Performance of Time Series Interpolation Algorithms in the Presence of Noise

A thesis submitted to the Committee on Graduate Studies

in partial fulfillment of the requirements

for the Degree of Masters of Science in the Faculty of Arts and Science

Trent University

Peterborough, Ontario, Canada

Applied Modelling and Quantitative Methods M.Sc. Graduate Program

January 2025

Abstract

Performance of Time Series Interpolation Algorithms in the Presence of Noise

Roxana Niksefat

The spectral properties of time series data reveal underlying processes but require complete datasets, often unavailable due to missing values and irregular sampling.

This thesis uses a computational simulations framework to evaluate the performance of the Hybrid Wiener Interpolator [3], a novel method designed to reconstruct nonstationary time series data, thus making said data amenable for spectrum analysis. This research evaluates the Hybrid Wiener Interpolator's ability to handle nonstationary data and data gaps, comparing its performance to other interpolation methods under different stationarity and data integrity conditions.

The results illuminate the robustness of this interpolator in scenarios typical of scientific datasets, offering a promising approach for enhancing spectrum estimation in the presence of non-ideal data conditions [5].

Keywords: *Time Series, Interpolation, Multitaper, Spectrum, Stationarity, ARIMA Models, Data Imputation, Time Series Simulations*

Acknowledgements

To those who supported me, I am indebted: This journey would not have been possible without the guidance of my mentor, Dr. Wesley Burr. His profound wisdom, patience, and unwavering support have not only shaped this research but also inspired me deeply. Every day, I aspire to be more like him — his brilliance, intellectual curiosity, and unwavering dedication continue to motivate me. I am eternally grateful for his belief in my potential and his enduring encouragement.

To my parents, and most notably my father, who first taught me the beauty of mathematics - thank you for your boundless love and support. You have been my first teachers and my steadfast pillars. Your sacrifices and unconditional love have been the bedrock of my spirit and resolve.

I am profoundly thankful to my friends, who have been my sanctuary during the most tumultuous times. Your compassion and presence provided solace and strength when I needed it most, helping me navigate through the darkest waters of this journey.

I would also like to extend my heartfelt gratitude to Trent University, whose academic environment and resources have greatly contributed to the completion of this thesis. The university has provided me with a platform to grow academically and professionally, and for that, I am deeply thankful.

Lastly, I dedicate this thesis to my beloved grandfather, whose guidance continues to light my path. Since my childhood, his exemplary virtues have inspired me to pursue greatness and make a meaningful difference in the world. I am deeply grateful

for his presence in my life, which has shaped me into the person I am today. His influence is a constant reminder of the profound impact one life can have on another, and I am proud to carry his legacy forward with every step I take.

Table of Contents

List of Figures

1 Introduction

The challenge of reconstructing incomplete data has been a central problem across many scientific disciplines for centuries. Interpolation, the name often assigned to this challenge in the context of engineering and time series, addresses the common issue of missing data, often resulting from measurement errors, instrument failures, or irregular sampling. In many real-world applications, incomplete datasets are the norm rather than the exception. Despite numerous advances in interpolation methods, no single technique has proven universally optimal, as the performance of each approach is highly dependent on the characteristics of the data under consideration.

This thesis focuses on the problem of interpolation in time series analysis, where the sequential nature of data introduces unique complexities. Time series data consists of measurements ordered over time, with each observation representing a realization of an underlying stochastic process. Unlike standard datasets, time series often exhibit dependencies across observations, periodic components, and nonstationary behavior, which complicate both analysis and interpolation. These complexities are further exacerbated when missing data must be filled in before performing key analyses, such as spectrum estimation.

A common assumption in time series analysis is that the data are stationary, meaning that its statistical properties do not vary over time. Additionally, many interpolation techniques assume that data is sampled at regular intervals. However, real-world time series often violate these assumptions, with data being nonstationary

and irregularly sampled. Such conditions pose significant challenges to conventional interpolation methods, which are typically not designed to handle nonstationary, gappy time series.

This thesis evaluates the Hybrid Wiener Interpolator (HWI), an advanced interpolation method designed to address these challenges. The HWI is an iterative algorithm that combines elements of classical Wiener interpolation with multitaper spectral estimation to reconstruct missing data in nonstationary time series. This research systematically tests the performance of the HWI across different gap structures and time series characteristics, providing a detailed comparison with other standard interpolation techniques. The goal of this work is to assess the robustness and accuracy of the HWI in the presence of noise.

2 Literature Review

This chapter provides an overview of existing research and methodologies related to time series interpolation. It covers fundamental concepts of time series analysis and various interpolation techniques. The chapter analyzes the strengths and limitations of these methods, reviews comparative studies, and explores theoretical underpinnings, setting the stage for the contributions of this thesis and identifying potential areas for further investigation.

Before we start, it is essential to differentiate between extrapolation and interpolation. Extrapolation involves estimating points beyond the known data range, while interpolation involves estimating missing points within the range of a discrete set of known values. In this review, we specifically focus on interpolation algorithms suitable for repairing stochastic time series containing embedded line components. This differentiation is crucial, as it helps select the appropriate methods and ensures that our efforts are directed toward the most relevant techniques for addressing the challenges of time series interpolation.

Norbert Wiener's [32] groundbreaking work is regarded as the first significant modern text on interpolating stationary stochastic processes. It provides the basis for the Hybrid Wiener Interpolator (HWI) developed later by Wesley Burr in his PhD thesis [3]. Written from a time series and communications engineering perspective, the book presented a more technical approach to interpolation, offering a valuable tool for engineers in these fields. The text enabled the experimental solution of practical

prediction problems, primarily focusing on the approximation of random variables using finite linear combinations of a given stationary time series within a Hilbert space framework. This rigorous mathematical approach provided the foundational techniques and theoretical insights necessary for developing advanced interpolation algorithms, such as the Hybrid Wiener Interpolator (HWI).

The appendices of Norbert Wiener's seminal work [32] include two significant contributions by Norman Levinson [13, 14], The first of which addressed the problem of eliminating random noise (filtering), while the second involved reworking Wiener's earlier mathematical theory of prediction and filtering into a more algorithmic procedure. Wiener's book solved the problem of univariate prediction in the linear case but did not extend these solutions to multivariate or nonlinear processes. Additionally, it is essential to note that Wiener's methods apply only to stationary stochastic processes. This limitation is a common challenge among many interpolation algorithms reviewed in this thesis, as they often need help with non-stationary data.

The works of Wiener and Kolmogorov had a significant influence on the problem of interpolating stationary time series [5], and led to work by Swedish statistician Harold Cramér. Cramér extended Wiener's prediction theory [32] to encompass discrete and continuous nonstationary processes [7]. However, while Cramér's work was theoretically significant, it did not offer the same practical application as Wiener's seminal text. The theoretical nature of Cramér's contributions, while valuable for advancing the mathematical foundations of time series analysis, limited its immediate applicability to practical problems in the same way Wiener's work did.

The interpolation problem has been particularly significant in the field of econometrics, with numerous advancements in this area stemming from works published by economists. A relatively recent review by Pavía-Miralles [21] provides a detailed chronology of several algorithms developed by economists to interpolate, extrapolate, and distribute time series data. One of the key objectives discussed in this review

is temporal disaggregation, which involves decomposing aggregated data from a low-frequency series into a high-frequency series (for example, converting quarterly data into monthly data). While these techniques are valuable, they are of lesser relevance to our research, which focuses primarily on repairing missing data through interpolation rather than processing aggregated data.

Canadian-born statistician and Professor Emeritus at Queen's University, David J. Thomson, made significant advances in spectrum estimation across his career. While working as a member of the technical staff at Bell Labs in the 1980s, Thomson developed the multitaper spectral estimation method. This method addressed the common problem of bias-variance tradeoff in spectral estimation by using multiple orthogonal data tapers to generate independent, approximately maximum-likelihood estimates. These estimates possess numerous desirable statistical properties. The multitaper method was comprehensively detailed in a special issue of the Proceedings of the IEEE [27] and has since been extensively validated by numerous experts in the field. The Hybrid Wiener Interpolator (HWI) incorporates the robust theoretical foundation of the multitaper spectral estimation procedure into its framework, leveraging these advanced techniques to enhance its performance.

A particularly influential work for the development of the Hybrid Wiener Interpolator (HWI) was the 1988 paper by Mohsen Pourahmadi [22]. In this paper, Pourahmadi demonstrated that any missing value problem in the time domain could be simplified into a series of multistep prediction problems using the Wold decomposition, basic matrix properties, and the concept of orthogonality in Hilbert spaces. His approach involved a modification of the Kolmogorov-Wiener method, framed within an Expectation-Maximization (EM) algorithm. This algorithm proved to be highly versatile, capable of being applied to any nondeterministic stationary time series with missing values occurring in various patterns. Pourahmadi's method offered solutions

that were both rapid and efficient, significantly enhancing the ability to address complex interpolation challenges in time series analysis.

Going back in time several decades, Parzen in 1963 [20] was the first to approach 'gappy' time series as an amplitude modulation (AM) of the complete series, where the AM is represented as a binary sequence, taking a value of one when the series is observed and zero when an observation is missing. However, Parzen's initial work focused on cases where missing observations followed a systematic pattern. This approach was later extended by Scheinok, who addressed the occurrence of missing observations in a random fashion, modeling the amplitude modulation using a Bernoulli random scheme [24]. This approach has been picked up by several recent authors such as Flores, Manousopoulos and Mariethoz.

Flores developed a new method for imputing missing values in air pollution particulate matter time series by combining deep learning with traditional interpolation techniques. They approached the imputation task as a classification problem, categorizing missing values into two groups - those suited for polynomial interpolation and those better handled by flipped polynomial interpolation. This method was applied to hourly $PM_{2.5}$ data from Ilo City, Peru, and compared with models such as ARIMA, LSTM, GRU, and standard polynomial interpolation. The results showed that their approach outperformed these baseline models, particularly for shorter gaps in the data, with R^2 improvements ranging from 2.43% to 19.96%. This recent study illustrates how combining deep learning with traditional methods can significantly improve time series imputation, especially in the context of environmental data analysis [8].

In their work on financial time series modeling, Manousopoulos explored the application of Fractal Interpolation Functions (FIF) [16] as an alternative to traditional interpolation methods. They argued that conventional approaches, such as polynomial interpolation, often fail to adequately capture the inherent irregularities and

fluctuations in financial data, which tend to exhibit non-smooth behavior. By utilizing fractal interpolation, the authors proposed a method better suited to handle the complexities found in financial time series, such as stock prices or GDP figures. The study evaluated the effectiveness of FIF on several datasets, including Bitcoin prices, and compared its performance with commonly used models like ARIMA and GARCH. The findings demonstrated that FIF delivers a more precise fit, particularly for data characterized by volatility and sudden changes, suggesting that fractal interpolation could serve as a valuable tool in financial time series analysis where conventional methods may prove inadequate.

Gregoire Mariethoz introduced a feature-preserving method for interpolation and filtering of environmental time series, which addressed missing data and interference in continuous monitoring applications. Their non-parametric approach relies on iterative pattern matching, where corrupted or missing values are replaced using unaffected portions of the time series, known as training data. This method contrasts with traditional techniques, such as spline interpolation, which tend to smooth the data and may not preserve the signal's intrinsic variability and statistical relationships between variables. The results demonstrated that the method effectively reconstructs the spectral characteristics of the original signal and provides uncertainty estimates through multiple realizations. This approach is particularly valuable for applications where preserving the signal's inherent features is essential [17].

In their comprehensive review, Li and Heap explored [15] the performance of various spatial interpolation methods within environmental sciences, categorizing them into non-geostatistical, geostatistical, and hybrid approaches. The review emphasizes that no single interpolation technique is universally optimal, as each method's performance varies depending on factors such as the spatial distribution of data, sampling design, and the correlation between variables. For instance, while geostatistical

methods like kriging are well-suited for datasets with spatial autocorrelation, non-geostatistical approaches such as thin plate splines (TPS) and regression models offer advantages in scenarios where the spatial structure is less pronounced. Li and Heap also highlight the limitations of certain methods in handling complex or anisotropic datasets, reinforcing the importance of method selection based on specific data characteristics. This perspective aligns with other works, such as that of Mariethoz [17], who underscored the need for feature-preserving methods when dealing with environmental time series data. Together, these studies suggest that the choice of an appropriate interpolation technique should be tailored to the dataset in question, particularly in the presence of spatial or temporal irregularities.

In evaluating various time series interpolation techniques, Dr. Wesley Burr and his student Sophie Castel [6] introduced the `interpTools` package, a systematic framework for simulating time series with structured missingness. This allows for the evaluation of interpolation methods based on specific gap configurations and data characteristics. Their research primarily focused on the Hybrid Wiener Interpolator (HWI), comparing its performance with methods like the Kalman Filter (KAF) and Exponential Weighted Moving Average (EWMA). The study highlighted the HWI's robustness, especially when handling large gaps and nonstationary time series, where other methods tend to perform suboptimally. Castel and Burr's findings align with previous studies [12] that emphasize the variability in the performance of interpolation methods based on the structural properties of the time series. As such, this underscores the critical need to carefully select interpolation techniques that are well-suited to the specific dataset at hand.

The literature indicates that many methods are available for interpolating time series data, and their performance varies depending on the characteristics of the data. However, there have been few significant innovations in this area over the past two decades, apart from developing the Hybrid Wiener Interpolator (HWI) [3]. This

suggests a significant opportunity for further research on time series interpolation from both time-domain and frequency-domain perspectives. Exploring new approaches and improving existing methods could [5] significantly improve the accuracy and efficiency of interpolating time series data.

3 Background

This section outlines fundamental concepts and definitions for analyzing data using various methods and techniques. Note that this is not restricted to data of the type relevant to time series interpolation, The methods discussed here include quantitative, qualitative, and mixed methods. Quantitative methods involve the collection and analysis of numerical data, often using statistical techniques to identify patterns and relationships. Qualitative methods, on the other hand, focus on understanding the experiences, perspectives, and behaviors of research participants through in-depth exploration and interpretation. Mixed methods combine both quantitative and qualitative approaches, leveraging the strengths of each to provide a more comprehensive understanding of the research problem.

Each method has its own strengths and limitations, and researchers must carefully consider which is most appropriate for their research question and objectives. Quantitative methods are well-suited for testing hypotheses and identifying statistical relationships, while qualitative methods excel at exploring complex social phenomena and generating rich, contextual data. Mixed methods can provide a more holistic understanding by integrating the strengths of both approaches.

Additionally, this section will cover the ethical considerations, potential biases associated with each method, and strategies for mitigating these issues. For example, quantitative methods may be susceptible to sampling bias or measurement error, while qualitative methods may be influenced by researcher subjectivity or participant

reactivity. Understanding the various methods available for data analysis is crucial for producing reliable and valid research findings that can inform policy, practice, and future research.

3.1 Time Series Data

A time series is a set of random variables, observed over an observation index (often time), that can be classified into two main categories based on its nature: discrete and continuous. In the discrete form, observations are defined over a countable domain, while the continuous counterpart involves an infinite set of observations within any given interval.

For discrete time series, it is expressed as:

$$x_t = \{x_0, x_1, x_2, \ldots, x_{N-1}\} \quad \text{for } t = 0, \ldots, N-1 \quad \text{(discrete)} \tag{3.1}$$

whereas for continuous:

$$x(t) = \{x : t \in [0, N-1]\} \quad \text{(continuous)}. \tag{3.2}$$

In these equations:

- x_t is the value of the time series at time t,
- t is the discrete time index, where $t = 0, 1, 2, \ldots, N-1$,
- N is the total number of observations in the discrete time series, and
- $\{x_0, x_1, x_2, \ldots, x_{N-1}\}$ is the set of all observed values over the entire period.

Although the term 'time series' implies a temporal ordering, it is noteworthy that spatial coordinates may also organize such data sequences. However, for the purpose of this thesis, our focus remains exclusively on time-ordered sequences.

3.1.1 Continuous Time series

Continuous time series, often called analog series, represent functions of a continuous-time argument. In this context, the function's value is defined over infinitesimally small increments, offering infinite resolution. This characteristic signifies that between any two-time points, an infinite number of intermediate measurements exist, rendering the domain of an analog series an uncountable set.

This continuous nature of time series data presents distinct advantages and challenges in analysis. The unbounded, continuous domain requires specialized techniques for processing and interpretation, which will be discussed in subsequent sections. Additionally, we should note here that no continuous time series examples exist in reality due to the Planck constant implying that one some level all time series are discrete, even if the discretization time step is minuscule.

3.1.2 Discrete Time Series

Discrete-time series, characterized by countable domains, are derived from two primary sources: sampling from an analog process or recording inherently discrete events. These signals have diverse applications, ranging from stock closing prices to digitized audio waves to midday temperature recordings. Due to their discrete nature, these signals are commonly represented as step functions. They are often visualized using continuous line graphs for better interpretability. While the discrete nature simplifies certain aspects of analysis, it also poses challenges in processing and modeling.

Proper continuity is a theoretical ideal rather than a practical reality in physical phenomena. The act of measurement or observation inherently imposes discretization on what may seem continuous. Take, for instance, light, often considered a quintessential example of continuity, travels in discreetly quantized packets known as photons. This fundamental insight challenges the notion of true continuity in the

physical world.

In practical terms, as mentioned above, continuous functions are primarily a mathematical abstraction, serving as valuable tools for modeling and analysis. Thus, when dealing with time series data, it is often more pragmatic to adopt a discrete framework. This approach acknowledges the inherent discretization introduced by the act of measurement or observation, ensuring a more realistic representation of the underlying physical processes.

3.1.3 Sampling Frequency

Time series data, whether discrete or continuous, tracks how a process changes over time. In discrete samples, the accuracy of the approximation to the original process depends on the sampling rate (also known as sampling frequency), denoted as f_s. This rate is defined as the number of observations taken per unit of time:

$$f_s = \frac{1}{\delta t} \tag{3.3}$$

where: δt is the time interval between consecutive samples.

Take the example of ambient audio waves. To convert this sound into a digital format, a microphone samples the air pressure amplitude at regular time intervals. The sampling rate plays a crucial role here, as it determines the highest frequency that can be accurately represented with these evenly-spaced samples. If the sampling rate is too low, the resulting digital wave will have a frequency (or pitch) that is significantly lower than that of the original wave. This discrepancy, known as aliasing, is a real-world application of the concept of sampling rate.

3.2 Stochastic Processes

A fundamental aspect of time series analysis involves studying stochastic processes, which are systems whose evolution is governed by probabilistic rules. Mathematically, a stochastic process is explained as a collection of random variables ordered according to specific temporal or spatial coordinates:

$$\{X_t\}_{t=0}^{N-1} = \{X_0, X_1, X_2, \ldots, X_{N-1}\} \tag{3.4}$$

where, again, t is the discrete time index, $t = 0, 1, 2, \ldots, N-1$, and N is the total number of observations.

In our context, we will focus on those ordered by time. Stochastic processes generally describe systems that exhibit random variability. The values of these random variables are determined through observation. The observed dataset $\{x_t\}$ of size N, recorded over discrete time points $t = 0, 1, \ldots, N-1$, represents a specific realization of the stochastic process:

$$\{x_t\}_{t=0}^{N-1} = \{X_0 = x_0, X_1 = x_1, X_2 = x_2, \ldots, X_{N-1} = x_{N-1}\}. \tag{3.5}$$

The observed time series represents just one of an endless number of potential outcomes. This endless collection is known as the ensemble. The main interest is usually in identifying the statistical characteristics of the ensemble.

3.3 Analysis in the Time Domain

The analysis of time series data often involves examining its behavior in the time domain, providing insights into the temporal distribution of the stochastic process realization. A time-domain representation presents a concrete realization of the stochastic process, showcasing how the system evolves.

3.3.1 Stationarity

The properties of a time series in the time domain can be understood by analyzing its statistical properties as the series progresses. A process is considered **strictly stationary** if its finite-order probability distributions remain consistent over time. As a result, its statistical attributes, such as the mean and variance, are also independent of time.

A time series is considered weakly stationary (also known as wide-sense, second-order, or covariance stationary) if:

1. $\mathbb{E}[|X_t|^2] < \infty \quad$ for all $t \in \mathbb{Z}$
2. $\mathbb{E}[X_t] = m \quad$ for all $t \in \mathbb{Z}$
3. $\mathbb{E}[X_t X^*_{t+\tau}] = \mathbb{E}[X_{t+\nu} X^*_{t+\tau+\nu}] \quad$ for all $t, \nu, \tau \in \mathbb{Z}$

In this context, $\mathbb{E}$ stands for the expectation operator, and m is a constant value. In this thesis, the term 'stationarity' will be used to indicate the less restrictive form: finite variance; time-independent mean; and covariance dependent only upon the lag between the elements. The validity of many time series analysis techniques, including interpolation, depends on the assumption of stationarity. Several basic transformations are often employed to induce approximate stationarity in non-stationary time series:

1. Applying a power transformation, square root, or logarithmic transformation to the data can help stabilize non-constant variance or eliminate exponential trends.
2. Differencing the data—expressing the data as the difference between consecutive observations—will remove a linear trend:

$$D^1_t = D^0_t - D^0_{t-1}, \quad t = 0, 1, \ldots, N-1$$

Equation 3.7 illustrates the scenario where $\{D_t^0\} = \{x_t\}$ is differenced once to produce the differenced series $\{D_t^1\}$, known as first-order differencing. More generally, the d-th order differenced series is defined by:

$$D_t^d = D_t^{d-1} - D_{t-1}^{d-1}, \quad t = 0, 1, \ldots, N-1 \tag{3.6}$$

where $\{x_t\}$ is differenced d times until stationarity is achieved.

While such transformations may be sufficient to achieve stationarity, they pose challenges in maintaining data integrity. Specifically, the differencing procedure results in a loss of information, as $\{D_t^d\}$ will have d fewer observations than the original series. This can complicate the interpretation of subsequent analyses. However, as we will soon demonstrate, a significant advantage of the Hybrid Wiener Interpolator is its applicability to certain types of nonstationary data, eliminating the need for prior transformative manipulations [3].

3.3.2 Autocovariance & Autocorrelation

Many inferential testing methods in statistics operate under the assumption that observations in a sample are independent - meaning the occurrence of event A does not affect the probability of event B occurring. However, in time series analysis, sample values often exhibit a natural temporal order, leading to observations that are temporally closer being more likely to be dependent on one another as the process unfolds. Remember that the covariance between two random variables X and Y quantifies their joint variability, defined as:

$$\operatorname{cov}(X, Y) = \mathbb{E}\left[(X_i - \mu_x) \cdot (Y_i - \mu_y)\right] \tag{3.7}$$

with estimator

$$\hat{\operatorname{cov}}(X, Y) = \frac{\sum_{i=1}^{N} (X_i - \bar{X})(Y_i - \bar{Y})}{n} \in (-\infty, \infty). \tag{3.8}$$

In these, the μ_i elements are the theoretical distribution means, and the $\bar{X}$ elements are the sample estimates of the same. By scaling the covariance with the product of the sample standard deviations of X and Y, denoted as S_X and S_Y, we convert it into a more interpretable measure called the correlation. The correlation quantifies the strength of the linear relationship between X and Y:

$$\text{cor}(X, Y) = \frac{\text{cov}(X, Y)}{S_X S_Y} \in [-1, 1] \tag{3.9}$$

The autocovariance function (ACVF) and the autocorrelation function (ACF) build upon the principles of covariance and correlation to analyze the relationships between observations at different times within a stochastic process. The autocovariance function, denoted by $\gamma(\tau)$, is defined as follows:

$$\gamma(\tau) = \text{cov}[X(t), X(t + \tau)], \tau \in \mathbb{R}. \tag{3.10}$$

This includes the condition that $X(t)$ is stationary, leading to the definition of the autocovariance function. The variance function is a particular case of the ACVF when $\tau = 0$. By standardizing the ACVF, we obtain the autocorrelation function, $\rho(\tau)$:

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)} = \frac{\gamma(\tau)}{\sigma^2}. \tag{3.11}$$

Note that in practice the requirement that a time series be weakly stationary can be equated to requiring that the ACVF is independent of the time variable t.

3.4 Modelling a Time Series

Time series modeling involves analyzing sequential data to understand patterns and make forecasts. Key components include trend, seasonality, cyclic patterns, and noise. Stationarity is often necessary and can be achieved through transformations like differencing. Common models are Autoregressive (AR), Moving Average (MA), AutoRegressive Moving Average (ARMA), and AutoRegressive Integrated Moving Average

(ARIMA). The process includes identifying the model, estimating parameters, checking diagnostics, and forecasting. Time series modeling is widely applied in finance, economics, and environmental science, aiding in decision-making based on historical data.

A stationary time series can be formally described using the Wold decomposition, which breaks down the series into two uncorrelated processes: one deterministic and the other stochastic. This concept represents the idea that the outcome of any ordered process at a given time is both entirely predictable and inherently random. In different contexts, the deterministic part is often referred to as the signal, while the stochastic part is called the noise. The signal-to-noise ratio (SNR) assesses the relative power between the signal and noise:

$$\text{SNR} = \frac{\sigma^2_{\text{signal}}}{\sigma^2_{\text{noise}}},$$

as the ratio of the variances of each component.

The main objective of statistically analyzing such a process is to deconstruct the physical phenomenon into its components and analyze and describe each one separately. It's important to note that the focus isn't always solely on identifying the signal. Several types of stochastic processes can be used to model time series, and the following section will provide a brief discussion of the most pertinent ones for our purposes.

3.4.1 White Noise Process

A white noise process, also referred to as a purely random process, denoted as $\{Z_t\}$ (and abbreviated as WN), is a sequence of random variables that are mutually independent and identically distributed. In a Gaussian white noise process, these random variables are also assumed to be normally distributed with a zero mean and finite

variance σ_Z^2. The assumption of independence implies that [5]:

$$\gamma(k) = \text{cov}(Z_t, Z_{t+k}) = \begin{cases} \sigma_Z^2 & \text{if } k = 0 \\ 0 & \text{if } k = \pm 1, \pm 2, \dots \end{cases} \tag{3.12}$$

Consequently, Gaussian white noise processes exhibit strict stationarity.

3.4.2 Moving Average Process

A moving average process of order q, denoted as $MA(q)$, is defined by the equation:

$$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q} \tag{3.13}$$

for the set $\{\beta_0, \cdots, \beta_q\} \in \mathbb{R}$, typically, as fixed coefficients. In this model, X_t is expressed as a linear combination of latent random variables, where the predictors are outcomes of a purely random process.

3.4.3 Autoregressive Process

Let $\{Z_t\}$ be a purely random process with a mean of zero and variance σ_Z^2. The process $\{X_t\}$ is called an autoregressive process of order p, denoted as $AR(p)$, if:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t. \tag{3.14}$$

Similar to the MA process above, the set $\{\alpha_1, \cdots, \alpha_p\} \in \mathbb{R}$ is a set of fixed coefficients. This equation represents a linear combination of random variables, where X_t is regressed on its past values. The coefficients α_k and β_k can be determined computationally by solving the Yule-Walker equations. The specifics of this algorithm are discussed in Chapter 8.1 of the book by Brockwell & Davis [2].

3.5 ARMA Models

An ARMA process of order (p, q) is the combination of an AR(p) and MA(q) process, given by the equation:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q} \tag{3.15}$$

3.6 ARIMA models

As mentioned earlier, a simple strategy for achieving stationarity is to difference the data. If X_t is replaced by $W_t = \nabla^d X_t$ in Equation 3.12, where ∇^d is a d-th order difference operator applied to X_t, then an ARIMA model of order (p, d, q) can be written as [2]:

$$W_t = \alpha_1 W_{t-1} + \cdots + \alpha_p W_{t-p} + Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q} \tag{3.16}$$

which is equivalent to an ARMA(p, q) model, but with d^{th} order differencing.

3.7 Other Models

A number of other models exist, continually stacking complexity on top of the framework of the ARIMA. The *ARIMAX* model combines the autoregressive (AR), integrated (I), and moving average (MA) components of ARIMA with additional external regressors, making it useful for forecasting when external influences are present. The *Seasonal Autoregressive Integrated Moving Average (SARIMA)* model is an extension of the ARIMA model that incorporates a seasonal component to handle time series data exhibiting periodic behavior. While ARIMA models are well-suited for non-seasonal data, they are often insufficient for datasets where seasonality plays a key role in the underlying patterns. The SARIMA model addresses this by integrating

seasonal differencing and seasonal autoregressive and moving average terms.

AutoRegressive Conditional Heteroskedasticity (ARCH) models are commonly used for modeling time series with time-varying volatility, especially in financial data where the variance of returns tends to change over time. The model assumes that the current variance of a time series depends on past variances. These are further extended, to account for persistent volatility in time series, especially in financial markets, as the *Generalized AutoRegressive Conditional Heteroskedasticity* (GARCH) models, extending the ARCH model by incorporating lagged conditional variances. In a GARCH model, the current variance depends not only on past squared residuals but also on its own previous values. Other models such as *Threshold Autoregressive (TAR)* models (nonlinear extension of the autoregressive (AR) model that allows different regimes based on the value of a threshold variable) and *Vector Autoregressive (VAR)* models (extends the univariate autoregressive (AR) process to the multivariate case) have also been developed, especially by practitioners in the econometrics field, to tackle complex nonlinear nonstationary data from economic and finance sources.

3.7.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to handle sequential data by maintaining a memory of previous inputs. Unlike traditional feedforward neural networks, RNNs have loops that allow information to persist, making them well-suited for tasks involving time-dependent data, such as time series forecasting and interpolation.

An RNN processes sequential data step by step, with each step depending on both the current input and the hidden state from the previous step. This can be mathematically expressed as:

$$h_t = f(W_x x_t + W_h h_{t-1} + b)$$

where:

- h_t represents the hidden state at time t,
- x_t is the input at time t,
- W_x and W_h are weight matrices for the input and hidden state, respectively,
- b is the bias term, and
- f is the activation function (typically a non-linear function such as tanh or ReLU).

RNNs are particularly effective in time series applications because they can learn dependencies across time steps. This makes them useful for both short-term and long-term forecasting tasks, where the relationship between data points evolves over time. In particular, RNNs are potentially useful for:

- **Time Series Forecasting**: RNNs can model sequential patterns in time series data, allowing them to predict future values based on past observations. However, due to their limited ability to retain long-term dependencies, RNNs can struggle with data that involves extended temporal relationships, making them less effective for some complex time series tasks compared to more advanced models like LSTMs.
- **Time Series Interpolation**: RNNs are also used for interpolation tasks, where missing data points in a time series are predicted based on the context provided by surrounding data. Although RNNs can capture short-term temporal patterns, they may struggle with interpolating missing values over longer periods due to their inherent limitation in handling long-term dependencies.

Despite their ability to model sequential data, RNNs suffer from the *vanishing gradient problem*, where gradients diminish during backpropagation through time, limiting their effectiveness in capturing long-range dependencies.

3.7.2 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to overcome the limitations of traditional RNNs, particularly their inability to retain long-term dependencies in sequential data. Introduced by Hochreiter in 1997 [10], LSTMs are equipped with memory cells that can store information over long periods, making them highly effective for tasks involving sequential data, such as time series forecasting and interpolation.

In contrast to simple RNNs, which suffer from issues like vanishing gradients when learning long-range dependencies, LSTM networks introduce gates that regulate the flow of information. The three primary gates in an LSTM cell are:

- **Input Gate**: Controls how much of the new input is stored in the memory cell.
- **Forget Gate**: Determines how much of the previous memory should be retained.
- **Output Gate**: Controls how much of the memory is used to compute the output at the current time step.

The cell state, which is modified by these gates, acts as a conveyor belt that stores information, allowing the LSTM network to retain relevant information over long time horizons.

LSTM networks are highly versatile and have been applied to a wide range of time series tasks, including:

- **Time Series Forecasting**: LSTMs are particularly effective in forecasting tasks, where they can capture both short-term and long-term dependencies in the data, such as seasonal patterns or trends.

- **Time Series Interpolation**: LSTMs can also be used for interpolation, where missing values in a time series are predicted based on the surrounding data points. The ability of LSTMs to learn temporal patterns makes them well-suited for filling in missing values in irregularly sampled or noisy time series data.

The architecture of LSTM networks allows them to effectively handle non-stationary time series data, making them a valuable tool for both interpolation and forecasting in complex real-world applications such as finance, weather prediction, and healthcare. Their ability to learn patterns from data without requiring strict assumptions about stationarity or linearity makes them a powerful alternative to traditional time series models like ARIMA.

LSTMs have recently been shown [4] to have strong potential for interpolation of time series, and so may be worth considering in this work.

3.8 Analysis in the Frequency Domain

A time series can be analyzed not only in the time domain, but also in the frequency domain. Understanding the relationship between these two perspectives relies on fundamental principles from Fourier Theory.

The Fourier transform (FT) changes a time-domain function into a frequency-domain function

For a discrete function $x(t)$ representing data with equal spacing, the discrete FT is defined as:

$$y(f) = \sum_{t=0}^{N-1} x(t) e^{-i2\pi ft}, \tag{3.17}$$

with the inverse given by:

$$x(t) = \int_{-1/2}^{1/2} y(f) e^{i2\pi ft} \tag{3.18}$$

The functions $y(f)$ and $x(t)$ are related as a pair of Fourier transforms, where $y(f)$ represents the Fourier transform of $x(t)$. It is important to observe that this transformation acts as a trivially sufficient statistic, meaning that the transform operation preserves all the original information without any loss [28]. The FT serves as the mathematical bridge between the time domain and the frequency domain.

Any function or waveform can be approximated using a Fourier series

This implies that functions which are sufficiently smooth can be approximated over a finite interval by a weighted sum of sinusoids with progressively increasing frequencies [22]. This finding suggests that any time series can be represented by a finite sum of oscillatory functions. Naturally, the focus may shift towards identifying the underlying frequencies, which can be uncovered through the calculation of the spectrum.

3.8.1 Harmonic Analysis

Harmonic analysis is a mathematical technique used to study and represent functions or signals as the combination of fundamental waves, typically sine and cosine functions. Harmonic analysis in the context of time series involves breaking down a time series into its fundamental frequency components, often called *line components*.

This process helps identify periodic patterns in the data, making it a valuable tool for signal processing, climatology, and economics. By transforming the time series into the frequency domain using methods such as Fourier Transform, harmonic analysis provides insights into the cyclical behavior and underlying periodicities, enabling more accurate modeling, forecasting, and interpretation of complex time series data.

3.9 Significant Interpolation Techniques

There is a wide variety and complexity of methods available for repairing discontinuities in time series data. The subsequent section provides an overview of several interpolation algorithms that will be employed in our later comparative study. The choice of these interpolators is influenced by the findings of Van Bussel [29] and Castel [5].

3.9.1 Polynomial Interpolation

This class of interpolators includes various methods, with one of the most straightforward being linear interpolation, which connects known data points by constructing a straight line between them:

$$\hat{x}_t = \frac{x_A - x_B}{(a - b)}(t - b) + x_B$$

where a and b represent the index positions of the endpoints x_A and x_B across a given gap, and $\hat{x}_t$ is the interpolated value at index t. This method can be extended to higher-order degrees to account for nonlinear phenomena. Various algorithms fall under this class of interpolators, all sharing the same fundamental approach but differing in the degree of the polynomial used and the continuity of their derivative functions [12].

3.9.1.1 Splines

Raising the degree of the interpolating polynomial does not necessarily lead to greater accuracy. In fact, it may result in non-convergence and increased error due to excessive oscillations near the endpoints of the interval — a phenomenon known as Runge's Phenomenon [23], and similar in behaviour to Gibb's phenomenon for Fourier series. To address this issue, a special type of piecewise polynomial known as a spline can be employed. Splines consist of multiple low-degree polynomials that are connected and pass through the existing data points. A widely used option is the cubic spline, which utilizes polynomials of degree no greater than three. Cubic splines are favoured for their ease of implementation, and within this category, several methods are available.

1. **Natural cubic splines** are characterized by the condition that their second derivatives at the endpoints are zero, which causes the ends of the spline to taper off smoothly into a straight line.

2. **Hermite cubic splines** are employed when there is a need to specify the slope at the endpoints of each segment, allowing control over the smoothness of the piecewise polynomial. This method requires that the derivatives at the endpoints be known up to a certain order, resulting in an interpolant with a continuous first derivative.

3. **The FMM cubic spline** method, introduced in 1977 by Forsythe, Malcolm, and Moler [9], determines the end conditions by fitting an exact cubic through four points at each boundary of the interpolation regions. Utilizing spline techniques instead of high-degree polynomials not only reduces interpolation error and enhances convergence properties, but also helps to prevent overfitting.

3.9.2 The Kalman Filter

The Kalman filter is a recursive algorithm that predicts past, present, or future values by integrating measurement data from a time series, which may include statistical noise and inaccuracies, with prior knowledge of the system. It produces estimates of unknown variables by computing their joint probability distribution at each point in time [11].The algorithm operates in two stages. During the prediction step, it estimates the current state variables and their uncertainties. When a new measurement is observed, the estimates are updated using a weighted average, with the weights determined by the confidence in each estimate. The filter is considered 'optimal' because it minimizes statistical error [30].

The fundamental assumption is that the process being analyzed can be sufficiently described by a linear model [18]. In this case, the true state at time k can be modeled based on the previous state at time $k-1$, as follows:

$$\hat{x}_k = F_k x_{k-1} + B_k u_k + w_k$$

where F_k represents the state transition model applied to the previous state x_{k-1}, and B_k is the control-input model applied to the control vector u_k. The term w_k is assumed to be zero-mean multivariate Gaussian noise with covariance Q_k, such that $w_k \sim \mathcal{N}(0, Q_k)$. If this assumption does not hold, a shaping filter can be applied to ensure that the model remains valid. The use of a linear model is often preferred due to its practicality [18]. In practice, many real-world dynamical systems, particularly in engineering, exhibit nonlinear behavior. To address this, the Kalman filter has been adapted for nonlinear applications. One simple variant is the Kalman ARIMA Filter (KAF), which is quite commonly used as a time series interpolator.

3.9.3 Exponential Weighted Moving Average

The Exponentially Weighted Moving Average (EWMA) algorithm is a distance-based weighting method used to estimate missing values by computing a weighted average of the previous k observations, where $1 \leq k \leq t-1$. The weights decrease exponentially as t approaches zero. The interpolated value is calculated as:

$$\hat{x}_t = \frac{(1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + \cdots + (1-\alpha)^k x_{t-k}}{1 + (1-\alpha) + (1-\alpha)^2 + \cdots + (1-\alpha)^k}$$

In this method, α takes a value within the range $[0, 1]$, representing the rate at which the weights decrease, or how quickly older observations are given less importance. While economists have traditionally used this approach to predict stock prices based on historical data, it tends to perform poorly when applied to highly volatile time series. For the purpose of this study, the algorithm is applied retrospectively to a set of realized data, serving as an interpolator rather than a forecasting tool. Other moving average methods, such as the Simple Moving Average and the Linear Moving Average, use constant or linearly decreasing weighting factors. However, preliminary research by Van Bussel [29] demonstrated that exponential weighting is the most effective of the three approaches for interpolating univariate time series, as discussed by Castel [5].

3.10 The Hybrid Wiener Interpolator

The Hybrid Wiener interpolation procedure, developed by Wesley Burr [3], is an iterative algorithm within an Expectation-Maximization (EM) framework. It is designed to impute missing samples in a time series. This method involves a three-part decomposition of the time series into the mean, periodic components, and residual noise. Leveraging each aspect effectively improves the accuracy of the interpolated values.

Much of the framework of the Hybrid Wiener algorithm is focused on transforming

a stochastic process to and from a stationary state. The underlying philosophy is that a time series can be modeled as a stochastic process with a stationary background and a complex periodic additive structure. By adopting this approach, trends and periodic components can be estimated and subsequently removed, uncovering the underlying stationary process. This transformation is crucial for accurately imputing missing values and enhancing the reliability of the interpolation results.Once the series has been made stationary, we apply the classic Wiener interpolating procedure. This step utilizes the established statistical properties of the stationary process to accurately estimate and fill in the missing values in the time series.The Hybrid Wiener algorithm has a notable advantage in its flexibility, as it does not heavily rely on prior assumptions. This makes it particularly versatile and applicable to a wide range of time series data, especially those of a scientific nature. The following provides a concise overview of the entire algorithm.

3.10.1 The E Step

The algorithm starts with the **E** (expectation) step, where an initial estimate of the power spectrum for the time series $\{x_t\}_{t=0}^{N-1}$ is calculated using the multitaper technique [27]. This multitaper power spectrum is then inverted to produce a consistent estimate of the ACVF, which will be utilized later in the interpolation step. Note that the multitaper spectrum estimation itself requires contiguity. Therefore, the first iteration of this algorithm will necessitate some form of crude interpolation of a copy of the series to fill the missing data in some way. The specific choice of interpolator for this step is not crucial, as it is assumed that the final estimates will eventually converge. This contiguous pilot series is denoted as $Z_t^{(0)}$.

3.10.2 The M Step

The Maximization (M) step is considerably more complex. The core of the algorithm lies in how a model for a general time series is conceptualized:

$$X_t = M_t + T_t + \xi_t \tag{3.19}$$

In this model, M_t is conceptualized as a fully deterministic polynomial function of time, representing the mean of the series. The term T_t is defined as a fully deterministic finite linear combination of sinusoids, which captures the periodic trend component. The residual ξ_t is modeled as a generalized zero-mean wide-sense stationary stochastic process, specifically considered here as an auto-regressive (AR) process of unspecified order. This modeling approach is particularly advantageous as it provides the flexibility to represent complex, real-world scientific time series that go beyond the simplistic assumption of stationary stochastic processes. The key task, therefore, involves estimating each component of the series: determining the degrees and coefficients of M_t, identifying the amplitudes and phases of T_t, and estimating the realization of ξ_t.

3.10.3 Estimating M_t: The Mean Component

The mean component is divided into two parts: a constant, non-varying element and a time-varying polynomial trend. The constant mean is estimated through the zeroth frequency transfer functions derived from the Slepian sequences, while the polynomial trend is approximated by performing a polynomial expansion using a Slepian projection basis [26, 28, 27]. Once the mean component is estimated, it is subtracted from the time series, resulting in a zero-mean process.

3.10.4 Estimating T_t: The Trend Component

In the following step, sinusoidal trends are estimated and separated from the background noise. This is achieved using a spectrum-based algorithm called the harmonic F-test [27], which identifies significant frequencies through harmonic analysis. The eigencoefficients of the series are utilized to obtain estimates of the amplitude and phase of the line components.

3.10.5 Sub-Iteration of M_t and T_t Estimation

The estimation of M_t and T_t is repeated on the residual series until it converges according to a chosen metric. In this context, the convergence criterion is the maximum pointwise difference between successive iterations:

$$\max\left\{\max_t\left|M_t^{(j)} - M_t^{(j-1)}\right|, \max_t\left|T_t^{(j)} - T_t^{(j-1)}\right|\right\} < \epsilon \tag{3.20}$$

for a suitable ϵ. Once M_t and T_t are sufficiently estimated, they are removed from X_t to yield an approximately white series.

3.10.6 The Interpolation Step

The final task in the M step involves using the best mean-square error linear interpolator [2] on the residual series to estimate the missing data points. This step produces the approximately white, zero-mean contiguous series W_t.

3.10.6.1 Global Iteration

The missing values are substituted with $\hat{x}_t = M_t + T_t + W_t$, resulting in the new contiguous interpolated series $\hat{Z}_t^{(1)}$. This series is then reintroduced into the E step, where the multitaper power spectrum and ACVF are re-estimated.It is important to note that the missing values interpolated during the final stage of the M step are solely

Global Algorithm (Hybrid Wiener Interpolator)

Initialize Step

1. Linearly interpolate the gappy original data series, label this $Z_t^{(0)}$.
2. Compute a pilot estimate of the spectrum and invert to obtain the pilot ACVF.
3. Estimate $\hat{M}_t^{(0)}$ using $Z_t^{(0)}$ as input.
4. Estimate $\hat{T}_t^{(0)}$ using $Z_t^{(0)} - \hat{M}_t$ as input.
   - **Fix the frequencies obtained in this zeroth step for the duration of the algorithm**.
   - Iterate the M_t, T_t sequence until convergence, obtaining final estimates of $\hat{M}_t^{(0)}$ and $\hat{T}_t^{(0)}$.
5. Estimate $\hat{W}_t^{(0)}$ using $Z_t^{(0)} - \hat{M}_t^{(0)} - \hat{T}_t^{(0)}$ as input.
6. Take $\hat{M}_t^{(0)}$, $\hat{T}_t^{(0)}$ and $\hat{W}_t^{(0)}$ and add them together to give $\hat{Z}_t^{(1)}$. (**E** step)

Convergence Step k

1. Estimate the spectrum (and corresponding ACVF) of $\hat{Z}_t^{(k)}$. (**E** step)
2. Estimate $\hat{M}_t^{(k)}$ using $\hat{Z}_t^{(k)}$ as input. (**M** step)
3. Estimate $\hat{T}_t^{(k)}$ using $\hat{Z}_t^{(k)} - \hat{M}_t^{(k)}$ as input.
   - Iterate the M_t, T_t steps until convergence, obtaining final estimates of $\hat{M}_t^{(k)}$ and $\hat{T}_t^{(k)}$.
4. Estimate $\hat{W}_t^{(k)}$ using $\hat{Z}_t^{(k)} - \hat{M}_t^{(k)} - \hat{T}_t^{(k)}$ as input.
5. Take $\hat{M}_t^{(k)}$, $\hat{T}_t^{(k)}$ and $\hat{W}_t^{(k)}$ and add them together to give $\hat{Z}_t^{(k+1)}$. (**E** step)
6. Evaluate the maximum absolute difference δ between $\hat{Z}_t^{(k+1)}$ and $\hat{Z}_t^{(k)}$. If $\delta > \epsilon_{\text{fixed}}$ then iterate.

used to derive a new estimate of the power spectrum and are not carried forward to subsequent iterations.The algorithm continues to iterate until it converges, halting when the maximum absolute difference δ between $\hat{Z}_t^{(k)}$ and $\hat{Z}_t^{(k+1)}$ is less than ϵ_{fixed}. Upon meeting this criterion, the final estimate of the interpolated series is taken to be $\hat{Z}_t^{(k+1)}$.

3.11 Interpolation Algorithms Considered

In this thesis, we will begin with an initial set of algorithms for interpolation, sourced from [29] and [5], who in turn drew inspiration from [12] – one of the most comprehensive works on interpolation from the last decade.

Most of these are simple algorithms: NN (Nearest Neighbour) fills gaps with the closest known data point; LI (Linear Interpolation) connects points on sides of a gap with a straight line fit; LOCF (Last Observation Carried Forward) does just as it says, as does NOCB (Next Observation Carried Backward), and SMA (Simple Moving Average). We discussed the KAF (Kalman Filter using ARIMA) above, along with the EWMA (Exponential Weighted Moving Average) and HWI (Hybrid Wiener Interpolator).

All of these algorithms have been implemented in `interpTools` and are trivially applied to data containing gaps. We will discuss results from the application of these algorithms in the following chapters. Also note that this table is included in the thesis a second time as an appendix for ease of reference.

Table 3.1: A set of interpolation algorithms that have are supported innately in `interpTools`, as provided by [5]. Algorithms considered in [5] (shown in bold) were based on the results of Van Bussel [29]; we primarily consider those underlined in this work.

Package	Function	Algorithm name
`interpTools`	`nearestNeighbor()`	Nearest Neighbor (NN)
`zoo`	`na.approx()`	Linear Interpolation (LI)
`zoo`	`na.spline()`	**Natural Cubic Spline (NCS)**
`zoo`	`na.spline()`	**FMM Cubic Spline (FMM)**
`zoo`	`na.spline()`	**Hermite Cubic Spline (HCS)**
`imputeTS`	`na_interpolation()`	Stineman Interpolation (SI)
`imputeTS`	`na_kalman()`	**Kalman - ARIMA (KAF)**
`imputeTS`	`na_kalman()`	Kalman - StructTS (KKSF)
`imputeTS`	`na.locf()`	Last Observation Carried Forward (LOCF)
`imputeTS`	`na.locf()`	Next Observation Carried Backward (NOCB)
`imputeTS`	`na_ma()`	Simple Moving Average (SMA)
`imputeTS`	`na_ma()`	Linear Weighted Moving Average (LWMA)
`imputeTS`	`na_ma()`	**Exponential Weighted Moving Average (EWMA)**
`imputeTS`	`na_mean()`	Replace with Mean (RMEA)
`imputeTS`	`na_mean()`	Replace with Median (RMED)
`imputeTS`	`na_mean()`	Replace with Mode (RMOD)
`imputeTS`	`na_random()`	Replace with Random (RRND)
`tsinterp`	`interpolate()`	**Hybrid Wiener Interpolator (HWI)**

4 Research Framework

To properly understand the workings of interpolation methods, we must subject them to a variety of tests. The rationale behind testing these methods through simulations lies in our goal to draw comparisons between the true values and the estimates provided by interpolation tools. Therefore, our first step involves creating a simulation to test different time series analysis techniques. To accomplish this, we will utilize the `interptools` package for the `R` programming language developed by Sophie Castel [5] in 2020, and discussed further in [6]. Theoretical results for interpolators have been derived for simple cases (such as single missing points or limited count forecasts), but are technically very challenging in real-world cases due to the unknown and highly complex joint probability distribution of the time series. Thus, using simulations is a strategic approach that allows us to examine interpolation processes and compare their performance against known outcomes, using a tool specifically designed for such analyses.

4.1 Summary of the structure of `interptools`

The package `interptools` is a research framework for evaluating the performance of the Hybrid Wiener Interpolator [3] on complex time series data with varying gap structures [5]. It highlights two main objectives: assessing the interpolator's performance under different conditions and comparing it to other sophisticated interpolators

available in R. The package documentation outlines a systematic approach involving generating a time series, applying a gap structure, interpolating the data, and comparing the interpolated series to the original. The package allows users to simulate time series data with specific mean, trend, and noise components, impose gap structures, interpolate the gappy data, and evaluate performance using a range of metrics.

4.2 Functions

Several custom functions were developed for this work to facilitate the analysis and visualization of time series interpolation methods. These functions were designed to streamline key tasks such as loading and splitting datasets by method, summarizing data, and generating both static and interactive visualizations. The core functions include `loadAndSplitData`, which efficiently organizes the dataset by dividing it into subsets based on different interpolation methods, and `summarizeSplitData`, which computes summary statistics for each subset, allowing for a more focused analysis of key metrics like the MSE. Visualization functions, such as `heatmapp` and `plotlySurfacePlotWithColors`, provide dynamic and insightful visual representations, using heatmaps and interactive 3D surface plots to highlight performance differences among the methods. These tools enable a comprehensive, structured, and efficient approach to data handling and analysis, ensuring clarity and precision in the comparison of interpolation techniques. Together, these functions contribute to a robust framework for analyzing and evaluating time series interpolation methods.

4.2.1 Splitting the simulation Data

The `loadAndSplitData` function is a critical part of our research framework's data preprocessing stage. It loads a dataset and divides it into subsets based on specified grouping variables. This allows for independent analysis of each subset, which is

useful for comparing the performance of different methods under various parameter settings.

4.2.2 Summarizing the split data frame

The `summarizeSplitData` function was designed for the data analysis stage. It splits the dataset based on 'Method', 'P', and 'G', using the `loadAndSplitData` function. These parameters are in-built arguments for the `interpTools` package, and describe interpolation method, proportion missing, and gap length, respectively. The function then summarizes each subset using a specified summary function. The result is a summarized data frame providing key statistics for each method and parameter combination in the dataset. This data can be further processed or visualized to compare different methods under various conditions.

4.3 Data Visualization Techniques

We further developed various data visualization techniques to effectively represent time series data and the results of different interpolation methods. Data visualization is essential for the analysis process as it helps in identifying patterns, trends, and anomalies within the data and conveying complex information. We will start by looking at fundamental visualization tools such as surface plots, which are crucial for illustrating time series data and comparing interpolation results. Additionally, we will explore more advanced visualization methods, including heatmaps and interactive dashboards, which provide dynamic and multidimensional views of the data. These advanced techniques allow for a more detailed exploration and interpretation of the data, enabling users to interact with and manipulate visual representations to gain deeper insights.

4.3.1 Surface Plots

Surface plots are a popular way of visualizing three-dimensional data in a clear and easy-to-understand manner, allowing researchers and analysts to gain valuable insights into complex relationships and patterns within their datasets. One of the key advantages of surface plots is their ability to effectively communicate the uncertainty and variability inherent in the data, which is crucial for making informed decisions and drawing accurate conclusions [25].

Plotly is a robust, open-source data visualization library that excels in creating interactive, high-quality graphs suitable for both web-based and offline applications. It supports a diverse range of chart types, including line plots, scatter plots, bar charts, histograms, heatmaps, and 3D surface plots, making it exceptionally versatile for comprehensive data analysis and presentation. One of Plotly's key strengths is its interactive capabilities, allowing users to zoom, pan, and hover over data points to reveal additional insights, which enhances the exploration and understanding of complex datasets. We use Plotly in our functions to do the actual generation of the surfaces.

Visualization of the surface plots are generated using the developed function `createMultiMethodSurfacePlot`, where the user can specify the methods of interest and the specific colours for each method. The function summarizes the data, filters it by the specified methods, and then creates a combined surface plot with unique colour scales for each method. The user can easily compare the different methods within a single plot. This approach provides an integrated and visually distinct representation of the specified methods, allowing for an effective comparison of their performance.

4.3.2 Heatmaps

Heatmaps are a versatile data visualization technique used to display the magnitude of data values across a two-dimensional grid. By utilizing a colour gradient, heatmaps allow for the quick identification of patterns, trends, and anomalies. Each cell is coloured according to its value, facilitating intuitive analysis of complex datasets. Widely used in fields such as finance, biology, and geography, heatmaps effectively convey correlations, density distributions, and other critical metrics, making them essential for data scientists and analysts. In the following chapters we use the `heatmapp` function to convert three-dimensional surfaces into heatmap.

4.4 Simulations

Simulations are crucial because they enable researchers and practitioners to model complex systems, predict outcomes, and conduct safe, cost-effective experiments in a controlled environment. They provide flexibility to explore multiple scenarios, including extreme and rare events, and support iterative development and optimization. Simulations enhance understanding and education by visualizing concepts and providing interactive learning experiences. Additionally, they generate valuable data for analysis, hypothesis testing, and benchmarking different methods. Ultimately, simulations assist decision-making processes by providing insights into potential outcomes and enabling informed choices across various fields.

We choose to use simulations instead of real world data to evaluate the performance of interpolators for several reasons. Firstly, simulations provide a controlled environment where all variables are known and can be manipulated, allowing for precise analysis and experimentation. They are also reproducible, ensuring that other researchers can verify and validate results. Simulations offer flexibility, enabling the

study of various scenarios, including rare or extreme events that might not be sufficiently represented in real data. Additionally, simulations are cost-effective and time-efficient, avoiding the expense and logistical challenges of collecting real-world data. Finally, as discussed in Chapter 3, time series can be modeled in a number of coherent ways, allowing us to use these models to take real-world time series and develop replica time series for simulation purposes, with the replicas sharing fundamental properties with the real-world examples, but also being contiguous, and therefore suitable for evaluation of performance of an interpolator.

Having said this, we start our results in Chapter 5 with a real-world example, to demonstrate what is possible using these methods, and emphasize the necessity of having access to contiguous, flexible examples via simulation for full understanding of the performance of different interpolators.

Note that for all simulations considered in this thesis, for sake of computational load, we have considered a restricted set of P, G and K parameters for the interpolators. P refers to the proportion of the time series missing, with restricted values $P \in \{0.1, 0.2, 0.3, 0.4\}$, G refers to the maximum length of a gap, with restricted values $G \in \{5, 10, 25, 50\}$. The number of replicates per (P, G) pairing will be set to $K = 250$ for all simulations.

4.4.1 Interpolators Considered

As mentioned earlier in Chapter 3, a number of interpolators will be considered in the following, as detailed in Appendix A. In particular, the most reasonable that are retained for much of the analysis are the Kalman ARIMA Filter (KAF), the Exponentially Weighted Moving Average (EWMA), and the Hybrid Wiener Interpolator (HWI), which were all discussed in the previous. All specifics of other acronyms are left to the appendix for brevity.

5 Analyzing the Sunspot Dataset

5.1 Sunspots and Their Characteristics

Sunspots are temporary, dark features on the Sun's surface that appear darker than the surrounding areas. The darker regions result from localized decreases in surface temperature, caused by concentrated magnetic fields that disrupt the Sun's convection processes. Sunspots typically occur within active regions on the Sun and often appear in pairs with opposite magnetic polarities. The number of sunspots varies systematically with the approximately 11-year solar cycle, which is driven by the Sun's internal magnetic dynamo.

The study of sunspots has provided valuable insights into the mechanisms that drive solar activity. The historical record of sunspots, dating back to the early 17th century when they were first observed through telescopes, offers a direct way to characterize long-term changes in solar activity, or 'space climate'. Analyzing sunspot patterns and variations can help scientists better understand the complex processes occurring on the Sun's surface and within its interior, which have significant impacts on the Earth's atmosphere and climate. As such, contiguous sunspot records are of significant value for scientific endeavours, and are a useful time series to be able to interpolate.

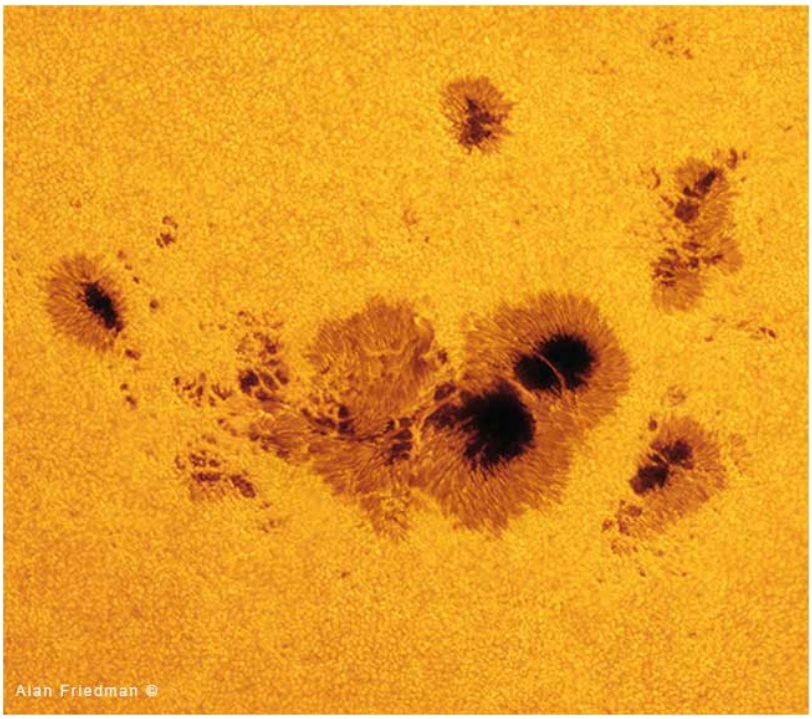

Figure 5.1: Closeup of sunspots using a white light solar filter. The number of sunspots varies on the surface of the Sun peaking at solar max in the 11 year sunspot cycle. Picture taken by amateur astronomer Alan Friedman and processed extensively. [1]

5.1.1 Sunspots dataset

The sunspot dataset is a historical record of sunspot counts observed on the Sun's surface, often spanning several centuries. The dataset includes the date of observation and the sunspot count, which fluctuates significantly over time. Sunspot data is scientifically compelling for interpolation due to its extensive historical record and predictive value. Sunspots influence space weather, affecting Earth's climate and technology, so accurate interpolation is essential for understanding and mitigating these impacts. The irregular sampling of sunspot observations poses challenges requiring advanced interpolation techniques. The cyclic nature of sunspot activity, following an approximately 11-year solar cycle, also necessitates sophisticated methods to capture these patterns. Accurate interpolation enhances the completeness and reliability of

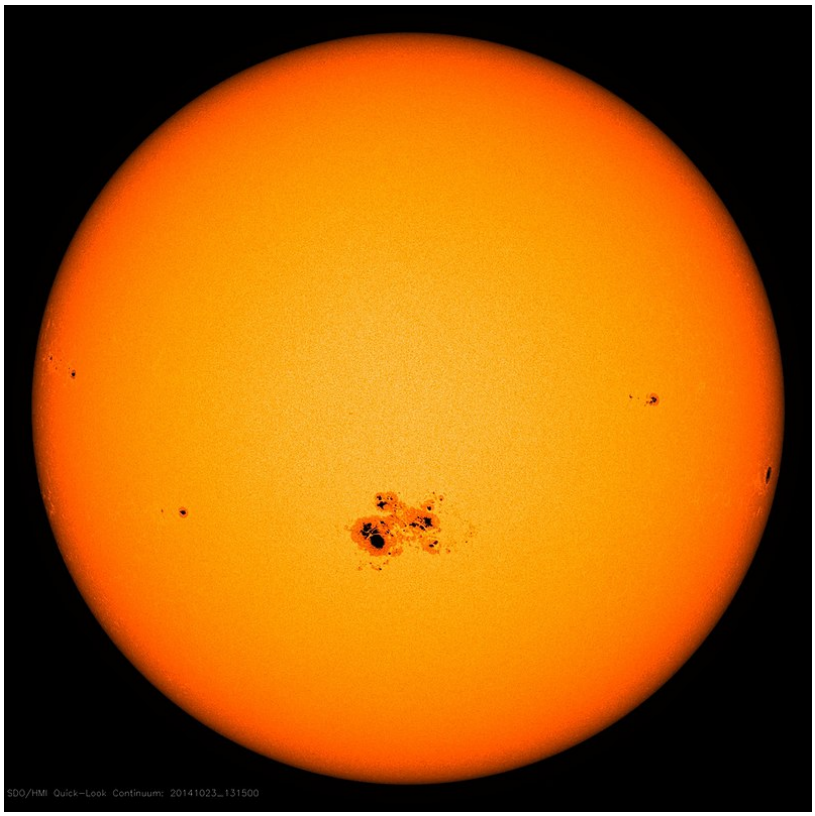

Figure 5.2: On Oct. 18, 2014, a sunspot rotated over the left side of the sun, and soon grew to be the largest active region seen in the current solar cycle, which began in 2008. Currently, the sunspot is almost 80,000 miles across – ten Earths could be laid across its diameter. Sunspots point to relatively cooler areas on the sun with intense and complex magnetic fields poking out through the sun's surface. Such areas can be the source of solar eruptions such as flares or coronal mass ejections. Picture obtained from [19].

sunspot datasets, aiding research on solar dynamics, magnetic field variations, and astrophysical phenomena. These factors make sunspot data interpolation vital for constructing reliable models and advancing understanding of solar behavior.

5.2 Interpolators

For this analysis, we will consider a large number of interpolators supported by the `interpTools` package innately. These include all interpolators detailed in Appendix A, with initialisms provided there.

To analyze the performance of interpolation methods on this target data set, we will first visually examine the generated plots to observe the overall patterns and trends. This includes identifying how closely each interpolator approximates the original data, the smoothness of the interpolated lines, and the presence of any anomalies or artifacts. We will then compare the interpolators based on specific performance metrics such as accuracy, which can be quantified using error measures like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or Mean Absolute Percentage Error (MAPE). By carefully evaluating these factors, we can decide the strengths and weaknesses of each interpolator on this class of time series, and the most suitable will be selected as our primary methods.

5.3 Performance Analysis and Comparison

This section presents a series of plots to visually compare the performance of the set of supported interpolators (NN, LI, NCS, FMM, HCS, SI, KAF, KKSF, LOCF, NOCB, SMA, LWMA, EWMA, RMEA, RMED, RMOD, RRND, HWI, NNI). Our analysis found that several interpolation methods, including RRND, RMOD, NCS, RMEA, RMED, and FMM, performed inconsistently and produced unsatisfactory results across different scenarios. As a result, these methods are not recommended for further use. Moving forward, we will focus on the interpolation methods that demonstrated consistent and reliable performance. By concentrating on these sufficient interpolators, we aim to enhance the overall reliability and accuracy of our interpolation results, leading to more dependable and precise outcomes in future applications.

The overall performance is depicted using surface plots, with each surface illustrating the performance of a specific interpolation algorithm. These individual surfaces are merged into one comprehensive plot, showcasing the collective performance of the

interpolation methods on a single dataset.

5.4 Surface Plot Analysis

In this section, the evaluation of the selected interpolation methods — KAF, HWI, EWMA, FMM, NCS, and HCS — is conducted through a series of detailed visual analyses. Surface plots are utilized to represent the performance of each interpolation algorithm, with the surfaces capturing key aspects of their effectiveness across the same dataset. The best of these methods is the Hybrid Wiener Interpolator (HWI), so it is used as the benchmark for comparison to the other methods. Each plot represents a direct comparison between HWI and another method, allowing for a visual evaluation of their respective strengths and weaknesses.

5.4.0.1 Summary of Comparison

The 3D surface plot comparing HWI and EWMA reveals that the HWI surface is significantly less elevated and dynamic across the various proportion missing ($p = 0.1$ through 0.4),gap length ($G = 10$ through 50) and $k = 250$. This is true for all plots (FMM, HCS, NCS, and RMOD) excepting KAF. Consider Figure 5.3, bottom pane, where there is some evidence of the KAF having superior error performance (that is, reduced MSE) over the HWI for low values of P and higher values of G. This indicates that while the HWI may be a strong method, the KAF can produce similar or superior results under certain circumstances. The slope of the surfaces does indicate that the KAF is extremely vulnerable to increases in missingness, however - as P increases, the rate of increase of the KAF surface (in green, Figure 5.3) increases exponentially.

All five other plots show dominant performance from the HWI, with methods FMM, NCS and HCS being quite bad, method RMOD being quite horrific, and

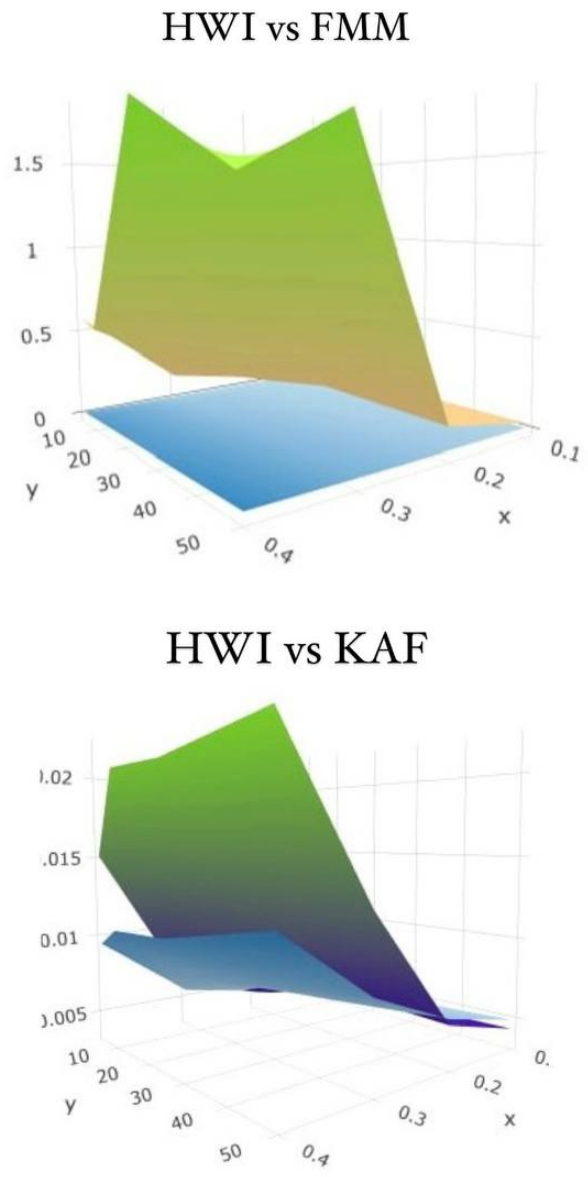

Figure 5.3: Comparison of the HWI with various interpolators (FMM and KAF). Each 3D surface plot illustrates the performance differences between HWI and the respective interpolation method, highlighting HWI's smooth and adaptive behavior across various data variations. The HWI consistently demonstrated superior performance compared to the other methods.

EWMA showing evidence of being a less capable variant of KAF. That is to say, none of the methods were *good*, and should not be considered for interpolation of classes of time series similar to the sunspot data.

The visual comparison of the six interpolation methods shows that the Hybrid Wiener Interpolator (HWI) is the most adaptable. HWI captures a wider range of data variations without overfitting or underfitting. Methods like EWMA and RMOD tend to smooth the data too much, while HCS and NCS risk overfitting because of their fluctuations. KAF and FMM perform well but don't match HWI's balance of accuracy and flexibility. Overall, HWI is the most reliable and effective method for capturing data trends while staying stable.

5.5 Heatmap Analysis of Method Efficiency and Accuracy

In this section, we present a detailed analysis of the performance of various interpolation methods using heatmap visualizations. These heatmaps provide a comprehensive visual representation of the accuracy of each method under a range of different conditions. The horizontal and vertical axes of the heatmaps represent two key parameters: G, which denotes the gap size, and P, which represents the proportion of missing data within the dataset. The colour intensity in each cell of the heatmap indicates the performance of the respective interpolation method, with lighter colours representing better performance and darker colours indicating poorer performance. This visual analysis allows for a quick and intuitive comparison of the strengths and weaknesses of the different interpolation techniques across various scenarios of data sparsity and missing values.

5.5.1 Interpretation of Heatmaps

Each heatmap corresponds to a different interpolation method, providing a comparative visualization of their performance. All results are shown in Figures 5.6-5.8.

SMA, SI, NOCB, NNI, and LI

These methods generally show lighter colours across the heatmaps, indicating relatively good performance with smaller gap sizes and lower proportions of missing data. However, as the gap size and proportion of missing data increase, the performance tends to degrade, evident from the gradual darkening of the heatmap cells.

NCS and RRND

These methods exhibit a noticeable decline in performance with increasing gap sizes and proportions of missing data. The dark red areas in their heatmaps indicate significant performance issues under these conditions. This suggests that NCS and RRND are less robust to larger gaps and higher levels of missing data.

RMOD, RMED, RMEA

These methods show intermediate performance, with some resilience to varying gap sizes and missing data proportions. Their heatmaps display a mix of light and dark cells, suggesting that while they perform adequately in some scenarios, they struggle in others.

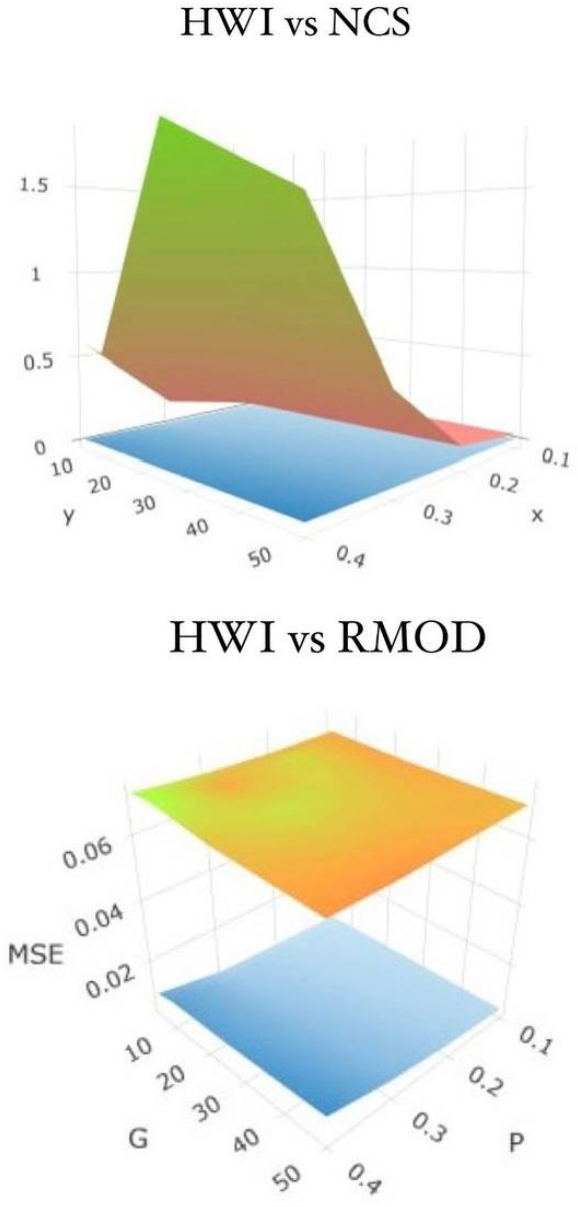

Figure 5.4: Comparison of HWI Interpolation Method with Various Interpolators (NCS and RMOD). In these plots, the blue surface is the HWI, the yellow is the RMOD, and the Green to pink variant is NCS. The 3D surface plots illustrate the performance differences between HWI and the respective interpolation methods, highlighting HWI's smooth and adaptive behavior across various data variations. The HWI consistently demonstrated superior performance compared to these methods.

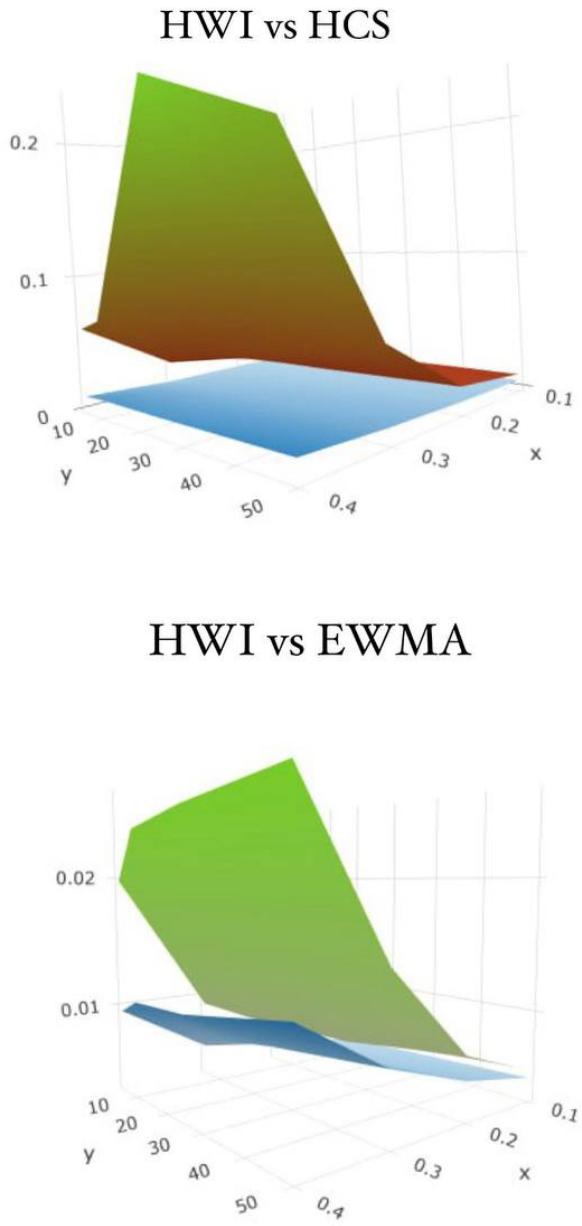

Figure 5.5: Comparison of the HWI with various vnterpolators (HCS and EWMA). The blue is the HWI, the green the EWMA and the green-to-red variant of colours is HCS. The 3D surface plots illustrate the performance differences between HWI and the respective interpolation methods, highlighting HWI's smooth and adaptive behavior across various data variations. The HWI consistently demonstrated superior performance compared to these methods.

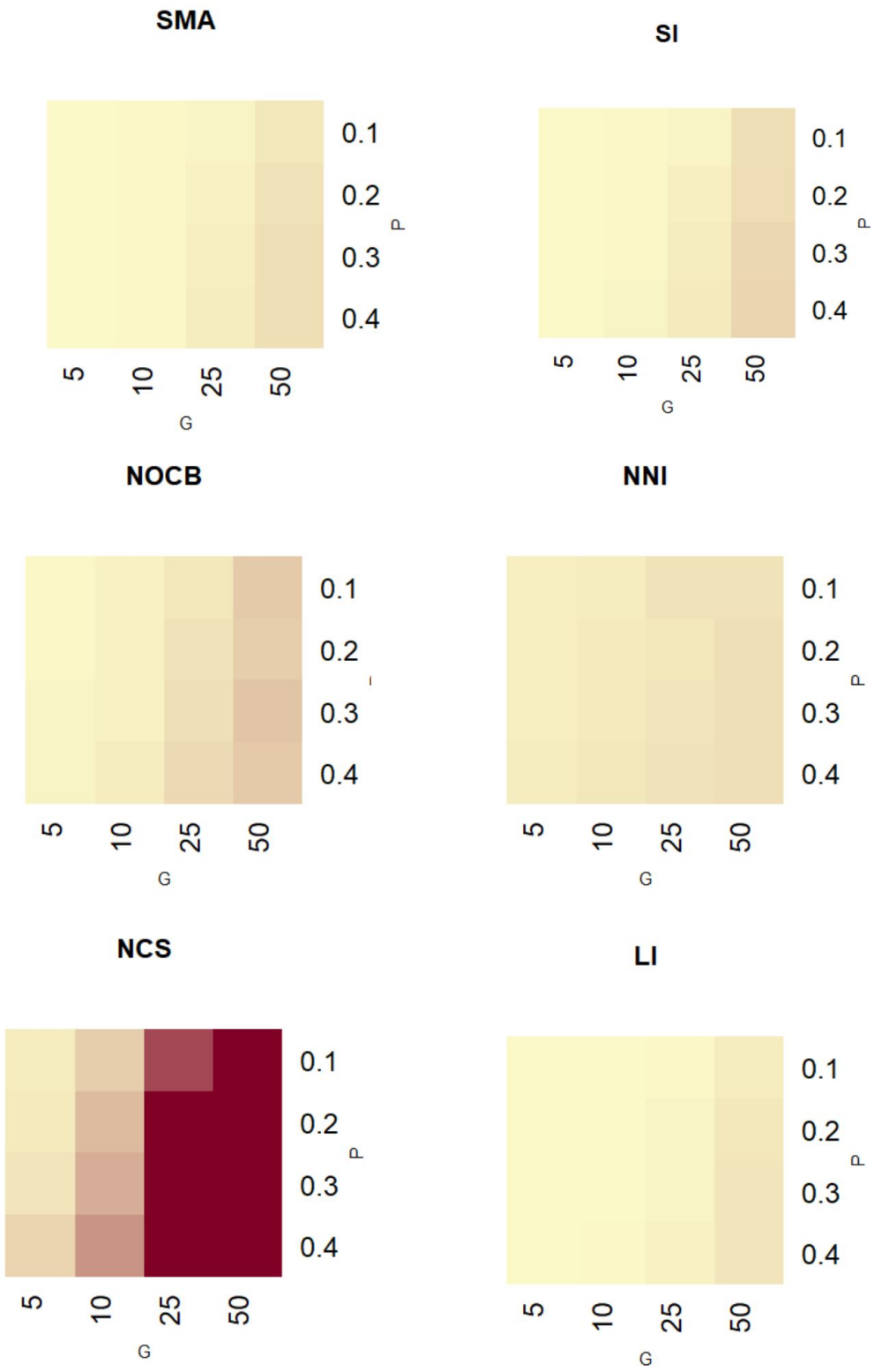

Figure 5.6: Six methods of interpolation shown in heatmaps.Note that the colours indicated in these heatmaps are compressed because the overall range from lowest to highest MSE is so large, it makes it difficult to show variation.

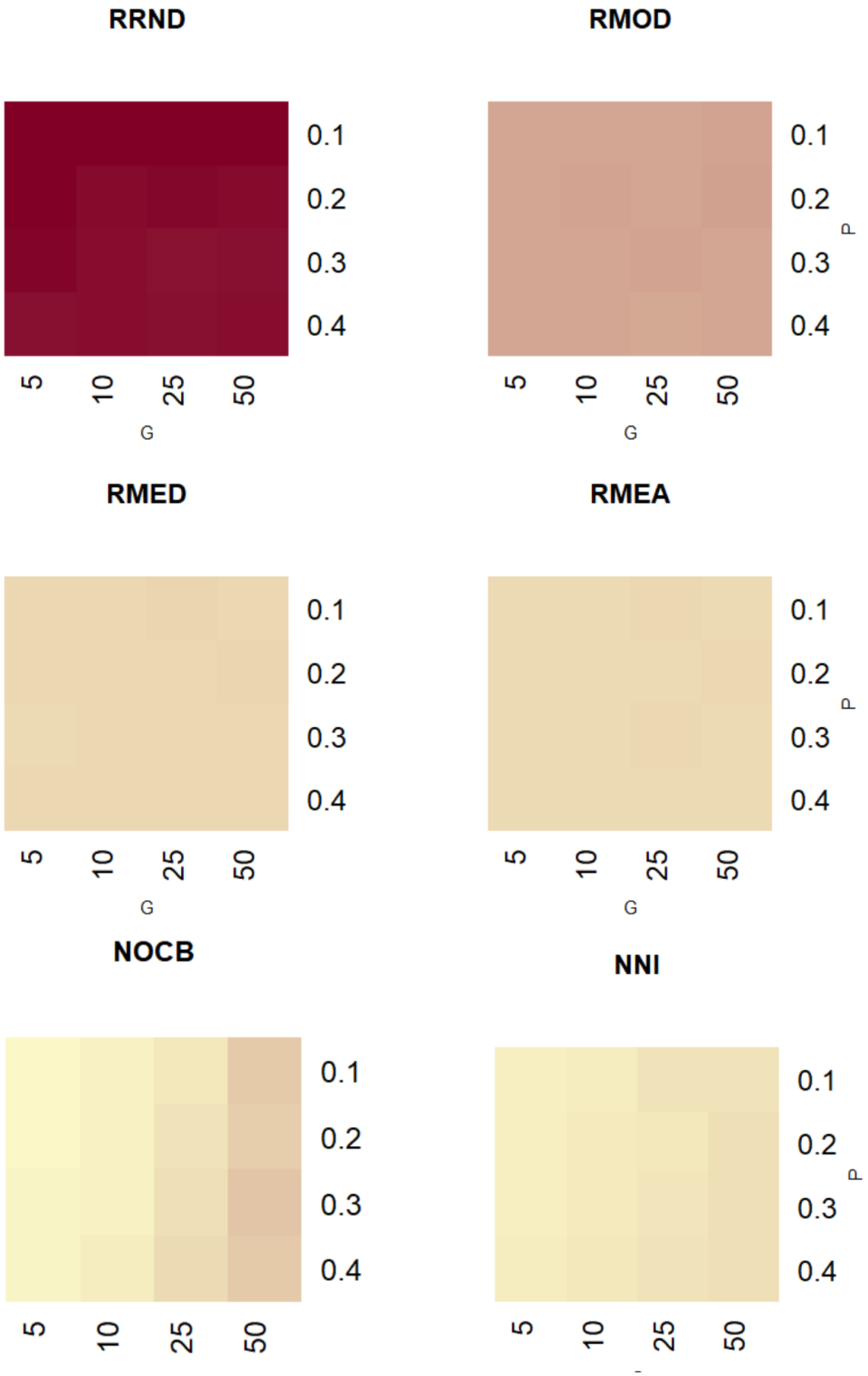

Figure 5.7: Six methods of interpolation shown in heatmaps.Note that the colours indicated in these heatmaps are compressed because the overall range from lowest to highest MSE is so large, it makes it difficult to show variation.

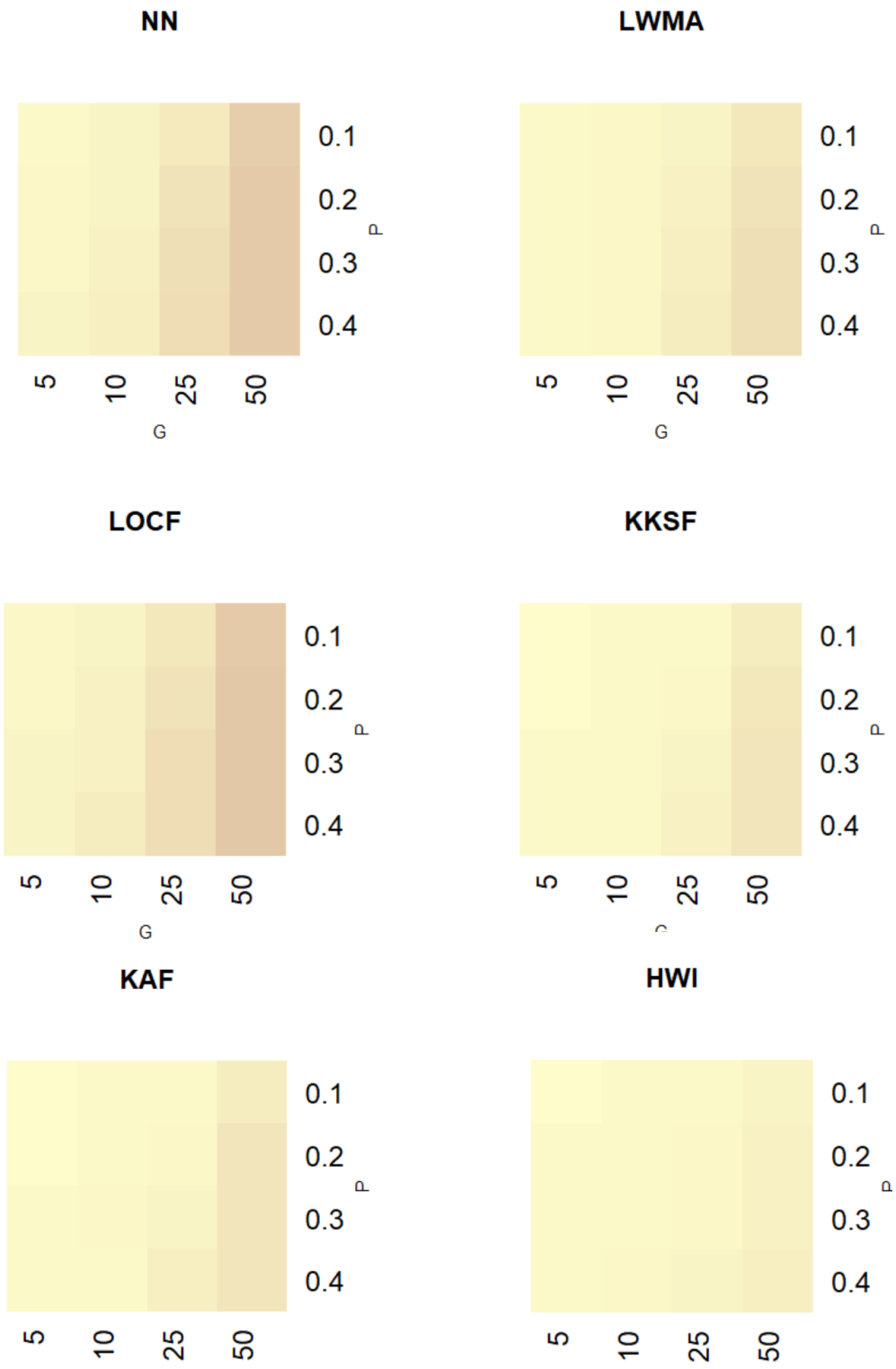

Figure 5.8: Six methods of interpolation shown in heatmaps.Note that the colours indicated in these heatmaps are compressed because the overall range from lowest to highest MSE is so large, it makes it difficult to show variation.

NN, LWMA, LOCF, KKSF, KAF, and HWI

The Hybrid Wiener Interpolator (HWI) method along with other sophisticated methods like KAF and KKSF, demonstrate more consistent performance across different conditions. The heatmaps for these methods show lighter cells across a wider range of gap sizes and missing data proportions, indicating their robustness and reliability

5.5.2 Comparative Performance

From the heatmap analysis across the 18 heatmaps contained in Figures 5.6 - 5.8, it is evident that the HWI method performs consistently well across various scenarios. Unlike simpler methods that degrade significantly with increasing gap sizes and missing data proportions, HWI maintains a high level of accuracy. This is particularly notable in scenarios with larger gaps and higher proportions of missing data, where HWI outperforms other methods, as indicated by the lighter cells in its heatmap.Note that the colours indicated in these heatmaps are compressed because the overall range from lowest to highest MSE is so large, it makes it difficult to show variation

5.6 Conclusion

Based on the results presented in here for this real-world scientific data set, we can come to some general conclusions and recommendations for practitioners. First, the HWI seems to be a consistent and reasonably robust method for time series of this class: that is to say, time series with strong periodic structure contained in them, although potentially against a background of cyclo- or non-stationary noise. Second, many of the classic methods are essentially useless and should be completely discounted. Of all 18 methods tested on this fairly simple example of a real-world series, only the Kalman ARIMA Filter (KAF) and Exponentially Weighted Moving

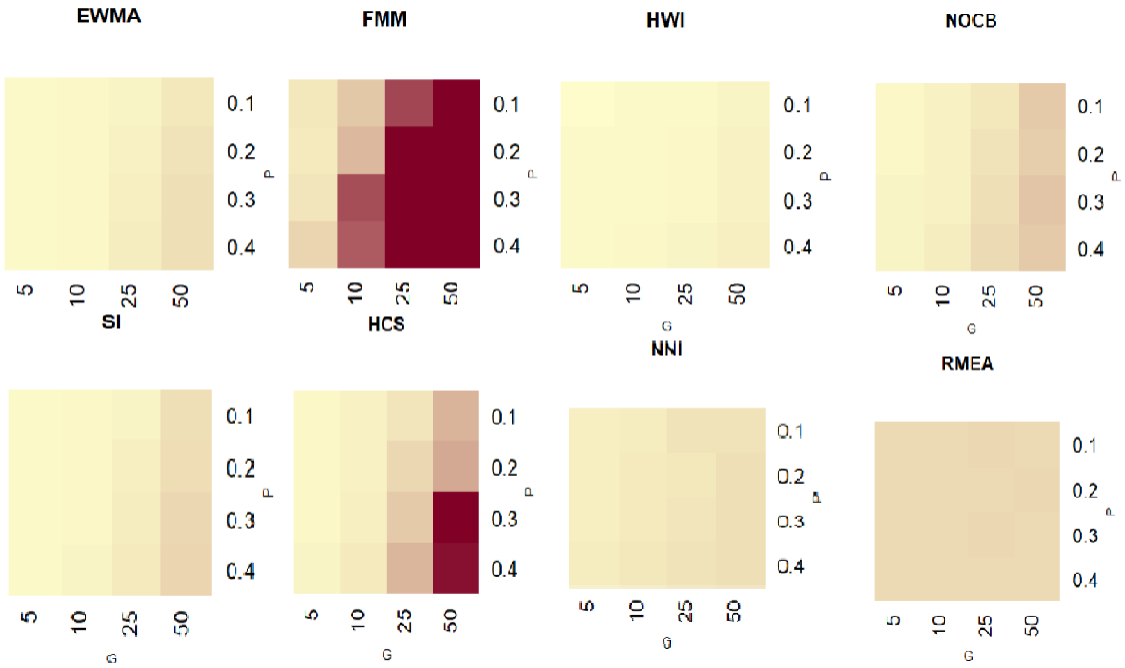

Figure 5.9: The MSE values for the $K = 250$ simulated interpolations are displayed for each (P, G) gap structure applied to the original datasets, with colour intensity indicating the value and scaled across the entire dataset. Each row shows the performance of a specific interpolation method (EWMA, FMM, HWI, NOCB, SI, HCS, NNI, RMEA), while each column represents a different dataset. HWI consistently outperformed the other methods across all datasets and gap structures.

Average (EWMA) showed performance reasonable enough to consider recommending them for general use. All other methods performed poorly on at least one sub-area of missingness considered.

This raises an interesting question, that we will consider in the following chapter: at what point does the HWI cease to perform well, when considered against a gradient of changing noise? That is, as the Signal-to-Noise ratio decreases, and noise becomes increasingly dominant in a time series, at what point does the HWI begin showing degradation? We will consider this in the following chapter.

6 Varying Signal-to-Noise Ratio Simulations

6.1 Signal-to-Noise Ratio

In the context of scientific research, the pursuit of high-quality data and reliable results is of utmost significance. Gathering data with a high signal-to-noise ratio (SNR) is a crucial aspect that enables researchers to obtain accurate and meaningful insights. SNR is a measure employed in science and engineering to compare the magnitude of a desired signal to the level of background noise. It is defined as the ratio of signal power to noise power, frequently expressed in decibels (dB). A ratio greater than 1.0 (or greater than 0 dB) indicates that the signal is stronger than the noise, which is desirable when trying to examine signals in data, and for drawing valid conclusions. Researchers emphasize the importance of maintaining a high SNR in various applications, as it is essential for ensuring the validity and reliability of scientific findings, e.g., [31]. An example of varying SNR when applied to two-dimensional image processing is shown in Figure 6.1.

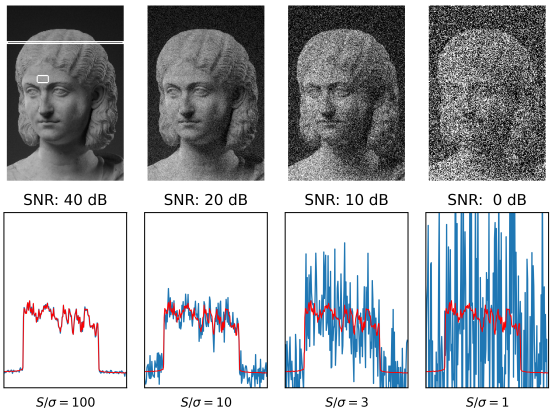

Figure 6.1: This figure presents a grayscale image subjected to varying levels of signal-to-noise ratios (SNR). The SNR values correspond to the rectangular region highlighted on the subject's forehead. The plots below each image depict the intensity profile along the indicated row, with the original signal in red and the noisy signal in blue.

6.2 The Simulation Environment

To generate the simulation time series, we use a function implemented in `interpTools`, `simXt`. In this function, data are generated component-wise based on an additive model for a general time series:

$$x_t = m_t + t_t + \xi_t$$

where m_t is a fully deterministic polynomial function of time, representative of the mean of the series, t_t is a fully deterministic finite linear combination of sinusoids representing a periodic trend component, and ξ_t is a generalized zero-mean wide-sense stationary stochastic process. To start, we consider a designated SNR of 2.0, which is an additional parameter able to be set in the function.

The arguments specified within `simXt` are piped into three sub-functions within the package, each generating the components independently as numeric vectors of specified length, and then combined to give x_t. The parameter ψ is used internally in the package to denote the number of independent periodic components simulated as

part of t_t: for all analyses in this chapter we have used $\psi = 50$ as a mid-point between 'no signals' and 'many signals' – real-world physical science datasets can have dozens, hundreds, or even tens of thousands of actual periodic signals embedded in them.

6.3 Performance Analysis and Comparison

This section provides a comprehensive visual comparison of the performance of the three selected interpolation methods for the SNR 2.0 case: Kalman ARIMA Filtering (KAF), Exponentially Weighted Moving Average (EWMA), and Hybrid Wiener Interpolator (HWI). As a reminder, these three methods were the only methods to have performed at a sufficiently acceptable level across the sunspot simulations presented in Chapter 5. The analysis that follows employs heatmaps and surface plots to illustrate the behavior of each method across varying conditions and datasets. By examining these visualizations, key performance metrics and trends become more apparent, enabling a deeper understanding of the strengths and limitations inherent to each approach. This comparative analysis serves as a foundation for evaluating which interpolator is most suitable for different scenarios and datasets in time series analysis.

6.4 Efficiency and Accuracy: Visualization

As discussed above, we consider an example series created using the `interpTools` framework with SNR set to 2.0, and 50 randomly selected periodic signals embedded in a white noise background and a quadratic polynomial trend. The performance of the three selected interpolation methods (KAF, HWI and EWMA) are visually compared using heatmaps (Figure 6.2). For this simulation, we considered three levels of each of P and G: $P = 0.1, 0.2, 0.3$ and $G = 5, 10, 20$, with $K = 250$ replicates per

(P, G) pairing. As a reminder, this means missingness of 10, 20 or 30%, and gap lengths of 5, 10 or 20 points.

In the following, we use colour as a proxy for performance, as the heatmaps are standardized against a common colour scheme. Lighter, more yellow colours are low, while darker, more magenta colours are high. Low in this case is good, as the metric being used is the Mean-Square Error (MSE). Thus, low MSE means accurate interpolation and corresponds to more yellow colours; high MSE corresponds to magenta.

6.4.1 KAF Performance

The KAF method shows a moderate level of performance, sitting between HWI and EWMA. The heatmap reveals a mix of medium shades, indicating moderate errors across different configurations. While KAF is more sensitive to parameter changes than HWI, it still offers relatively stable results with fewer significant variations. This balance between stability and adaptability places KAF as a reasonable middle-ground option, suitable for scenarios that require both reliability and some level of flexibility.

6.4.2 HWI Performance

The HWI method demonstrates the best performance among the three methods, as indicated by the brighter, more yellow, and uniform shades in its heatmap. The low intensity of the colours suggests minimal interpolation errors across various grid sizes and noise levels. HWI's ability to maintain low errors consistently makes it the most reliable choice, regardless of changing data conditions across P and G. The smooth colour gradients further underscore its stability for this high SNR case.

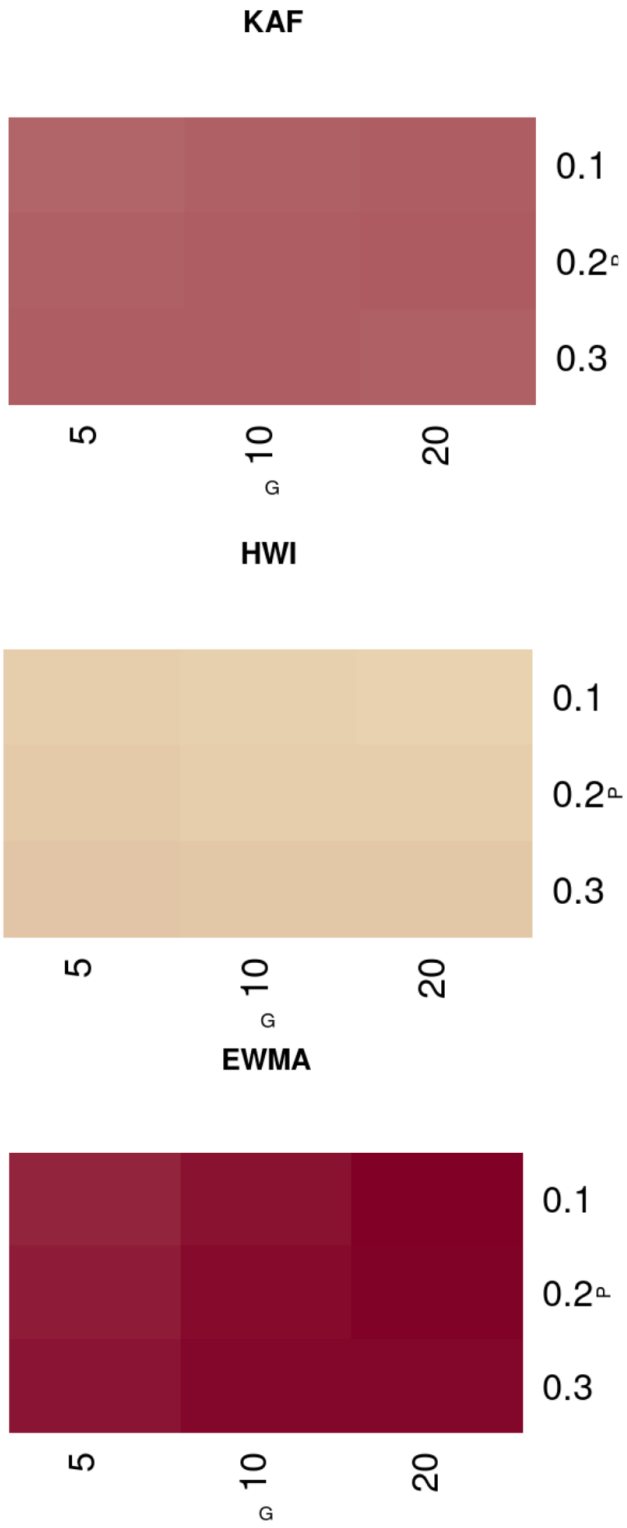

Figure 6.2: Performance Comparison of KAF, HWI, and EWMA methods for a starting simulation with SNR of 2.0.Note that the colours indicated in these heatmaps are compressed because the overall range from lowest to highest MSE is so large, it makes it difficult to show variation.

6.4.3 EWMA Performance

The EWMA method clearly performs the worst, as reflected by the darker shades in its heatmap, which signal higher interpolation errors. The method shows a noticeable sensitivity to both noise and grid size, leading to inconsistent performance. Although it can achieve reasonable results under ideal conditions (high grid density and low noise), its overall instability and higher error levels make it the least favourable option. The variability in the heatmap colours suggests that EWMA struggles to maintain accuracy across a range of settings, making it the least reliable method in this comparison.

In conclusion, the heatmap analysis highlights HWI as performing the best in this case, which is to say, a highly structured time series with periodicities, and a SNR of 2.0. This is perhaps not surprising, as this is the type of time series the HWI was designed to work very well on.

We also present a 3D variant of the previously displayed heatmaps in Figure 6.3, which reinforces that the HWI is consistently superior in its error structure over the other two methods.

6.4.4 Effect of SNR on Performance

This section presents a visual and quantitative comparison of three interpolation methods: KA, HWI, and EWMA, across varying levels of SNR. As mentioned earlier, a reasonable question is: does decreasing SNR correspond to decreasing (and eventually, unacceptable) performance by the HWI. The 3D surface plots are shown in Figure 6.4, and illustrate the performance of these methods by again displaying the Mean Squared Error (MSE) as a function of the interpolation parameters P and G under varying levels of signal-to-noise ratio (SNR). The SNR values in this analysis range from 0.5 to 2.0, providing insights into how the noise level in the data influences

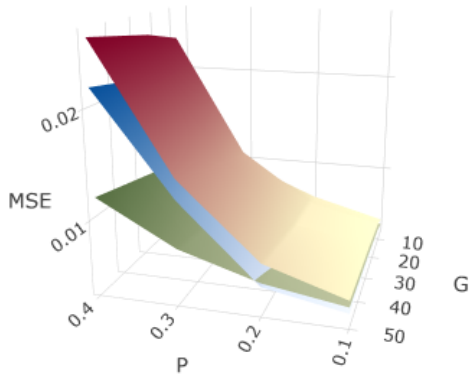

Figure 6.3: 3D Surface Plot Comparing Interpolation Methods: KAF(Blue), HWI(Green), and EWMA(red variate) for SNR 2.0 and the parameters discussed in Chapter 6.4. The plot visualizes the Mean Squared Error (MSE) as a function of parameters P and G, illustrating the performance of each method. The surface heights reflect the error levels, with KAF and HWI showing lower errors compared to EWMA, indicating their superior adaptability in high signal-to-noise ratio data. The colours here do not match the heatmaps, as they do not have a common scale: green is the HWI; blue is the KAF; and Red/Yellow gradient is the EWMA.

the effectiveness of each interpolation technique.

6.4.4.1 General Observations

Upon examining the surface plots across varying SNR levels, it is clear that the HWI consistently outperforms both KAF and EWMA in terms of maintaining lower MSE values. The error surfaces associated with HWI are consistently lower across all parameters, indicating its superior robustness and adaptability to the noise present in the data. This consistent performance suggests that HWI is the most reliable method among the three, particularly when handling both moderate and high SNR values.

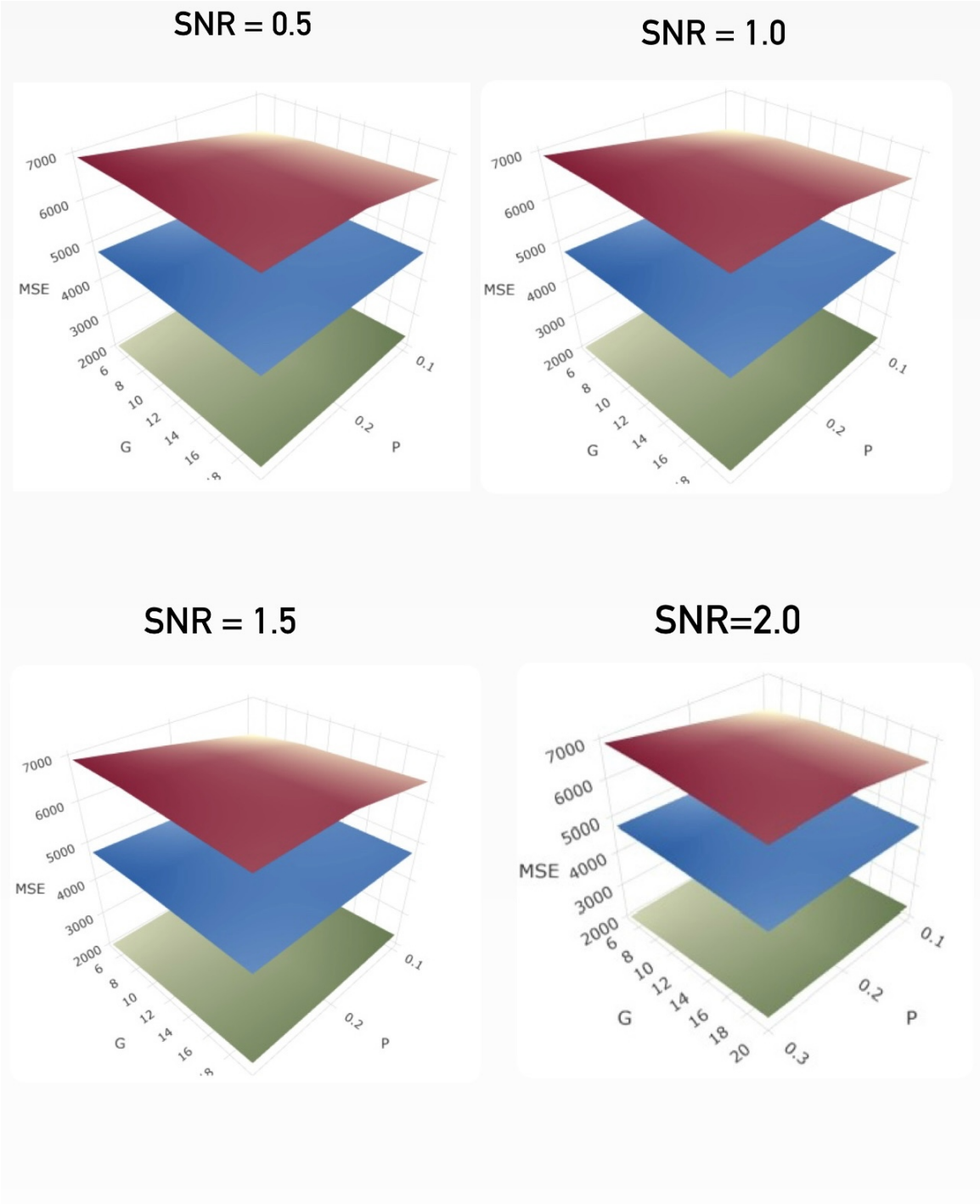

Figure 6.4: 3D Surface Plot Comparing Interpolation Methods: KAF, HWI, and EWMA Across Different Signal-to-Noise Ratios (SNRs). The plots visualize the Mean Squared Error (MSE) as a function of parameters P and G for each SNR level (0.5, 1.0, 1.5, and 2.0). The surface heights represent the error levels, with HWI consistently exhibiting lower errors compared to EWMA and KAF, highlighting better performance and adaptability across the varying levels of SNR.

While KAF also demonstrates relatively strong performance in some cases, its MSE values are slightly higher than those of HWI, particularly in regions where the parameters P and G increase. EWMA, in contrast, exhibits the highest MSE values overall, indicating that it is more sensitive to noise and less effective in adapting to varying noise levels compared to HWI and KAF.

It is also important to note that, when viewed broadly, the differences in the performance of all three methods across the different SNR levels are not as pronounced as one might expect. While there is a general improvement in MSE as SNR increases, the relative ranking of the methods remains consistent, and the variation in error surfaces between SNR values of 0.5 and 2.0 is relatively modest. This suggests that, in practice, the choice of SNR may not drastically alter the relative performance of the interpolation methods.

6.5 The HWI across Different SNR

We now consider only the performance of HWI under different SNRs. The individual 3D surface plots are provided in Figure 6.5, and again visualize the MSE as a function of the interpolation parameters P and G across the four different SNR levels: 0.5, 1.0, 1.5, and 2.0. This plot offers a demonstration of how the HWI behaves under different noise conditions and demonstrate its adaptability to varying levels of noise in the data.

However, as the scales are similar, but not identical, it is not entirely clear from this set of figures how the HWI actually changes in performance with differing SNR. To demonstrate this, we normalize all of these surfaces, and then display them on a common set of axes in Figure 6.6.

This last figure is quite interesting, as it shows that the behaviour of the HWI relative to P and G is largely unaffected. Each of the surfaces displayed in Figure 6.6

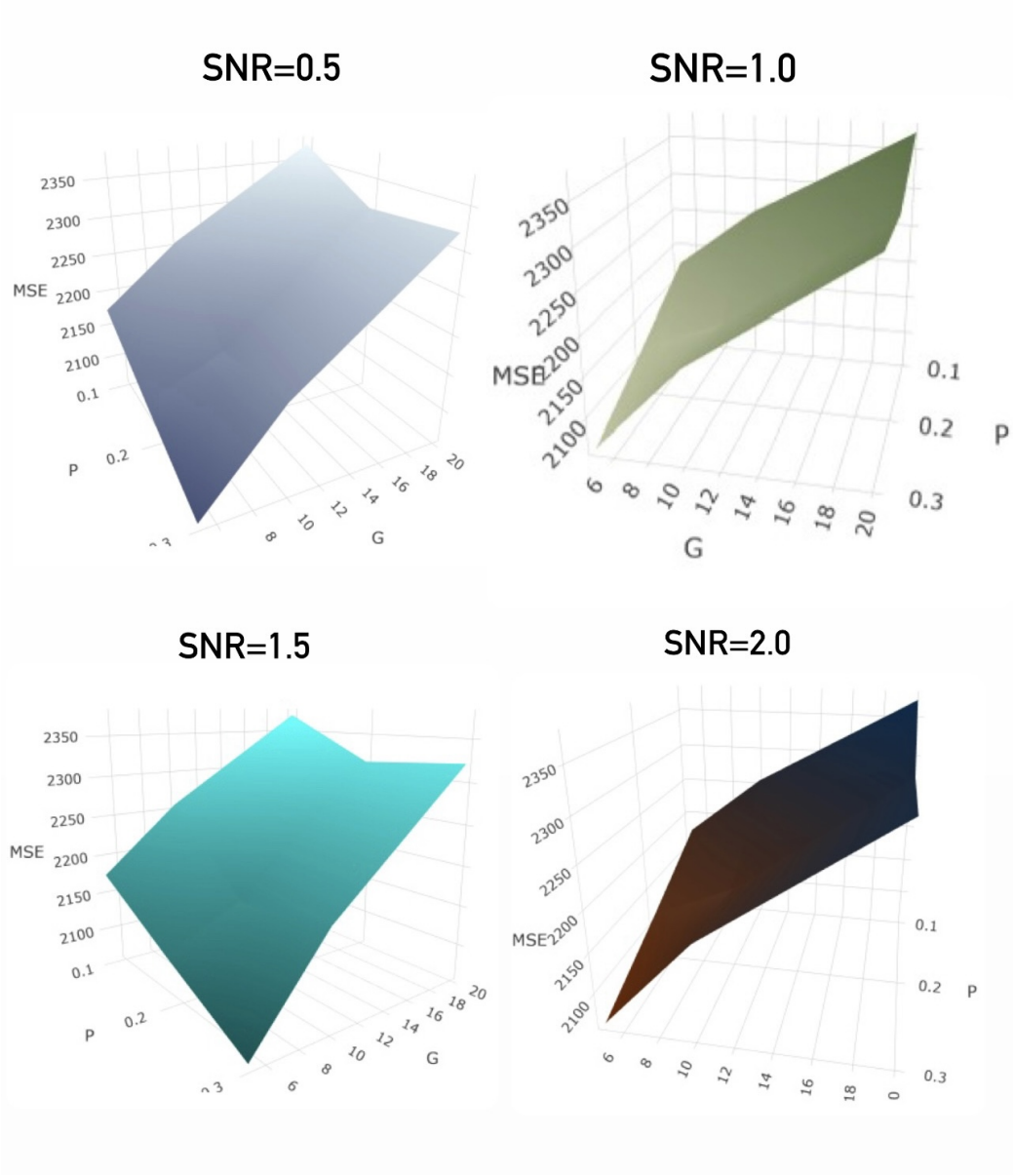

Figure 6.5: 3D Surface Plot visualizing the MSE as a function of parameters P and G across SNR levels 0.5, 1.0, 1.5, and 2.0. As the SNR increases, there is a noticeable decrease in MSE, indicating improved performance of the HWI method in lower-noise environments. The plots highlight how HWI adapts to varying noise levels, showing more stability and accuracy at higher SNR values.

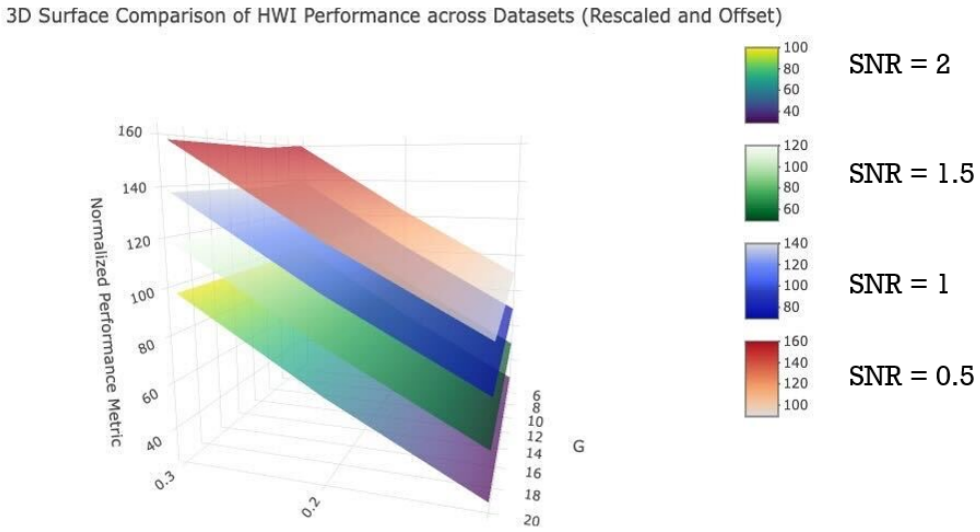

Figure 6.6: Set of normalized 3D Surface Plots visualizing the MSE as a function of parameters P and G across SNR levels 0.5, 1.0, 1.5, and 2.0. As the SNR increases, there is a clear decrease in MSE, indicating improved performance of the HWI method in lower-noise environments.

looks to have the same shape and structure, just slid vertically. As the surfaces are all normalized, we can see that the impact of the decrease in SNR corresponds to the normalized drop of approximately 20 units. Further, each subsequent change in SNR corresponds to approximately the same drop (or rise).

From this, we see that at least in the range of SNR considered, the HWI is quite robust to the noise, with the errors increasing approximately in line with the increase in the variance of the noise term. This implies that each of these cases is actually interpolating the signal very accurately, and it is only the prediction of the noise that varies, example to example. This is a piece of evidence for the strong performance of the HWI in the presence of line components.

6.5.1 General Observations

Across all SNR levels, the Hybrid Wiener Interpolator consistently performs well, demonstrating its ability to adapt to various noise conditions. As the SNR increases,

the MSE steadily decreases, reflecting HWI's suitability for time series with higher signal-to-noise ratios. Even at the lowest SNR (0.5), HWI remains functional, although the error rates are higher due to the increasing dominance of noise. This robustness to noise levels highlights HWI's versatility, particularly in handling scientific datasets where noise levels can vary significantly.

Although the differences in performance across the different SNR levels are observable, they are not highly pronounced. HWI delivers stable and reliable results regardless of the SNR level, with only modest variations in error. This further confirms HWI's effectiveness as a general-purpose interpolation method, capable of handling both high-noise and low-noise time series with considerable accuracy.

7 Conclusion

In this thesis, we explored the problem of missing data in time series analysis, specifically focusing on the interpolation of a particular class of nonstationary time series with embedded line components, in the presence of varying levels of noise. The complexities associated with real-world data — such as irregular sampling intervals, nonstationarity, and embedded trends — present significant challenges for conventional interpolation methods, which often rely on assumptions of stationarity and contiguity. To address these limitations, this thesis examined the performance of the Hybrid Wiener Interpolator, an advanced interpolation algorithm designed to handle such data imperfections.

Through a series of simulations on both synthetic and real-world datasets, we systematically tested the performance of the HWI in comparison with other state-of-the-art interpolation techniques. The results demonstrated that the HWI consistently outperformed traditional methods in reconstructing missing data for several classes of nonstationary time series, particularly in scenarios where gaps and irregularities were prevalent. By leveraging multitaper spectral estimation and a flexible iterative framework, the HWI proved to be robust across varying gap structures and different levels of signal-to-noise ratios.

While the HWI has shown promising results, there are still areas for future research. One potential direction is to further optimize the algorithm's computational

efficiency, making it more suitable for large-scale datasets: of the three primary methods considered, the HWI is by far the most computationally intensive. Additionally, exploring the integration of machine learning-based approaches with traditional interpolation methods may lead to the development of hybrid models that can better adapt to diverse data characteristics. Future research could also investigate the performance of the HWI in more specialized domains, such as high-dimensional or multivariate time series, where the complexity of the data poses additional challenges.

In conclusion, the Hybrid Wiener Interpolator presents a significant tool for the interpolation of time series data with embedded periodic structure, offering a flexible and reliable tool for handling missing data. This work contributes to the ongoing development of more robust and adaptable methods in time series analysis, laying the groundwork for further innovations in the field.

Bibliography

[1] Robert Berdan. Astrophotography of the Sun by Alan Friedman. https://www.canadiannaturephotographer.com/alan_friedman.html. Accessed June 25, 2024.

[2] David Brillinger. *Statistical Inference for Irregularly Observed Processes*, pages 38–57. hh, 01 1984.

[3] Wesley Burr. *Air Pollution and Health: Time Series Tools and Analysis*. PhD thesis, Queen University, 2012.

[4] Evan Callaghan. Using a neural network autoencoder framework for time series interpolation. Master's thesis, Queen's University, 2024.

[5] Sophie Castel. A framework for testing time series interpolators. Master's thesis, Trent University, 2020.

[6] Sophie Castel and Wesley S. Burr. Assessing statistical performance of time series interpolators. *Engineering Proceedings*, 5, 2021.

[7] Harald Cramér. On the linear prediction problem for certain stochastic processes. *Arkiv för Matematik*, 4, 1960.

[8] Anibal Flores, Hugo Tito-Chura, Deymor Centty-Villafuerte, and Alejandro Ecos-Espino. PM2.5 time series imputation with deep learning and interpolation. *Computers*, 12, 2023.

[9] George Forsythe, Michael Malcolm, and Cleve Moler. Computer methods for mathematical computations. *SERBIULA (sistema Librum 2.0)*, 01 1977.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[11] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960.

[12] M. Lepot, J.-P. Aubin, and F. Clemens. Interpolation in time series: an introductive overview of existing methods, their performance criteria and uncertainty assessment. *Water*, 9(10):796, October 2017.

[13] Norman Levinson. The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction. *Studies in Applied Mathematics*, 25, 1946.

[14] Norman Levinson. A Heuristic Exposition of Wiener's Mathematical Theory of Prediction and Filtering. *Studies in Applied Mathematics*, 26, 1947.

[15] Jin Li and Andrew D. Heap. Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53:173–189, 3 2014.

[16] Polychronis Manousopoulos, Vasileios Drakopoulos, and Efstathios Polyzos. Financial time series modelling using fractal interpolation functions. *AppliedMath*, 3, 2023.

[17] Gregoire Mariethoz, Niklas Linde, Damien Jougnot, and Hassan Rezaee. Feature-preserving interpolation and filtering of environmental time series. *Environmental Modelling & Software*, 72:71–76, 10 2015.

[18] Peter Maybeck. Stochastic models, estimation and control. *SERBIULA (sistema Librum 2.0)*, 1982.

[19] SDO NASA Goddard Space Flight Center. NASA's SDO Observes Largest Sunspot of the Solar Cycle. https://artsandculture.google.com/asset/nasa-s-sdo-observes-largest-sunspot-of-the-solar-cycle/_gHOnefEWgEMuw?hl=en, accessed July 1, 2024.

[20] Emanuel Parzen. On spectral analysis with missing observations and amplitude

modulation. *The Indian Journal of Statistics, Series A*, 25, 1963.

[21] Jose Manuel Pavía-Miralles. A survey of methods to interpolate, distribute and extrapolate time series. *Journal of Service Science and Management*, 03, 2010.

[22] Mohsen Pourahmadi. Estimation and interpolation of missing values of a stationary time series. *Journal of Time Series Analysis*, 10, 1989.

[23] Carl Runge. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. *Zeitschrift für Mathematik und Physik*, 46(224-243):20, 1901.

[24] Perry A. Scheinok. Spectral analysis with randomly missed observations: The binomial case. *The Annals of Mathematical Statistics*, 36, 1965.

[25] Steven S. Skiena. *Visualizing Data*. 2017.

[26] D. Slepian. Prolate spheroidal wave functions, Fourier analysis, and uncertainty—V: The discrete case. *Bell System Technical Journal*, 57(5):1371–1430, 1978.

[27] David J. Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70, 1982.

[28] David J. Thomson. Inverse-constrained projection filters. In *Wavelets: Applications in Signal and Image Processing IX*, volume 4478, 2001.

[29] Melissa Van Bussel. Time series interpolation algorithms: Honours Thesis, Trent University. 2019.

[30] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. *In Practice*, 7, 2006.

[31] M. Welvaert and Y. Rosseel. On the Definition of Signal-To-Noise Ratio and Contrast-To-Noise Ratio for fMRI Data. *PLOS ONE*, 8(11):e77089, 2013.

[32] Norbert Wiener. The extrapolation, interpolation and smoothing of stationary time series, with engineering applications. *Journal of the Royal Statistical Society. Series A (General)*, 113, 1950.

Appendix

Table 1: A set of interpolation algorithms that have are supported innately in `interpTools`, as provided by [5]. Algorithms considered in [5] (shown in bold) were based on the results of Van Bussel [29]; we primarily consider those underlined in this work.

Package	Function	Algorithm name
`interpTools`	`nearestNeighbor()`	Nearest Neighbor (NN)
`zoo`	`na.approx()`	Linear Interpolation (LI)
`zoo`	`na.spline()`	**Natural Cubic Spline (NCS)**
`zoo`	`na.spline()`	**FMM Cubic Spline (FMM)**
`zoo`	`na.spline()`	**Hermite Cubic Spline (HCS)**
`imputeTS`	`na_interpolation()`	Stineman Interpolation (SI)
`imputeTS`	`na_kalman()`	**Kalman - ARIMA (KAF)**
`imputeTS`	`na_kalman()`	Kalman - StructTS (KKSF)
`imputeTS`	`na.locf()`	Last Observation Carried Forward (LOCF)
`imputeTS`	`na.locf()`	Next Observation Carried Backward (NOCB)
`imputeTS`	`na_ma()`	Simple Moving Average (SMA)
`imputeTS`	`na_ma()`	Linear Weighted Moving Average (LWMA)
`imputeTS`	`na_ma()`	**Exponential Weighted Moving Average (EWMA)**
`imputeTS`	`na_mean()`	Replace with Mean (RMEA)
`imputeTS`	`na_mean()`	Replace with Median (RMED)
`imputeTS`	`na_mean()`	Replace with Mode (RMOD)
`imputeTS`	`na_random()`	Replace with Random (RRND)
`tsinterp`	`interpolate()`	**Hybrid Wiener Interpolator (HWI)**