate $\alpha_{\text{eff}}^{(\boldsymbol{\Upsilon})}$ and $D_{\text{eff}}^{(\boldsymbol{\Upsilon})}$ via least-squares:

$$\alpha_{\text{eff}}^{(\boldsymbol{\Upsilon})} = \frac{\sum_{n\in I_{\boldsymbol{\Upsilon}}}(y_n-\bar{y})(x_n-\bar{x})}{\sum_{n\in I_{\boldsymbol{\Upsilon}}}(x_n-\bar{x})^2}, \qquad D_{\text{eff}}^{(\boldsymbol{\Upsilon})} = \tfrac{1}{2}\exp(\bar{y}-\alpha_{\text{eff}}^{(\boldsymbol{\Upsilon})}\bar{x}),$$

where $\bar{x} = \frac{1}{|I_{\boldsymbol{\Upsilon}}|}\sum_{n\in I_{\boldsymbol{\Upsilon}}} x_n$ and $\bar{y} = \frac{1}{|I_{\boldsymbol{\Upsilon}}|}\sum_{n\in I_{\boldsymbol{\Upsilon}}} y_n$ are the corresponding averages over the indices in $I_{\boldsymbol{\Upsilon}}$.

3. The subdiffusion timescale $\boldsymbol{\Upsilon}$ is determined by solving the constrained optimization problem

$$\arg\max_{\boldsymbol{\Upsilon}} |\log(t_{\max}) - \log(t_{\min})|$$
$$\text{subject to} \qquad \max_{n\in I_{\boldsymbol{\Upsilon}}} \left| \frac{\alpha_{\text{eff}}^{(\boldsymbol{\Upsilon})}\cdot x_n + \log(2D_{\text{eff}}^{(\boldsymbol{\Upsilon})}) - y_n}{y_n} \right| < \kappa,$$

where κ is a tolerance for departure from a perfect power law over the subdiffusive range. In Figure E.1 and the calculations of Section 3.4.2 we have used $\kappa = 1\%$. This optimization problem can be solved in $\mathcal{O}(N^2)$ steps by trying all combinations of $t_{\min}$ and $t_{\max}$ in the set $\{t_0, \ldots, t_N\}$.

Appendix F

Superfast Computation of the Fisher Information

In order to compute the Fisher information

$$\boldsymbol{F}_i = E\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i} \right],$$

where $\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)$ is the likelihood of the location-scale model (4.2.4)

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i) = -\frac{1}{2}\mathrm{tr}\left\{\boldsymbol{\Sigma}^{-1}(\Delta\boldsymbol{Y}_i - \boldsymbol{G}\boldsymbol{\beta})'\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}(\Delta\boldsymbol{Y}_i - \boldsymbol{G}\boldsymbol{\beta})\right\} - \log|\boldsymbol{V}_{\boldsymbol{\varphi}}| - \frac{N}{2}\log|\boldsymbol{\Sigma}|,$$

we look into the element of the Hessian matrix

$$\left[\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i)\right]_{nm} = \frac{\partial^2}{\partial\boldsymbol{\theta}_n\partial\boldsymbol{\theta}_m}\ell(\boldsymbol{\theta} \mid \boldsymbol{Y}_i),$$

whose computation is exactly presented in Equation (2.3.6), where

$$\begin{aligned}\frac{\partial^2}{\partial\theta_n\partial\theta_m}\ell(\boldsymbol{\theta}\mid\boldsymbol{X}) = &-\operatorname{tr}\left\{\underbrace{\boldsymbol{\Omega}_m\boldsymbol{Z}_n'\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{Z}}_{H_1}+\underbrace{\boldsymbol{\Omega}\boldsymbol{Z}_{nm}'\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{Z}}_{H_2}+\underbrace{\boldsymbol{\Omega}\boldsymbol{Z}_n'\boldsymbol{\zeta}_m\boldsymbol{Z}}_{H_3}+\underbrace{\boldsymbol{\Omega}\boldsymbol{Z}_n'\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{Z}_m}_{H_4}\right\}\\ &-\frac{1}{2}\operatorname{tr}\left\{\underbrace{\boldsymbol{\Omega}_m\boldsymbol{Z}'\boldsymbol{\zeta}_n\boldsymbol{Z}}_{H_5}+\underbrace{\boldsymbol{\Omega}\boldsymbol{Z}_m'\boldsymbol{\zeta}_n\boldsymbol{Z}}_{H_6}+\underbrace{\boldsymbol{\Omega}\boldsymbol{Z}'\boldsymbol{\zeta}_{nm}\boldsymbol{Z}}_{H_7}+\underbrace{\boldsymbol{\Omega}\boldsymbol{Z}'\boldsymbol{\zeta}_n\boldsymbol{Z}_m}_{H_8}\right\}\\ &-\frac{1}{2}\operatorname{tr}\left\{\underbrace{\boldsymbol{\Omega}_{nm}\boldsymbol{Z}'\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{Z}}_{H_9}+\underbrace{\boldsymbol{\Omega}_n\boldsymbol{Z}_m'\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{Z}}_{H_{10}}+\underbrace{\boldsymbol{\Omega}_n\boldsymbol{Z}'\boldsymbol{\zeta}_m\boldsymbol{Z}}_{H_{11}}+\underbrace{\boldsymbol{\Omega}_n\boldsymbol{Z}'\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{Z}_m}_{H_{12}}\right\}\\ &-\frac{N}{2}\operatorname{tr}\left\{\boldsymbol{\Omega}\boldsymbol{\Sigma}_{nm}-\boldsymbol{\Omega}\boldsymbol{\Sigma}_m\boldsymbol{\Omega}\boldsymbol{\Sigma}_n\right\}-\operatorname{tr}\left\{\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{V}_{nm}-\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{V}_m\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{V}_n\right\},\end{aligned}$$

and

$$\begin{aligned}&\boldsymbol{Z}=\Delta\boldsymbol{Y}_i-\boldsymbol{G}\boldsymbol{\beta},\quad \boldsymbol{Z}_n=-\boldsymbol{G}\frac{\partial\boldsymbol{\beta}}{\partial\boldsymbol{\theta}_n},\quad \boldsymbol{Z}_{nm}=-\boldsymbol{G}\frac{\partial^2\boldsymbol{\beta}}{\partial\boldsymbol{\theta}_n\partial\boldsymbol{\theta}_m}\\ &\boldsymbol{V}_n=\frac{\partial}{\partial\theta_n}\boldsymbol{V}_{\boldsymbol{\varphi}},\quad \boldsymbol{V}_{nm}=\frac{\partial^2}{\partial\theta_n\partial\theta_m}\boldsymbol{V}_{\boldsymbol{\varphi}},\quad \boldsymbol{\Sigma}_n=\frac{\partial}{\partial\theta_n}\boldsymbol{\Sigma},\quad \boldsymbol{\Sigma}_{nm}=\frac{\partial^2}{\partial\theta_n\partial\theta_m}\boldsymbol{\Sigma},\\ &\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1},\quad \boldsymbol{\Omega}_n=\frac{\partial}{\partial\theta_n}\boldsymbol{\Omega}=-\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_n\boldsymbol{\Sigma}^{-1},\quad \boldsymbol{\zeta}_{\boldsymbol{\varphi}}=\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1},\quad \boldsymbol{\zeta}_n=\frac{\partial}{\partial\theta_n}\boldsymbol{\zeta}=-\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\boldsymbol{V}_n\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\\ &\boldsymbol{\Omega}_{nm}=\frac{\partial^2}{\partial\theta_n\partial\theta_m}\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_m\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_n\boldsymbol{\Sigma}^{-1}+\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_n\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_m\boldsymbol{\Sigma}^{-1}-\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{nm}\boldsymbol{\Sigma}^{-1}\\ &\boldsymbol{\zeta}_{nm}=\frac{\partial^2}{\partial\theta_n\partial\theta_m}\boldsymbol{\zeta}_{\boldsymbol{\varphi}}=\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\boldsymbol{V}_n\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}+\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\boldsymbol{V}_n\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\boldsymbol{V}_m\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}-\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}\boldsymbol{V}_{nm}\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}.\end{aligned}$$

The 12 marked components $(H_1,\ldots,H_{12})$ in $\frac{\partial^2}{\partial\theta_n\partial\theta_m}\ell(\boldsymbol{\theta}\mid\boldsymbol{X})$ can be divided into 3 categories:

$$(1)\quad \boldsymbol{A}\boldsymbol{Z}'\boldsymbol{B}\boldsymbol{Z},\qquad (2)\quad \boldsymbol{A}\boldsymbol{Z}_n'\boldsymbol{B}\boldsymbol{Z},\qquad (3)\quad \boldsymbol{A}\boldsymbol{Z}_n'\boldsymbol{B}\boldsymbol{Z}_m,$$

where $H_5, H_7, H_9, H_{11} \in (1)$, $H_1, H_2, H_3, H_6, H_8, H_{10}, H_{12} \in (2)$ and $H_4 \in (3)$.

For type (1), to compute $E\left[\operatorname{tr}\{\boldsymbol{A}\boldsymbol{Z}'\boldsymbol{B}\boldsymbol{Z}\}\right]$, we notice that $\boldsymbol{A}$ can be $\{\boldsymbol{\Omega},\boldsymbol{\Omega}_n,\boldsymbol{\Omega}_n,\boldsymbol{\Omega}_{nm}\}$, meaning that $\boldsymbol{A}=\left[\begin{smallmatrix}A_1 & A_2\\ A_2 & A_3\end{smallmatrix}\right]$ is a 2×2 symmetric matrix. Similarly we notice that matrix $\boldsymbol{B}$

can be $\{\boldsymbol{\zeta}, \boldsymbol{\zeta}_n, \boldsymbol{\zeta}_{nm}\}$ and $\boldsymbol{Z} = [\, \boldsymbol{z}_1 \; \boldsymbol{z}_2 \,]$ is a $N \times 2$ matrix with columns $\boldsymbol{Z}_1, \boldsymbol{Z}_2 \overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}_{\boldsymbol{\varphi}})$. As a result, we have

$$
\begin{aligned}
E\left[\operatorname{tr}\{\boldsymbol{A}\boldsymbol{Z}'\boldsymbol{B}\boldsymbol{Z}\}\right] &= E\left[\operatorname{tr}\{\begin{bmatrix} A_1 & A_2 \\ A_2 & A_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{Z}_1' \\ \boldsymbol{Z}_2' \end{bmatrix} \boldsymbol{B} \, [\, \boldsymbol{z}_1 \; \boldsymbol{z}_2 \,]\}\right] \\
&= E\left[\operatorname{tr}\{\begin{bmatrix} A_1 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_1 + A_2 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_1 & A_2 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_2 + A_1 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_2 \\ A_2 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_1 + A_3 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_1 & A_3 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_2 + A_2 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_2 \end{bmatrix}\}\right] \\
&= E\left[A_1 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_1 + A_2 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_1 + A_3 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_2 + A_2 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_2\right] \\
&= E\left[A_1 \cdot \boldsymbol{Z}_1'\boldsymbol{B}\boldsymbol{Z}_1 + A_3 \cdot \boldsymbol{Z}_2'\boldsymbol{B}\boldsymbol{Z}_2\right] \\
&= E\left[A_1 \cdot \operatorname{tr}\{\boldsymbol{B}\boldsymbol{Z}_1\boldsymbol{Z}_1'\} + A_3 \cdot \operatorname{tr}\{\boldsymbol{B}\boldsymbol{Z}_2\boldsymbol{Z}_2'\}\right] \\
&= A_1 \cdot \operatorname{tr}\{\boldsymbol{B} \times E\left[\boldsymbol{Z}_1\boldsymbol{Z}_1'\right]\} + A_3 \cdot \operatorname{tr}\{\boldsymbol{B} \times E\left[\boldsymbol{Z}_2\boldsymbol{Z}_2'\right]\} \\
&= (A_1 + A_3) \cdot \operatorname{tr}\{\boldsymbol{B}\boldsymbol{V}_{\boldsymbol{\varphi}}\},
\end{aligned}
$$

where the computation of $\operatorname{tr}\{\boldsymbol{B}\boldsymbol{V}_{\boldsymbol{\varphi}}\} = N$ or $\operatorname{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_n\}$ or $\operatorname{tr}\{\boldsymbol{\zeta}\boldsymbol{V}_n\boldsymbol{\zeta}\boldsymbol{V}_m\}$ is shown to be superfast in Section 2.3.3.

For type (2), we find that

$$
E\left[\operatorname{tr}\{\boldsymbol{A}\boldsymbol{Z}_n'\boldsymbol{B}\boldsymbol{Z}\}\right] = \operatorname{tr}\{\boldsymbol{A}\boldsymbol{Z}_n'\boldsymbol{B} \times E\left[\boldsymbol{Z}\right]\} = 0.
$$

For type (3), we have that

$$
E\left[\operatorname{tr}\{\boldsymbol{A}\boldsymbol{Z}_n'\boldsymbol{B}\boldsymbol{Z}_m\}\right] = \operatorname{tr}\{\boldsymbol{A}\boldsymbol{Z}_n'\boldsymbol{B}\boldsymbol{Z}_m\},
$$

which can be analytically computed.

Generally, the 12 marked parts $(H_1, \ldots, H_{12})$ can all be computed in $\mathcal{O}(N \log N)$ steps with $\boldsymbol{V}_{\boldsymbol{\varphi}}^{-1}$ provided. As is demonstrated in Section 2.3.3, the computation of the remaining parts $\operatorname{tr}\{\boldsymbol{\Omega}\boldsymbol{\Sigma}_{nm} - \boldsymbol{\Omega}\boldsymbol{\Sigma}_m\boldsymbol{\Omega}\boldsymbol{\Sigma}_n\}$ and $\operatorname{tr}\{\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{V}_{nm} - \boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{V}_m\boldsymbol{\zeta}_{\boldsymbol{\varphi}}\boldsymbol{V}_n\}$ are also superfast. In conclusion, the elements of the Fisher information $\boldsymbol{F}_i$ can be computed in superfast speed.