University of Strathclyde

Department of Mechanical & Aerospace Engineering

An evaluation of noise reduction algorithms for particle-based fluid simulations in multi-scale applications

Małgorzata Jadwiga Zimoń

A thesis presented in fulfilment of the requirements

for the degree of Doctor of Philosophy

2015

Declaration of author's rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Małgorzata J. Zimoń

June 2015

Abstract

Particle-based fluid simulations can be utilised to study phenomena ranging from galaxy-scale, using smoothed particle hydrodynamics, plasma physics with particle-in-cell methods, aerospace re-entry problems, using direct simulation Monte Carlo, down to chemical, biological and fluid properties at the nanoscale with molecular dynamics and dissipative particle dynamics. The information generated by particle methods, such as molecular dynamics, is converted to macroscopic observables by means of statistical averaging. A significant drawback of nano- or micro-scale modelling is the substantial noise associated with particle techniques, which disturbs the analysis of the results. The uncertainty in the mean of the ensemble is due to fluctuations caused e.g. by additional forcing terms (thermostats). Extracting the genuine information from indirect, noisy measurements is analogous to solving the ill-posed statistical inverse problem, where the object of interest is not easily accessible. The presence of noise in the data can be reduced by averaging over a large number of samples, but the computational intensity of the simulations would then be substantially increased.

In order to improve the efficiency of estimating the unknown structure from the disturbed observations, a number of decomposition techniques have been applied, including: proper orthogonal decomposition, singular spectrum analysis, random QR de-noising, wavelet transform, and empirical mode decomposition. In the present work, the strengths and weaknesses of each approach, and their extensions to solving statistical inverse problems for particle simulations, are evaluated. Furthermore, we propose several novel combinations of these methods, that have the capability to improve the signal-to-noise ratio and reduce the computational cost.

"Most people, if you describe a train of events to them, will tell you what the result would be. They can put those events together in their minds, and argue from them that something will come to pass. There are few people, however, who, if you told them a result, would be able to evolve from their own inner consciousness what the steps were which led up to that result. This power is what I mean when I talk of reasoning backwards, or analytically. (...) Now let me endeavour to show you the different steps in my reasoning. To begin at the beginning."

Sherlock Holmes to Dr. Watson in *A Study in Scarlet*
Sir Arthur Conan Doyle, 1887.

Acknowledgements

Completing a project like this was a big challenge. However, I have been lucky not to have had to overcome it alone. I am indebted to many individuals for their support, inspiration and encouragement, which made my PhD studies such a rich experience.

First of all, my sincere gratitude goes to my supervisors, Prof. David Emerson and Prof. Jason Reese, for their support throughout my PhD. I am truly grateful to them for giving me the opportunity to work with them and enabling me achieve my dream. They have been great bosses to whom I am thankful for all the time and patience they have put into accompanying and advising me in my studies. My special appreciation goes to David for creating an excellent working environment and providing me with lots of cakes! In the same breath, I would like to thank Dr. Leopold Grinberg for his inspirational work that sparked my interest in this research. I am very grateful for his support and guidance.

I would like to extend my heartfelt appreciation to the whole Computational Engineering Group and other amazing individuals at STFC Daresbury Laboratory that made me truly enjoy my work! I am specially indebted to:

- Stefano, whose enthusiastic and energetic Italian personality gave me the inspiration for studying noise reduction,
- Charles, for eating all my chocolate and not allowing me to get fat,
- Sebastian, with whom I could always have a serious conversation and a good laugh,
- Yue, for sharing with me the experience of being a PhD student and challenging me in running,

- Yoann, for allowing me to work in his office and being an amazing flatmate,
- Rob Barber, Benzi John, Xiaojun Gu, Jianpeng Meng, Gregory Clover whom I admire for their passion for research and their hard work,
- Aleš, Benjamin, Henry, Jian, Sèthy and Stephen for all the fun that they brought to the Daresbury community,
- all the runners, rugby players and gym members who kept me moving throughout my PhD!

Furthermore, I am grateful to all the colleagues from the University of Strathclyde and University of Warwick that I have had the privilege to work with. In particular, I would like to acknowledge Matthew Borg, Konstantinos Ritos, William Nicholls, Duncan Lockerby, Špela Ivekovič and Craig White for their invaluable support, instructions and inspiring conversations.

Finally, I would like to express my very special thanks to Julio for being such a great friend whom I admire. I know he has a new interesting nickname for me now!

And moving on to French! Pierre, mon chéri, merci de m'avoir sortie du lit chaque matin et aussi pour ta cuisine! Grâce à ton aide et ta nourriture, j'ai fini cette thèse. Tu es si important pour moi! De par ton amour, tu fais de moi la plus heureuse des femmes.

And some Polish! Kochana rodzinko, tę pracę dedykuję Wam, ponieważ bez Waszego wsparcia i miłosci nie byłabym w stanie nic osiągnąć. Mamusiu, tatusiu, jesteście moją inspiracją. Wasza dobroć, pasja, zapał, pracowitość, to tylko nieliczne z cech, za które Was podziwiam. Madziu, moja najwspanialsza siostrzyczko, zawsze bylaś i będziesz dla mnie wzorem do naśladowania! Mam nadzieję, że możesz być ze mnie chociaż w części tak dumna, jak ja jestem z Ciebie. Strzelczuczku, dziękuję, że tak wspaniale się Madzią opiekujesz. Babciu Jadziu, babciu Zosiu, moi niezastąpieni chrzestni, wujku Tadeuszu, ciociu Olu, moi kuzyni, dziekuję Wam za Wasze wsparcie. Bardzo Was podziwiam i jestem szczęśliwa, że mam taką wspanialą rodzinę. Zawsze jesteście w moim sercu i myślach.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

1D, 2D	One-dimensional, two-dimensional
CWT	Continuous wavelet transform
DPD	Dissipative particle dynamics
DWT	Discrete wavelet transform
EEMD	Ensemble empirical mode decomposition
EMD	Empirical mode decomposition
EMD-DT	Empirical mode decomposition direct thresholding
EMD-IIT	EMD iterative interval thresholding
EMD-IT	EMD interval thresholding
EOF	Empirical orthogonal function
FFT	Fast Fourier transform
FWT	Fast wavelet transform
GPU	Graphics processing unit
IFFT	Inverse fast Fourier transform
IMF	Intrinsic mode function

IWT	Inverse wavelet transform
L-J	Lennard-Jones
LGA	Lattice gas automata
DSMC	Direct simulation Monte Carlo
MD	Molecular dynamics
MDPD	Many-body dissipative particle dynamics
MEMS	Micro-electro-mechanical-systems
MRA	Multiresolution analysis
MSE	Mean square error
rQRd	Random QR de-noising
SNR	Signal-to-noise ratio
STD	Standard deviation
STFT	Short time Fourier transform
TSVD	Truncated singular value decomposition
urQRd	Uncoiled random QR de-noising
1D/2D-WAV	One- or two-dimensional wavelet thresholding
WT	Wavelet transform

Greek Symbols

α	Temporal POD coefficient
β	Relationship between dimensions of $N \times M$ matrix, $\beta = \frac{N}{M}$
δ_2	Error in L_2 norm

δ_F	Error in Frobenius norm
δ_{ij}	Kronecker delta
ϵ_{L-J}	Constant determining the depth of the potential well in Lennard-Jones model
$\epsilon_{r,Ar}, \sigma_{r,Ar}$	Reduced Lennard-Jones parameters for argon
$\epsilon_{r,H_2O}, \sigma_{r,H_2O}$	Reduced Lennard-Jones parameters for water
γ_D	Dissipation strength of dissipative force in DPD/MDPD
λ	Eigenvalue, $\lambda = s^2$
Ω	Set of P random vectors
ϕ	Spatial POD mode
ϕ_s	Scaling function (father)
Ψ	Functional domain
ψ	Mother wavelet
$\psi_{j,n}$	Wavelets formed by stretching and shifting mother wavelet (children)
Σ	Diagonal matrix with singular values
σ_n	Noise level, standard deviation (square root of variance)
σ_R	Noise strength of random force in DPD/MDPD
σ_{L-J}	Constant denoting particle's diameter and corresponds to where the potential energy is zero (in Lennard-Jones model)

Latin Symbols

A_C	Interaction strength parameter of conservative force in DPD
A_c	Circulant matrix

B_C	Interaction strength (repulsive parameter) of conservative force in MDPD
$\mathbb{C}$	Complex numbers
C_R	Covariance matrix in MC-SSA
$c_{j_0,n}$	Approximation coefficients at level j_0 and shift $n2_0^j$ obtained with wavelet transform
$d_{j,n}$	Detail coefficients at level j and shift $n2^j$ obtained with wavelet transform
$db3, db4, \ldots$	Filters based on Daubechies' orthogonal wavelets with $3, 4, \ldots$ number of vanishing moments
E	Total energy of all eigenvalues
E_x	Projection of C_R onto the original data EOFs
E_λ	Percentage of variance held by each eigenvalue
$E_{\mathrm{IMF}}^{(p)}$	Energy of p-th IMF
$\bar{f}$	Statistical average of f
$\mathbf{F}^{(i)}$	Molecule force vector
$\mathbf{F}_C^{(ij)}$, $\mathbf{F}_R^{(ij)}$,$\mathbf{F}_D^{(ij)}$	Conservative, random (stochastic) and dissipative forces, respectively.
$\tilde{f}$	Approximation of f
$\widehat{f}$	Fourier transform of f
f	Random field, signal or a function; if degraded, consists of true signal and noise: $f = f_{\mathrm{true}} + f_{\mathrm{noise}}$
$f^s(x, t_s)$	Snapshot $f^s(x) = f(x, t_s)$, obtained from a successive measurement at time t_s

f_w	Wavelet transform of a function
f_{noise_w}	Approximation of noise expressed in wavelet domain
f_{true_w}	Approximation of true signal expressed in wavelet domain
H_{filter}	High-pass filter
H_{in}	Hurst index
H_j	Hankel matrix built from j-th column of X_{2D}
$\text{IMF}_p(t)$	t-th element of p-th IMF function
K	The number of L-lagged vectors obtained from X, $K = M - L + 1$
k	Mode number; number of significant EOFs, frequencies; rank of the low-dimensional subspace $\widetilde{X}$
k_B	Boltzmann constant
L	Window length for SSA analysis
$L_x \times L_y$	Dimensions of two-dimensional window
L_{filter}	Low-pass filter
M	Length of a data series, function
$m^{(i)}$	i^{th} particle's mass
N, M	Dimensions of matrix A, number of rows (observations) and columns (samples), respectively; N signals of length M
N_p	Number of particles in the system
p	Degree to which polynomials are reproduced with a wavelet
P_k	Number of random vectors in rQRd/urQRd calculations
p_k	Oversampling parameter, $p_k = P_k - k$
Q, R	Components obtained with QR decomposition

$\mathbb{R}$	Real numbers
$\mathbf{r}^{(i)}$	Position vector of the i^{th} particle
$\mathbf{r}^{(ij)}$	Separation distance between two particles (i, j)
P_{ff}, P_{tt}, P_{nn}	Input, noise-free signal and noise power spectra, respectively
r_{cut}	Cut-off radius
R_{ff}, R_{tt}, R_{nn}	Autocorrelation matrices of the noisy signal, noise-free signal and noise, respectively
r_{ff}, r_{tt}, r_{nn}	elements of autocorrelation matrices of the noisy signal, noise-free signal and noise, respectively
s	Singular value; diagonal entry of Σ ordered in a decreasing manner
s_{in}	Statistical inefficiency
$sym3, sym4, \ldots$	Symlets wavelets, modified nearly symmetric version of Daubechies wavelets with $3, 4, \ldots$ number of vanishing moments
$T_H(.), T_S(.)$	Hard and soft wavelet thresholding
t_s	Sampling time, temporal coordinate
T_u	Universal threshold
T_{IMF}	IMF-dependent threshold value in EMD-based de-noising, $T_{\text{IMF}}^{(p)} = C_t\sqrt{E_{\text{IMF}}^{(p)} 2\ln(t_{\max})}$
th	Threshold value for truncation of singular values (or eigenvalues)
U_{L-J}	Lennard-Jones potential
U, V	Set of left and right singular vectors, respectively
u, v	Orthonormal singular vectors corresponding to the columns of the matrices U and V
$\mathbf{v}^{(i)}$	i^{th} particle's velocity

w	Frequency
W_{Chop}	WienerChop filter, improvement of Wiener filter, W_{filter}
$\widetilde{X}$	De-noised time series obtained from data array X, $\widetilde{X} \approx S$
X	Noisy time series containing *true* signal S and disturbances R_N, $X = S + R_N$
X_i	i-th lagged vector formed from X
X_{2D}	Matrix of noisy data
Y	Trajectory matrix, e.g. Hankel or Toeplitz
Y_{2D}	Hankel block Hankel matrix
Y_Ω	Matrix defined as a product of Y and Ω
$\mathbb{Z}$	Set of integers
Z	Joint trajectory matrix

Subscripts

$*$	in MD simulations, properties in reduced units
2D	Two-dimensional data, matrix
r	in MD simulations, reference properties in SI units

Superscripts

$*$	Complex conjugate of a matrix
$+$	Moore-Penrose pseudoinverse of a matrix
-1	Inverse of a matrix
$\dagger$	Transpose of a matrix

Chapter 1

Introduction

"If we knew what it was we were doing, it would not be called research, would it?"

Albert Einstein, (1879-1955).

Nowadays, large amounts of data are processed and the task of extracting important information from the measurements, i.e., solving inverse problems, becomes one of the central issues. Direct observations on the system of interest are not always possible, and only estimates can be derived. The whole process becomes more challenging when the available data-sets are perturbed by some uncontrollable element, often referred to as *noise*. This thesis investigates the numerical treatment of such *ill-posed* statistical inverse problems that arise in the field of computational nanofluidics. Similar challenges are also encountered in many other disciplines such as geophysics [1], acoustics [2], image and signal processing [3, 4], and astrophysics [5]. Therefore, the discussed procedures are versatile and inter-disciplinary in nature.

The concept of an ill-posed problem was introduced by Hadamard [6] in cases where the solution was non-unique, unstable, or discontinuous with respect to the data, i.e., an arbitrarily small perturbation in the data leads to a large modification in the model. Hadamard believed that such cases were *artificial* and they resulted only from an incorrect physical representation of the system. However, this is not the case, and ill-posed problems were shown to exist in the form of inverse problems in many areas of science and engineering [7]. An interesting example is the story of Hubble Space Telescope

which was launched in 1990[1] as a joint project of the North American and European space agencies, NASA and ESA. This optical observatory was supposed to provide space images with a very high spatial resolution and to mark the most significant advance in astronomy since Galileo's telescope. Unfortunately, it happened to have a manufacturing error (flawed main mirror) which resulted in fuzzy, almost out-of-focus images sent back to Earth. Before the telescope could be fixed in 1993, the astronomers had been improving the blurred images by numerical reconstruction, i.e., by solving an inverse problem [8].

Analogous to the given example, the ill-posedness in simulation data comes from the fact that we want to obtain specific information about the system, such as the molecular velocity distribution, which can often be distorted by effects such as thermal fluctuations or finite sampling. Computing derivatives or other properties of a function specified by contaminated samples further amplifies the discrepancies, making the calculation very sensitive to the data. To extract any meaningful information the data first needs to be filtered, or de-noised, to uncover any underlying structures masked by errors and enable further execution of data-dependent tasks.

This thesis attempts to find solutions (approximations) to these discrete inverse problems, which are encountered in many particle-based simulations, through applying the following methods: proper orthogonal decomposition (POD), singular spectrum analysis (SSA), random QR de-noising (rQRd), wavelet thresholding, wavelet-based WienerChop filtering, empirical mode decomposition (EMD) interval thresholding, and their combinations. These methodologies originate from different constraints or regularisation procedures, but they have a common denominator – they search for a domain wherein signal and noise, that initially appeared inextricably *tangled*, can be separated. Apart from being useful alternatives to statistical averaging, the transforms offer additional information about the nature of the signal and the simulation.

[1] More details on the story can be found on NASA website, `www.nasa.gov/mission_pages/hubble/main`.

1.1 Motivation

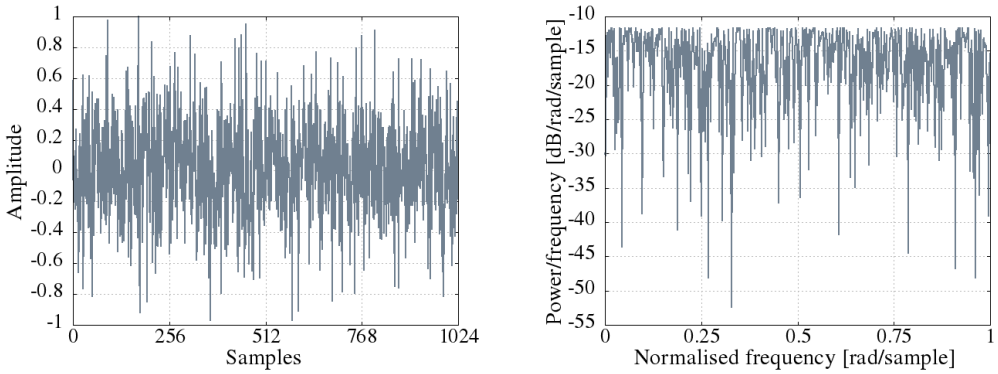
Numerical simulation techniques are indispensable tools for gaining a better understanding of many physical phenomena that can be difficult to describe with analytical methods or experimental studies. The statistical mechanics of complex systems is often analysed with molecular dynamics (MD) or Monte Carlo methods, e.g. direct simulation Monte Carlo (DSMC) [9]. These procedures can be used to accurately resolve dynamics at the atomistic scale and are widely used to simulate nano/microfluid flows e.g. confined in channels such as nanotubes [10, 11]. In addition, information obtained from molecular simulations forms the basis of new emerging hybrid multi-scale modelling methods for physical and biological applications (see Mohamed and Mohamad [12] for a review). Examples demonstrating the ubiquity of multi-scale, multi-physics applications include the dynamics of complex fluid flows [13], such as the physiology of red blood cells and blood flow [14], the classical turbulence problem [15], meteorological predictions [16], chemical and biological reactions [17], and emergent rheology [18]. Moreover, there is significant potential to apply multi-scale techniques to sociological problems, such as crowd and traffic flow [19]. The central problems with all particle-based and multi-scale modelling are the computational cost and the accurate transfer of information across disparate length and time scales; there currently exist many sources of uncertainty and noise disturbing this intra-scale transfer, with a concomitant loss of simulation fidelity.

Multi-scale modelling strategies to couple molecular simulations to continuum dynamics require smooth gradients and accurate particle distribution descriptions. For many problems, the conversion of the microscopic information to macroscopic observables is done through simple averaging. This procedure can be a poor choice, however, due to low resolution and statistical noise; circumventing this problem requires large samples and long averaging periods, resulting in bottlenecks in intra-scale communication and computationally expensive calculations. There is a clear and growing need for a systematic, mathematically rigorous de-noising approach for extracting coherent structures (and hence, emergent macro-scale fields) from particle data in stationary and non-stationary fluid flow simulations.

1.1.1 Noise classification

Noise can be defined as an unwanted disturbance that interferes with the measurement and processing of a signal, or a system's communication. It can cause calculation errors, and disrupt data transmission and analysis; therefore noise reduction is an important part of computational modelling. Noise can be classified as white or coloured depending on its frequency spectrum, obtained via Fourier analysis (called *sampled spectrum* or *periodogram* [20]), or time characteristics.

White noise is defined as an uncorrelated random process with a flat power spectrum, i.e. equal power, variance distribution, at all frequencies (see Fig. 1.1). It is a theoretical



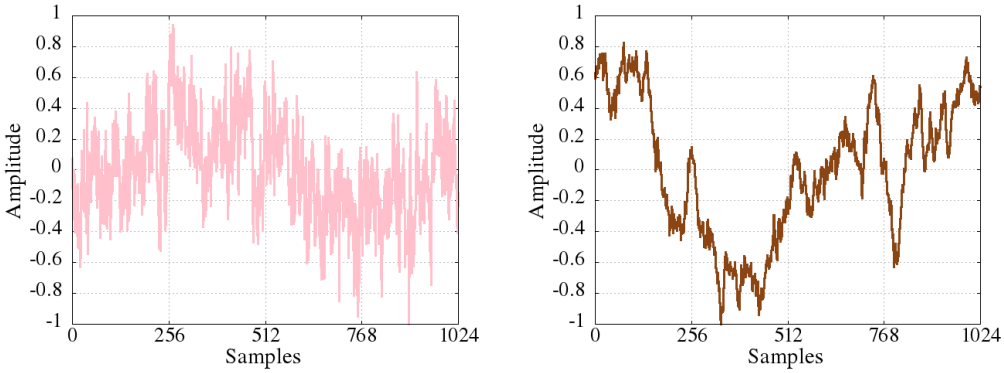
(a) White noise. (b) Periodogram of white noise.

Figure 1.1: White noise characteristics.

concept since its existence would require an infinite energy to cover an infinite range of frequencies. In addition, discrete-time signals by necessity need to be band-limited [21]. Therefore, in practice, a random signal is considered as *white* if it has a flat spectrum over a definite bandwidth (analogous to white light which contains all the visible frequencies at equal amplitudes). Thermal noise, which is generated by the random movement of thermally energised particles, is an example of fluctuations with a white (flat) spectrum. Additive white Gaussian noise is a basic noise model used to mimic the effect of many random processes that occur in nature. Gaussian noise has a probability density function equal to that of the normal (Gaussian) distribution in the time domain, with an average time domain value of zero. Other popular models include uniform, Laplace, and Cauchy distributions [22].

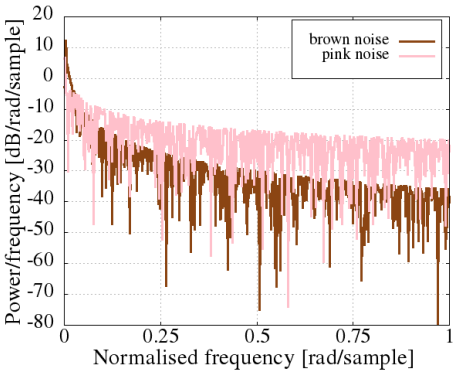
In reality, analysed systems are often contaminated with noise that is correlated

with itself or with the signal, not continuous, and not stationary. Fluctuations that are non-white are often referred to as coloured or $1/f^{\alpha}$ noise[2]. The latter name derives from the observation that they have spectra growing with low frequency as $1/f^{\alpha}$, where f is the cyclic frequency and α is a real number usually between 0 and 2. Two classic examples of noise *colours* are the pink (with $\alpha = 1$) and brown (also called red, with $\alpha = 2$) shown in Fig. 1.2 . The long *memory*, or correlation property of some disturbances



(a) Pink noise, $1/f$.

(b) Brown noise, $1/f^2$.



(c) Comparison of spectra.

Figure 1.2: Correlated noise: pink and brown.

has been observed in many diverse fields such as economics, music, traffic, physics, and engineering [23]. In this thesis, it is shown that data obtained from particle-based simulations is also subject to variations, with energy shifted towards lower frequencies due to finite sampling, additional force terms, and small characteristic time scales of the system. More specifically, during this research it was observed that artificial effects

[2]They are sometimes also referred to as flicker, burst, low frequency divergent, and fractional noise [23].

from MD thermostats, e.g. the Berendsen thermostat, contribute to correlated fluctuations, which do not exist independently of the underlying physical process. Recent work by Sanghi and Aluru [24] on noise extracted from MD simulation without temperature control reveals that, for both bulk and confined fluids, the distribution function of the thermal force is not strictly Gaussian. The fact that noise present in results obtained with particle-based simulations can be correlated is of importance when solving inverse problems. Most of the estimators assume randomness of the noise and they treat coherence as the information to be recovered. Due to the fact that both the data and noise *have* a pattern, identification of coloured fluctuations could be enhanced in a transform domain that can adapt to the signal's locality. On the other hand, such procedures are often conditioned by a number of parameters. In this thesis, different methodologies are applied to particle-based nanofluid flow simulations with the goal of assessing their effectiveness in separating different types of noise encountered in the data.

1.1.2 Previous work

To the author's best knowledge, little work has been done on noise cancellation in particle-based fluid flow simulations. The main contribution in the field is the comprehensive research performed by Grinberg [25], in which he introduced the proper orthogonal decomposition method with an adaptive time-window for analysing MD, DPD, and multi-scale data. Habasaki [26, 27] mentions using singular spectrum analysis, to separate signals of physical interest from contamination of thermal noise, in MD modelling of ion dynamics in ion-conducting materials and ionic liquids. The wavelet transform is known to the molecular dynamics community and was applied to characterise many-body dynamics in early work by Li *et al.* [28]. However, wavelet-based thresholding has not been much applied for filtering the observables obtained from particle simulations. An exception is the work of Albert *et al.* [29] in which wavelet de-noising was employed in shear crack modelling performed with MD. No research results have been published that use EMD interval thresholding, a WienerChop filter, or rQRd for processing measurements obtained from nanofluidic simulations.

1.2 Research objectives

The objective of this thesis is to investigate the capabilities of various mathematical transforms, introduced by different scientific communities, in assisting noise reduction in particle-based modelling. Different evaluation criteria, including processing time, signal-to-noise ratios, errors in norms, and data-adaptivity, are employed to evaluate their performance. A number of benchmark fluid flow problems were designed to investigate the usefulness of the considered methodologies and provide guidelines on how the algorithms can be successfully utilised. In addition to analysing the strengths and weaknesses of existing methods, the aim is to develop novel and efficient de-noising techniques to improve the quality of simulations results. The focus of the research is on evolving procedures that can provide rapid, adaptive, noise-free coarse-graining of micro-scale phenomena, and can further be employed in molecular-continuum simulations. This project directly tackles the important challenge of extracting information from the data without the need for long averaging periods.

Two of the methods, windowed proper orthogonal decomposition and its coupling with wavelet thresholding (using the C++ wavelet library[3]), have been implemented in the molecular dynamics code that is a part of the Open Field Operation and Manipulation (OpenFOAM) C++ fluid dynamics toolbox[4]. In addition, the other filtering utilities, apart from EMD routines which were written by Kopsinis and McLaughlin [30], and Rilling and Flandrin [31], were developed during this research as MATLAB scripts (some based on the original codes written in different programming languages) and can be applied to post-process the simulation results.

1.3 Thesis outline

This study introduces to computational nanofluidics methodologies, such as rQRd, urQRd, EMD-IT, which were utilised in other fields of science but never before used to reduce discrepancies in particle-based simulations. In addition, based on the research conducted, we propose a number of novel methods referred to as POD+ techniques which, for the first time, combine different de-noising approaches with POD. Moreover,

[3] `www.wavelet2d.sourceforge.net`
[4] `www.openfoam.org`

we test analytical procedures, only recently developed by the statistics community, for automated truncation of singular values in order to design simplified data filtering approaches for multi-scale modelling.

In Chapter 2, the molecular dynamics (MD), dissipative particle dynamics (DPD), and direct simulation Monte Carlo (DSMC) methods used to perform the fluid flow simulations in this thesis are briefly introduced. Chapter 3 is a review of all the mathematical procedures and their extensions applied in the research for data processing. We discuss their strengths and weaknesses and provide guidelines on how to utilise them for signal de-noising. The theoretical basis is followed by the results of employing the filtering methodologies to synthetic signals in Chapter 4. A comparison of the performance of each technique in extracting information from particle-based simulations is presented in Chapter 5, followed by concluding remarks on the new developments. The key findings of this research are summarised in Chapter 6 together with a discussion of future work.

Chapter 2

Computational Methods for Particle-Based Simulations

> "I would like to describe a field, in which little has been done, but in which an enormous amount can be done in principle. (...) What I want to talk about is the problem of manipulating and controlling things on a small scale."
>
> Richard Feynman, 1959.

Feynman's famous talk *There's plenty of room at the bottom* [32] was delivered over 50 years ago at the annual meeting of the American Physical Society, and referred to what later became known as *nano-science* and *nanotechnology*. The Nobel prize winner stressed the importance of miniaturisation and molecular manufacturing that gives the ability to, as Feynman put it, "manoeuvre things atom by atom". This famous lecture with its visionary power began a string of remarkable achievements, including development of small-scale machines referred to as MEMS (Micro Electro-Mechanical Systems). The increasing number of applications for microsystems, which are characterised by large surface-to-volume ratios, causing boundary effects to be significant, has led to the emerging technology of nano/microfluidics [33]. Even though progress in nanotechnology is tremendous, there is still plenty of room for advancement.

Particle-based simulation methods serve an important function in understanding fluids at the small scale, and enable the analysis of numerous physical phenomena in the field of nanofluidics (e.g. slip at a fluid-solid interface) that are very challenging to

study experimentally. Much of the interesting physics encountered in nanoflows can be understood by studying simple fluids, such as Lennard-Jones model liquids, confined in nanochannels. As the modelling techniques tend to be computationally expensive, a common approach is to utilise parallel-processing allowing the entire algorithm to run simultaneously on many computer processors. Moreover, GPU computing offers the potential of additional performance gains. This chapter provides a brief introduction to the modelling techniques utilised in this work for simulating nano-scale flows: molecular dynamics for liquid flows; its meso-scale counterpart, dissipative particle dynamics; and the direct simulation Monte Carlo method for rarefied gas dynamics.

2.1 Basics of molecular dynamics

Today, computer simulations play a significant role in nearly all branches of scientific research. Molecular dynamics is thought to be conceptually the simplest approach to study many-body problems, especially if non-equilibrium states and the evolution of the system are considered. The method is suitable for simulating very small systems with linear dimensions on the order of 100 nm or less, for time-scales of several tens of nanoseconds [9]. In this section, the methodology behind molecular dynamics is presented based on exhaustive reviews found in [9, 34–37].

The basis of the MD approach is classical mechanics[1] – solving numerically Newton's equation of motion for an interacting (through, e.g., pair potentials) multi-particle system. If a system of N_p particles with Cartesian coordinates $\mathbf{r}^{(i)} = \left(x^{(i)}, y^{(i)}, z^{(i)}\right),\ i = 1, \ldots, N_p$, is considered, the dynamics can be described in the following form:

$$\frac{d}{dt}\mathbf{r}^{(i)}(t) = \dot{\mathbf{r}}^{(i)} = \mathbf{v}^{(i)}(t), \tag{2.1}$$

$$m^{(i)}\frac{d}{dt}\mathbf{v}^{(i)}(t) = m^{(i)}\dot{\mathbf{v}}^{(i)} = \mathbf{F}^{(i)}(t), \tag{2.2}$$

with $\mathbf{v}^{(i)}$, $m^{(i)}$, $\mathbf{F}^{(i)}$ denoting a particle's velocity, mass, and acting force, respectively. Simulations performed with MD, therefore, consider a set of particles (here atoms, or

[1] It should be stated that quantum mechanics describes the basic physics of condensed matter, and not the classical approach. However, solving the Schrödinger equation (with the Born-Oppenheimer approximation) for multi-particle systems is still not entirely feasible [35]. For that reason, classical MD serves as a good alternative.

molecules) moving in a defined space. The idea behind this technique is to calculate trajectories of the nuclei. The time averages of observables (e.g., temperature, pressure, diffusion constant) are extracted along the trajectories in the system and linked to macroscopic properties with the use of statistical mechanics. The ergodicity hypothesis relates the ensemble average, defined as a collection of all possible systems differing in microscopic states, but having an identical macroscopic or thermodynamic state, to measurements carried out for a single equilibrium during the system's evolution. It is then assumed that the average of a process property over time and the average over the statistical ensemble should be the same. Different ensembles of statistical mechanics can be realised, by introducing a coupling to appropriate *thermostats* or *barostats*, including

- Microcanonical ensemble, NVE – The thermodynamic state characterised by a fixed number of particles (denoted here by N), volume (V), and a fixed energy (E); it corresponds to an isolated system.
- Canonical Ensemble, NVT – This is a collection of all systems with a thermodynamic state defined by a constant number of particles (N), volume (V), and temperature (T); such systems are considered in the simulations that follow.
- Isobaric-isothermal ensemble, NPT – System having a fixed number of particles (N), pressure (P) and temperature (T).

In the non-equilibrium case, the time evolution of the system is considered. Initial molecular positions within the ensemble are usually random and the initial velocities are assigned according to the Maxwell-Boltzmann distribution. The two main categories of molecular models are soft and hard sphere. The first MD implementation sampled the phase space of a system of hard spheres, and was performed by Alder and Wainwright [38] over 50 years ago. Hard sphere systems define molecular interactions through impulsive, instantaneous collisions, with free movement between them. In soft sphere systems, particles interact via a potential model, such as the Lennard-Jones potential (L-J).

In addition to classical MD, other methods for studying nanofluidics include the Monte Carlo (MC) techniques, the *ab initio* MD, or Car-Parrinello method. A major advantage of some of these procedures is the lack of dependency on the effective inter-

particle potential, which often is not precise and is chosen without any firm chemical foundation [35]. Nevertheless, for most fluids, the classical MD soft sphere approach is fast and generally effective and is utilised for many modern applications.

2.1.1 MD integration algorithms

The Verlet algorithm is a widely used method to solve Eq. (2.1) and (2.2) directly from the Taylor expansions of $\mathbf{r}^{(i)}(t)$:

$$\mathbf{r}^{(i)}(t+\Delta t) = \mathbf{r}^{(i)}(t) + \Delta t\dot{\mathbf{r}}^{(i)}(t) + \frac{(\Delta t)^2}{2}\ddot{\mathbf{r}}^{(i)}(t) + \frac{(\Delta t)^3}{6}\dddot{\mathbf{r}}^{(i)}(t) + O(\Delta t^4), \quad (2.3)$$

$$\mathbf{r}^{(i)}(t-\Delta t) = \mathbf{r}^{(i)}(t) - \Delta t\dot{\mathbf{r}}^{(i)}(t) + \frac{(\Delta t)^2}{2}\ddot{\mathbf{r}}^{(i)}(t) - \frac{(\Delta t)^3}{6}\dddot{\mathbf{r}}^{(i)}(t) + O(\Delta t^4). \quad (2.4)$$

Adding Eq. (2.3) to (2.4) yields for the time evolution:

$$\mathbf{r}^{(i)}(t+\Delta t) = 2\mathbf{r}^{(i)}(t) - \mathbf{r}^{(i)}(t-\Delta t) + \Delta t^2\ddot{\mathbf{r}}^{(i)}(t) + O(\Delta t^4). \quad (2.5)$$

While this is quite straightforward, it is not very accurate. To reduce the influence of numerical rounding error occurring in Eq. (2.5), the Verlet Leapfrog algorithm was introduced. The simulation sequence proceeds as follows for one time-step $t \to t+\Delta t$ [37]:

- *Step 1*: Calculate projected velocity at time $t + \Delta t/2$ for all N_p particles

$$\mathbf{v}^{(i)}\left(t + \frac{\Delta t}{2}\right) = \mathbf{v}^{(i)}(t) + \frac{1}{2}\dot{\mathbf{v}}^{(i)}(t)\Delta t, \quad (2.6)$$

  where $\dot{\mathbf{v}}^{(i)}(t)$ is the acceleration.

- *Step 2*: Update position of particles at $t + \Delta t$

$$\mathbf{r}^{(i)}(t+\Delta t) = \mathbf{r}^{(i)}(t) + \mathbf{v}^{(i)}\left(t + \frac{\Delta t}{2}\right)\Delta t. \quad (2.7)$$

- *Step 3*: Compute the interparticle forces acting on all molecules in the system

$$\mathbf{F}^{(i)}(t+\Delta t) = \sum_{j=1(\neq i)}^{N_p} -\nabla U_p\left(r^{(ij)}\right) = \sum_{j=1(\neq i)}^{N_p} \mathbf{F}^{(ij)}, \quad (2.8)$$

where $\mathbf{r}^{(ij)} = \mathbf{r}^{(i)} - \mathbf{r}^{(j)}$ at time-step $(t + \Delta t)$ and $\nabla U_p\left(r^{(ij)}\right)$ is the potential difference.

- *Step 4*: Determine the acceleration of every particle

$$\dot{\mathbf{v}}^{(i)}(t + \Delta t) = \frac{\mathbf{F}^{(i)}(t + \Delta t)}{m^{(i)}}. \tag{2.9}$$

- *Step 5*: Obtain the particle's velocity

$$\mathbf{v}^{(i)}(t + \Delta t) = \mathbf{v}^{(i)}\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2}\dot{\mathbf{v}}^{(i)}(t + \Delta t)\Delta t. \tag{2.10}$$

- *Step 6*: Repeat from the first step above until the simulation is statistically converged. A widely used approach to determine this is to inspect the evolution of observables (e.g. temperature, density, mass flux).

The above procedure is the version of the Verlet Leapfrog method that is implemented in the MD solver used for this research.

2.1.2 Force calculation: Potential energy model

In a soft sphere MD simulation, particle interaction is based on the potential energy function, and the force is a gradient of the potential, as shown in Eq. (2.8). Two classes of interaction can be distinguished: non-bonded, and bonded. Bonded interactions model strong chemical bonds; the potentials used for the simulations are bond angle potentials and torsion potentials [35]. Non-bonded interactions between particle pairs commonly include the Coulomb (long range) and van der Waals (short range) interactions.

The Lennard-Jones potential is often used as an approximate model of the short range pair-wise interactions in MD simulations. It is also utilised to simulate the behaviour of noble gases (its first application was related to liquid argon). For the purpose of theoretical research, the Lennard-Jones parameters are sometimes used to describe a system based on unrealistic materials (tuning the coefficients to obtain desired interactions without chemical consideration). The L-J potential is defined as

$$U_{L-J} = 4\epsilon_{L-J}\left(\left(\frac{\sigma_{L-J}}{r^{(ij)}}\right)^{12} - \left(\frac{\sigma_{L-J}}{r^{(ij)}}\right)^{6}\right), \tag{2.11}$$

where the depth of the potential well (strength of the interaction) is determined by ϵ_{L-J}, and the distance at which the interparticle potential is zero by σ_{L-J}.

In MD simulations incorporating the L-J potential, physical quantities are typically computed using reduced units. Table 2.1 summarises the units used for various quantities, with k_B denoting the Boltzmann constant. In all of our results the quantities are reported in reduced units.

Fundamental quantities		
length	$l_* = l/(\sigma_{L-J})_r$	$(\sigma_{L-J})_r = 0.34 \cdot 10^{-9}$ m
energy	$\epsilon_* = \epsilon_{L-J}/(\epsilon_{L-J})_r$	$(\epsilon_{L-J})_r = 1.65678 \cdot 10^{-21}$ J
mass	$m_* = m/m_r$	$m_r = 6.6904 \cdot 10^{-26}$ kg
Derived quantities		
time	$t_* = t/t_r$	$t_r = \sqrt{m_r(\sigma_{L-J})_r^2/(\epsilon_{L-J})_r} = 2.16059 \cdot 10^{-12}$ s
force	$F_* = F/F_r$	$F_r = (\epsilon_{L-J})_r/(\sigma_{L-J})_r = 4.87288 \cdot 10^{-12}$ N
acceleration	$a_* = a/a_r$	$a_r = m_r(\epsilon_{L-J})_r/(\sigma_{L-J})_r = 7.28340 \cdot 10^{-13}$ ms^{-2}
velocity	$v_* = v/v_r$	$v_r = (\sqrt{(\epsilon_{L-J})_r/m_r}) = 157.364$ ms^{-1}
density	$\rho_* = \rho/\rho_r$	$\rho_r = (\sigma_{L-J})_r^{-3} = 1702.22$ m^{-3}
temperature	$T_* = T/T_r$	$T_r = (\epsilon_{L-J})_r/k_b = 120$ K
pressure	$p_* = p/p_r$	$p_r = (\epsilon_{L-J})_r/(\sigma_{L-J})_r^3 = 42.153 \cdot 10^6$ Nm^{-2}
viscosity	$\eta_* = \eta/\eta_r$	$\eta_r = \sqrt{(\epsilon_{L-J})_r m_r}/(\sigma_{L-J})_r^2 = 9.10753 \cdot 10^{-5}$ kgm^{-1}s^{-1}

Table 2.1: Table of reduced units used in atomistic simulations performed with MD. Lennard-Jones potential quantities for argon serve as the reference values $()_r$, which allow to convert a property, originally in SI units, into reduced units $()_*$.

According to Eqs. (2.8) and (2.11), the force of interaction between particles is defined by:

$$\mathbf{F}^{(ij)} = 4\epsilon_{L-J}\left(12\left(\frac{\sigma_{L-J}}{\mathbf{r}^{(ij)}}\right)^{12} - 6\left(\frac{\sigma_{L-J}}{\mathbf{r}^{(ij)}}\right)^{6}\right)\left(\frac{\mathbf{r}^{(ij)}}{\left(r^{(ij)}\right)^2}\right). \tag{2.12}$$

This Lennard-Jones force decays reasonably fast for increasing particle separation. Nev-

ertheless, all particle pairs should be considered to compute the forces on a single atom. The computational cost of the force calculation, which is an $\mathrm{O}\left(N_p^2\right)$ procedure, is the main drawback of MD. Approximations are widely introduced to improve the efficiency of the method, including the use of a cut-off radius, r_{cut}, beyond which the force vanishes. The potential truncated at a separation distance, r_{cut}, defines a spherical region within which particles interact. Avoiding double-counting enables a reduction in the computational cost by computing the force between particles i and j only once, as the pair-potential calculations are symmetric. Details of the implementation of molecular dynamics have been given in previous works by Rapaport [36], Borg [37], and Nicholls [39].

2.1.3 Errors in MD

The systematic errors present in molecular dynamics are mainly due to finite system sizes, poor equilibration, and the possible influence of the random number generators [9, 34]. Moreover, statistical errors due to finite sampling in the presence of thermal fluctuations introduced by thermostats strongly affect the simulation results. In particle simulations, due to the finite time-steps on the scale of femtoseconds, there is no guarantee that instantaneous estimates are sufficiently independent. This makes the statistical analysis more difficult as the variance in the mean of a property, given by

$$s_N^2(\langle A\rangle_{run}) = s_N^2\left(\frac{1}{\tau_{run}}\sum_{\tau=1}^{\tau_{run}} A(\tau)\right) = \frac{1}{\tau_{run}} s_N^2(A), \tag{2.13}$$

where

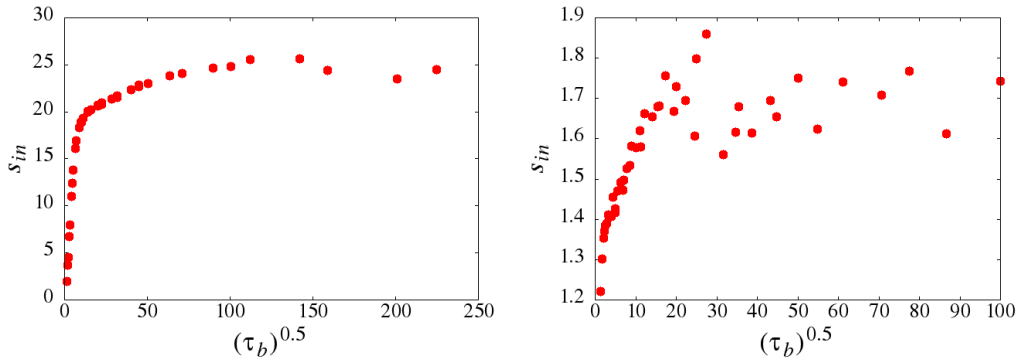
$$s_N^2(A) = \frac{1}{\tau_{run}}\sum_{\tau=1}^{\tau_{run}} (A(\tau) - \langle A\rangle_{run})^2, \tag{2.14}$$

is underestimated. In Eq. (2.13), $\langle A\rangle_{run}$ is a mean of the data and $A(\tau)$ is a measurement at time, τ. In order to avoid lack of accuracy due to strong correlations within the measurements, the statistical inefficiency method presented by Friedberg [40], and Allen and Tildesley [34], also referred to as *block-averaging* by Rapaport [36], should be utilised. In this procedure, the sequence of steps is broken up into n_b blocks each of length τ_b, so that $n_b\tau_b = \tau_{\mathrm{run}}$ gives the total size of the data set. The statistical

inefficiency s_{in} is then defined as:

$$s_{in} = \lim_{\tau_b \to \infty} \frac{\tau_b s_N^2(\langle A \rangle_b)}{s_N^2(A)}, \tag{2.15}$$

where $s_N^2(\langle A \rangle_b)$ is the block variance. The result is plotted against $\tau_b^{0.5}$ until a plateau is eventually reached; according to Rapaport [36], the point at the start of the plateau is an indication of the extent to which the samples are correlated, i.e. *lower bound*. For a particle simulation time-step, Δt, the plateau value of s_{in} indicates that data should be sampled at every time interval, $t_w = \Delta t \cdot s_{in}$, to ensure statistical independence. The statistical inefficiency informs which data can be ignored to reduce redundancy. Figure 2.1(a) shows the plateau for correlated MD data, and in Fig. 2.1(b) the statistical inefficiency is plotted for the sequence consisting of every 24th time-step. It can be seen that estimation of an effective time-step, in this case $t_w = 24\Delta t$, ensures an appreciable difference among successive measurements. Figure 2.1(a) suggests that the system was sampled too often and repetitions of the same state are present. In contrast, in Fig. 2.1(b) the plateau is below $s_{in} = 2$ indicating that now every sample provides new information.



(a) Data sampled at every time-step.

(b) Statistical inefficiency for every 24th successive measurement.

Figure 2.1: The calculation of statistical inefficiency s_{in} with approach to the plateau for velocity data from MD simulation of Poiseuille flow of water.

2.2 Dissipative particle dynamics

Dissipative particle dynamics is a stochastic particle-based technique for simulating systems at scales often not reachable by other atomistic methods. This mesoscopic model was originally introduced by Hoogerbrugge and Koelman [41] in 1992 as an off-lattice version of lattice gas automata (LGA). As noted by Hoogerbrugge and Koelman [41], lattice models introduce two fundamental problems: isotropy and Galilean invariance[2] are both broken, which DPD overcomes. In 1995, the method was put into a formal statistical mechanics context [43]. In the case of DPD, interactions are soft, repulsive, and short-ranged. The soft potential enables the use of a time-step that is up to an order of magnitude larger than the values typically used in MD simulations. Moreover, the soft interactions reduce the complexity of the DPD method by decreasing the number of degrees of freedom for particles. In the original technique, each dissipative particle is regarded not as a single molecule of the fluid (as with MD), but rather as an assembly or collection of molecules.

2.2.1 Methodology of DPD

Dissipative particle dynamics can be seen as a coarse-graining of molecular dynamics [44]. The method's simplicity and flexibility make it a competitive technique in the field of modelling complex fluids. Similar to molecular dynamics, the time evolution in DPD simulations is governed by Newton's equations of motion (see Eq. (2.1) and (2.2)). However, in the standard DPD method, particles interact with a pairwise additive force defined as a sum of three contributions [45]:

$$\mathbf{F}^{(i)} = \sum_{j \neq i} \left(\mathbf{F}_C^{(ij)} + \mathbf{F}_R^{(ij)} + \mathbf{F}_D^{(ij)} \right), \tag{2.16}$$

where $\mathbf{F}_C^{(ij)}$, $\mathbf{F}_R^{(ij)}$ and $\mathbf{F}_D^{(ij)}$ are the conservative, random (stochastic) and dissipative pair forces, respectively. The conservative repulsive force is related to the soft interaction

[2] Galilean invariance implies that motion is the same in a coordinate system that moves with constant velocity, which is equivalent to the conservation of total linear momentum [42].

potential and is represented as follows:

$$\mathbf{F}_C^{(ij)} = A_C^{(ij)} w_C\left(r^{(ij)}\right)\frac{\mathbf{r}^{(ij)}}{r^{(ij)}}, \tag{2.17}$$

where $r^{(ij)} = \left|\mathbf{r}^{(ij)}\right| = \left|\mathbf{r}^{(i)} - \mathbf{r}^{(j)}\right|$, and $A_C^{(ij)}$ is the interaction strength (maximum repulsion parameter). The switching function $w_C\left(r^{(ij)}\right)$ is equal to $1 - r^{(ij)}/r_{\text{cut}}$ for $r^{(ij)} < r_{\text{cut}}$, and vanishes for $r^{(ij)} \geq r_{\text{cut}}$, where r_{cut} is a cut-off radius. The dissipative and random forces are given by

$$\mathbf{F}_D^{(ij)} = -\gamma_D^{(ij)} w_D\left(r^{(ij)}\right)\left(\mathbf{r}^{(ij)}\cdot\mathbf{v}^{(ij)}\right)\frac{\mathbf{r}^{(ij)}}{\left(r^{(ij)}\right)^2}, \tag{2.18}$$

$$\mathbf{F}_R^{(ij)} = \sigma_R^{(ij)} w_R\left(r^{(ij)}\right)\zeta^{(ij)}\Delta t^{-\frac{1}{2}}\frac{\mathbf{r}^{(ij)}}{r^{(ij)}}, \tag{2.19}$$

where Δt is the time-step, $\mathbf{v}^{(ij)} = \mathbf{v}^{(i)} - \mathbf{v}^{(j)}$ is the relative velocity, and $\zeta^{(ij)}$ is a random number with zero mean and unit variance; $\gamma_D^{(ij)}$ and $\sigma_R^{(ij)}$ are the dissipation strength and noise strength, respectively. The terms $w_D\left(r^{(ij)}\right)$ and $w_R\left(r^{(ij)}\right)$ are weighting functions which are given by $w_D\left(r^{(ij)}\right) = \left(w_R\left(r^{(ij)}\right)\right)^2 = \left(1 - r^{(ij)}/r_{\text{cut}}\right)^2$, if $r_{(ij)} < r_{\text{cut}}$, and zero for $r^{(ij)} \geq r_{\text{cut}}$. The dissipative term, $\mathbf{F}_D^{(ij)}$, can be considered as friction that acts on the relative velocities of particles. The random force is related to temperature. The force $\mathbf{F}_R^{(ij)}$ can compensate for the loss of energy due to the dissipation. Therefore, both drag and random forces may be used as a system thermostat provided that the weight functions and amplitudes obey a fluctuation-dissipation theorem [46]:

$$\sigma_R^2 = 2\gamma_D k_B T. \tag{2.20}$$

All the interactions in DPD are pairwise, which means that the method obeys Galilean invariance and isotropy. The mass and momentum are conserved to preserve the hydrodynamics of the system [47, 48] and the three types of forces satisfy Newton's Third Law.

2.2.2 Many-body DPD

Despite the method's versatility, it was noted by Louis *et al.* [49] that the quadratic equation of state derived from the conservative force does not contain a van der Waals

loop necessary for describing a vapour-liquid coexistence e.g. for a droplet study. In DPD, the effective force is solely repulsive, which makes it impossible to consider systems with free surfaces [50]. Many-body DPD is a technique which provides a large range of thermodynamic behaviours for the DPD particles [47, 48].

Many Body Dissipative Particle Dynamics (MDPD) is a promising mesoscopic method for simulating fluid interfaces. Based on the work of Pagonabarraga and Frenkel [51], and further developed by Trofimov [52, 53] and Warren [46], MDPD introduces a density-dependence into conservative forces:

$$\mathbf{F}_C^{(ij)} = \left[A_C^{(ij)} w_C + B_C^{(ij)} (\bar{\rho}^{(i)} + \bar{\rho}^{(j)}) w_d \right], \tag{2.21}$$

where $\bar{\rho}^{(i)}$ represents the average local density at the position of particle i defined as

$$\bar{\rho}^{(i)} = \sum_{j \neq i} w_\rho \left(r^{(ij)} \right). \tag{2.22}$$

The weight function $w_\rho \left(r^{(ij)} \right)$ is normalised so that $\int_0^\infty 4\pi r^2 w_\rho(r) dr = 1$ [54]. The other function, w_d, is chosen to be equal to $1 - \frac{r^{(ij)}}{r_d}$ for $r^{(ij)} < r_d$ and 0 if $r^{(ij)} \geq r_d$, where r_d is a new cut-off radius. This modification divides the conservative force into two parts: an attractive component obtained by setting the parameter $A_C^{(ij)} < 0$ with cut-off usually being $r_{\text{cut}} = 1$, and a repulsive force with $B_C^{(ij)} > 0$ and shorter cut-off e.g. $r_d = 0.75 r_{\text{cut}}$. According to Warren [55] the force law is not conservative unless $B_C^{(ij)}$ is a constant matrix. This has been defined as the *no-go* theorem. In most of the simulations described in the literature the change in particle interaction is obtained by tuning the value of $A_C^{(ij)}$, with $B_C^{(ij)}$ being unchanged [54, 56, 57]. MDPD enables simulation of vapour/liquid interfaces: allowing the study of surface wetting properties [58], droplets [45], calculation of surface tension [54], capillary problems, such as the distribution of liquids in porous media, and water-oil displacements [50]. In the current study, the simulation of liquids confined in a channel were the main focus, and dissipative particle dynamics (not MDPD) was the preferred numerical choice.

2.3 Direct simulation Monte Carlo

Gas flows can be characterised by a parameter known as the Knudsen number, Kn, which measures the degree of non-equilibrium of the fluid flow. This dimensionless quantity is defined as

$$Kn = \frac{\lambda_{Kn}}{L}, \tag{2.23}$$

where λ_{Kn} is a molecular mean free path, and L denotes a characteristic length scale of the system. The continuum fluid formulation breaks down when the geometric length L is comparable to the mean free path of the molecules in the gas. In general, the continuum description based on partial differential equations is inadequate when $Kn > 0.1$, and a particle-based approach is needed. Despite significant progress in solving the Boltzmann equation [59], the direct simulation Monte Carlo technique, developed by Bird in the 1960's for kinetic scale simulations [60, 61], remains the dominant numerical method for the simulation of dilute gases. Since the technique was introduced, it has been quickly adopted for many problems where the Knudsen number is significant. Initially the focus was on hypersonic aerospace applications, such as the simulation of re-entry flight in the upper atmosphere. Today DSMC is also used to study gaseous flows in microfluidic devices, as the small length scales of such systems, even under atmospheric pressure, suggest the presence of rarefaction effects and the continuum assumptions of the Navier-Stokes equations are no longer valid.

DSMC models the gas at a microscopic level using particles representing a large number of physical molecules or atoms. The method defines the physics of the gas through the motion of particles in a discretised computational domain, and collisional interactions between them. A fundamental assumption of DSMC is that particle movement can be decoupled from particle collisions, which is valid for dilute gases. Particles then move according to discrete changes in velocity and energy caused by collisions. A successful DSMC simulation is dependent on the following conditions [9, 62–64]:

- The time-step Δt is chosen to be smaller than the mean collision time.
- The DSMC cell size is a fraction of the mean free path, λ_{Kn}; the average cell size is about $\lambda_{Kn}/3$.

- There is a sufficient number of simulated particles per cell (≥ 20).
- Collision-rate and reaction-probabilities are defined correctly.

As described in [9, 62–66], the DSMC algorithm can be broken down into the following operations:

- *Step 1*: Move all particles according to their molecular velocity for a given time-step, Δt.
- *Step 2*: Index and track the particles, calculate boundary interactions i.e. generate/move inflow particles, remove outflow particles, and process reflections at solid boundaries.
- *Step 3*: Compute collisions via a probabilistic approach.
- *Step 4*: Sample macroscopic flow properties within the cell, and iterate the whole procedure.

When a pair of particles is selected for a collision, momentum and energy are exchanged between them. The scattering angle and degree of inelasticity are defined statistically in order to generate post-collision properties. The collision cross-section schemes that have been utilised in this work (for simulating argon gas flow) are the hard and variable hard sphere models, as they are sufficient for monatomic gases, for which the vibrational and rotational non-equilibrium effects are negligible [9]. In the hard sphere approach, the interparticle force, defined as the inverse of the repulsive power force,

$$F_n = \frac{C_p}{r_p^{\eta}}, \tag{2.24}$$

has $\eta \to \infty$ [65], where r_p is the distance between the colliding particles, and C_p is a constant. The hard sphere model is considered to be the simplest and most efficient.

In the last step of the DSMC procedure, the macroscopic properties are obtained through time-averaging for steady-state flows, or by calculating the mean ensemble of many independent calculations. One of the main drawbacks associated with DSMC is a significant statistical scatter in the results, particularly in low velocity applications; the uncertainty in the recovered velocity is inversely proportional to the square of the Mach

number [67]. For that reason, DSMC velocity measurements are used in this research, together with MD and DPD data, to study the performance of de-noising techniques.

Chapter 3

Noise Reduction Techniques

This chapter provides a review of all the algorithms utilised in the present work for noise reduction. It is divided into seven sections, each introducing a different technique. The first section is concerned with proper orthogonal decomposition and how it can be used for filtering simulation data. The second part of the review focuses on singular spectrum analysis, which can be seen as a modification of proper orthogonal decomposition for analysing one-dimensional data arrays (signals). Improvement of the method, referred to as random QR de-noising, is discussed in Sec. 3.3. A brief mathematical description of the wavelet transform and wavelet-based thresholding is then presented. Section 3.5 discusses recent applications of empirical mode decomposition for removing noise from signals. Novel couplings of proper orthogonal decomposition with other algorithms are introduced in Sec. 3.6. The methods presented in this chapter enhance de-noising of simulation data, as shown in the chapters that follow. For completeness, the final section is concerned with dynamic mode decomposition: this recently developed approach has promising features that should be the subject of future studies.

3.1 Proper orthogonal decomposition

Proper orthogonal decomposition is a statistical method that finds a low-dimensional approximate description of high-dimensional data containing a large number of interrelated variables. In addition to reducing order, POD is also used for feature extraction by revealing coherent structures within the data (e.g. simulation results, experimental

measurements). The method was introduced to the turbulence community by Lumley (1967) [68]. However, the same procedure was developed independently by several scientists and is known under different names depending on the area of application. Its exact origins are therefore difficult to trace.

The review by Stewart [69] focuses on the early history of singular value decomposition (SVD), which is closely connected to POD, and mentions the contributions of five mathematicians in its development. These are Betrami (1835-1899), Jordan (1838-1921), Schmidt (1876-1959), Sylvester (1814-1897), and Weyl (1885-1955) who established "one of the most fruitful ideas in the theory of matrices". In statistics, POD is referred to as principal component analysis (PCA) and originated with the work of Pearson [70] on fitting lines and planes to a set of points in n-dimensional space. It was also independently proposed and named by Hotelling [71] in 1933. In signal processing, the method is known as the Karhunen-Loève decomposition (KLD) or theorem, due to the work of Karhunen [72] from 1946 and Loevè [73] from 1948. In some papers it is also referred to as the Kosambi-Karhunen-Loève theorem [74], in order to acknowledge the earlier contribution of Kosambi [75]. Other names that have appeared in the literature include empirical orthogonal function (EOF) in oceanography [76] and meteorology [77], and factor analysis in social sciences [78, 79].

The method has been utilised for different applications in each mentioned field of science. In this thesis, however, noise reduction is the main interest and the POD technique has been investigated in the context of its capabilities to extract high quality information from numerical data. This section begins with the mathematical formulation of POD. Different means of performing POD are explained in 3.1.2. The last part discusses the issue of determining the number of *components* that account for significant structures contained in the data.

3.1.1 Mathematical formulation

The following mathematical description of proper orthogonal decomposition is based on the formulations presented in [80–83]. We wish to describe a real function $f(x,t)$,

which is a random field on a domain, Ψ, as a finite sum of its variables:

$$f(x,t) \approx \sum_{i=1}^{r} \alpha_i(t)\phi_i(x), \tag{3.1}$$

where the coefficients $\alpha(t)$ represent temporal information and ϕ contains spatial components (t denotes time and x space). In terms of fluid mechanics, the space Ψ can be considered as a domain of flow-fields at a given instant in time. An element of Ψ is therefore a snapshot of the flow. When r (the number of elements) approaches infinity, the estimate becomes exact. Classical orthogonal approaches allow for such a decomposition by establishing a set of orthonormal basis functions, $\{\phi_i(x)\}$, such that its first $k < r$ terms provide the best approximation of the function $f(x,t)$.

Seeking a structure that resembles an ensemble of observations $\{f^s\}$ of the field $f(x,t)$ corresponds to seeking a function ϕ such that the average squared error between f^s and its projection onto ϕ is minimised:

$$\min \left\langle \left\| f^s - \frac{(f^s,\phi)}{\|\phi\|^2}\phi \right\|^2 \right\rangle, \tag{3.2}$$

where f^s is a snapshot, $f^s(x) = f(x,t_s)$, obtained from successive measurements at time t_s during the simulation, and $(f^s,\phi) = \int_\Psi f^s\phi d\Psi$ denotes the real inner product in Ψ; $\langle . \rangle$ indicates an appropriately defined ensemble average, e.g. time, space, or phase average; $\|.\| = (.,.)^{\frac{1}{2}}$ is the *induced* norm. This reasoning is equivalent to finding ϕ to maximise the averaged projection:

$$\max \frac{\left\langle |(f^s,\phi)|^2 \right\rangle}{\|\phi\|^2}, \tag{3.3}$$

where $|.|$ represents the modulus. Equation (3.3) means that if f^s is projected along ϕ, the average energy content is greater than if it is projected along any other basis. It should be noted that the *energy* associated discussed here is not a physical energy. However, for some cases, e.g. velocity measurements in fluid dynamics, this energy is strongly related to the kinetic energy of the system. Proper orthogonal decomposition is concerned with orthonormal basis functions, i.e. functions that are both orthogonal ($(\phi_i,\phi_j) = 0$ if $i \neq j$) and normalised, $\|\phi\|^2 = 1$, which satisfies Eq. (3.3), resulting in

the best approximation to the elements of $\{f^s\}$. Holmes [80] explained that according to spectral theory, for Eq. (3.3) to be satisfied, ϕ needs to be an eigenfunction of a two-point correlation tensor:

$$\int_{\Psi} \left\langle f^s(x) f^s(x') \right\rangle \phi(x') dx' = \lambda \phi(x). \tag{3.4}$$

A prime in Eq. (3.4) denotes evaluation of the field at the displaced position $x' = x + \Delta x$, and λ is an eigenfunction. This constraint[1] is a result of solving a condition for extrema of a corresponding Lagrange multiplier with normalised $\|\phi\|^2 = 1$:

$$J[\phi] = \left\langle |(f^s, \phi)|^2 \right\rangle - \lambda \left(\|\phi\|^2 - 1 \right). \tag{3.5}$$

Setting the functional derivative to zero for all variations of ϕ leads to the solution of Eq. (3.3), which is given by the orthogonal eigenfunctions $\{\phi_k(x)\}$, often called the POD modes, or empirical orthogonal functions (EOFs), and the corresponding eigenvalues, λ_k, of Eq. (3.5). The modal coefficients $\alpha_k(t)$ in Eq. (3.1) are uncorrelated, i.e.,

$$\langle \alpha_k \alpha_{k'} \rangle = \delta_{kk'} \lambda_k, \tag{3.6}$$

and are determined by $\alpha_k(t) = (f(x,t), \phi_k(x))$; $\delta_{kk'}$ is the Kronecker delta, i.e.

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{3.7}$$

There is an infinite number of solutions to Eq. (3.4). The energy, E, contained in the data having r elements (matrix of rank r) is defined as the sum of the eigenvalues,

$$E = \sum_{i=1}^{r} \lambda_i. \tag{3.8}$$

The percentage of variance contained in the k-th eigenvalue is given as

$$E_{\lambda}^{(k)} = \frac{\lambda_k}{E}. \tag{3.9}$$

[1] In the calculus of variations, Eq. (3.4) is known as the Fredholm integral equation of the second kind, and it is an ill-posed problem [84].

Since the eigenvalues are arranged in a specific order, the index k can be called the mode number. Having the eigenvalues ordered by $\lambda_k \geq \lambda_{k+1}$, the optimal vector to approximate the ensemble of snapshots is the one corresponding to the first eigenvalue, $\lambda_{k=1}$. The mode associated with the second greatest eigenvalue, $\phi_{k=2}$, is the optimal solution to characterise the ensemble of snapshots, but is restricted to the space orthogonal to $\phi_{k=1}$. This reasoning is extended to all the modes. It is desired to find the basis $\{\phi_k(x)\}$ which determines all the variations in the function $f(x,t)$ with a small number of modes k in a least squares sense, i.e., capturing most of the energy contained in the data.

3.1.2 Finite-dimensional case: computation of the decomposition

For practical purposes, a discretised set of data in time and space is considered. In the case of a large number of points, the direct method of finding the eigenfunctions becomes challenging. Sirovich [85] stressed that the temporal correlation matrix will yield the same dominant spatial modes, but is a less computationally expensive eigenproblem. The approach proposed is generally referred to as the *method of snapshots*. For simplicity, consider a real matrix A, which is a collection of M simultaneous measurements of some variable at N instants of time. In such an arrangement, the element $A_{i,j}$ of the $N \times M$ matrix is a measurement from the j-th probe taken at the i-th time instant. Finding POD modes and corresponding eigenvalues is associated with eigensolutions of a symmetric matrix $C = AA^\dagger$ (or $A^\dagger A$ if $N > M$). The superscript $\dagger$ indicates matrix transpose as we are analysing a real data-set; if the matrix is complex then the conjugate transpose is performed instead, i.e $C = AA^*$, where the A^* is obtained by taking the transpose and then calculating the complex conjugate of each complex entry. The discrete formulation of Eq. (3.4) involves computing the eigenvectors of the equation

$$Cv = \lambda v, \tag{3.10}$$

where v is an eigenvector and λ is a scalar eigenvalue defined previously. Diagonalisation of the symmetric matrix gives the following decomposition:

$$C = V \Lambda V^\dagger, \tag{3.11}$$

where V is a set of eigenvectors corresponding to eigenvalues contained in the diagonal matrix Λ. This representation of a square matrix is referred to as eigenvalue decomposition (EVD).

Proper orthogonal decomposition can also determine the optimal approximation of the matrix, A_k ($A_k \approx A$), by first performing singular value decomposition (SVD) of the original real $N \times M$ matrix A:

$$A = U\Sigma V^{\dagger}, \tag{3.12}$$

where, in case of full SVD, U is an $N \times N$ orthogonal matrix, V is an $M \times M$ orthogonal matrix, and Σ is an $N \times M$ diagonal matrix. Columns of U and V are left and right singular vectors, respectively. The singular vectors can be considered as rotations and reflections, and Σ as a stretching matrix. Diagonal entries of Σ, called singular values of A, are non-negative numbers (greater than or equal to zero) arranged in decreasing order:

$$s_1 \geq s_2 \geq \ldots \geq s_r \geq 0. \tag{3.13}$$

In Appendix A it is explained how SVD can be computed manually. It should be stressed that SVD analysis allows us to unveil some useful facts about the matrix A, e.g.

1. The number of non-zero singular values defines the rank of the original matrix A ($r = min(N, M)$).

2. If A is a square $N \times N$ matrix, the absolute value of its determinant is equal to the product of its singular values,

$$|\det(A)| = \prod_{i}^{N} s_i, \tag{3.14}$$

   as U and V are unitary matrices.

3. SVD can be used to calculate a pseudoinverse of A, $A^{+}{=}V\Sigma^{+}U^{\dagger}$, where Σ^{+} is formed by replacing every non-zero diagonal entry by its reciprocal and transposing the resulting matrix [86].

4. The matrix norms can be calculated based on singular values, $\|A\|_2 = s_1$, $\|A\|_F = \sqrt{s_1^2 + s_2^2 + \ldots + s_r^2}$, where $\|.\|_F$ and $\|.\|_2$ indicate Frobenius and L_2 norms, respectively.

Equation (3.12) can also be expressed in the form

$$A = QV^\dagger = \sum_{i=1}^{r} q_i v_i^\dagger, \tag{3.15}$$

where $Q = U\Sigma$, A represents the function $f(t,x)$ from Eq. (3.1), q_i is a column matrix corresponding to $\alpha_k(t)$, and $v_i^\dagger$ is a vector matrix representing $\phi_k(x)$. The description in Eq. (3.15) is an accurate approximation as the data set has a finite size. To construct an optimal lower-rank estimate of A, for a determined value $k < r$, the matrix Σ_k is obtained by setting $s_{k+1} = s_{k+2} = ... = s_r = 0$,

$$A_k = U\Sigma_k V^\dagger = \sum_{i=1}^{k} s_i u_i v_i. \tag{3.16}$$

To save computational time, U and V can be replaced by matrices consisting of their first k columns and Σ_k can be replaced by its $k \times k$ principle minor. Throughout this thesis, the rank k of the best matrix approximation will be referred to as the number of the *dominant* modes.

Geometric interpretations

Singular value decomposition can be better understood by visualising how the $N \times M$ matrix A deforms an M-dimensional space to an N-dimensional space [87]. Since the matrix V is unitary (i.e. $V^\dagger = V^{-1}$), Eq. (3.12) can be rearranged:

$$A = U\Sigma V^\dagger \Longleftrightarrow AV = U\Sigma. \tag{3.17}$$

If each vector is considered separately, Eq. (3.17) can be expressed as

$$Av_i = s_i u_i, \quad i = 1, \ldots, M. \tag{3.18}$$

It can be seen that A maps the set of unit vectors of an orthogonal coordinate system v_i, onto a new *scaled* system, $s_i u_i$. The unit sphere with respect to the matrix L_2 norm (the square root of sum of squares of elements) gets mapped to an ellipsoid in a new N-dimensional space. Singular values s_i define the lengths of the semi-axes which are spanned in the direction u_i. Figure 3.1 summarises the whole procedure for $M = 3$ and $N = 2$; $M - N$ dimensions of the domain collapse, then the remaining dimensions are stretched and rotated. As the higher singular values can be equal to zero, it should be specified that SVD maps a unit sphere onto an r-dimensional space.

Another way to interpret the decomposition comes directly from the least squares optimisation defined in Eq. (3.2). Chatterjee [83] explained that the matrix A can be seen as a collection of coordinates of N points in M-dimensional space. We are seeking a new k-dimensional subspace ($k \leq M$) which has the smallest mean square distance from the set of points. The solution is a projection of the points on the new subspace. A basis for this subspace is given by the first k columns of V.

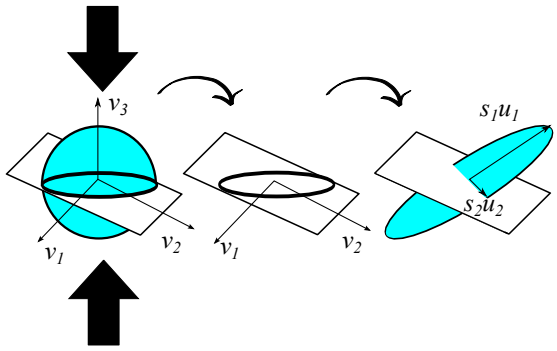


Figure 3.1: Geometrical interpretation of singular value decomposition.

SVD vs eigenvalue decomposition

There is a direct relation between SVD and eigenvalue decomposition; for A which can be mean-centred, i.e. the mean of A is subtracted from each column, or have non-zero mean, the principal components can be determined from the symmetric matrix, $C = AA^\dagger$ or $C = A^\dagger A$. Making use of the orthonormality of U and V, SVD can be

performed by solving two eigenvalue problems:

$$C = AA^{\dagger} = U\Sigma V^{\dagger}V\Sigma U^{\dagger} = U\Sigma^2 U^{\dagger}, \tag{3.19}$$

$$C = A^{\dagger}A = V\Sigma U^{\dagger}U\Sigma V^{\dagger} = V\Sigma^2 V^{\dagger}, \tag{3.20}$$

where V is a matrix of eigenvectors of the $M \times M$ square matrix of A, $A^{\dagger}A$, and the left singular vectors U are the eigenvectors of the symmetric $N \times N$ matrix $AA^{\dagger}$. The squares of singular values are the r largest eigenvalues,

$$\lambda = s^2, \tag{3.21}$$

common to both symmetric matrices of A.

Kalman [87] stressed that the EVD and the SVD are the same for real and symmetric matrices, except that the singular values are the absolute values of the eigenvalues. The elements of SVD remain within a real domain whenever A is real. However, according to Chatterjee [83], eigenvalues and eigenvectors of unsymmetric real matrices can be complex. In case of rectangular matrices, numerical analysts consider SVD as being superior to EVD because it is more accurate [87, 88], since the formation of matrix cross-products in EVD can lead to round-off errors. In contrast, SVD can be directly applied to the original arbitrary matrix A, avoiding degradation of results. On the other hand, EVD is computationally less expensive. Kerschen *et al.* [82] and Hansen [84] claim that another advantage of using SVD is that more information on the spectral properties of the data can be obtained through singular vectors.

Mean-centring before SVD and EVD

It is quite common to mean-centre the data in A before performing SVD. If A is visualised as a set of N points, subtracting the mean from the data is equivalent to shifting the mean of the point cloud towards the origin of the coordinate system. This is equivalent to performing eigenvalue decomposition on a covariance matrix (or, after normalising, a correlation matrix). Following this procedure does not influence the main calculation, but it can affect the interpretation of the results.

It has been widely debated whether mean-centring is a necessary step [83, 89, 90].

Chen *et al.* [90] argued that refraining from subtracting the mean is advantageous in many practical applications, e.g. while analysing in-cylinder engine flows. In their work, it is demonstrated that performing POD without subtracting the ensemble average is beneficial as the coefficient of the first mode can reveal the extent to which the mean flow is present and its cycle-to-cycle variability. This could also be important if SVD is used to assess whether the simulation has reached a steady-state. For the purpose of solving a statistical inverse problem, no benefit is seen in performing mean-centring of the noisy measurements. The comparison of the first POD mode, which represents the mean, with the other modes can assist in partial reconstructions which are performed when solving statistical inverse problems. In Chapter 4, examples are shown which confirm that mean-centring the data before performing SVD is unnecessary, and does not lead to any improvements in terms of the quality of results or processing time. In the present thesis, SVD has only been applied to the non-zero mean data.

3.1.3 Noise filtering with proper orthogonal decomposition

The two methods discussed, SVD and EVD, may also be considered as *energy decompositions* which have the capability to filter out low energy spectra (i.e. noise) from raw data. If the previously considered matrix, A, is now a collection of M noisy measurements at N instants of time, it can be represented as a composition of the form

$$A = A_{\text{true}} + B, \tag{3.22}$$

where A_{true} is the $N \times M$ matrix that contains the *true* signal, and B is a matrix that denotes the unwanted noise. In general, we only know the original matrix A and we need to remove the noise to extract the true information contained in A_{true}. For the special case of a synthetically generated matrix A, we will know A_{true} and corrupt the signal with artificially generated noise, represented by matrix B. Using POD, we can remove the noise from the matrix A by creating a corresponding approximation matrix of rank k, A_k, that contains all the correlations from the original data, i.e. $A_k \approx A_{\text{true}}$. The approximation is obtained by truncating the singular values as described in Eq. (3.16); such a procedure is referred to as truncated singular value decomposition (TSVD). The solution is optimal with respect to the induced L_2 matrix norm (spectral), defined

previously as the largest singular value of the matrix, and the Frobenius norm, calculated as the square root of the sums of squares of the entries (or singular values). In other words, POD yields the matrix A_k with the lowest possible Frobenius (or spectral) error.

In the case of simulation data, Grinberg [25] developed an extension to POD for particle simulations, which is based on *time-windows* and generally referred to as WPOD:

$$T_{\text{POD}} = N_{\text{POD}} N_{ts} \Delta t, \tag{3.23}$$

where N_{POD} is the number of time averages used, N_{ts} defines how many observations are in one average, and Δt is the simulation time-step. In this work, WPOD is utilised to filter out noise in molecular simulations based on the approach presented by Grinberg. For a set of noisy observations (snapshots), $A(t_s, x)$, defined as a field at positions in space $x \epsilon \mathbb{R}^d$, $d = 1, 2, 3$ and at discrete times t_s, $s = 1, 2, ..., N_{\text{POD}}$, WPOD calculates a set of orthogonal basis modes by applying SVD to the POD window, SVD($A = T_{\text{POD}}$). The estimation of statistical inefficiency (see Sec. 2.1.3) may be used to determine the window size, where $N_{\text{POD}} = n_b$ and $N_{ts} = \tau_B$ from Eq. 2.15. Throughout this thesis the singular vectors are referred to as spatial and temporal POD modes as they contain information either about the shape of the signal or its time nature. In the case of the matrix $A(t_s, x)$, a temporal POD mode (related to temporal coefficient $\alpha_k(t)$ from Eq. (3.1)) corresponding to mode number k is a left singular vector u_k (column k of matrix U) and a spatial mode is a right singular vector, v_k. In terms of the eigenvalue problem, the temporal information is held by the eigenvectors of a correlation matrix, $C = AA^\dagger$, and the spatial mode is defined as the product of the eigenvector and the original matrix, A.

Choosing a subset of significant modes

If we consider again the noisy $N \times M$ matrix A, defined in Eq. (3.22), we can re-write the equation as:

$$A = A_{\text{true}} + \sigma_n B_r, \tag{3.24}$$

where the noisy matrix, B, is now defined as the product of a matrix, B_r, whose entries are independent, zero-mean variables, and σ_n represents noise level. The matrix, A_{true},

is to be estimated from the noisy measurement contained in matrix A. As mentioned previously, the default technique for solving this statistical inverse problem is a truncation of singular values, i.e. setting to zero the singular values which correspond to noise and calculating the approximation, A_k, according to Eq. (3.16). To obtain the best estimate of A_{true}, the number, k, of singular values required for data reconstruction needs to be carefully determined.

The main rationale of using SVD (or EVD) to filter out noise is based on the assumption that, unlike unwanted fluctuations, important (coherent) structures are energetic. One natural criterion for choosing k is to select the cumulative percentage of total energy (e.g. 90%) that the selected modes contribute. In other words, the number of singular values (or eigenvalues) used for data reconstruction is defined by the smallest value of k for which this chosen percentage is exceeded:

$$\frac{\sum_{i=1}^{k} E_{\lambda}^{(i)}}{E} 100\% \geq 90\%, \tag{3.25}$$

where E_λ, explained in Eq. (3.9), can be expressed in terms of singular values by replacing $\lambda_k = s_k^2$. This is valid for both approaches, when SVD is applied to mean-centred and the original data. The energy threshold is often chosen arbitrarily, and it can depend on some practical details of a considered data set. In the case of particle simulations, defining a cut-off based on energy poses difficulties, as it often happens that structures of interest contain very little energy when compared to dominant features. Additional analysis needs to be performed in order to establish an appropriate subset of significant modes. The following techniques can be applied to both singular values and eigenvalues, based on the equivalence defined in Eq. (3.21).

In factor analysis, Kaiser [91] proposed a new rule in 1960, in which he states that any eigenvalue smaller than a threshold, $th = 1$, contains less information than the original component and should be rejected. It was argued that despite the simplicity of this method, *Kaiser's rule* is quite inefficient and either overestimates or retains too few modes; different cut-off values were also proposed following Kaiser's work, e.g. $th = 0.7$ by Jolliffe [92]. The rule is specifically designed for correlation matrices, and if the covariance matrix is of interest, the threshold should be set to the statistical mean of

the eigenvalues $th = \bar{s^2}$. This implementation can be found in commercial packages, e.g. MATLAB.

Another popular approach, discussed and named by Cattell (1966) [93], involves a visual study of a plot of eigenvalues against their position index, or mode number. In the *scree* graph, the eigenvalues are plotted in descending order and can be linked with a line. When the line creates a steep change, as in Fig. 3.2, or where the line levels off, a breaking point (an *elbow*) defines that all eigenvalues located above it are significant. Cattell [93] even suggested that beyond that point of change the eigenvalues should follow an almost straight line. The logic behind this procedure is that such a break in the plot divides the important or dominant factors from the trivial elements. The spectrum needs to be truncated at the first point (rank) when a plateau in the singular values begins. Donoho and Gavish [94] explained that when A_{true}, in Eq. (3.24), is exactly or approximately low-rank, and the entries of B_r are of zero mean and unit variance, the empirical distribution of singular values of the $N \times M$ matrix, A, forms a quarter-circle bulk with a boundary edge defined by $\left(1 + \sqrt{\frac{N}{M}}\right)\sqrt{M}\sigma_n$. The singular values that are larger than that boundary are visible in the scree-graph above the *elbow*, which represents the edge. This method is sometimes referred to as *bulk-edge hard thresholding* [94]. However simple, the scree graph has been criticised for being subjective, and difficult to interpret in cases where various drops and possible cut-off points appear in the data [89].

An alternative but similar approach to the scree plot is a log-eigenvalue diagram (LEV), and its first description is often associated with the work of Craddock and Flood (1969) [95]. Instead of plotting the eigenvalues against the index number, their logarithms are considered, $\log(\lambda = s^2)$. Choosing the truncation based on LEV is motivated by the idea that, if higher modes represent uncorrelated noise, then the corresponding eigenvalues should decay exponentially with increasing mode number [96]. The *smoother*[2] the kernel $\langle f^s(x) f^s(x') \rangle$ from Eq. (3.4), the faster the eigenvalues should decay to zero [84]. Craddock and Flood showed that, in meteorology, eigenvalues representing noise are decaying in a geometric progression, almost forming a straight line on a LEV diagram. In other words, the slow-decaying eigenvalues represent short

[2] *Smoothness* is measured by the number of continuous partial derrivatives.

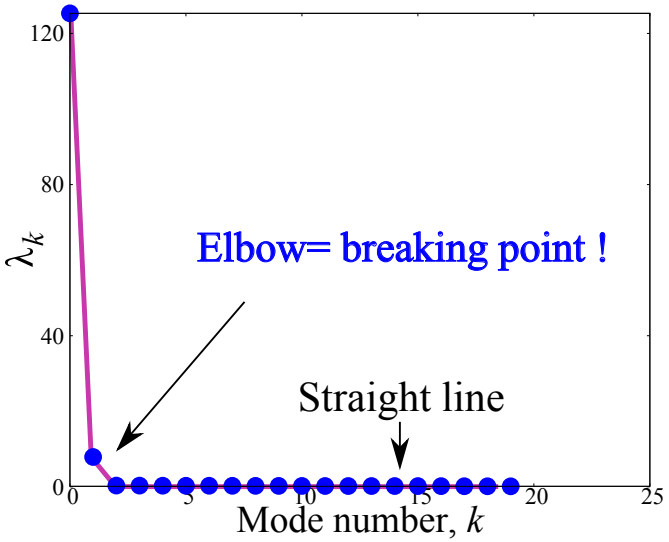


Figure 3.2: Scree diagram of synthetically generated signal; two eigenvalues emerge as dominant modes.

correlation times.

Choosing the number of significant singular values (or eigenvalues) is equivalent to determining which singular vectors (or eigenvectors) define the signal's subspace. When any low-rank data is corrupted by statistical fluctuations, higher singular values and corresponding modes are associated with the unwanted disturbances. Grinberg [25] suggested that for filtering simulation data, it is helpful to investigate the temporal modes directly, and assess which of them can be accurately approximated with a smooth function, e.g., high-order polynomials. For dynamic problems, information about the evolution of the signal is clearly represented by dominant temporal modes (or eigenvectors), as shown in Fig 3.3. The smaller the singular values, the more oscillations are contained in the singular functions u_k and v_k. It is a consequence of the fact that the singular value system is the analogue of the Fourier expansion of the kernel $\langle f^s(x) f^s(x') \rangle$, from Eq. (3.4), in the sense that low frequency singular functions correspond to the large singular values and higher frequency singular functions correspond to smaller singular values [84, 97]. Rejecting all the vectors containing high frequencies eliminates noise from the system. In this thesis, the smoothness of temporal modes (or eigenvectors) is taken into account.

Grinberg [25] also proposed another criterion based on an analysis of the temporal modes, u_k. He considered flows simulated with atomistic solvers, in which only the

fluctuating component resulting from the temperature control (thermostat) contributes to the slow converging modes. In the case of controllers which employ random forces to regulate the system's energy, it is expected that the thermal disturbances will have a zero mean. Consequently, the temporal modes corresponding to noise will also have zero mean. The standard deviation of a temporal mode representing unwanted fluctuations effectively is defined as

$$\mathrm{STD}(v_k) = \sqrt{\frac{\sum_i^{N_{\mathrm{POD}}} \left(v_k(\tau^i) - \langle v_k \rangle\right)^2}{N_{\mathrm{POD}} - 1}} = \frac{1}{\sqrt{N_{\mathrm{POD}} - 1}}, \tag{3.26}$$

as $v_k \cdot v_k \equiv 1$, $\langle v_k \rangle = N_{\mathrm{POD}}^{-1} \sum_i^{N_{\mathrm{POD}}} v_k\left(\tau^i\right)$ and N_{POD} is defined in Eq. (3.23).

Recent work [94, 98] has focused on finding a universal threshold value for truncating singular values. Donoho and Gavish [94] try to establish a singular value hard threshold (SVHT) which successfully adapts to unknown rank and noise level, and performs as well if the rank of A_{true} was given. Their study is based on the asymptotic mean square error in a framework where the matrix size is large compared to the rank that should be recovered. Donoho and Gavish [94] show that the optimal choice of the truncated SVD, a *hard threshold*, in the case of a square $N \times N$ matrix, A, with white noise level σ_n, is exactly

$$th = \frac{4}{\sqrt{3}}\sqrt{N}\sigma_n \approx 2.309\sqrt{N}\sigma_n, \tag{3.27}$$

when the noise variance is known, or when σ_n is unknown

$$th = 2.858 \cdot y_{\mathrm{med}}, \tag{3.28}$$

where y_{med} is a median empirical singular value. For a non-square $N \times M$ matrix A, the thresholding coefficients are replaced with different values based on a relation $\beta = \frac{N}{M}$:

$$th(\beta) = \sqrt{2(\beta+1) + \frac{8\beta}{(\beta+1) + \sqrt{\beta^2 + 14\beta + 1}}} \cdot \frac{y_{\mathrm{med}}}{\sqrt{\mu_B}}, \tag{3.29}$$

where μ_B is the Marčenko-Pastur distribution [94]. If the noise level is known, $\frac{y_{\mathrm{med}}}{\sqrt{\mu_B}}$ in Eq. (3.29) is replaced with $\sqrt{N}\sigma_n$, accordingly. In the case of a real inverse problem, the noise level is not given and in most cases, the simulation data forms a rectangular

matrix and Eq. (3.29) is preferred and used in the present work. However, in the case of synthetically generated signals, the equations for known variance can be used to verify the results obtained.

This list of criteria presented for determining the number of significant modes, k, is by no means exhaustive. A large amount of research has been done in order to improve extraction of a true low-rank approximation of the noisy data. More examples can be found in the literature [89, 99]. However, even though more complex rules are available, the simplest tests still seem to be the most beneficial and widely used in different fields of science. In the present thesis, most of the data is analysed with the LEV diagram, by analysing the energy content of eigenvalues, investigating the smoothness of temporal modes, and in some cases verified with the SVHT value defined in Eq. (3.29).

In order to visualise how some of the tests are being performed to aid the process of noise filtration, the following example of a disturbed synthetic signal is presented in Fig. 3.3. The original smooth data matrix was generated as follows:

$$A_{\text{true}}(t,x) = \sin\left(\frac{\pi x}{M}\right)\cos\left(\frac{\pi t}{N}\right) + 0.8e^{(-3\pi x/M)}\sin\left(\frac{9\pi x}{M}\right), \tag{3.30}$$

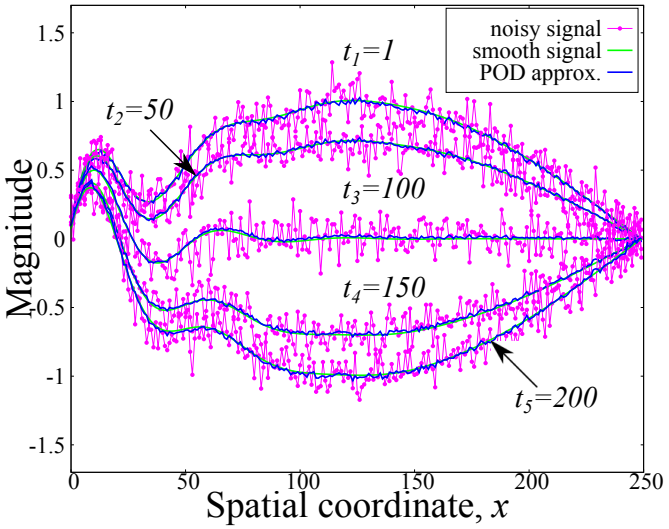
looping over $t = 1:N$ and $x = 1:M$. Proper orthogonal decomposition was applied to the matrix with one-dimensional (1D) signals of length set to $M = 250$ and the number of observations was set to $N = 200$. The initial smooth non-steady signals were disturbed by adding Gaussian noise (with zero mean and unit variance) using MATLAB's pseudo-random number generator, $B = \sigma_n B_r = 0.1\text{randn}(N, M)$, constructing a noisy matrix A with signal-to-noise ratio[3], SNR = 12.4752 dB.

Previously described criteria were employed to select dominant POD modes. In this case, only two modes, $k = 2$, were selected for approximation of the original matrix as a result of applying the smoothness test of temporal modes and investigating the rate of decay of eigenvalues. Our reconstructed matrix had SNR = 30.2052 dB, about 142% higher than the original noisy signals, and is plotted in Fig. 3.3(a) with blue lines against A_{true} presented with green lines. The first and the second eigenvalue, $\lambda_{k=1} = s_1^2$ and $\lambda_{k=2} = s_2^2$, were the most energetic, containing together 96.42% of the total variance. In practice, it is common to select levels of energy threshold between 70% to 95% [92].

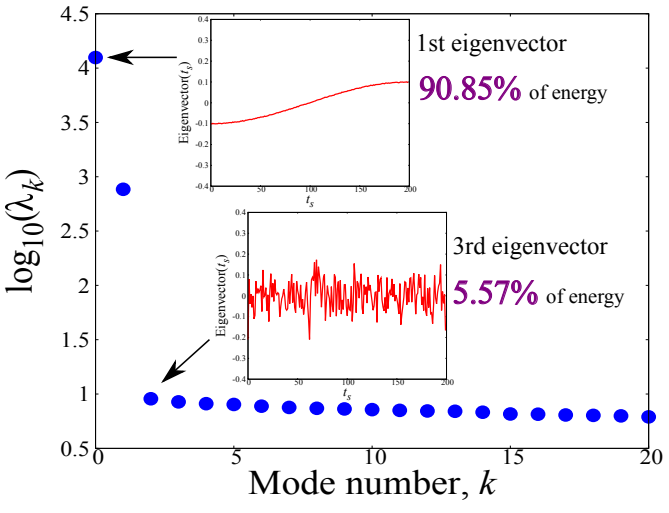
[3]Defined as the ratio of signal power to the noise power, expressed in decibels.

It is clear that the first two modes retained most of the significant information, and the third eigenvalue corresponded to only 0.066% of the cumulative energy. When the LEV diagram of $y(k) = \log_{10}(\lambda_k)$ was plotted, the first two eigenvalues also appeared to be fast-decaying (see Fig. 3.3(b)), while the other points formed almost a straight line. The corresponding eigenvectors were smoother than other temporal modes which clearly contained higher frequencies.

The choice of k was further confirmed by applying the SVHT, which retained a



(a)



(b)

Figure 3.3: Result of applying POD to synthetically generated noisy signals, and investigating criteria for the choice of significant k: (a) filtered signal plotted against noisy and smooth profiles, (b) LEV diagram with the 1st and 3rd eigenvector highlighted.

threshold of $th = 3.4601$ for the known noise ($\sigma_n = 0.1$), and similar $th = 3.4583$ for unknown variance and $\beta = \frac{N}{M} = \frac{200}{250} = 0.8$; the third singular value was smaller than the threshold, $s_3 = 3.0014 < th$ resulting in only two modes being recovered. It is important to consider all the criteria because any single test on its own may not provide enough information to capture the significant phenomena. To achieve higher confidence in selecting an appropriate k it is advisable to analyse POD (or WPOD for simulation data) results with at least two tests.

3.2 Singular spectrum analysis

Proper orthogonal decomposition can successfully extract trends and oscillations of the signal from its noise-contaminated measurements. However, when applied to stationary data, it appears to be no more efficient in separating unwanted components from the mean ensemble than statistical averaging. In order to reduce the computational cost of de-noising the results from steady-state simulations, another method needs to be applied. Utilising wavelet thresholding for recovering the data's mean is a possible alternative to classical orthogonal basis methods (see Sec. 3.4). The main drawback of using wavelet transforms and multiresolution analysis is the number of parameters that need to be considered *a priori*, e.g. mother wavelet, number of vanishing moments, and levels of decomposition. The choice of an appropriate model is often problematic and may lead to data misinterpretation if any deviations from it appear. Having a technique that uses a basis that is adaptive to the signal instead of an *a priori* basis is then desired. This statistical method, related to the Karhunen-Loéve transform, was described by Broomhead and King [100, 101] and is known as a singular spectrum analysis. It is also based on SVD, hence its name. Sometimes referred to as Cadzow's method[4] [103], SSA aims at decomposing the original temporal or spatial series into a collection of a slowly varying trends, oscillatory components, and noise. By applying singular value decomposition analysis to the ensemble mean, it results in more efficient noise reduction.

[4]Under certain conditions SSA may be considered as one iteration of Cadzow's basic algorithm. Performing many iterations does not necessarily lead to better results than the basic SSA [102].

3.2.1 Basic SSA

The method has been widely used in the analysis of climatic, meteorological, geophysical, and electrical data series, where its de-noising and forecasting abilities are employed. The basic scheme of SSA can be found in Ref. [104, 105] and is presented graphically in Fig. 3.4. The algorithm consists of four main steps (two for the decomposition stage, and two for the reconstruction): *embedding*, SVD, *eigentriple grouping* and *diagonal averaging*. In the embedding stage, the series X of length M is broken into a sequence of *lagged vectors* of size L by forming

$$X_i = (x_i, \dots x_{i+L-1})^\dagger, \quad (1 \leq i \leq K), \tag{3.31}$$

where $K = M - L + 1$. As a result, a *trajectory matrix* Y of series X is formed:

$$Y = [X_1 : ... : X_K] = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ x_2 & x_3 & x_4 & \dots & x_{K+1} \\ x_3 & x_4 & x_5 & \dots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & ... & x_M \end{pmatrix}. \tag{3.32}$$

In other words, by sliding a window of length L, often chosen to be $2 < L < \frac{M}{2}$, the data contained in X is mapped onto a $L \times K$ matrix, permitting further analysis. The new data set Y is a structured matrix with constant values on the diagonals, hence it is a Hankel matrix. For a stationary series, it is recommended to perform a centring procedure before processing (i.e. subtracting the mean from the data), and constructing a Toeplitz matrix for analysis using SSA [106–108].

In the second key step of the decomposition process, the trajectory matrix is subject to singular value decomposition (or eigenvalue decomposition of a symmetric matrix $YY^\dagger$). As explained in Sec. 3.1.2, SVD produces a set of left and right singular vectors U, V, also known as empirical orthogonal functions, or modes, and singular values Σ. As the SVD may be costly to perform, it can be substituted with other algorithms, e.g. QR factorisation, or the Lanczos method [109]. The Lanczos method, however, leads to reduced numerical accuracy even for relatively small matrix sizes, most likely due to

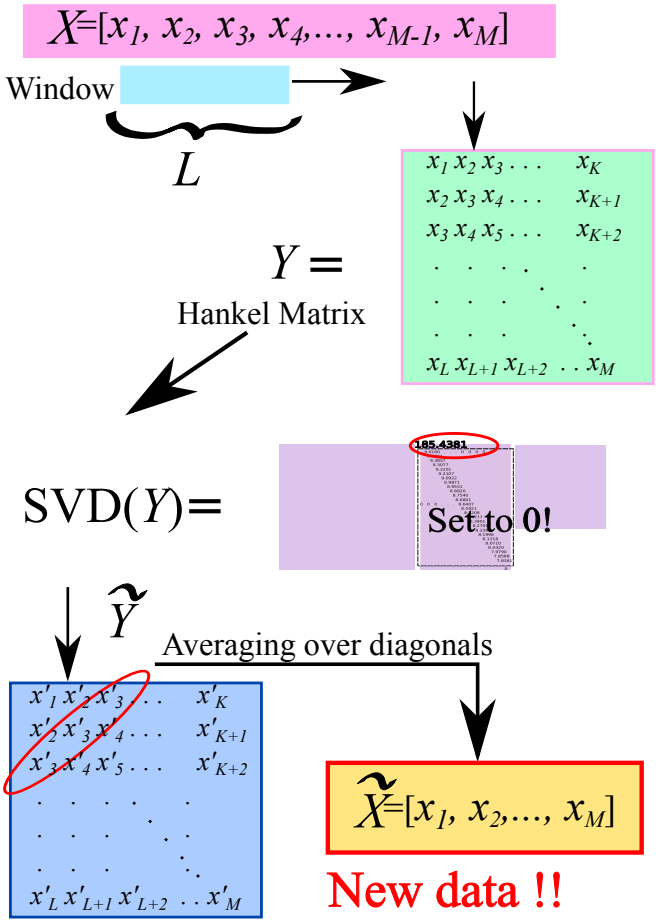


Figure 3.4: Schematic diagram of SSA algorithm.

the loss of mutual orthogonality of the vectors generated by the algorithm [110].

The following reconstruction process reduces the rank of the trajectory matrix based on finding a threshold of a noise *floor* in a scree diagram of singular values (or eigenvalues given in a descending order). In the case of noisy data sets, the Hankel matrix has a full-rank because of the partial de-correlation of the disturbed data points [111]. Otherwise its rank is limited. Analysis of the eigenspectrum enables finding the number k of significant EOFs, which is equivalent to the rank of a *clean* matrix. The outcome of the *grouping* stage, is then a collection of $k < L$ orthogonal functions (or *eigentriples*),

$$U_{(L\times k)}\Sigma_{(k\times k)}V^{\dagger}_{(M\times k)} = \Sigma_{i=1}^{k} u_i s_i v_i^{\dagger}. \tag{3.33}$$

The symbols are the same as in Sec. 3.1. Eigentriples determine a k-dimensional subspace in $\mathbb{R}^L$ of dominant frequencies, onto which the Hankel matrix is then projected

forming a new data-set, $\widetilde{Y}$. This projection onto the highest-ranked EOFs provides an optimal linear filter for white (uncorrelated) noise. The subsequent averaging over its diagonals, performed in the last stage of the reconstruction process, yields a new series $\widetilde{X}$, which is an approximation of X. In Appendix B it is shown how to perform SSA on a short signal. The rank k of the new matrix should ensure that the distance $\left\|X - \widetilde{X}\right\|$ is *small*. If X is assumed to be a set of noisy measurements, a composition of the true signal, S, and some random residual (noise), R_N, $X = S + R_N$, then SSA provides the estimate of the component of interest, $\widetilde{X} \approx S$. The residual is defined as $R_N = X - \widetilde{X} = \Sigma_{i=k+1}^{L} u_i s_i v_i^{\dagger}$. A good reconstruction of the signal is obtained in the case of its approximate separability from the unwanted components [108]. All the steps can be applied iteratively, starting each cycle with the reconstructed series. The computational cost of one iteration of basic SSA can be up to $O(M^3)$. An improved implementation with the use of the fast Fourier transform was presented by Korobeynikov [112], and leads to reduced complexity of the order of $O(kM\log(M) + k^2M)$, where k is the number of eigentriples.

A scree diagram, described in Sec. 3.1.3 as a plot of λ_k against the mode number k, is often used in the literature as a graphical method for selecting the significant structures. However, it can also be defined as a plot of total relative *energy* captured, E_λ, which represents the percentage variance held by each eigenvalue. The latter approach is preferred in this thesis as it gives information about the cumulative portion of variance contained in the first modes. It is essential to stress the importance of not relying solely on any single examination. The *quality* of the eigenspectrum strongly depends on the window length chosen, and in SSA, as well as in POD, additional tests need to be performed to make a confident selection of empirical orthogonal functions (modes) for signal reconstruction. This issue is further discussed in Sec. 3.2.2.

Any knowledge of the nature of a signal can help in extracting useful information from noisy measurements. Objects, such as a time-series or image, produce a trajectory matrix with a finite rank, which is equivalent to the number of components (frequencies) that they contain. Reconstruction of the desired data depends on the proper choice of the SVD modes (grouping), and defining an accurate rank of the true object. For example, to obtain the exponential trend, only one SVD component is needed, which

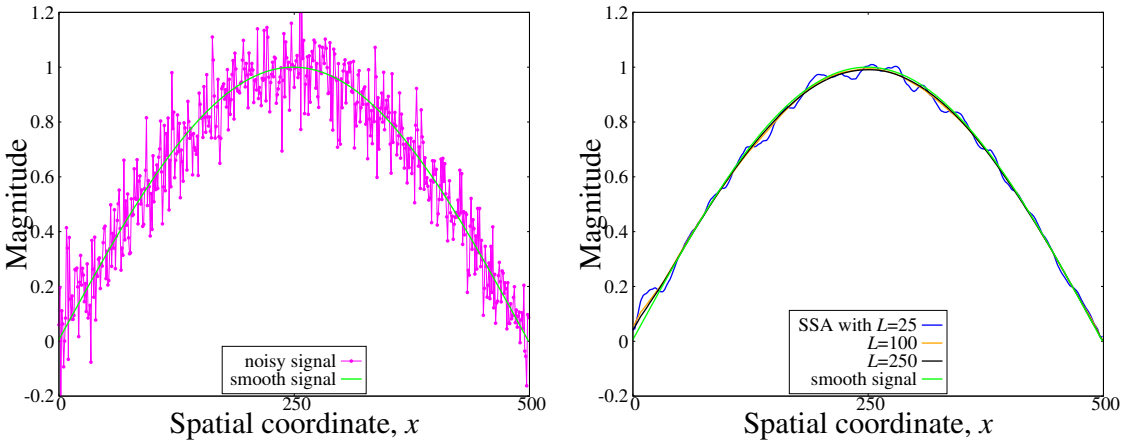
means that the data has rank one [113]. If the rank of the information that we want to extract is known prior to de-noising, the signal reconstruction can be very effective. If an assumed rank number is incorrect, the signal approximation can be polluted with spurious artifacts.

3.2.2 Window length and separability

The window length, L, is the only parameter that has to be determined prior to the SSA. Its choice may result in a weaker (or stronger) separability between genuine information and noise, influencing the effectiveness of the signal reconstruction process. There is no universal rule regarding the optimal value of L, but several general principles have been described by Golyandina and Zhigljavsky [105]:

- Theoretical results show that the window should not be greater than half of the length M of the analysed series, $L \leq \frac{M}{2}$ [114];
- For larger values of $L \in \left[0; \frac{M}{2}\right]$, the decomposition retains more details, with $L = \frac{M}{2}$ producing the most detailed description;
- Smaller window lengths serve as a linear smoothing filter;
- If there is a known periodic component in the processed data, then L should be chosen to be proportional to its period.

In order to explain the influence of the window length on noise reduction, the following examples are considered. The signal in Fig. 3.5(a), which is a smooth parabola, was corrupted with white noise obtaining SNR =16.9625 dB. To extract the original simple profile from its noisy observation, relatively large values of L are recommended [105]. However, because the parabolic trend that we wished to recover dominated the noisy measurement, the choice of L did not strongly influence the smoothing (see Fig. 3.5(b)). In Fig. 3.5, the plot of the signal of length $M = 500$ reconstructed with three different values of $L \in \{25, 100, 250\}$ are compared. For each case considered, the highest signal-to-noise ratio was achieved by extracting the rank $k = 2$, as shown in Fig. 3.6. The best de-noising was obtained with the window length equal to half of the length of the signal; the reconstructed signal with $L = 250$ had SNR = 37.9592 dB. For this problem,



(a) Noisy data. (b) Approximations with different L.

Figure 3.5: Application of SSA to a signal of length $M = 500$ corrupted with white noise (SNR = 16.9625 dB). Three window lengths were considered: $L \in \{25, 100, 250\}$. The best reconstruction of the parabolic trend was obtained with the largest window size, $L = 250$, resulting in SNR = 37.9592 dB.

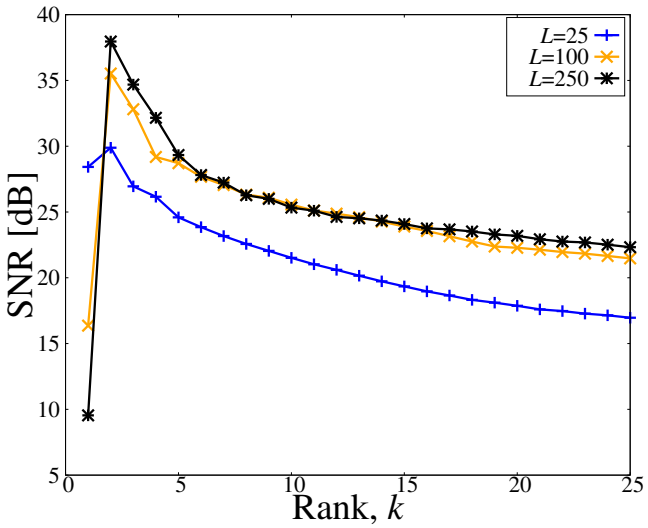
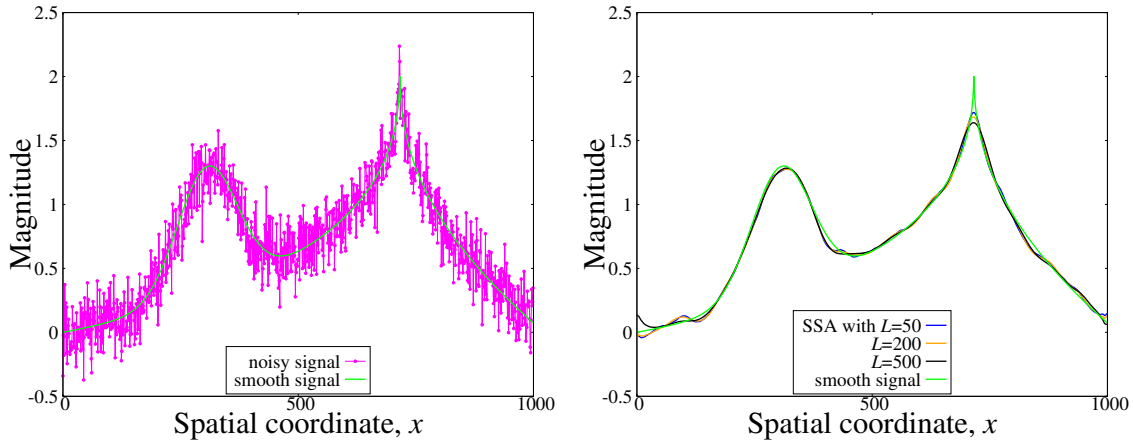


Figure 3.6: Comparison of SNR gained by SSA as a function of the rank k for different window lengths. Highest values were achieved for $k = 2$. The analysed signal was of length $M = 500$ and corrupted with additive noise (SNR = 16.9625 dB), see Fig. 3.5.

the signal reconstructed with $L = \frac{M}{5}$ also provided a good approximation with SNR = 35.5250 dB. This suggests that for smooth and simple trends the window size can be taken from a relatively wide range of values without much loss of accuracy, provided the value chosen is not too small. The smallest window size selected, $L = 25\,(\frac{M}{20})$, resulted in artifacts (SNR = 29.8848 dB). This is due to the fact that the spectrum was too coarsely decomposed, weakening the separability of the trend from the noise.

Similar results were also observed for longer data series. Using a signal with twice the length, $M = 1000$, with $L \in \{50, 200, 500\}$ gave SNR $= \{31.4632, 33.8126, 35.5614\}$ dB, respectively, when applied to noisy measurements with SNR $= 16.9768$ dB.

Smaller window sizes can produce better signal reconstruction in the case of more complex shapes, such as MATLAB *cuspamax* pictured in Fig. 3.7. For this problem,



(a) Noisy data. (b) Approximations with different L.

Figure 3.7: Reconstruction of the MATLAB cuspamax function of length $M = 1000$ from noisy measurements with SNR $= 14.8258$ dB. Three window lengths were considered: $L \in \{50, 200, 500\}$. The best approximation was received for the smallest window size, $L = 50$, resulting in SNR $=$ 28.7562 dB. Extraction of the signal from larger windows required a substantial number of small singular values, that could easily be mistaken for noise.

the signal length was $M = 1000$ and 3 windows, $L \in \{50, 200, 500\}$, were applied to the series with SNR $= 14.8258$ dB. The smallest L extracted the trend with SNR $=$ 28.7562 dB. The windows $L = 200$ and $L = 500$ provided SNR $==$ 28.0926 dB and SNR $= 27.3526$ dB, respectively. Using SSA with $L = \frac{M}{2}$ in this case performed the worst, as the decomposition was too fine, confusing small elements of signal with noise. The examination of all possible ranks showed, that for $L = 50$, $L = 200$, and $L = 500$, the highest SNR was obtained with $k = 2$, $k = 6$, and $k = 9$, respectively. Figure 3.8 shows the value of SNR for the first 50 ranks. It should be noted that the values of SNRs for any other rank (higher than $k = 2$) were smaller for $L = 50$ in comparison with the other window sizes.

In the real situation we do not know the *true* signal, but only noisy measurements, and often we are not sure of its type. We have to rely solely on examination of the eigenspectra in order to establish an adequate number of k for the approximation. For

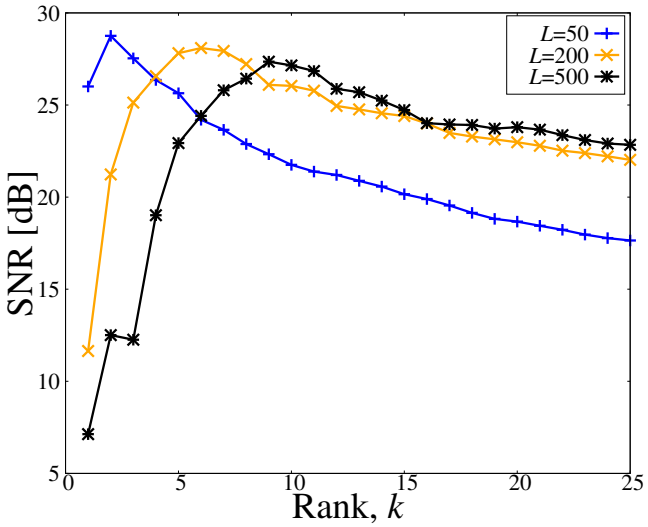
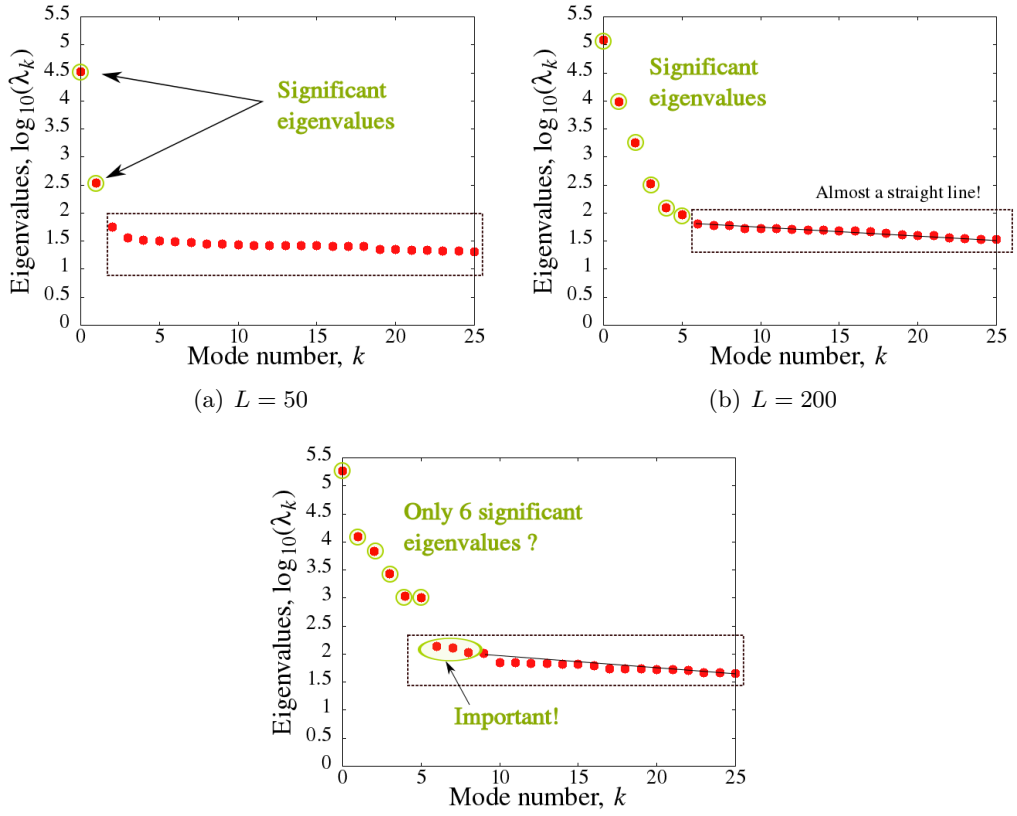


Figure 3.8: Comparison of SNR gained by SSA as a function of the rank k for window lengths $L \in \{50, 200, 500\}$. Highest values were achieved for $k = 2$, $k = 6$, and $k = 9$, respectively. The analysed signal was of length $M = 1000$ and corrupted with additive noise (SNR = 14.8258 dB), see Fig. 3.7.

the $L = 50$ used on the last considered signal, two squared singular values (eigenvalues, λ_k) separated from the *floor* in the LEV plot (see Fig. 3.9(a)), confirmed that they represented the true information. Studying of the semi-log distribution for $L = 200$ given in Fig. 3.9(b) indicates that $k = 6$ should be retained. However, a smaller number of significant eigenvalues was suggested from the scree diagrams of the eigenspectrum e.g. as presented for $L = 200$ in Fig. 3.10; only $k = 3$ eigenvalues appeared to be significant. If the original smooth signal is not known prior to de-noising, an analysis of the scree diagram alone for large L might not be sufficient to provide good trend extraction. In the case of the largest window, the estimation of the rank k was burdensome. The semi-log plot and the scree-diagram failed to determine the right number of elements, suggesting a smaller value of k (e.g. $k = 6$ in Fig. 3.9(c) instead of $k = 9$).

As discussed in Sec. 3.1.3, to gain confidence in separating noisy subspaces from any signal, the results from SVD (or EVD) should be analysed by at least two tests. Plotting singular values (or eigenvalues) in semi-logarithmic scale, in order to determine which s_k (or λ_k) are fast-decaying, should be utilised together with an investigation of the corresponding eigenvectors. Goylandina *et al.* [113] stated that the visual inspection of eigenvectors can improve detection of SVD components related to the desired information. The eigenvectors are believed to repeat the properties of the signal, e.g.



(a) $L = 50$ (b) $L = 200$

(c) $L = 500$

Figure 3.9: Semi-log plot of eigenspectrum of synthetic signal corrupted with Gaussian noise (SNR = 14.8258 dB) analysed with three window lengths, L.

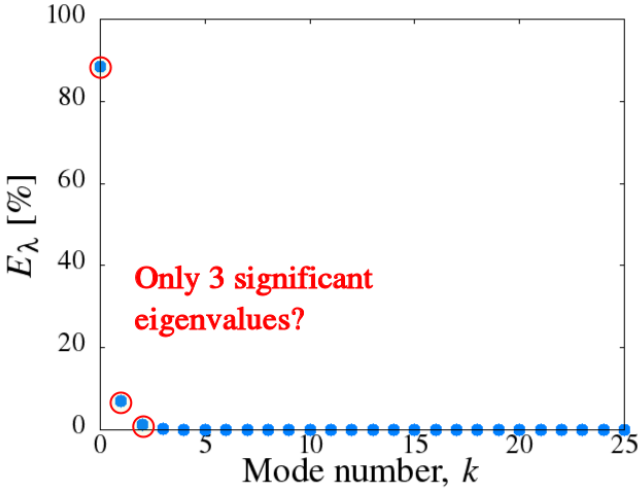
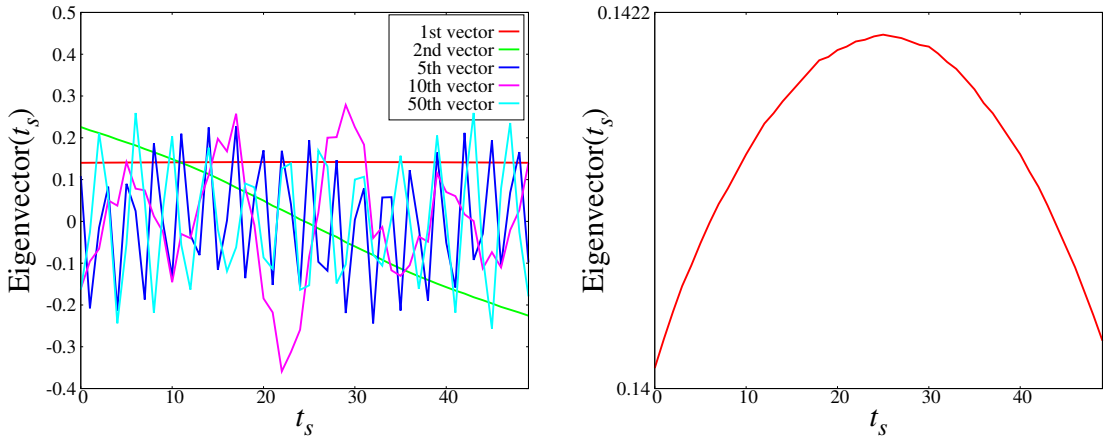


Figure 3.10: Scree diagram of the eigenspectrum obtained with SSA with the window length set to $L = 200$.

slowly-varying eigenvectors correspond to slowly-varying elements of the signal. In Fig. 3.11(a), several EOFs obtained with $L = 50$ are plotted. It can be seen that the first $k = 2$ of them represent information related to the signal; the plots of higher eigenvectors contain noise. When the first eigenvector is plotted separately in Fig. 3.11(b), it is



(a) 1st, 2nd, 5th, 10th and 50th eigenvector.

(b) 1st eigenvector.

Figure 3.11: Eigenvectors of the corrupted synthetic signals produced with $L = 50$.

clear that it contains part of the signal. The same test was applied to the other window lengths to verify the reconstruction. For $L = \frac{M}{2}$, an inspection of the eigenvectors confirms that the appropriate rank for the reconstruction is $k = 9$. This example shows that, for the largest window, the decomposition was too fine and components of the signal were represented by eigenvalues of a very small amplitude, making the extraction of the trend more difficult. In general, we recommend using smaller window sizes for more complex trends. Nevertheless, for many signals, the value of $L = \frac{M}{2}$ appears to be adequate [105, 108, 115, 116].

3.2.3 Extensions of SSA

The separation of noise from the underlying true signal by inspecting the eigenvalues yields good results as long as the unwanted disturbances are *structureless*, i.e. uncorrelated. When the noise is white, the high-ranked EOFs, which correspond to the largest eigenvalues (or singular values) obtained from the corrupted data, provide a consistent estimate of EOFs of the genuine signal [117]. When the system is affected by coloured fluctuations, the dominant eigenvalues represent not only the variance of the genuine

data but also the noise. The truncation of the eigenspectrum at the high-ranked position is unreliable, as EOF shapes depend as much on the properties of noise as on the signal.

The difficulties that appear with correlated disturbances led to the development of statistical significance tests, referred to as *Monte Carlo* SSA (MC-SSA) [107]. One role of MC-SSA is to assess whether the SSA spectrum can reject the *null hypothesis* that the time series is red noise, or any linear stochastic process in which the power spectrum declines with frequency ($1/f^{\alpha}$ fluctuations). A general description of one of the original tests consists of the following stages [106, 107]: (1) estimation of red noise parameters with the same variance and auto-covariance as the observed series, X, using a maximum-likelihood criterion; (2) generation of an ensemble of simulated red-noise, and for each realisation (called *surrogate data*), computation of the covariance matrix C_R; (3) obtaining the eigenspectra from the projection of C_R onto the original data EOFs, E_x: $\Lambda_R = E_x^{\dagger} C_R E_x$; (4) validation of the red-noise null hypothesis. The statistical distribution of the elements of Λ_R gives confidence intervals, outside of which the null hypothesis can be rejected. In other words, the modes of the signal that do not appear within the limits determined from surrogate eigenspectra, with certain confidence, can be considered as different from red-noise. Performing MC-SSA tests is a time-consuming process, and not very suitable for analysing particle data during a simulation run, especially if coupling across scales is of interest.

Singular spectrum analysis can be applied to more than one series at a time by performing *multivariate* or *multi-channel* SSA (MSSA). If we consider two sets of data, $X1$ and $X2$, we can create a joint trajectory matrix $Z = (Y1; Y2)$ or $Z = (Y1; Y2)^{\dagger}$, where $Y1$ and $Y2$ are the trajectory matrices of the individual series. The other steps are the same as for univariate SSA. The difference is that the approximated matrix is now a *block-Hankel* matrix. Moreover, the window length can be $L > \frac{(M+1)}{2}$ for trend extraction, whereas for SSA it is not recommended [113]. Multivariate SSA can be useful if there is a common pattern between analysed series. In this case, applying MSSA may improve the quality of the results compared to performing individual univariate SSA. However, for a large number of series, such a process becomes costly.

Another extension of SSA for decomposition of two-dimensional data, $X_{2\mathrm{D}}$, of size

$N \times M$ is *2D singular spectrum analysis* (2D-SSA). This method was introduced by Danilov and Zhigijavsky [118], and further developed by Golyandina and Usevich [119], and consists of similar stages as the basic SSA. However, the trajectory matrix in 2D-SSA is a *Hankel block Hankel* matrix, and processing is computationally intensive.

The algorithm of 2D-SSA is based on two-dimensional window of size $L_x \times L_y$, where $1 \leq L_x \leq N$, $1 \leq L_y \leq M$, $1 < L_x L_y < NM$, and $K_x = N - L_x + 1$, $K_y = M - L_y + 1$ [119]. All possible $L_x \times L_y$ sub-matrices of the data matrix are considered. Each sub-matrix obtained by sliding the window is vectorised in order to create a column of the trajectory matrix, $Y_{2\mathrm{D}}$. It was shown by Golyandina and Usevich [119] that the constructed matrix is of the form:

$$Y_{2\mathrm{D}} = \begin{pmatrix} H_1 & H_2 & H_3 & \dots & H_{K_y} \\ H_2 & H_3 & H_4 & \dots & H_{K_y+1} \\ H_3 & H_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ H_{L_y} & H_{L_y+1} & \dots & \dots & H_M \end{pmatrix}, \tag{3.34}$$

where every H_j is an $L_x \times K_x$ Hankel matrix built from the j-th column of the 2D-array, $X_{2\mathrm{D}}$. The following step is the SVD of Hankel block Hankel matrix, $Y_{2\mathrm{D}}$; after the eigentriple grouping, the approximation $\widetilde{X}_{2\mathrm{D}}$ of the original matrix is obtained by first averaging over diagonals within each Hankel block, and next performing *hankelisation* to the whole matrix, i.e. averaging over blocks in the diagonals. The reverse operation can also be performed.

The computational cost of performing 2D-SSA is severe. Using MSSA for a large number of signals is also high. Although faster implementations of both methods by means of the R-package has been proposed by Golyandina *et al.* [113], neither technique seems suitable for analysing big sets of particle data. In this thesis, we propose another approach to make the MSSA (or SSA) applicable not only to the mean ensemble, but also to non-stationary simulation results which form large matrices. To improve the efficiency of de-noising and avoid using complex 2D-SSA, we combined the MSSA (or SSA) with POD, by applying the multivariate SSA to dominant spatial modes. Using MSSA instead of SSA for de-noising singular vectors means we perform the truncation of

eigentriples only once. This is discussed in Sec. 3.6. In addition, an improvement in the quality of the results for more complex problems can be obtained when MSSA is applied twice to the same spatial modes, but with two different window lengths, L_1 and L_2, and the average of the result is used for the reconstruction. Utilising different window sizes enables *seeing* the signal at different resolution. This procedure is similar to the *sequential* SSA described by Golyandina *et al.* [104] and Golyandina and Zhigljavsky [105], where the trend is extracted from the series with a small window length and then periodicity is analysed from the residual using a large L.

3.3 rQRd/urQRd as more efficient SSA for large data-sets

Production of big data is common in the field of nano-fluid mechanics. There is a growing need for efficient de-noising algorithms designed specifically to handle large sets of measurements. In the recent paper by Chiron *et al.* [111], a new method has been described, which, similar to SSA, seeks a low-rank approximation of the Hankel matrix in order to extract more signal than noise. It is claimed that the technique offers a substantial improvement in the processing time of big data-sets by using random sub-sampling of the matrix, and utilising fast QR decomposition, hence its name – random QR de-noising. A further improvement of the algorithm, called uncoiled random QR de-noising (in short, urQRd), has also been proposed by Chiron *et al.* [111], differing from rQRd only in the implementation. We have studied these new methods in order to compare them with the basic SSA procedure in terms of SNR and processing time.

In the case of SSA, the correlations from data are captured by truncating the singular values, whereas rQRd uses random projections of the Hankel matrix to separate the signal from the noise. Random sampling has recently received lots of attention as an alternative dimensionality reduction tool which is significantly less expensive than SVD or EVD [120–122]. In random QR de-noising, a matrix Y_Ω, containing most of the significant information of the Hankel matrix Y (of size $L \times K$), is obtained by calculating the product of Y and a set of P_k random vectors stored in a matrix Ω:

$$Y_{\Omega(L\times P_k)} = Y_{(L\times K)} \times \Omega_{(K\times P_k)}. \tag{3.35}$$

As $P_k \leq L$, and $L \leq K$, the matrix Y_Ω is smaller than the Hankel matrix; L can here be referred to as the *order of the analysis*. The following step of the rQRd method is a QR decomposition of the new matrix, $Y_\Omega = QR$. The factorisation is performed in order to construct a projection of the original Hankel matrix onto the reduced rank orthonormal basis Q:

$$\widetilde{Y} = QQ^\dagger Y, \tag{3.36}$$

where $\widetilde{Y}$ has a rank equal to P_k. A de-noised time series $\widetilde{X}$ is then obtained in the same way as described in Sec. 3.2.1. Graphical representation of the method is shown in Fig. 3.12. According to Halko *et al.* [121], the approximation error for such a procedure satisfies

$$\left\| Y - \widetilde{Y} \right\|_2 \leq [1 + 9\sqrt{P_k} \cdot \sqrt{L}] s_{k+1}, \tag{3.37}$$

with the probability of at least $1 - 3 \cdot p_k^{-p_k}$, having the oversampling parameter $p_k = P_k - k$, and s_{k+1} being the $(k+1)$ greatest singular value of Y.

3.3.1 Introduction to rQRd/urQRd

It is recommended to have a small p_k, however, in real situations the number of components k, contained in the signal, is unknown, making estimation of an adequate oversampling parameter more difficult. The solution given by rQRd might be less optimal than the approximation constructed with SVD, as the latter gives the closest result in the Frobenius or spectral (L_2) norm. However, the rQRd is much faster when applied to large data-sets. Singular value decomposition scales as $O(LK^2)$ operations, while the processing cost of rQRd is only $O(ML)$ [111].

Chiron [111] proposed a further improvement of rQRd in terms of processing time by utilising fast Hankel matrix-vector multiplications. The alternative implementation, called uncoiled rQRd (urQRd), performs the same analytical procedure, but offers a reduction in cost of the product of Y and Ω from $O(P_k LK)$ to $O(P_k M \log(M))$. The calculation of Y_Ω is a fundamental step in the rQRd algorithm, and it is important to perform it efficiently. For Hankel (or Toeplitz) matrices the fast matrix-vector multiplication is based on the fast Fourier transform (FFT). Hankel or Toeplitz matrices can be embedded into a *circulant* matrix of double size, which can be then multiplied by a

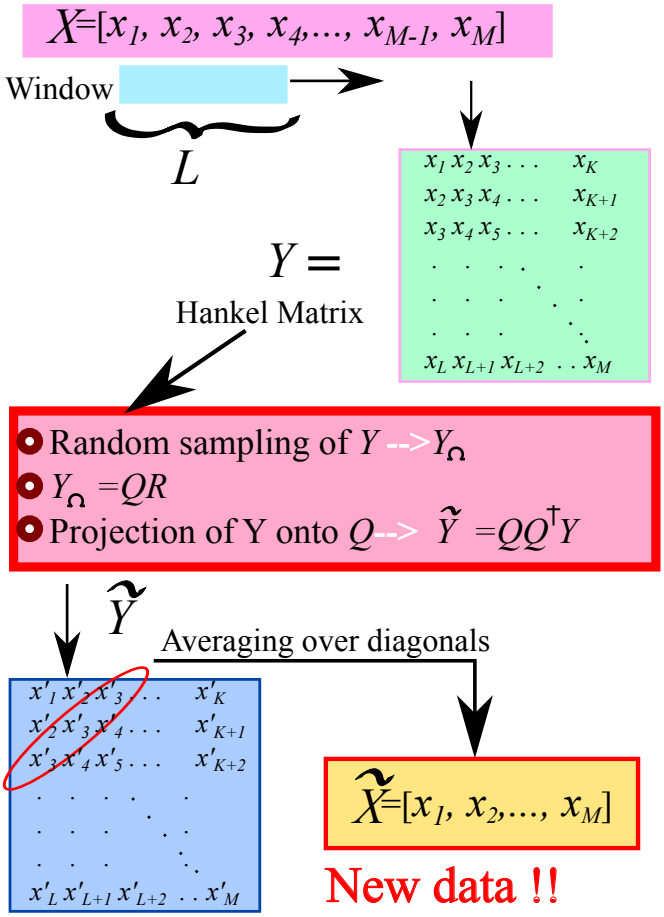


Figure 3.12: Schematic diagram of rQRd algorithm.

vector with the use of FFT and inverse FFT (IFFT). An $n \times n$ matrix is called *circulant* if it has the following form [110]:

$$A_c = \begin{bmatrix} a_1 & a_n & \cdots & a_3 & a_2 \\ a_2 & a_1 & \cdots & a_4 & a_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_n \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \end{bmatrix}. \tag{3.38}$$

The A_c is defined by its first column $a = (a_1, \cdots, a_n)^\dagger$. If the circulant matrix is to be multiplied by a vector $g = (g_j)_{j=1}^n$, an efficient FFT-based calculation can be performed:

$$\widehat{y} = \text{IFFT}(\text{FFT}(a) * \text{FFT}(g)), \tag{3.39}$$

where $*$ is an element-wise multiplication. If a Hankel matrix, constructed from an array of size $M = 2n - 1$, $X = [X_1, X_2, \cdots, X_{2n-1}]$, is embedded in $2n \times 2n$ circulant matrix, the fast Hankel matrix-vector multiplication can be computed analogous to Eq. (3.39), but with a being part of the Hankel array, X, and the vector, g, with n elements extended by appending $M - n$ zeros, $g_{\text{extended}} = (0, \cdots, 0, g_n, g_{n-1}, \cdots, g_1)^\dagger$. The desired product is then given by the first n components.

3.3.2 rQRd/urQRd v SSA

One of the benefits of utilising the rQRd algorithm, instead of SSA, is the greater flexibility in determining the rank of the Hankel matrix. In singular value decomposition, if the number k is smaller or greater than the number of the components (frequencies) contained in the signal, the reconstruction is disturbed by artifacts or missing elements. The random QR de-noising can produce some *defects* even for the rank set to the exact number of significant EOFs. However, it allows for a wider range of values, k, which produce high SNR. Moreover, iterating rQRd can further broaden the set of k providing SNR *gain*[5]. Noise reduction is improved by re-computing the product of Y and Ω, with a new Hankel matrix at each iteration; the number of iterations performed is a balance between the quality of results and processing time. Performance of the basic SSA is compared in Fig. 3.13 with rQRd performed only once, iterated twice, three times and four times. The noisy signal was MATLAB cuspamax of size $M = 1000$, analysed previously in Fig. 3.7.

Chiron *et al.* [111] explain that the most important feature of rQRd and urQRd is the fact that they can process large matrices much faster than classical methods. To confirm that assumption, we applied SSA, rQRd, and urQRd, to the Doppler signal presented in Fig. 3.14, which is one of the benchmark functions used for wavelet thresholding by Donoho and Johnstone [123]. A signal's length, $M = 1024$, was systematically increased by integer $i = 1, 2, ..., 20$. The window size (or order of the analysis) was also changing proportionally to the length of the signal, $L = 500 \cdot i$, and the number of of random vectors was kept constant, $P_k = 35$. Figure 3.15 compares SSA, rQRd and urQRd in terms of processing time against the length of the analysed signal with $k = 25$; the

[5]Gain in SNR $= (\text{SNR}_{\text{noisy}} - \text{SNR}_{\text{approximated}})/\text{SNR}_{\text{noisy}}$; absolute value is taken if $\text{SNR}_{\text{noisy}} < 0$.

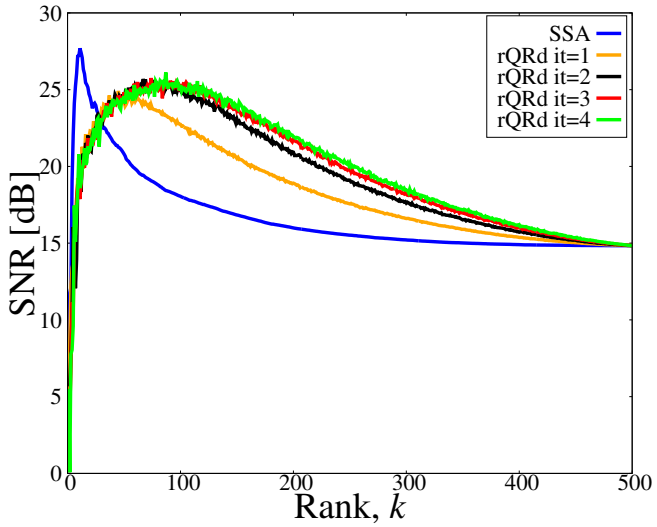


Figure 3.13: Values of SNR obtained with SSA and rQRd iterated it=1, it=2, it=3, and it=4 times. The signal considered was the MATLAB cuspamax function having SNR $\approx$ 14.8 dB.

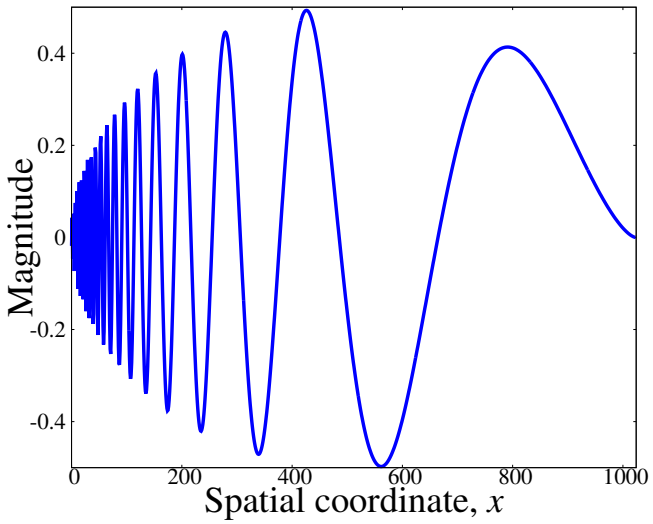


Figure 3.14: Benchmark function first utilised in wavelet de-noising analysis.

oversampling parameter was set to $p_k = P_k - k = 10$. It can be easily observed that with the higher value of i, the time to process the data with SSA dramatically increases. For the longest length, $M = M \cdot 20 = 202480$, the cost of reconstructing the signal with k EOFs was about 21 times greater than for rQRd, and 640 times higher than for urQRd which employs the FFT for matrix-matrix computations.

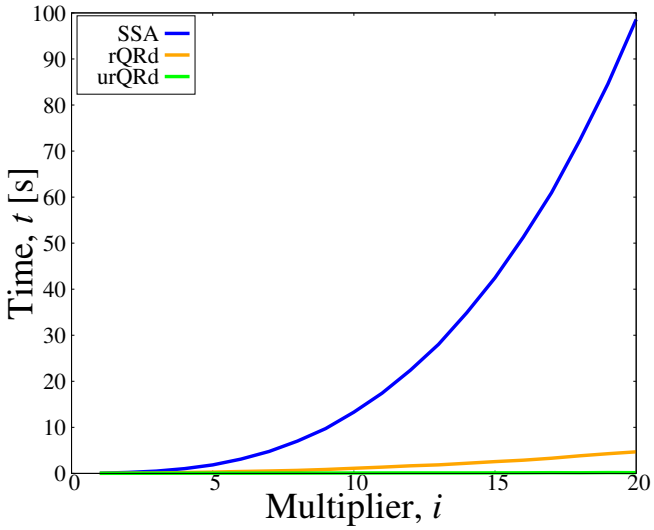


Figure 3.15: Comparison of processing time in signal reconstruction with SSA, rQRd, and urQRd. Different lengths of the signal were considered, $M = 1024 \cdot i$, where $i = 1, 2, ..., 20$.

3.4 Wavelet transform

Wavelets are an extension of Fourier analysis. The Fourier transform is a useful tool to indicate the frequency components of a signal, but it lacks a time-localisation of the events, i.e. the Fourier transform (or, if the original signal is periodic, Fourier series) displays frequencies, but hides time (or space) information. The short-time Fourier transform (STFT) utilises a moving window which analyses the function in both time and frequency domain. Unfortunately, the constant window size limits the frequency resolution; using small windows gives good time resolution, allows for better analysis of sudden changes, but it is blind to low frequencies of the signal [124]. The wavelet transform accommodates this shortcoming by automatically adapting to different components of the analysed function, using a small window for recognising brief changes, high-frequencies, and a large window to look at long-lasting, low-frequency components. The main aim in using wavelets is to transform the information of a signal into coefficients that can be easily manipulated, stored, transmitted and later used for reconstruction of the original data. In an analogous manner, POD decomposes the matrix data into orthogonal components that can be processed in order to create an approximation of the original set (see Sec. 3.1).

3.4.1 Theoretical background

The following mathematical introduction to wavelet theory is largely based on descriptions presented in [124–127]. Wavelet analysis is best understood with the continuous wavelet transform (CWT), in which a function, ψ, is used to create a family of *little wavelets*:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi^* \left(\frac{t-b}{a} \right), \tag{3.40}$$

by compressing or stretching the *mother wavelet* with a real number a, and displacing (translating) it by a real number, b. The functions defined by Eq. (3.40) have a changing time-frequency window because of scaling. For small a ($a < 1$), $\psi_{a,b}(t)$ will be short and of high frequency, while for large a ($a > 1$) the wavelet will be long and of low frequency.

A wavelet is an oscillating function of zero mean

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0. \tag{3.41}$$

It is also normalised, $\|\psi\| = 1$, and well localised (it exhibits a fast decay for $|t|$ tending to infinity). To meet this description, the mother wavelet needs to fulfil the admissibility condition:

$$C_\psi = \int_0^{+\infty} \frac{\left|\widehat{\psi}(w)\right|^2}{w} dw < +\infty, \tag{3.42}$$

where

$$\widehat{\psi}(w) = \int_{-\infty}^{+\infty} \psi(t) e^{-iwt} dt, \tag{3.43}$$

denotes the Fourier transform which measures how many oscillations at the frequency w are in $\psi(t)$. To ensure that the integral in Eq. (3.42) is finite the Fourier transform of ψ at zero should be zero, $\widehat{\psi}(0) = \int_{-\infty}^{+\infty} \psi(t)dt = 0$, giving the zero-mean condition for wavelets. It is also required that the wavelet $\widehat{\psi}(w)$ is continuously differentiable, i.e. ψ has sufficient time decay, which implies smoothness:

$$\int_{-\infty}^{+\infty} (1 + |t|) \, |\psi(t)| \, dt < +\infty. \tag{3.44}$$

Higher order moments of ψ should also vanish, i.e.

$$\int_{-\infty}^{+\infty} t^p \psi(t) dt = 0, \tag{3.45}$$

suggesting that polynomials up to a certain degree, p, are reproduced. This is equivalent to the Fourier transform decaying smoothly at $w = 0$.

The continuous wavelet transform represents a signal $f(t)$ as a function with two variables – scale and time (or space), $f_w(a, b)$. To obtain the transform, the wavelet is scaled, shifted, multiplied by the original signal, and integrated over time:

$$WT(f) = f_w(a, b) = \langle f, \psi_{a,b} \rangle = \int f(t) \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right) dt. \tag{3.46}$$

A continuous wavelet transform is then a correlation between a wavelet at different scales and the signal, i.e. it is a measure of similarity indicating which parts of the signal look like the wavelet. The admissibility condition (Eq. (3.42)) of ψ indicates the existence of a finite energy reproducing kernel which is a necessary condition for reconstructing a function from its wavelet coefficients (inverse wavelet transform, IWT):

$$f(t) = IWT(f_w) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} f_w(a, b) \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right) db \frac{da}{a^2}. \tag{3.47}$$

When $f_w(a, b)$ is known only for certain small $a < a_0$, in order to recover f, complementary information is needed. This is obtained by a scaling function, also referred to as a father wavelet, ϕ_s, which is defined as an aggregation of wavelets at scales larger than 1.

Since a one-dimensional function is mapped into a two-variable function, this continuous representation is highly redundant. Applying a CWT is an endless task and generally not suitable for engineering applications. The discrete wavelet transform (DWT), however, allows for more practical analysis by shifting and scaling the mother wavelet by e.g. powers of 2 (dyadic grid) forming basis functions often referred to as *children*:

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - n2^j}{2^j} \right), \tag{3.48}$$

where 2^j is a discrete dilation parameter, an integer $j \in \mathbb{Z}$ represents a scale resolution,

and $n2^j$ is a discrete shift. This representation provides an orthonormal basis of $L^2(\mathbb{R})$. The use of orthogonal wavelets implies the use of the discrete wavelet transform, while a nonorthogonal wavelet function can be utilised with either CWT or DWT. Orthogonal wavelets (the wavelet basis) are then a special case of discrete wavelets making the reconstruction from transform coefficients possible. They eliminate the redundancies, provide perfect recovery of the original signal, and lead to fast algorithms. Construction of orthogonal wavelet transforms, however, had not been known until Mallat's and Meyer's multiresolution theory [128, 129] from 1988/89 which resulted in development of the fast wavelet transform (FWT).

Multiresolution introduces an orthogonal wavelet transform, where a signal is analysed at scales varying by a factor of 2. The mother wavelet is used together with a dilated and translated scaling function $\phi_{s_{j,n}}(t) = \frac{1}{\sqrt{2^j}}\phi_s\left(\frac{t-n2^j}{2^j}\right)$ to provide decomposition of the signal into coefficients representing its smoothed approximation and details at different resolutions:

$$f(t) = \sum_{n\in\mathbb{Z}} c_{j_0,n}\phi_{s_{j_0,n}}(t) + \sum_{j,n\in\mathbb{Z}} d_{j,n}\psi_{j,n}(t), \tag{3.49}$$

where $-J \leq j \leq 0$, $0 \leq n < 2^{-j}$, and $j < j_0 \leq 0$ with J denoting the maximum number of resolutions. The coefficients in Eq. (3.49) are obtained by integrating the product of the functions with the signal:

$$c_{j_0,n} = \int_{\mathbb{R}} f(t)\phi_{s_{j_0,n}}dt, \tag{3.50}$$

$$d_{j,n} = \int_{\mathbb{R}} f(t)\psi_{j,n}dt. \tag{3.51}$$

In other words, the details ($d_{j,n}$) and approximation ($c_{j_0,n}$) are projections of the signal onto certain subspaces. A multiresolution approximation of a function $f \in L^2(\mathbb{R})$ at the scale 2^j, which is entirely characterised by the scaling function, is defined as an orthogonal projection on a space $V_j \subset L^2(\mathbb{R})$ [126]. Let W_{j-1} be the orthogonal complement of V_j relative to V_{j-1}:

$$V_j = V_{j-1} \oplus W_{j-1}, \tag{3.52}$$

where the space W_{j-1} is associated with the wavelet, and $\oplus$ denotes orthogonal decomposition. In the case of finite data with information at resolution level j, a wavelet transform decomposes the functional space into a direct sum of orthogonal subspaces:

$$V_j = V_{j-1} \oplus W_{j-1} = V_{j-2} \oplus W_{j-2} \oplus W_{j-1}. \tag{3.53}$$

Orthogonal wavelets carry the details necessary to increase the resolution of a signal's approximation. The detail coefficients at given j are obtained by projecting the signal f onto a complementary subspace $W \subset L^2(\mathbb{R})$.

Mallat and Daubechies [130] established a link between filter banks in signal processing and wavelets, allowing for a fast decomposition. The fast wavelet transform algorithm does not make use of the wavelet and scaling function, but of the quadrature mirror filters (QMFs)[6] that describe their interaction. In fast WT, the signal is convolved with both a high-pass filter (H_{filter}, determining the wavelet function), which produces the details of the decomposition, and a low-pass filter, (L_{filter}, associated with the scaling function) which gives the approximation of the signal. The process is shown in Fig. 3.16. Given a signal f of length M, the fast wavelet transform can consist of

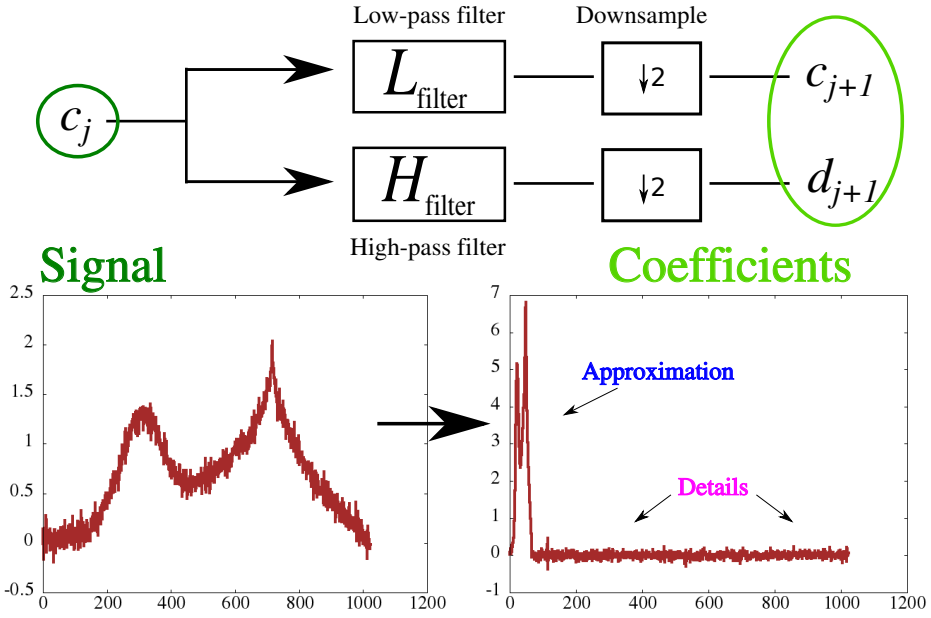


Figure 3.16: One step of DWT with filter banks (FWT).

[6]QMFs in digital signal processing are odd index alternated reversed versions of each other.

maximum $J = \log_2 M$ levels (e.g. for $M = 512$ only $J = 9$ stages of decomposition are possible). At each stage, two sets of coefficients, approximation and details, are produced by convolving the signal with the low and high-pass filters followed by dyadic decimation (downsampling, taking one sample out of two, often introduces distortions called aliasing) of approximation. The signal has then half of the number of samples which means that the scale is doubled[7]. Details are stored while the smoothed image (approximation) of the signal is again convolved with the QMFs resulting in another set of details at different scale and new approximation. The process is repeated until a desired level of decomposition is reached. This algorithm progressively drains the signal of its information. In the end, the original signal is represented by a vector containing J sets of details (each of different length due to downsampling) and only one set of approximation coefficients (see Fig. 3.17). The fast wavelet transform works then from the

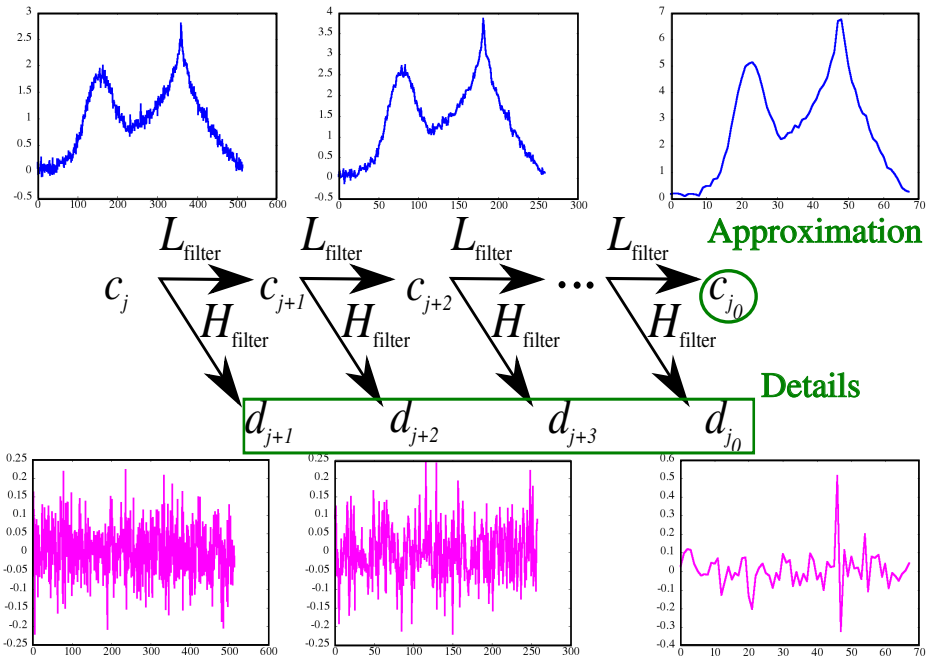


Figure 3.17: Schematic representation of FWT. Note that j is negative.

finest scale to the coarse. It is often perceived as a mathematical microscope allowing us to *see* the signal at different dyadic magnifications, offering a powerful way to decompose data into its elementary constituents across scales. Inverting the decomposition with

[7]This decomposition reduces the time resolution since only half of the number of points characterises the signal. However, at the same time, it doubles the frequency resolution, reducing the uncertainty in the frequency by half [131].

an inverse wavelet transform consists of inserting zeros (upsampling) between samples and convolving the results with the reconstruction filters which are flipped versions of decomposition QMFs.

The algorithm can be naturally extended to encode two-dimensional signals (e.g. images), where the scaling function, ϕ_s, is associated with a one-dimensional multiresolution approximation, $\{V_j\}_{j\in\mathbb{Z}}$. A two-dimensional multiresolution is defined by $\left\{V_j^2\right\}_{j\in\mathbb{Z}}$, where $V_j^2 = V_j \oplus V_j$, and W_j^2 denotes the detail space which is an orthogonal complement of the lower resolution approximation. In this case, three 2D wavelets are obtained from the 1D scaling function, ϕ_s, and the corresponding wavelet, ψ, by tensorial product [124]:

$$\psi^1(x) = \phi_s(x_1)\psi(x_2), \tag{3.54}$$

$$\psi^2(x) = \psi(x_1)\phi_s(x_2), \tag{3.55}$$

$$\psi^3(x) = \psi(x_1)\psi(x_2), \tag{3.56}$$

where for $1 \leq l \leq 3$

$$\psi_{j,n}^l(x) = \frac{1}{2^j}\psi^l\left(\frac{x_1 - n_1 2^j}{2^j}, \frac{x_2 - n_2 2^j}{2^j}\right). \tag{3.57}$$

Each new wavelet from the family $\left\{\psi_{j,n}^1(x), \psi_{j,n}^2(x), \psi_{j,n}^3(x)\right\}_{n\in\mathbb{Z}^2}$ measures variations in a different direction. This kind of two-dimensional transform leads to a decomposition of the signal into four components: approximation, and the details in three orientations (horizontal, vertical, and diagonal).

3.4.2 Signal de-noising with wavelets

To date, there exist many variants of wavelet-based thresholding, as reviewd by Mallat [126]. Our goal is to reduce the computational effort of de-noising for particle-based systems. For that reason, we focus only on the most basic estimators: soft and hard thresholding.

Donoho and Johnstone [123] pioneered the work on recovering information from noisy data using wavelet transforms. The procedure consists of decomposition of data

into wavelet coefficients, thresholding detail coefficients and applying the inverse transform to reconstruct the signal. Donoho and Johnstone proved that a nearly optimal non-linear estimator is obtained with soft thresholding (wavelet shrinkage):

$$T_S(d_{j,n}) = \begin{cases} \text{sgn}(d_{j,n})(|d_{j,n}| - T_u) & \text{if } |d_{j,n}| \geq T_u, \\ 0 & \text{otherwise.} \end{cases} \tag{3.58}$$

De-noising can also be performed with hard thresholding, defined as follows:

$$T_H(d_{j,n}) = \begin{cases} d_{j,n} & \text{if } |d_{j,n}| \geq T_u, \\ 0 & \text{otherwise.} \end{cases} \tag{3.59}$$

Similar to hard thresholding, chosen singular values (or eigenvalues) in POD are set to zero before data reconstruction.

In our study, for simplicity of analysis, the universal threshold (also called *VisuShrink*) is applied:

$$T_u = \sigma_n \sqrt{2\ln(M)}, \tag{3.60}$$

where the white noise level estimate is defined as

$$\sigma_n = MAD/0.6745, \tag{3.61}$$

with MAD being the median absolute value of the finest scale wavelet coefficients [132]. Soft thresholding has the ability to efficiently smooth the signal but with loss of some characteristics, e.g. peak heights, over-smoothing the edges. The hard threshold method generally reproduces the sharpness and discontinuities of the signal better, but at some cost in visual smoothness (can generate Gibbs-like oscillations) [132].

In the present work, filters associated with Daubechies family of wavelets and their nearly symmetric modification (Symlet wavelets) were utilised [130], mainly *db*3-*db*6 and *sym*4-*sym*8. The numbers, e.g. $3, \ldots, 8$, define how many vanishing moments are used. The p-th moment of a function f is the integral of the function multiplied by its variable raised to the power p, as given in Eq. (3.45). The number of vanishing moments determines what the wavelet cannot recognise, e.g. $p = 2$ makes a wavelet *blind* to linear and quadratic functions (i.e. wavelet coefficients are zero) [124]. In other words, the

choice of p controls how much information is concentrated in a relatively small number of coefficients. In this study, for Daubechies filters, $p = 3, \ldots, 6$ were applied as they provided a good compromise in terms of signal-to-noise ratio and smoothness of data reconstruction.

3.4.3 Empirical Wiener filter

In 1949, Norbert Wiener [133] formulated a continuous-time, optimal estimation analysis of time series. The extension of his theory to discrete time enabled its practical use. Since then, the Wiener filter and its modifications have been utilised in a range of applications, such as signal detection and noise reduction [21]. Assuming that an underlying signal is smooth, a Wiener filter minimises the mean square error between an estimated random process and a desired process. It strikes an optimal balance in the bias-variance trade-off, i.e. inverse filtering and noise smoothing. However, in real situations it is very challenging to choose a signal model for designing the filter because it requires an exact knowledge of the true signal and the noise statistics.

As derived by Vaseghi [21], the Wiener filter can be expressed in the form:

$$W_{\text{filter}} = R_{ff}^{-1} r_{ft}, \tag{3.62}$$

where R_{ff} is the autocorrelation matrix of the input signal (noisy), and r_{ft} is an element of the cross-correlation vector of the input and the desired signal. Given a degraded signal

$$f(t) = f_{\text{true}}(t) + f_{\text{noise}}(t), \tag{3.63}$$

consisting of true data corrupted by some noise that is uncorrelated with the signal, the filter can be defined by expressing the autocorrelation matrix, R_{ff}, as the sum of the autocorrelation matrix of the signal f_{true} and noise f_{noise}:

$$R_{ff} = R_{tt} + R_{nn}. \tag{3.64}$$

In Eq. (3.64), R_{tt} and R_{nn} are the autocorrelation matrices of the noise-free signal and the noise, respectively. The elements of the cross-correlation vector of the input and

desired signal in Eq. (3.62) can be replaced by elements of R_{tt}, r_{tt}, resulting in:

$$W_{\text{filter}} = \frac{r_{tt}}{R_{tt} + R_{nn}}. \tag{3.65}$$

Wiener filters are often applied in the frequency domain as such studies provide useful insight into data analysis. With Eq. (3.68) expressed in the form

$$\widehat{f}(w) = \widehat{f}_{\text{true}}(w) + \widehat{f}_{\text{noise}}(w), \tag{3.66}$$

where $\widehat{f}(w)$, $\widehat{f}_{\text{true}}(w)$, and $\widehat{f}_{\text{noise}}(w)$ are the input, true signal, and noise spectra, respectively, the Wiener filter is then given as

$$W_{\text{filter}}(w) = \frac{P_{tt}(w)}{P_{tt}(w) + P_{nn}(w)}, \tag{3.67}$$

with P_{tt} and P_{nn} denoting the signal and noise power spectra.

The main practical problem in the implementation of a Wiener filter is that the desired signal is not readily available. Ghael *et al.* [134] proposed a straightforward estimate of the signal and noise by utilising wavelet transforms. The wavelet-based empirical Wiener filtering, referred to as WienerShrink or WienerChop, performs two wavelet transforms; the first transform (WT_1) produces estimates of the desired data and noise. The approximations are then used to design an empirical Wiener filter, which de-noises the original signal in the WT_2 domain. An inverse transform, IWT_2, is then applied to obtain the new data. Figure 3.18 illustrates the WienerChop procedure.

In the wavelet domain, Eq. (3.68) becomes

$$f_w(i) = f_{\text{true}_w}(i) + f_{\text{noise}_w}(i), \tag{3.68}$$

with $f_w = WT(f)$, $f_{\text{true}_w} = WT(f_{\text{true}})$, $f_{\text{noise}_w} = WT(f_{\text{noise}})$, and WT denoting an orthonormal wavelet transform. The goal is to estimate the true signal wavelet coefficients, f_{true_w}, from the noisy observation, f_w. An approximation, $\widetilde{f}_{true_{w_1}}$, of the signal's coefficients, $f_{true_{w_1}} = WT_1(f_{\text{true}})$, is obtained in the domain of WT_1 by thresholding the wavelet coefficients. The noise level, σ_n, is calculated from the finest details according to Eq. (3.60), and an inverse transform is applied to the data, $\widetilde{f}_{\text{true}} = IWT_1(\widetilde{f}_{true_{w_1}})$.

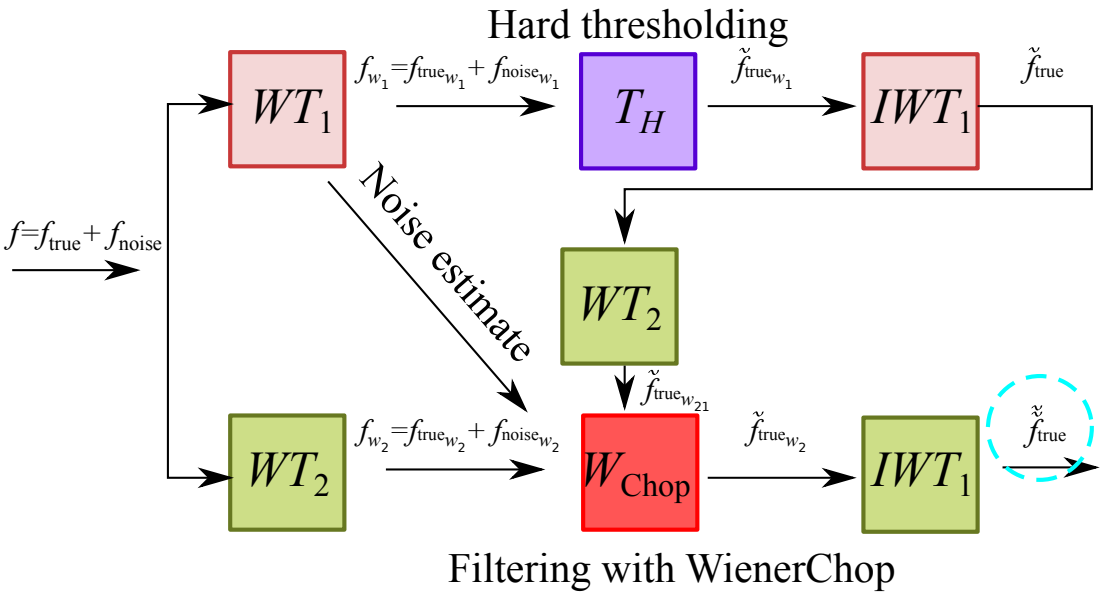


Figure 3.18: Wavelet-based empirical Wiener filtering, WienerChop.

A second transform is then performed, and the estimation of the signal in WT_2 with the noise variance is used to construct the wavelet-based Wiener filter:

$$W_{\text{Chop}} = \frac{\widetilde{f}_{true_{w_{21}}}}{\widetilde{f}_{true_{w_{21}}} + \sigma_n^2}. \tag{3.69}$$

In Eq. (3.69), the subscript w_{21} indicates that WT_2 was applied to the true signal estimate obtained from WT_1. The WT_2 of the original signal, $f_{w_2} = WT_2(f)$, allows to filter the coefficients with WienerChop. After de-noising, the inverse transform is applied to recover the final estimation of the true signal, $\tilde{\tilde{f}}_{\text{true}}$.

Choi and Baraniuk[135] explained that the rationale behind applying Wiener filtering to wavelet coefficients arises from the fact that the wavelet transform decorrelates data. Assuming perfect decorrelation of noisy coefficients, the Wiener filter is optimal in the sense of minimising the mean squared error (MSE). Ghael *et al.* [134] stated that the WienerChop is fairly insensitive to the choice of WT_1 and WT_2 as long as both transforms are adequate for classical wavelet thresholding procedures. While the WienerChop requires the calculation of two wavelet transforms, the increase in performance often outweighs the increase in computational cost, which is a constant factor. Although originally the method has been used with hard thresholding, wavelet shrinkage can also be utilised for obtaining the first estimate of signal coefficients. In this thesis,

the application of WienerChop for both one-dimensional and two-dimensional data is presented in Chapters 4 and 5. Additionally the coupling of POD with WienerChop is discussed in Sec. 3.6.

3.5 Empirical mode decomposition

In 1998, Huang *et al.* pioneered a nonlinear technique, referred to as empirical mode decomposition, for adaptively decomposing an unsteady signal into a finite sum of zero-mean AM (or amplitude modulation) and FM (or frequency modulation) elements, called intrinsic mode functions (IMFs), based on a direct extraction of the energy associated with various intrinsic time scales [136]. The filtered out functions form a complete and nearly orthogonal basis for the original signal, even though they are not necessarily orthogonal. An explanation for this is given by Huang [136]: "(...) the real meaning here applies only locally. For some special data, the neighbouring components could certainly have sections of data carrying the same frequency at different time durations. But locally, any two components should be orthogonal for all practical purposes." Orthogonality is a requirement only for linear decomposition systems. As Huang explains, it would not make physical sense for a nonlinear decomposition as in EMD.

3.5.1 Performing EMD

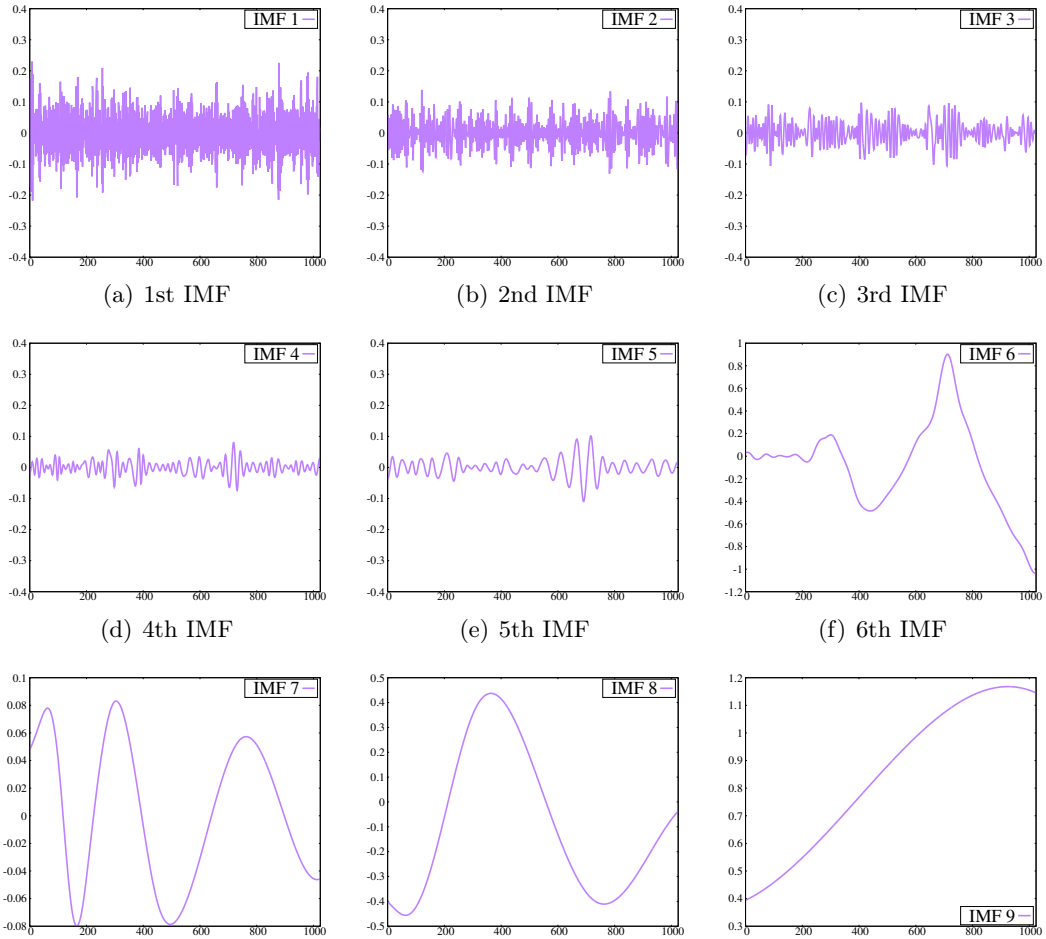
Although the method has been shown to be effective and quite versatile in many applications [31, 137, 138], the technique is essentially defined by an algorithm, and does not have an analytical formulation. The decomposition is based on the assumption that a signal has at least two extrema (minimum, t_-, and maximum, t_+). If the data does not contain any, it can be differentiated once or more times to reveal them. Empirical mode decomposition looks at the evolution of a signal, $f(t)$, between two consecutive extrema, and defines a local high-frequency part (called the detail), $h(t)$, and a low-frequency element (the trend), $r(t)$:

$$f(t) = r(t) + h(t),\ t_- \leq t \leq t_+, \tag{3.70}$$

The algorithm of EMD can be summarised in the following steps:

- *Step 1*: Identify the extrema of a given signal $f(t)$.
- *Step 2*: Obtain the envelopes, $e_{\min}(t)$ and $e_{\max}(t)$, by interpolating between minima and maxima, respectively.
- *Step 3*: Compute the mean of the two envelopes, $m_1^1(t) = \frac{e_{\min}(t)+e_{\max}(t)}{2}$.
- *Step 4*: Extract the detail by subtracting the mean from the signal, $h_1^1(t) = f(t) - m_1^1(t)$.
- *Step 5*: Examine whether the residual $h_1^1(t)$ satisfies the definition of intrinsic mode function according to a stopping criterion.
  - NO: Repeat n-times *Step 2* to *Step 5* until the conditions are met. Then: $\text{IMF}_1 = h_1^n(t) = h_1^{n-1}(t) - m_1^n(t)$. This refining procedure is referred to as a *sifting* process.
  - YES: The first IMF is found, $\text{IMF}_1 = h_1^1(t)$, which contains the shortest period component of the signal.
- *Step 6*: Iterate on the residual, $f(t) - \text{IMF}_1 = r_1$.

The number of extrema contained in each IMF is decreased when going from one residual to the next, as shown in Fig. 3.19. In other words, at first the finest local mode is separated from the data, and with an increasing number of modes the scale becomes more coarse. The necessary conditions for existence of intrinsic mode functions (in *Step 5*), that define a meaningful instantaneous frequency, are that the functions are symmetric with respect to the local zero mean, and have the same numbers (or differing at most by one) of zero crossings and extrema. All extrema in IMF appear as an alternation of local minima and maxima separated by only one zero-crossing. Flandrin *et al.* [139] stated that determining the average number of zero-crossings in a mode is a meaningful way of defining its mean frequency. The procedure is iteratively applied on the residual consisting of all local trends until either the components are smaller than a certain predetermined value, or the residue becomes a monotonic function from which no more IMFs can be extracted. The original signal can be recovered by simply



(a) 1st IMF (b) 2nd IMF (c) 3rd IMF

(d) 4th IMF (e) 5th IMF (f) 6th IMF

(g) 7th IMF (h) 8th IMF (i) 9th IMF

Figure 3.19: The successive empirical mode decomposition components of MATLAB cuspamax signal corrupted with noise.

summing up all the $p = 1, 2, 3, \ldots, P_{\max}$ empirical modes and residue:

$$f(t) = \sum_{p=1}^{P_{\max}} \mathrm{IMF}_p + r_{P_{\max}}, \tag{3.71}$$

where $r_{P_{\max}}$ can be either the mean trend or a constant. Although EMD is constructed solely from data and follows a simple algorithm, it cannot be considered as fully unique, as it depends on a number of user-defined settings, e.g. choice of the stopping criterion used in the sifting process, or the interpolation scheme for constructing the envelopes. Due to a lack of sound mathematical analysis, tuning the parameters for EMD might not be straightforward, but useful guidelines based on numerous experiments can be

found in the literature [140, 141].

Empirical mode decomposition can introduce mode mixing when the local minimum (or maximum) of two signals with different frequencies overlap. The IMFs consist then of oscillations of dramatically disparate scales, resulting in a lack of stability. In other words, a signal of a similar scale in a noisy data set could possibly be contained in the same IMF component. As the real data often contains a certain amount of intermittency, it is important to make sure that the decomposition is reliable and suitable for physical interpretation. To overcome this drawback a modified approach, named ensemble EMD, was introduced by Wu and Huang [140]. The ensemble empirical mode decomposition (EEMD) is a noise-assisted data analysis method that generates multiple noise realisations to keep the physical uniqueness of the IMFs (IMFs can be separated from each other). This new technique is based on the study presented in [137, 138], which showed that EMD is effectively an adaptive dyadic filter bank [8], resembling those involved in wavelet decomposition (see Sec. 3.4), when applied to white noise. The major steps of the procedure are:

- *Step 1*: Add white noise to the signal to obtain different measurements.
- *Step 2*: Decompose the corrupted data into IMFs.
- *Step 3*: Repeat steps 1 and 2 with different noise, and calculate the ensemble mean of the corresponding IMFs.

Finite amplitude white noise has to be used to exhaust all possible solutions. Since the noise in each trial is different, and it fills all the scale space uniformly, the added disturbances are averaged out with a sufficient number of samples. According to Wu and Huang [140], the only part that survives such a process is the component of the original (true) signal. Wu and Huang [141] compared the role of added noise in the EEMD to that of a catalyst in a chemical reaction, which only helps in the process but is not a part of the final result. For efficient application of the method, it is important to choose an adequate noise amplitude. It should not be too small, as it may not have the ability to change the extrema of IMFs. On the other hand, having

[8] A dyadic filter bank is a set of band pass filters with a constant band pass shape, which decompose a broadband signal into a collection of more band-limited components by repeatedly dividing the frequency range [140].

too strong fluctuations may require a larger ensemble in order to cancel them out, resulting in increased computational cost. The difference between the true signal and the result of the ensemble decreases as the inverse square-root of the number of samples. Although ensemble EMD offers an improvement over the original approach, it has some weaknesses. One important drawback of EEMD is that an ensemble solution does not fully meet the requirements of IMF. This can be tackled by performing sifting of the EEMD components.

3.5.2 EMD-based de-noising method

One of the useful applications of EMD is trend recovery by removing high frequency fluctuations that does not require any *a priori* basis, unlike e.g. wavelet thresholding. The most natural way to utilise this method for data filtering is to decide which IMFs are degraded, and which represent elements of the true signal, and perform partial reconstruction. A *significance test*, developed by Flandrin *et al.* [139] and Wu and Huang [137], provides IMF statistics in noise-only situations that can help identify important modes to make a confident choice of the IMFs that should be discarded. The procedure is based on comparison of the energy of IMFs from the decomposition of analysed data against that obtained from pure noise; having a relatively large discrepancy suggests that the particular IMF contains useful information. According to Flandrin *et al.*, the energy of IMFs obtained from a fractional Gaussian noise can be approximated as:

$$E_{\mathrm{IMF}}^{(p)} = \frac{E_{\mathrm{IMF}}^{(1)}}{\beta_H} \rho_H^{-2(1-H_{in})p},\ p = 2, 3, 4, \ldots, P_{\max}, \tag{3.72}$$

where $E_{\mathrm{IMF}}^{(1)}$ is the energy of the first IMF and β_H, ρ_H are parameters that are dependent on the number of sifting iterations and the Hurst index, H_{in}, which defines the type of noise (with long or short correlations) used. The fractional Gaussian noise or fractional Brownian motion (fGn/fBm) of index H_{in}, with $0 < H_{in} < 1$ being physically meaningful, is defined as a zero-mean Gaussian stationary process whose autocorrelation sequence $r_H(\Delta t) = \langle x(t)x(t + \Delta t)\rangle$ is

$$r_H(\Delta t) = \frac{\sigma_n^2}{2}\left(|\Delta t - 1|^{2H_{in}} - 2\,|\Delta t|^{2H_{in}} + |\Delta t + 1|^{2H_{in}}\right). \tag{3.73}$$

Equation (3.73) defines three classes of correlation:

- $0 < H_{in} < \frac{1}{2} \Rightarrow r_H(\Delta t)$ is negative,
- $H_{in} = \frac{1}{2} \Rightarrow r_H(\Delta t)$ is zero,
- $\frac{1}{2} < H_{in} < 1 \Rightarrow r_H(\Delta t)$ is positive long-range dependence,

where the arrow means *implies*. Therefore, the $H_{in} = \frac{1}{2}$ indicates discrete white noise, and for that special case Eq. (3.72) reduces to

$$E_{\mathrm{IMF}}^{(p)} = \frac{E_{\mathrm{IMF}}^{(1)}}{\beta_H} \rho_H^{-p}. \tag{3.74}$$

For such problems, Flandrin *et al.* [139] propose using $\beta_H = 0.719$ and $\rho_H = 2.01$, resulting in

$$E_{\mathrm{IMF}}^{(p)} = \frac{E_{\mathrm{IMF}}^{(1)}}{0.719} 2.01^{-p}. \tag{3.75}$$

In practise, the energy of IMF_1 can be estimated from the sum of its squared elements:

$$E_{\mathrm{IMF}}^{(1)} = \sum_{t=1}^{t_{\max}} (\mathrm{IMF}_1(t))^2. \tag{3.76}$$

Utilising all the above information, the significance IMF test can be summarised in the following steps:

- *Step 1*: Assuming that the first mode contains only noise, estimate the noise level in the processed data by computing from Eq. (3.76).
- *Step 2*: Establish the noise-only model with Eq. (3.72) and (3.76).
- *Step 3*: Compute the EMD of the noisy signal, and compare the IMF's energies to the ones obtained from the model at a selected confidence level.
- *Step 3*: Discard the IMFs having energies below the given threshold, and reconstruct the signal with the residual and the remaining modes.

The success of the described procedure relies on the fact that the decomposition provides noise or signal-only modes. However, in some cases noise can be distributed over all IMFs, making the method less effective.

Alternative EMD-based de-noising procedures inspired by level-dependent wavelet thresholding were proposed by Kopsinis and McLaughlin [30]. The first method directly thresholds the noisy IMFs in order to locally exclude low-energy parts that are expected to be corrupted. The IMF-dependent threshold value is defined in a similar manner as in Eq. (3.60):

$$T_{\mathrm{IMF}}^{(p)} = C_t \sqrt{E_{\mathrm{IMF}}^{(p)} 2\ln(t_{\max})}, \tag{3.77}$$

where C_t is a constant, and $t_{\max}$ denotes the length of a single IMF. The energy associated with the first IMF is established similarly to the noise estimate in Eq. (3.61):

$$E_{\mathrm{IMF}}^{(1)} = \left(\frac{\mathrm{median}(|\mathrm{IMF}_1|)}{0.6745} \right)^2. \tag{3.78}$$

The IMF energies of the remaining modes are determined according to Eq. (3.75). A direct EMD thresholding (EMD-DT) is described as

$$\widetilde{\mathrm{IMF}}_p(t) = \begin{cases} \mathrm{IMF}_p(t) & \text{if } |\mathrm{IMF}_p(t)| > T_{\mathrm{IMF}}^{(p)}, \\ 0 & \text{otherwise}, \end{cases} \tag{3.79}$$

for hard thresholding and

$$\widetilde{\mathrm{IMF}}_p(t) = \begin{cases} \mathrm{sgn}(\mathrm{IMF}_p(t)) \left(|\mathrm{IMF}_p(t)| - T_{\mathrm{IMF}}^{(p)} \right) & \text{if } |\mathrm{IMF}_p(t)| > T_{\mathrm{IMF}}^{(p)}, \\ 0 & \text{otherwise}, \end{cases} \tag{3.80}$$

in the case of soft thresholding. The reconstruction of the de-noised signal is given by

$$\tilde{f}(t) = \sum_{p=P_1}^{P_2} \widetilde{\mathrm{IMF}}_p(t) + \sum_{p=P_2+1}^{P_{\max}} \mathrm{IMF}_p(t), \tag{3.81}$$

where the parameters P_1 and P_2 were introduced by Kopsinis and McLaughlin [30] to provide some flexibility on the choice of modes to be discarded, or kept and thresholded.

Kopsinis and McLaughlin observed that such direct application of thresholding is not correct due to the oscillatory nature of IMFs. As the authors pointed out, even in a noiseless example, for any interval $Z_j^{(p)} = \left[z_j^{(p)}, z_{j+1}^{(p)} \right]$, with $j = 1, 2, 3, \ldots, P_z$, the absolute amplitude of the p-th IMF will be smaller than any non-zero threshold close to the zero-crossings, $z_j^{(p)}$ and $z_{j+1}^{(p)}$. Thresholding based just on the absolute amplitude

of IMF samples can result in some discontinuities. To overcome this difficulty, Kopsinis [30] proposed to analyse the intervals $Z_j^{(p)}$ and decide whether they are noise- or signal-dominant based on the single extrema $\mathrm{IMF}_p\left(e_j^{(p)}\right)$ that correspond to this interval. The improved procedure, referred to as EMD interval thresholding (EMD-IT), translates to

$$\widetilde{\mathrm{IMF}}_p\left(Z_j^{(p)}\right) = \begin{cases} \mathrm{IMF}_p\left(Z_j^{(p)}\right) & \text{if } \left|\mathrm{IMF}_p\left(e_j^{(p)}\right)\right| > T_{\mathrm{IMF}}^{(p)}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.82}$$

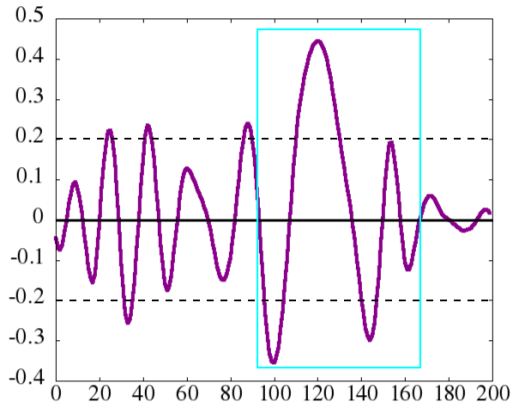
where $\mathrm{IMF}_p\left(Z_j^{(p)}\right)$ denotes samples of the p-th IMF in the interval $Z_j^{(p)}$, and $e_j^{(p)}$ indicates where extrema are reached between points $z_j^{(p)}$ and $z_{j+1}^{(p)}$. It can be seen that EMD-IT closely resembles wavelet thresholding, where the operation is performed to the wavelet coefficients instead of data samples. Figure 3.20 visualises the difference between EMD-DT and EMD-IT in the treatment of IMF intervals between zero-crossing. Incorporating the idea of soft thresholding for EMD-IT yields

$$\widetilde{\mathrm{IMF}}_p\left(Z_j^{(p)}\right) = \begin{cases} \mathrm{IMF}_p\left(Z_j^{(p)}\right) \dfrac{\left|\mathrm{IMF}_p\left(e_j^{(p)}\right)\right| - T_{\mathrm{IMF}}^{(p)}}{\left|\mathrm{IMF}_p\left(e_j^{(p)}\right)\right|} & \text{if } \left|\mathrm{IMF}_p\left(e_j^{(p)}\right)\right| > T_{\mathrm{IMF}}^{(p)}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.83}$$

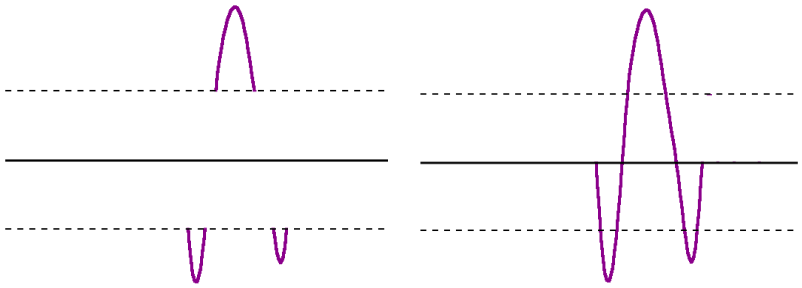
Equation (3.83) reduces in a smooth way all the IMF samples that correspond to zero-crossing intervals with extrema exceeding the threshold, in order for the extremum amplitude to be decreased exactly by the threshold value.

Further improvement of the EMD-IT method inspired by invariant wavelet-based thresholding was described by Kopsinis and McLaughlin [30]. In the neighbourhood of discontinuities, wavelet de-noising can exhibit pseudo-Gibbs phenomena that can be suppressed with so called *cycle spinning* introduced by Coifman and Donoho [142], which averages over de-noised cyclically-shifted versions of the signal or image. Incorporating the same idea for EMD requires construction of different noisy versions of the unknown true signal. Assuming that the first IMF represents mainly noise, this can be obtained by altering in a random way the positions of IMF_1 samples, and then adding the resulting mode to the sum of the remaining IMFs. For clarity, the iterative EMD interval-thresholding (EMD-IIT) is summarised as follows:

- *Step 1*: Decompose the signal with EMD.



(a) Section of 4th IMF of white noise to which thresholding is applied.



(b) The result of applying EMD-DT. Note that discontinuities are created.

(c) Outcome of hard thresholding with EMD-IT. The sections between zero-crossings, that contain extrema of greater absolute value than the threshold, are recovered. The threshold limits are indicated by dashed lines.

Figure 3.20: Comparison of EMD-DT treatment of IMF section against EMD-IT.

- *Step 2*: Perform partial reconstruction with all IMFs apart from the first one.
- *Step 3*: Randomly change the position of elements in IMF_1.
- *Step 4*: Construct a different noisy version of the signal by adding the altered mode to the sum of $P_{\max} - 1$ intrinsic functions.
- *Step 5*: Decompose the new noisy signal with EMD.
- *Step 6*: Perform EMD-IT de-noising on new IMFs.
- *Step 7*: Repeat from the *Step 3* to *Step 6* until there are no more IMFs.

- *Step 8*: Obtain the ensemble solution of all de-noised versions of the signal.

Kopsinis and McLaughlin [30] used two approaches for altering the data:

- Random circulation – the samples of IMF are circularly shifted.
- Random permutation – the positions of elements in the first IMF are randomly changed.

The article also describes the clear iterative EMD interval thresholding method that performs better than EMD-IIT in cases where the SNR is relatively high. However, in this work, most of the signals are degraded by high noise level, therefore this technique is not utilised. Both EMD-IT and EMD-IIT provide promising results for filtering particle data, and their performance is further studied in the following chapters.

3.6 POD+

The goal of this research is to not only utilise de-noising techniques from different fields of science for particle simulations, but also to improve available methods for this application. Proper orthogonal decomposition with SVD provides the most optimal (i.e. with the lowest rank) clean data reconstruction. However, it is not beneficial for de-noising steady-state simulation results as it only recovers an approximation of the mean, which could simply be obtained by cumulative averaging of all the observables. For that reason, alternative methods, such as SSA or EMD-IIT, should be utilised for steady-state simulations. Moreover, POD requires large data-sets in order to successfully separate noise in unsteady measurements from significant structures.

The need to enhance POD's efficiency has led to novel couplings of the classical orthogonal approach with other methods described in this thesis. Filtering procedures, including wavelet thresholding, EMD-IT and SSA/MSSA, are applied within SVD's domain to reduce the number of observations required for clean data recovery. Additional de-noising is performed, in the majority of problems, only on the dominant spatial modes obtained after SVD analysis. Combinations such as POD+wavelet thresholding, POD+EMD-IT, etc., result in improved filtering properties in comparison to applying a single method. The algorithm for POD+ techniques is explained below by taking the

example of coupling SVD with wavelet thresholding. The advantages and weaknesses of utilising each POD+ method are summarised in the chapters that follow.

In order to achieve better efficiency of POD in processing unsteady fields, we combined the method with wavelet-based filtering with fixed (universal) thresholding (see Eq. (3.60)). In this new procedure, wavelet thresholding is applied within POD's domain, hence the name WAVinPOD. The algorithm exhibits promising results when applied to both synthetic signals and particle data. In the following sections, it is shown that WAVinPOD outperforms the other estimators in de-noising synthetic data, achieving higher signal-to-noise ratios for a smaller number of observations.

Combining the wavelet transform with SVD (or EVD) has already been proposed in the literature. Most of the procedures involve using SVD (or EVD) for noise level estimation, transforming a signal to the wavelet domain and performing SVD (or EVD) on the chosen coefficients or the whole matrix (after IWT), as in Bakshi's Multiscale PCA [143] or its extension, multivariate wavelet de-noising developed by Aminghafari *et al.* [144]. However, these methods appear to be computationally expensive, considering the number of operations performed, making them unsuitable for de-noising particle-based simulations.

In the proposed WAVinPOD, the wavelet thresholding is performed only on spatial modes corresponding to the most energetic or/and fast decaying (dominant) singular values, as shown in Fig. 3.21. Applying wavelet thresholding in the SVD domain preserves the dimensionality reduction. Wavelet de-noising is used to eliminate spatial fluctuations from the dominant modes, which would require larger amounts of data for POD or WPOD alone.

More precisely, the general procedure for WAVinPOD de-noising is as follows:

- *Step 1*: Perform SVD on matrix data A.
- *Step 2*: Define adaptively the number k of dominant modes and set all the higher singular values to zero.
- *Step 3*: Perform a wavelet transform of the k spatial modes corresponding to the most energetic singular values.
- *Step 4*: Apply wavelet de-noising (soft or hard) with universal thresholding to the

detail coefficients and reconstruct the modes with the inverse wavelet transform.

- *Step 5*: Multiply the matrices according to Eq. (3.16) to obtain the data approximation.

When the method is applied to data during a simulation run, the moving window is used in the same manner as in WPOD (see Sec. 3.1.3).

Dominant temporal modes in WAVinPOD are left unchanged. Assuming that the noise is uncorrelated, and a sufficient number of observations are provided, the first k singular vectors, which define the behaviour of the signal over time, are less affected by fluctuations than the corresponding spatial modes. Applying the same wavelet thresholding to the u_k and v_k vectors does not result in enhanced de-noising, as shown by the following analysis. A matrix containing 20 oscillating signals with a mixture of sines and cosines, as described by Eq. (3.30) (see Fig. 3.22), each corrupted with different white noise of the same variance, was subjected to POD, wavelet thresholding, and both

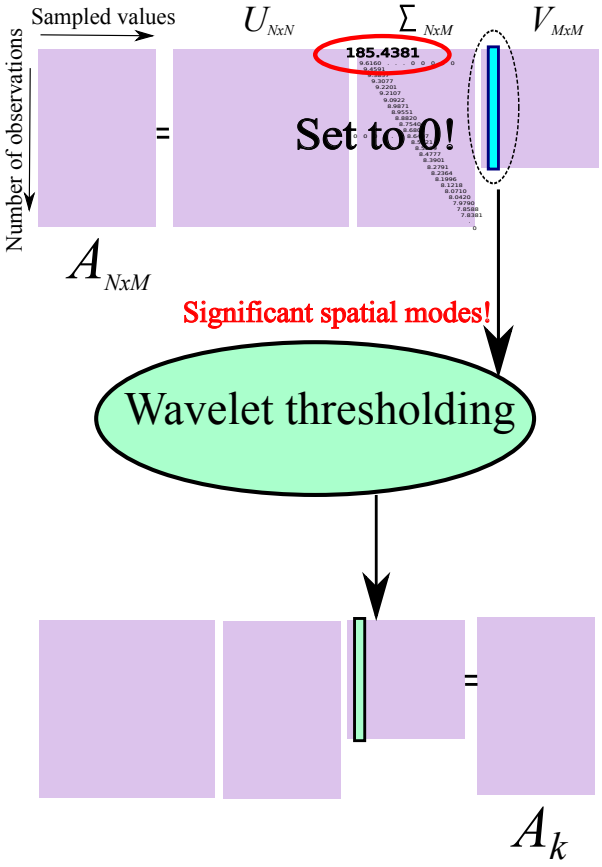
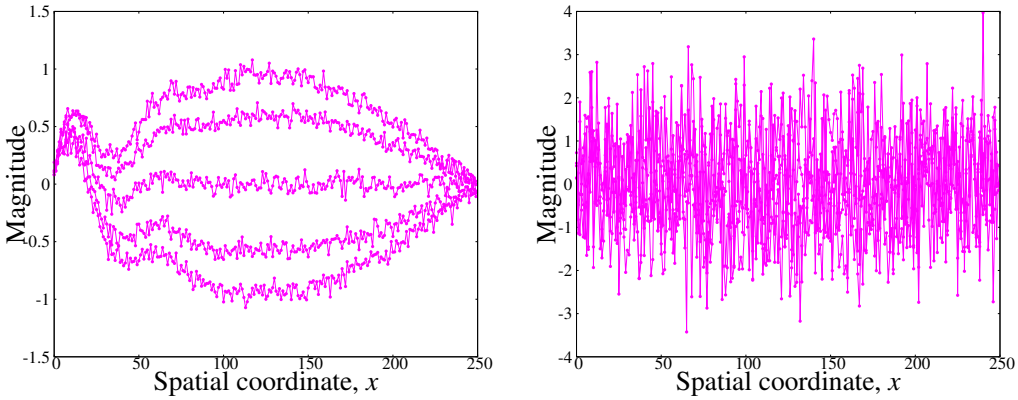


Figure 3.21: Graphical representation of WAVinPOD algorithm.



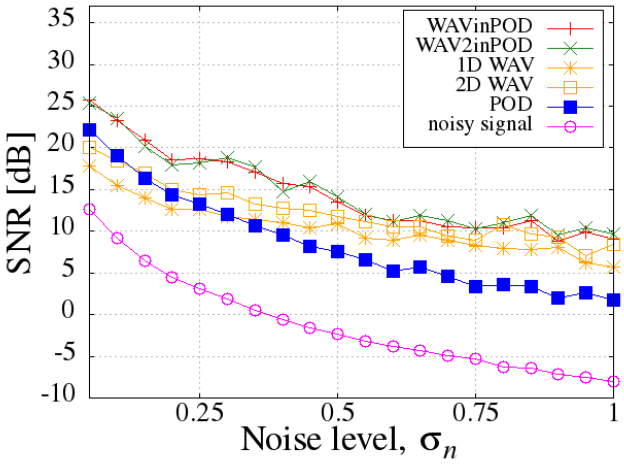
(a) Signals corrupted by white noise with σ_n=0.05.

(b) Resulting data with σ_n=1.

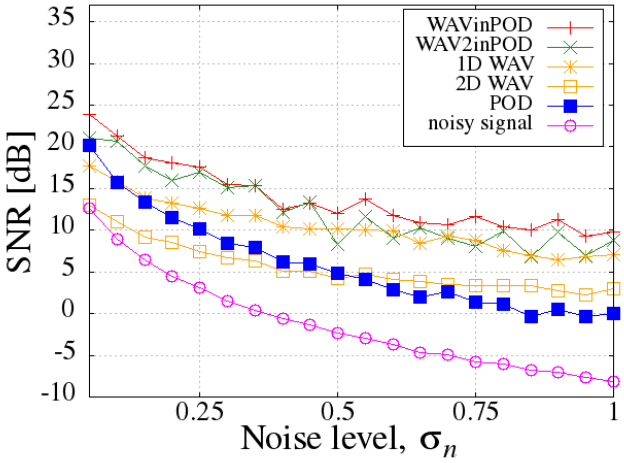
Figure 3.22: Noisy signals on which the filtering methods were tested.

versions of WAVinPOD. This data artificially mimics a set of observations, or profiles, that could be obtained from unsteady particle simulations. Figure 3.23(a) illustrates the performance of each method, measured by the SNR of the approximation data, for increasing noise level. The main observation is that WAVinPOD filters the signals much better then POD or wavelet thresholding alone. In addition, wavelet thresholding applied to dominant left and singular vectors (hereinafter referred to as WAV2inPOD) does not provide higher SNR gain than the original procedure. It is therefore preferable to use the less computationally expensive option, WAVinPOD. Moreover, when fewer observations of the signal are available, WAV2inPOD performs noticeably worse than WAVinPOD, as shown in Fig. 3.23(b). Similar conclusions were drawn for correlated noise. A more comprehensive comparison of noise reduction techniques applied to synthetic data is given in Chap. 4.

The reason why there is not much improvement in modifying both sets of modes lies in the fact that each transformation affects the orthogonality of the vectors. When wavelet thresholding is applied, the angle of the resulting modes is reasonably close to 90° (actually, between 88° and 91°). Therefore, using WAVinPOD preserves the orthogonality of the system quite well. However, de-noising both temporal and spatial modes may further weaken this property, resulting in unwanted aliases and eventually a lower SNR of the approximated data-set.



(a) Matrix containing $N = 20$ signals of length $M = 250$.



(b) Matrix containing $N = 10$ signals of length $M = 250$.

Figure 3.23: Performance of WAVinPOD and WAV2inPOD, compared with POD and wavelet thresholding for increasing noise variance. Considered matrix is a set of oscillating signals of length $M = 250$. For WAVinPOD and WAV2inPOD, 2 modes were subjected to soft wavelet thresholding with *db*3 filter and 6 levels of decomposition.

3.7 Other methods: Dynamic mode decomposition

In the course of our study, we came across a development in low-rank modelling and feature enhancement. However, due to certain properties of this technique, information extraction from contaminated data appeared to be challenging. As a result, the method has not been utilised for noise reduction in this thesis. A goal of future research would be to modify the procedure for data filtering purposes, and for that reason it is briefly discussed in this section.

In turbulence, proper orthogonal decomposition is used to extract deterministic functions associated with energetic structures in a flow and, together with the Galerkin method, derive a system of ordinary differential equations [81, 145]. A new procedure, referred to as dynamic mode decomposition (DMD), has been introduced by Schmid [146] to improve the derivation of reduced-models by extracting more relevant flow features. In contrast to POD, this method determines the energy of the fluctuations at particular frequencies, i.e. each DMD mode contains only a single frequency while POD modes contain several frequencies. It is therefore considered to be more suited for describing fluid-dynamical behaviours and transport processes [146, 147].

The dynamic decomposition method is based solely on snapshots of the flow, given by a matrix D_1^N defined as $D_1^N = (d_1, d_2, \ldots, d_N)$, where d_i stands for the i-th snapshot, the subscript 1 denotes the first member of the sequence and the superscript N indicates the last entry. One of the main assumptions in the procedure is that the sampling is carried out at a fixed time interval $t_{j+1} - t_j = \Delta t$. However, very recent work by Guéniat *et al.* [148] introduces a new algorithm for arbitrarily sampled systems. With O_l denoting the linear mapping that moves the domain from one time-step to the next, the following dependency holds

$$O_l D_1^{N-1} = D_2^N. \tag{3.84}$$

Performing singular value decomposition according to Eq. (3.12) on D_1^N

$$D_1^{N-1} = U\Sigma V^\dagger, \tag{3.85}$$

allows us to formulate Eq. (3.84) as

$$O_l U\Sigma V^\dagger = D_2^N, \tag{3.86}$$

or, using the Moore-Penrose pseudo-inverse, an alternative way

$$U^\dagger O_l U = U^\dagger D_2^N V\Sigma^{-1} \equiv \tilde{S}. \tag{3.87}$$

The eigenpairs (μ, y) of the $N-1 \times N-1$ companion matrix $\widetilde{S}$, on the right hand side

of Eq. (3.87), are computed to further establish DMD dynamic modes, $\phi^i_{\mathrm{DMD}} = Uy_i$, where y_i is the i-th eigenvector of $\widetilde{S}$, i.e. $\widetilde{S}y_i = \mu_i y_i$, and U is the set of right singular vectors of the snapshot sequence D_1^{N-1}. The eigenvalues and modes of DMD, therefore, represent approximate eigenvectors of the linear operator O_l projected onto the POD basis. This decomposition is able to extract coherent structures from a sequence of data fields, which indicates that it should work well on particle data.

In contrast to POD, which concentrates on a representation based on spatial orthogonality, DMD focuses on temporal orthogonality (frequency) [146]. In principle, it can allow us a more accurate and complete description of the flow features. Our preliminary analysis showed that indeed DMD produces a useful low-rank description of the data. However, when noise is present in the field, the eigenvectors of a companion matrix often lose their orthogonality, resulting in spurious artifacts. This observation was also confirmed by Duke *et al.* [149]. Consequently, in its original form, DMD is not a beneficial tool for noise reduction. However, there is potential to modify it in future and to improve its noise response and filtering properties by ensuring orthogonality preservation.

Chapter 4

Synthetic Data Analysis

This chapter presents the results of applying techniques described in Chapter 3 to de-noise synthetically generated data. These studies allow a straightforward comparison of the effectiveness of the various methods in signal processing. All data processing was performed using the commercial software package MATLAB R2014b (the MathWorks Inc., Natick, MA, 2014). Three objective measures, averaged signal-to-noise ratio (SNR), relative error in the L_2 matrix norm, δ_2, and error in the matrix Frobenius norm, δ_F, were applied to a range of accepted test problems. For each observation, SNR was calculated as a ratio of the summed squared magnitude of the true signal to that of the noise, and expressed in the logarithmic decibel scale. The relative errors in the L_2 and Frobenius norms are given as

$$\delta_2 = \frac{\|A_{\text{true}} - A_k\|_2}{\|A_{\text{true}}\|_2}, \quad \delta_F = \frac{\|A_{\text{true}} - A_k\|_F}{\|A_{\text{true}}\|_F}, \tag{4.1}$$

where A_k is a de-noised matrix obtained with one of the methods described in Chapter 3. The L_2 norm of matrix A is defined as the maximum singular value of A, $\|A\|_2 = s_1$, and the Frobenius norm is given by $\|A\|_F = \sqrt{\sum_{i=1}^{N} \left(\sum_{j=1}^{M} \|A_{i,j}\|^2 \right)}$. The error in L_2, i.e. δ_2, compares the energy content of A_k with the original matrix. The Frobenius norm takes all entries of the difference, $A_{\text{true}} - A_k$, as a single vector and measures its length. It then indicates which output has the shortest length of errors [150]. The approximation of one-dimensional signals was validated only with SNR.

Numerical modelling of fluid flow involves both steady-state, e.g. with constant

forcing, and time-varying simulations. The steady-state problems produce an ensemble prediction, e.g. mean velocity profile, while the latter focus on varying aspects of the studied phenomena. In order to mimic real data, we generated either a matrix with oscillating signals or a single array corrupted with white Gaussian noise (with zero mean and unit variance). In section 4.1, we analyse how SSA, rQRd, 1D wavelet thresholding, WienerChop filter, and EMD-based procedures perform when only one measurement is available. This study enables us to assess how efficiently a smooth ensemble solution can be extracted from a steady-state simulation. The section that follows is concerned with removing high frequencies from a collection of oscillating signals. The generated data are analysed with 2D extensions of the discussed methods, including the POD+ techniques. In Sec. 4.1 and 4.2, noise is considered to be white. However, particle data often contains disturbances which are not purely random. To address the issue of separating coloured noise from the *true* profiles, additional tests are performed in Sec. 4.3.

4.1 Signal processing

Donoho and Johnstone [123] analysed their de-noising procedure on four functions: Bumps, Blocks, HeaviSine, and Doppler, which are displayed in Fig. 4.1(a)-4.1(d). These signals are often used in signal processing because they imitate spatially-variable functions that can appear in different scientific fields. We have therefore incorporated these benchmark cases into our study. Using MATLAB's pseudo-random number generator, *randn()*, we added white noise to the data, keeping $\mathrm{STD}(f_{\mathrm{true}})/\sigma_n = 7$, and applied SSA, rQRd, 1D-wavelet thresholding (1D-WAV), EMD, and WienerChop to examine how efficiently they can recover the original signal, f_{true}. In other words, we wanted to see if an ideal reconstruction can be performed, while relying solely on noisy data, without any information about the true signal.

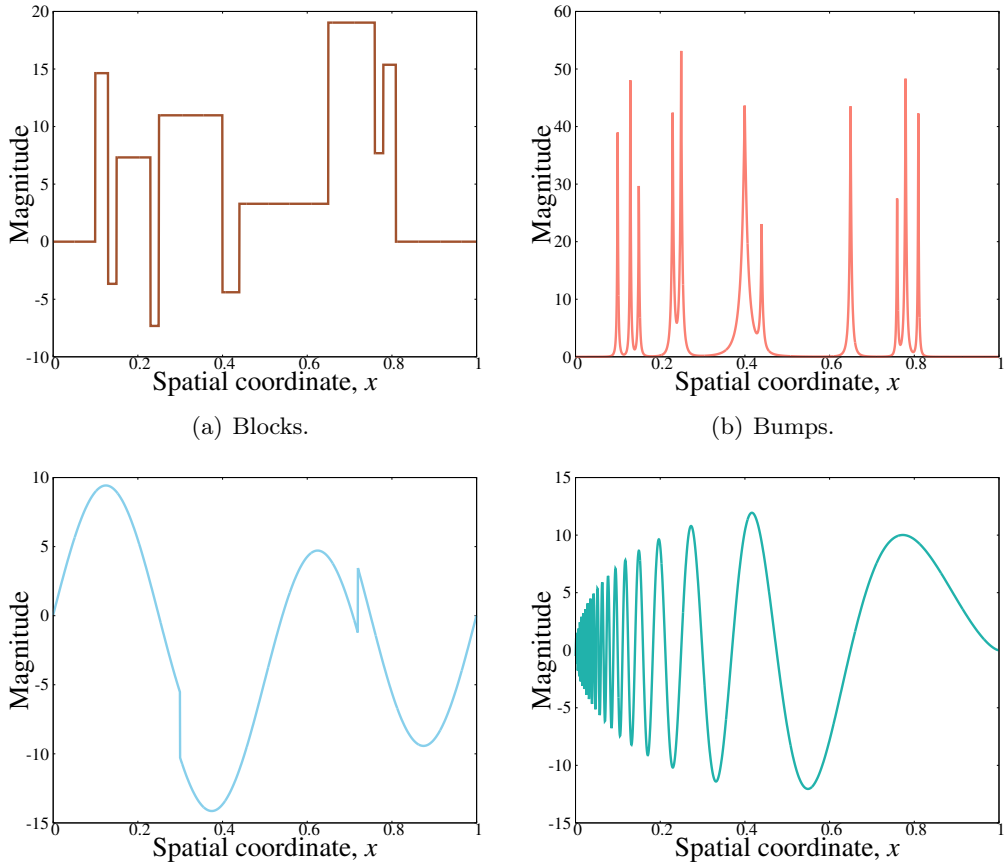
All the analysed signals were of length $M = 2048$. We did not apply urQRd because, according to the analysis in Sec. 3.3.2, the method is more efficient than the rQRd only for much longer data-sets. The oversampling parameter for rQRd procedure was kept constant, $p_k = 4$. A window for all the singular spectrum analysis was set to $L = \frac{M}{64} = 32$, as in Sec. 3.2.2 small values of L were recommended for recovering more

complex trends. The number of EOFs, k, for each case was determined prior to truncation of singular values. De-noising with EMD was performed by utilising EMD-IT and EMD-IIT, both discussed in Sec. 3.5.2, with a fixed number of sifting operations, $n = 7$, and using the hard thresholding approach. The methods are very similar; EMD-IT is the same as EMD-IIT when the number of iterations is set to one. Following Donoho and Johnstone [123], wavelet thresholding was performed with the Symlets filter, $sym8$, and 5 levels of decomposition. For estimating the coefficients of WienerChop, we utilised a wavelet transform with the same parameters. The second filter for WT_2 in the WienerChop operation was set to have half of the vanishing moments of the WT_1, i.e. $sym4$ was taken, and the resolution was one level higher.

Figure 4.2(a) depicts the results obtained with the five de-noising techniques applied to the four functions. All the parameters, including window length for SSA and oversampling parameter for rQRd, were kept constant. The overall dimensionless gain in signal-to-noise ratio, defined as

$$\text{Gain} = \frac{\text{SNR}_{\text{approximation}} - \text{SNR}_{\text{noisy}}}{\text{SNR}_{\text{noisy}}}, \tag{4.2}$$

where $\text{SNR}_{\text{noisy}}$ and $\text{SNR}_{\text{approximation}}$ are the SNR values of the original corrupted signal and de-noised data, respectively; the absolute value of the gain was considered when $\text{SNR}_{\text{noisy}} < 0$. The average processing times are presented relative to the computational cost of the rQRd method, which in most of the cases appeared to be the fastest. Results show that WienerChop extracted signals closest to the original functions, f_{true}, from Fig. 4.1. Wavelet thresholding performed similarly to EMD-IT. For functions f_{true} with smooth transitions, e.g. Doppler, higher SNRs were obtained with EMD-IIT, in this case with $it = 20$ iterations; the more repetitions, the better the noise removal. However, the improvement in the reconstructed data quality was not good enough to justify the substantially increased processing time. In general, for signals with sharp edges or peaks, EMD-IT was preferred, as averaging over many realisations resulted in the over-smoothing of peaks. In addition, EMD-IIT showed a tendency to recover small oscillations for hard thresholding, caused by permutations, if an insufficient number of iterations was performed. In contrast, Kopsinis and McLaughlin [30] showed that the method can be beneficial for sharp functions when the noise level is very high.



(a) Blocks.

(b) Bumps.

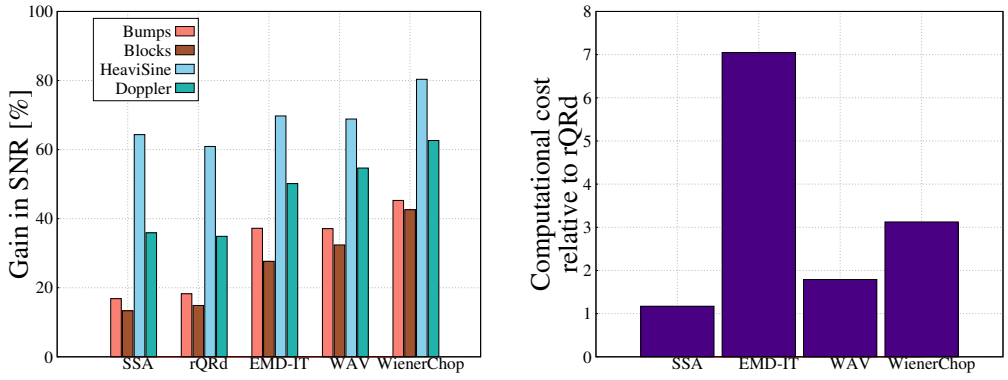
(c) HeaviSine.

(d) Doppler.

Figure 4.1: Four spatially variable test functions; $M = 2048$.

Random QR de-noising with SSA provided the poorest reconstruction, but its average processing time was the smallest (see Fig. 4.2(b)). It should be stressed that for SSA the number of significant EOFs was pre-defined. In a real situation, eigentriple grouping should be performed based on certain analysis, e.g. plotting an LEV diagram, which would then contribute to the processing time. Empirical mode decomposition, even not iterated, was the slowest. Utilising WienerChop also appeared to be computationally more expensive. However, WienerChop offered a good trade-off between the quality of data filtering and the time it took to perform the operation; the reconstructed functions with WienerChop are plotted against noisy signals in Fig. 4.3. From all the considered methods, SSA and rQRd appeared to be the weakest.

Figures 4.4(a) and 4.4(b) depict the performance of each method applied to the HeaviSine function and Bumps, respectively, for increasing noise level. All the parame-

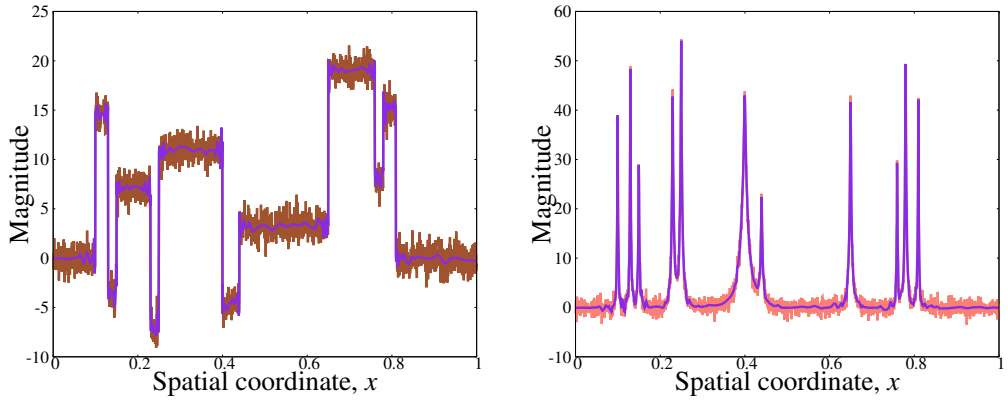


(a) Gain in SNR. (b) Average processing time.

Figure 4.2: Results of reconstructing corrupted signals with $\mathrm{STD}(f_{\mathrm{true}})/\sigma_n = 7$ using SSA, rQRd, EMD-IT, wavelet (hard) thresholding, and WienerChop; window length for SSA was $L = 32$ and oversampling parameter for rQRd $p_k = 4$; filter $sym8$ was used for 1D-WAV with 5 resolutions, and additional $sym4$ with 6 levels of decomposition for WienerChop.

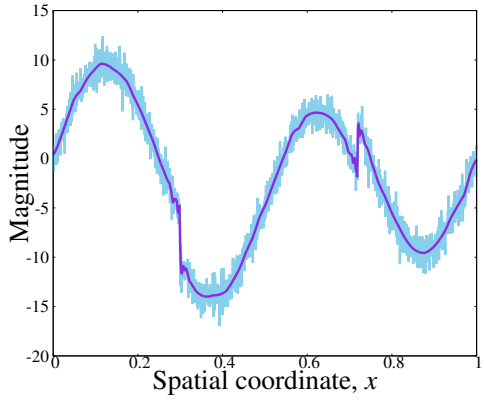
ters were kept the same as in the previous study, apart from the noise standard deviation changing from $\sigma_n = 0.05$ to $\sigma_n = 2$. A main observation is that for a high noise variance, there is a small discrepancy in the SNR of the approximated signals between the de-noising techniques, particularly for a smoother f_{true} (i.e. HeaviSine). When applied to HeaviSine, WienerChop provided slightly enhanced noise removal up to $\sigma_n = 1$. For higher σ_n, all the methods obtained comparable results. Iterated interval EMD performed the best for the more severely corrupted signals, but on average it was the most computationally expensive, as presented in Fig. 4.4(c). Different conclusions were drawn for the signal Bumps, which contained sharp transitions and peaks. Filtering it with WienerChop was the most effective in terms of recovered SNR, while EMD-IIT with 20 iterations provided the poorest approximations partially due to over-smoothing. One-dimensional wavelet thresholding and EMD-IT gave very similar results, but 1D-WAV was the fastest.

In conclusion, the results showed that if data quality is of importance, then the WienerChop filter or EMD-based de-noising should be applied; EMD-IT is recommended for trends with sudden transitions, edges and peaks, while EMD-IIT performs better for smooth, less complex trends. However, as will be shown in the following sections, in the case of less complex shapes, for which it is easier to define the dominant EOFs, the SSA algorithm is capable of outperforming the other procedures. If the com-

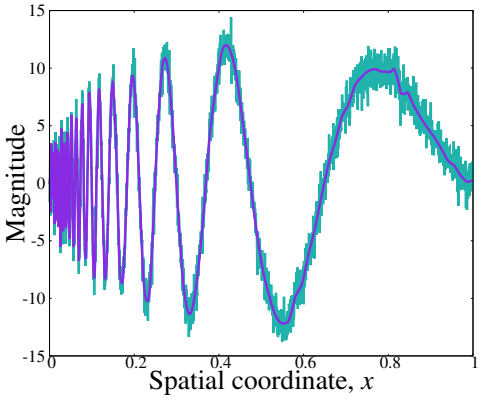


(a) Blocks.

(b) Bumps.



(c) HeaviSine.



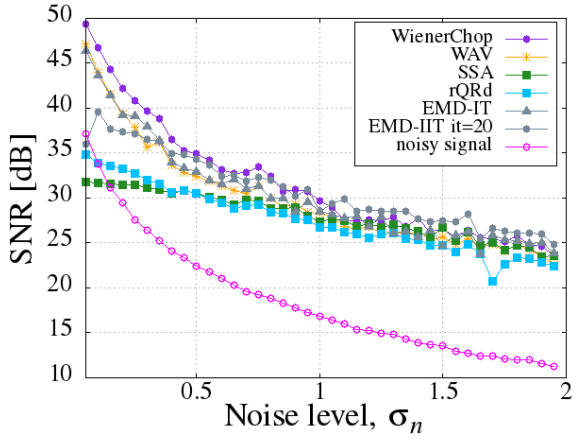
(d) Doppler.

Figure 4.3: Noisy functions and their reconstruction (purple solid line) obtained with the Wiener-Chop filter.

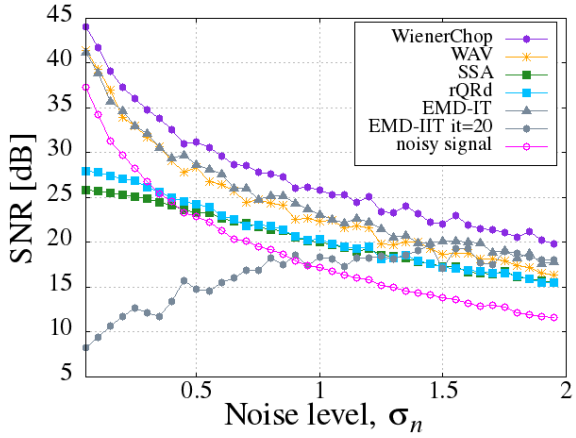
putational cost of noise-reduction has to be the lowest, then one-dimensional wavelet thresholding and rQRd are the preferred de-noising techniques.

4.2 De-noising of data ensemble

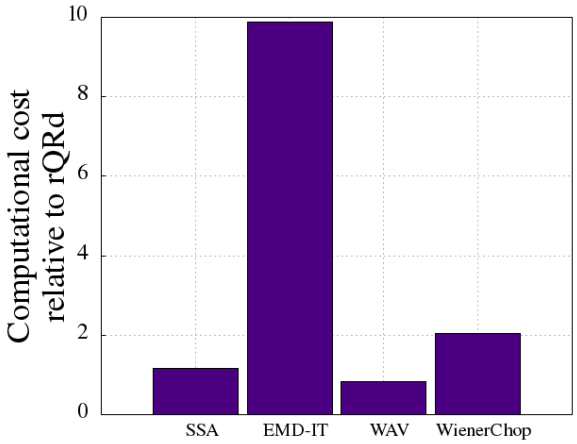
All the methods discussed in this thesis can be used to filter sets of data. They can be applied directly, treating a row (or a column) of a matrix as a one-dimensional signal, or their extensions to two-dimensional objects can be utilised. If only a small number of long data arrays is available ($N \ll M$), it is more beneficial to de-noise each signal separately, as there are not enough observations to look for correlations between them. In the case of large matrices, applying two-dimensional modifications of the algorithms is noticeably less expensive, and allows us to reduce the data's dimensionality. The



(a) Obtained SNRs for HeaviSine.



(b) Performance for Bumps.



(c) Average computational time.

Figure 4.4: Comparison of all the methods applied to the HeaviSine and Bumps functions with increasing noise level; in SSA $k = 2$ for HeaviSine and $k = 13$ for Bumps.

only exception is 2D-SSA, described in Sec. 3.2.3, because its computational cost is severe [113]. Our study showed that 2D-SSA can take over 100 times longer than POD. Consequently, only results obtained with SSA and POD+SSA were employed in this analysis.

Two different oscillating signals have been investigated. The first initially smooth $N \times M$ data matrix, previously described in Sec. 3.1.3, was generated using Eq. (3.30). The second set of signals, $A^{(2)}_{\text{true}}$, was constructed with the following MATLAB code:

```
s=0:0.01:(M*0.01-0.01);
y(1,s/0.01)=3*sin(s)+sin(0.5*s+40)+2*sin(3*s-60);
    for t=1:N
        for x=1:M
            A_true(t,x)=y(1,x)*cos(pi*t/N) +
            sin(0.5*t+40)*cos(pi*x/M)+(0.01*t+2+sin(t)).^2;
        end
    end
```

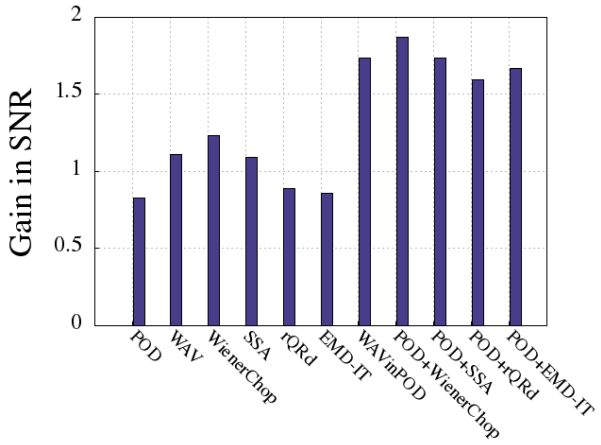
In both cases the signals were of length $M = 1024$ and initially only $N = 20$ observations were used. The number of time-samples required to de-noise the signal is of significance as it defines how long the simulation has to run to provide enough data. The ranks of smooth matrices, equal to $k_1 = 2$ and $k_2 = 3$ for $A^{(1)}_{\text{true}}$ and $A^{(2)}_{\text{true}}$, respectively, were increased by corrupting each signal with added white noise using MATLAB's pseudo-random number generator. The resulting noisy data-sets, A_1 and A_2, were full-rank because of the partial de-correlation of the disturbed data points. In a real situation we will not know the original signal, but only the corrupted measurements, and often we are not sure of their nature. For the analysis with SVD-based methods (POD, SSA and POD+) we have to rely solely on examination of the eigenspectra in order to establish an adequate number of k for the approximation. The previously described criteria were utilised here to find the number of significant modes.

At first the original matrix $A^{(1)}_{\text{true}}$ was corrupted with noise of standard deviation $\sigma_n = 0.1$, producing A_1 with SNR ≈ 12.47 dB. After applying SVD to the whole matrix, the criteria for determining k were utilised. The tests described in Sec. 3.1.3 managed to recover an accurate number of significant modes. In the case of SSA analysis, SVD

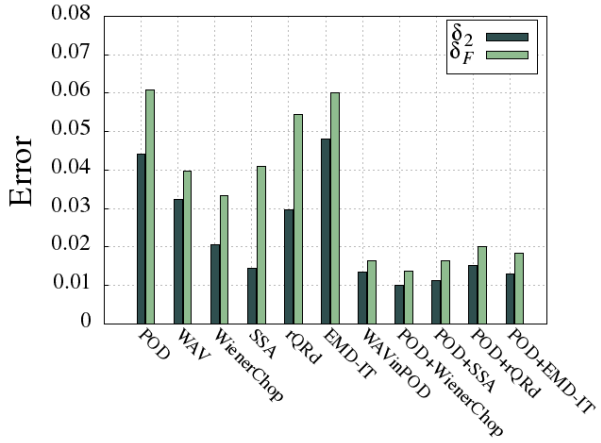
was applied to Hankel matrices built from each row of A_1; the same number $k = 2$ was retained for all the signals. The time required to determine which EOFs were significant was not included in the final estimation of the computational cost of each SVD-based method. In other words, the processing times presented here depict how long it took to perform the whole algorithm with a pre-defined k.

The main drawback of using wavelet transforms and multiresolution analysis is the number of parameters that need to be considered *a priori*, e.g. mother wavelet, number of vanishing moments, and levels of decomposition. The choice of an appropriate model is often problematic and may lead to data misinterpretation if any deviations from it appear. In our study, we know that the signals to be recovered do not have sharp transitions, so a higher number of vanishing moments is recommended, and wavelet shrinkage should be used instead of hard thresholding. The filters associated with Daubechies wavelet, *db*6, appeared here to give a good balance between the processing time and the SNR. When applying wavelet thresholding to spectral modes obtained with SVD, we do not have to be as careful with the choice of wavelet basis as when the transform is directly applied to noisy data. This is because WAVinPOD filters only the spectral components of the signal that have already been partially de-noised. For a clear comparison, the same wavelet transforms were used for wavelet thresholding applied to raw data (WAV), WAVinPOD and for determining WienerChop's coefficients (WT_1); for the second transform in the WienerChop procedure we again utilised a filter with half of the vanishing moments in WT_1, *db*3, and one level higher resolution. Parameters for the other methods were kept the same as described in Sec. 4.1.

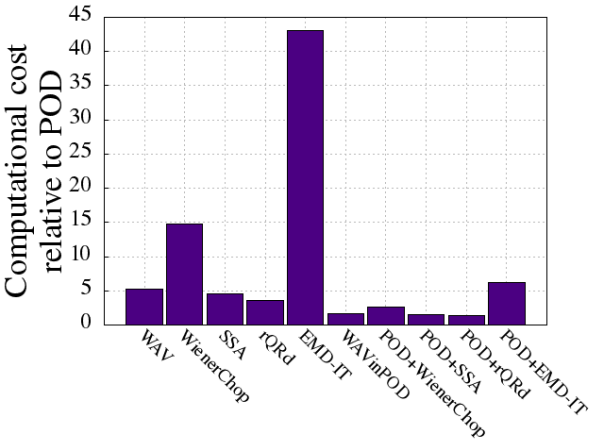
Figures 4.5(a) and Fig. 4.5(b) compare gains in signal-to-noise ratio and errors in the L_2 and Frobenius norms for each reconstruction; the time it took to perform every operation is summarised in Fig. 4.5(c) in relation to the computational cost of performing the fastest method, POD. As expected, POD did not recover the signals well for such a small number of observations; the reconstructed matrix had SNR = 22.81 dB (82.93% higher than the original), $\delta_2 = 0.0443$ and $\delta_F = 0.0610$. In contrast to previous results, the SSA method performed better than EMD-IT as the trend was less complex. Clearly, the POD+ techniques were the most successful, with POD+WienerChop filter providing the best approximation with over 187% higher SNR ($\approx$ 35.80 dB) than the



(a) Gain in SNR.



(b) Errors in the L_2 and Frobenius norms.



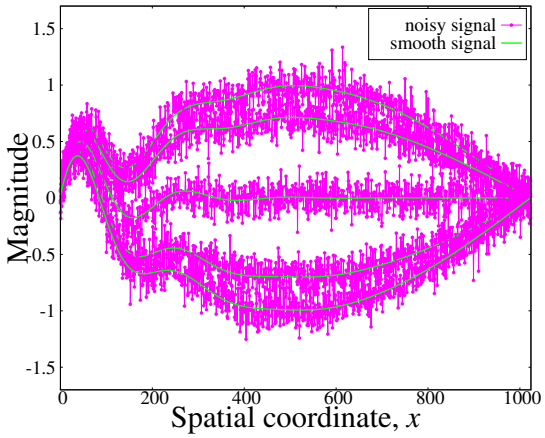
(c) Computational time.

Figure 4.5: Comparison of de-noising A_1 with $N = 20$, $\text{SNR}_{\text{noisy}} = 12.47$ dB, $\delta_2 = 0.0508$ and $\delta_F = 0.1945$.

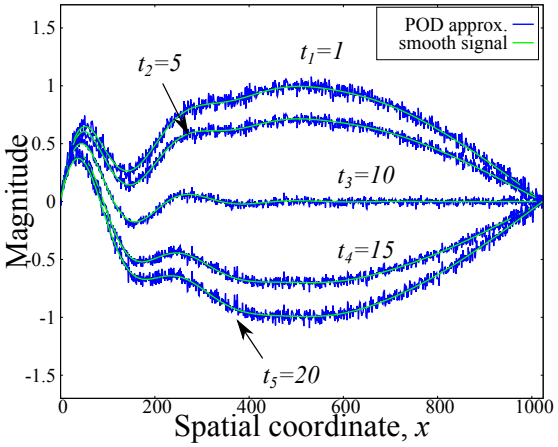
original noisy A_1 (12.47 dB). Moreover, combinations of POD and other methods were the least expensive to perform, with the exception of POD+EMD-IT due to the empirical mode decomposition being the most computationally intensive. Figures 4.6(b) and 4.6(c) depict the worst and the best approximations recovered from the same noisy data-set, A_1 (plotted in Fig. 4.6(a)), with POD and POD+WienerChop, respectively. Utilising the two-dimensional wavelet thresholding (2D-WAV) would in this case be $4\times$ faster than 1D-WAV. However, as mentioned before, for a small N the recovered SNR would be low; only 19.16 db (53.63% gain), while one-dimensional thresholding increased the SNR by 110.89%.

Another analysis was performed in order to determine how many time-samples of the signal with the same noise level, SNR = 12.47 dB, would be required for POD to achieve a comparable de-noising performance as POD+ methods did for $N = 20$. It was found that, around 400 samples (20 times more than for POD+) allowed POD to reach SNR = 34.38 dB, as illustrated in Fig. 4.7(a). De-noising with EMD-IT was confirmed to be expensive; performing this method on 400 signals took about 17.5 s, $57\times$ more than for POD+EMD-IT. The computational cost of the other methods is depicted in Fig. 4.7(b). The two-dimensional de-noising with wavelet shrinkage or WienerChop recovered better results than their one-dimensional counterparts in terms of gain in SNR and computational cost (see Fig. 4.7(a) and Fig. 4.7(b)). The extensions of POD again outperformed the other methods, reaching a higher SNR in less time. It was also shown that when a large number of observations is available, applying filtering with, e.g., wavelet thresholding to dominant left and right singular vectors may be beneficial. In such cases the additional transformation does not strongly affect the orthogonality of modes. For $N = 400$, WAV2inPOD recovered the best approximation with an averaged SNR = 42.45 dB. However, as our goal is to decrease the number of measurements required for data extraction, POD+ methods applied only to spectral modes are still the preferred de-noising approach.

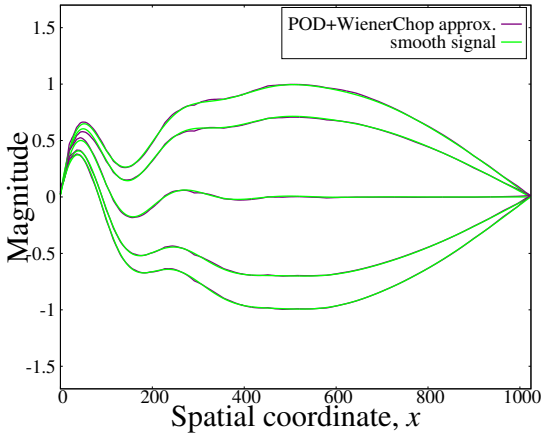
It is important to mention that the two-dimensional wavelet thresholding and WienerChop alone do not reduce the dimensionality of a matrix as well as POD and POD+. Methods based on SVD produce approximations with rank equal to the number of dominant modes. For the case of $N = 400$ observations, 2D-wavelet analysis produced a



(a) Noisy and the original smooth data.

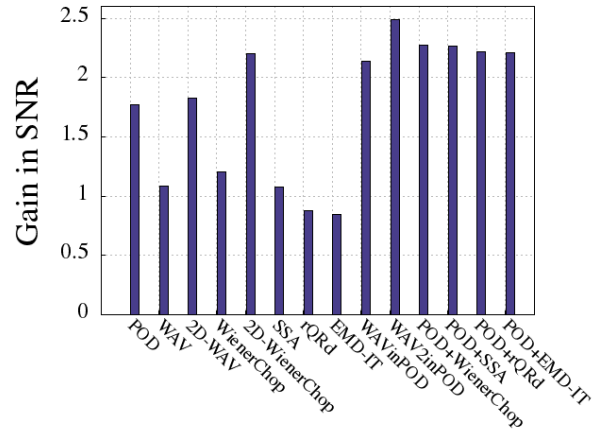


(b) POD plotted against the original smooth signals.

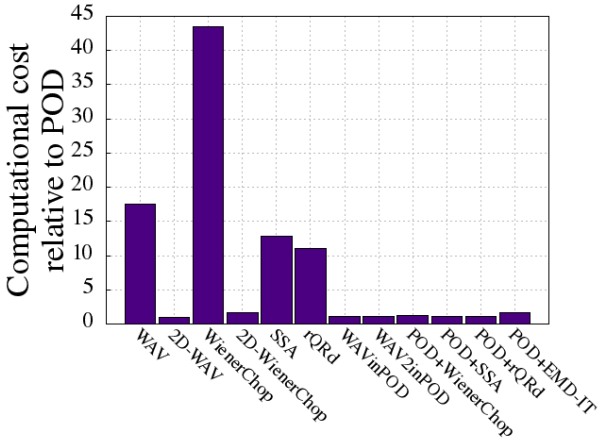


(c) POD+WienerChop approximation and the $A_k^{(1)}$.

Figure 4.6: Signals recovered with POD and POD+WienerChop. Note the improved de-noising quality for POD combined with WienerChop filter for the same number of observations $N = 20$.



(a) Gain in SNR for signals recovered from A_1.



(b) Computational time.

Figure 4.7: Comparison of de-noising A_1 with $N = 400$, $\text{SNR}_{\text{noisy}} = 12.47$ dB, $\delta_2 = 0.0160$ and $\delta_F = 0.1944$.

matrix of rank = 27 (for 1D-WAV the rank was 122, and 2D-WienerChop rank = 343), whereas POD and POD+ resulted in a matrix of rank = $k = 2$. Singular value decomposition gives the best rank reduction for all norms that are invariant under rotation [151]. The low rank of the matrix is important if compression of the data is of interest. Transferring the original matrix with rank = 400 requires sending $400 \times 1024 = 409600$ samples of information. When the rank is lowered to 27, the matrix can be represented by 3 components that together contain $400 \times 27 + 27 \times 27 + 1024 \times 27 = 39177$ elements, which is over 10 times lower than the original set. Having an approximation of rank = 2 further reduces the size of the matrices resulting in only 2852 samples of the data (about 143 times less than the original!). It should be stressed that using POD+

methods allows not only the successful extraction of information from disturbed data, but also reduces the computational cost of further processing or storage.

Considering that the data in A_1 is periodically oscillating, applying a statistical mean to the original observables in this case would fail to provide good results. In order to perform de-noising through averaging, copies of the matrix with its added noise need to be generated, which is equivalent to re-running a particle simulation. For a similar noise reduction as obtained with POD+ techniques, averaging over 150 of the corrupted copies was required. In other words, to de-noise a given signal with the mean method, a simulation would have to be re-run 150 times (or left to oscillate for 150 periods) in order to average over 150 sets of data. This could be a major computational bottleneck. Moreover, POD and POD+ are able to efficiently extract a smooth unsteady signal without any prior knowledge of its nature. They can perform comparably well even when a signal is oscillating in a non-periodic manner, with changing frequency.

Additional analysis was performed on matrix $A_k^{(2)}$, also corrupted with white noise to produce A_2 with an SNR ≈ 12.47 dB. After applying SVD, the criteria for determining k were utilised. The tests described in Sec. 3.1.3 managed to recover an accurate number of significant modes. The first, second and third eigenvalue, $\lambda_{k=1,\dots,3} = s_k^2$, were the most energetic, containing together 97.22% of the total variance ($E_\lambda^{(1)} = 87.42\%$, $E_\lambda^{(2)} = 9.06\%$ and $E_\lambda^{(3)} = 0.74\%$). In practice, it is common to select levels of energy threshold between 70% to 95% [92]. It is evident that the first three modes retained most of the significant information; the fourth eigenvalue corresponded to only 0.21% of the total energy. When the LEV diagram of $\log_{10}(\lambda_k)$ was plotted, the first three eigenvalues also appeared to be fast-decaying (see Fig. 4.8), while the other points formed almost a straight line. Corresponding eigenvectors were smoother than other temporal modes that clearly contained high-frequency oscillations. The choice of k was further confirmed by applying SVHT, which suggested a threshold of $th = 53.7104$ for known noise, and similarly a $th = 55.1929$ for unknown variance and $\beta = \frac{N}{M} = \frac{20}{1024} \approx 0.02$; the fourth singular value was smaller than the threshold, $s_4 = 41.7528 < th$ resulting in only three modes being recovered. We also propose using the σ_n estimated from the finest wavelet coefficients (see Eq. (3.61)) in cases where noise variance is not known *a priori*. In this approach, a slightly higher threshold was obtained, $th = 60.9337$, but

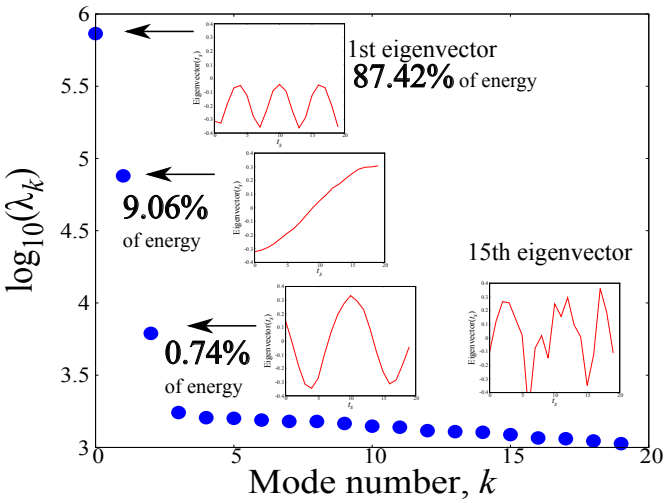
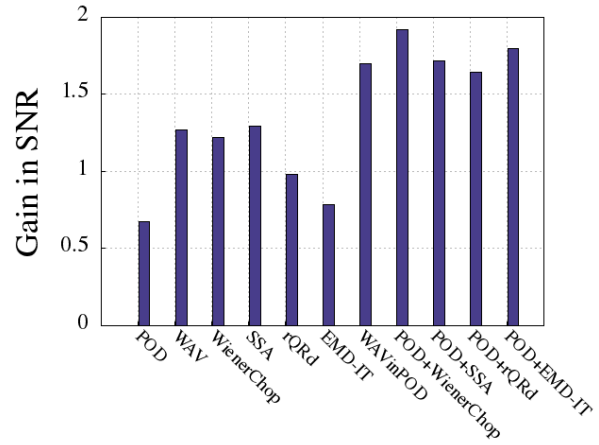


Figure 4.8: Investigating criteria for the choice of significant k; LEV diagram with the 1st, 2nd, 3rd and 15th eigenvectors shown and the relative energy of the first three eigenvalues.

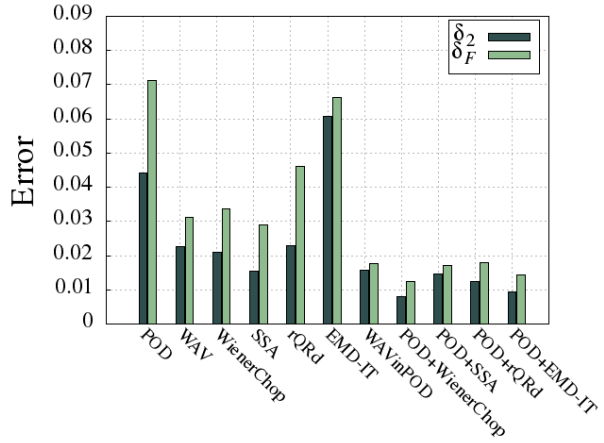
sufficient to retain the same number $k = 3$ as $s_3 = 78.4646$. As stressed in Chapter 3, it is important to consider all the criteria because any single test on its own may not provide enough information to capture the significant phenomena. To achieve higher confidence in selecting an appropriate k it is best to analyse the results using at least two tests. For SSA with $L = 32$, only the first eigenvalue was kept. Figure 4.9 summarises the values of gain in SNR and errors in the L_2 and Frobenius norms for each reconstruction, along with the time it took to perform each calculation. The best results were again produced with POD+ methods, with POD+rQRd being the fastest and POD+WienerChop, POD+EMD-IT and POD+SSA providing comparably high averaged SNRs. Initial smooth and noisy functions are shown in Fig. 4.10(a). The enhanced signals obtained with the POD technique are plotted every fifth observation in Fig. 4.10(b), and the best approximation constructed by POD+WienerChop is illustrated in Fig. 4.10(c).

4.3 Removing spatially correlated noise

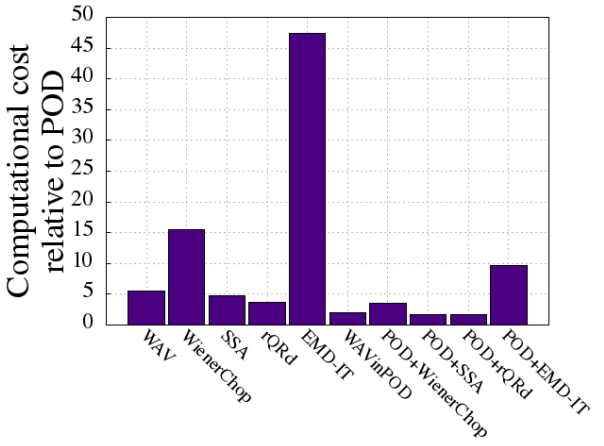
The main advantage of SVD-based methods is their ability to capture time-space correlations in the signal, making them an effective tool for processing data corrupted with uncorrelated random noise. However, particle-based simulations often produce results disturbed by temporally or spatially dependent fluctuations. For example, thermostats



(a) Gain in SNR.

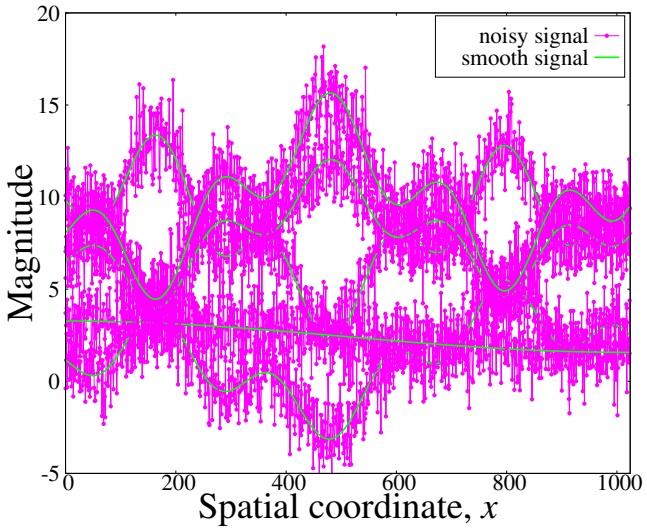


(b) Errors in the L_2 and Frobenius norms.

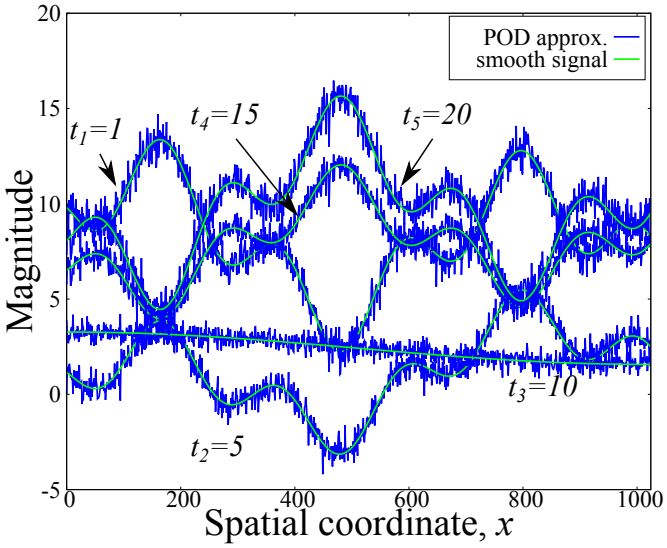


(c) Computational time.

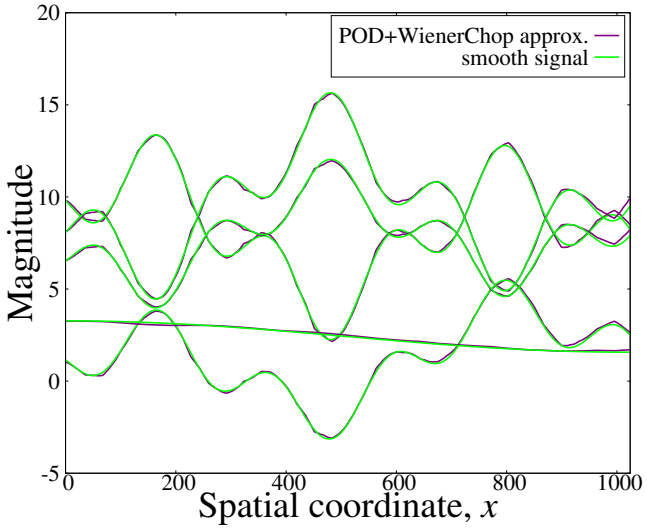
Figure 4.9: Comparison of de-noising A_2 with $N = 20$, $\text{SNR}_{\text{noisy}} = 12.47$ dB, $\delta_2 = 0.0489$ and $\delta_F = 0.1840$.



(a) Noisy signals.



(b) POD approximation.



(c) POD+WienerChop.

Figure 4.10: Signals recovered from A_2 with POD and POD+WienerChop for $N = 20$.

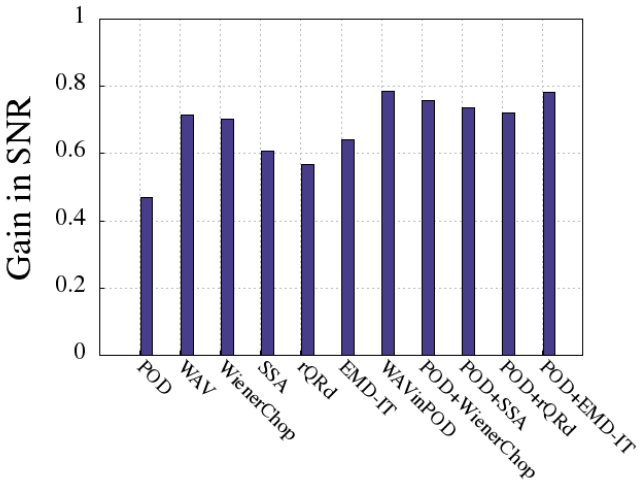
used in canonical ensemble are a potential source of correlated (or coloured) additive noise [25]. Proper orthogonal decomposition, based on the L_2 vector norm, suffers in the presence of such corruptions and can skew the features of the data by weighting them too heavily.

To address the issue of efficiency in separating coloured noise from the *true* profile, additional tests have been performed. The synthetic data, $A_k^{(1)}$, described previously, was corrupted with pink (flicker) noise, also referred to as $1/f$ noise as its power spectral density (i.e. energy or power per Hz) is inversely proportional to the frequency (here f) of the signal. The noisy matrix, A_1, with the same dimensions ($N = 20$, $M = 1024$) and SNR = 12.47 dB was treated with all the filtering techniques. For WienerChop analysis, the level-dependent noise estimation, introduced by Johnstone and Silverman [152] for signals corrupted with coloured noise, was used. Soft thresholding was applied for both EMD-IT and the wavelet-based methods. All the parameters were kept the same as in the previous study.

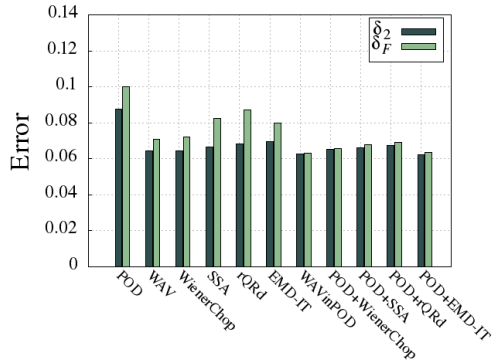
Most of the criteria for determining the significant modes retained an adequate number $k = 2$. However, a higher $k = 3$ was suggested by SVHT as the estimated thresholds were too low even when the noise level was determined from wavelet coefficients. This is due to the fact that the optimal threshold developed by Gavish and Donoho [94] was designed only for white noise. It was observed that, for the case of coloured additive noise, if the square root of the standard deviation calculated from Eq. (3.61) is inserted into the formula for SVHT with known variance, the same number k is retained as the value established with other tests. In addition, if the median of the singular values is replaced with the median of eigenvalues, more appropriate thresholds are also computed. This observation is further confirmed with the studies performed on simulation data in the next Chapter. Additional investigation and improvement of the optimal general threshold for de-noising is part of future work discussed in Chapter 6.

The results showed that the de-noising efficiency for each method was lower than in the case for white noise. This is due to the fact that statistically dependent variations have a pattern with relatively high energy which is difficult to separate from the original smooth shape of the signal; in other words, the error in the spectral norm is higher than in the case of white noise. Methods based on SVD do not perform well for

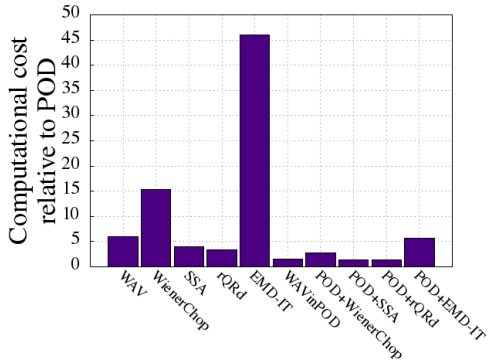
a small number of measurements, N, as they retain all the energetic principal components, including some unwanted structures. Applying additional de-noising in POD+ procedures lowers the errors in both norms, particularly δ_F as denoted in Fig. 4.11(b).



(a) Gain in SNR.



(b) Errors in the L_2 and Frobenius norms.



(c) Computational time.

Figure 4.11: Comparison of de-noising efficiency in processing $A_k^{(1)}$ corrupted with pink noise; $N = 20$, $\text{SNR}_{\text{noisy}} = 12.47$ dB, $\delta_2 = 0.0883$ and $\delta_F = 0.1961$.

Figure 4.11(a) shows that utilising wavelet thresholding, including WAVinPOD and POD+WienerChop, and EMD-IT are the most successful. A wavelet transform does not measure the energy content but the correlation between the wavelet and the signal, making it a more flexible, adaptive technique. If the chosen wavelet basis is close to the original smooth data, and adequate thresholds are established, the filtering performance can be significantly improved. In contrast, the high versus low frequency discrimination in EMD-IT applies only locally, which can potentially enhance detection

of unwanted correlations, and does not correspond to any pre-determined basis. In the case of coloured fluctuations, it is then expected that some information about the noise might be necessary in order to improve the filtering performance. Further study on reducing unwanted correlations from the data should be part of any future work.

Chapter 5

Removing Noise from Simulation Results

Grinberg [25] showed how non-stationary MD and DPD data can be successfully de-noised using only POD with time windows. In this chapter, we report the results of applying the filtering methods previously discussed in this thesis to velocity measurements and density profiles from stationary and time-dependent simulations performed with either MD, DPD, or DSMC. The aim is to investigate the benefits of applying de-noising techniques to simulation results, and compare the performance of POD+ methods with the other procedures in time-dependent modelling.

Application of SSA, rQRd, EMD-IT, 1D-WAV and WienerChop to an ensemble mean of data obtained from steady fluid channel flows is presented in Sec. 5.1. The results of utilising WPOD, POD+ methods, 2D extensions of wavelet thresholding, and WienerChop to statistically non-stationary simulations are discussed in Sec. 5.2. In all the problems considered, the spatial distribution of the observables was calculated using the binning method (see Allen and Tildesley [34]). In this approach, the system's domain is partitioned into a number of cells, or bins, and the averaged velocity and number of particles in each bin is computed based on their positions.

The molecular dynamics simulations were carried out using the open-source mdFOAM solver, built in OpenFOAM. The model fluids were either liquid argon in a krypton channel, or water flowing between two silicon walls. All of the parameters from the MD simulations are presented in reduced units; the reference values are linked

to the Lennard-Jones (L-J) potentials. For water, the quantities for length, energy and mass, respectively, were: $\sigma_{r,H_2O} = 3.1589 \cdot 10^{-10}$ m, $\epsilon_{r,H_2O} = 1.2868 \cdot 10^{-21}$ J, $m_{r,H_2O} = 2.987 \cdot 10^{-26}$ kg, and for argon: $\sigma_{r,Ar} = 3.405 \cdot 10^{-10}$ m, $\epsilon_{r,Ar} = 1.6568 \cdot 10^{-21}$ J, $m_{r,Ar} = 6.6904 \cdot 10^{-26}$ kg. We used the rigid TIP4P/2005 water model as described by Abascal and Vega [153]. This is a four-site model, consisting of a L-J interaction potential at the oxygen atom site, positive Coulomb charges at the two hydrogen sites, and a negative charge at a an additional massless site, located a small distance away from the oxygen. For the DPD modelling, this was performed using DL MESO[1] and the dimensionless parameters were converted to physical units with the reference values of the cut-off radius, $r_{cut_r} = 6.46 \cdot 10^{-10}$ m. This was calculated with one DPD particle representing 3 water molecules, based on the relation described by Ghoufi *et al.* [45]. The DPD energy was $k_B T_r = 4.114 \cdot 10^{-21}$ J (where k_B is Boltzmann's constant and $T_r = 298$ K), and the mass of one water molecule was $m_r = 2.987 \cdot 10^{-26}$ kg. Parameters for DPD that enforce proper water compressibility for the system were taken from Groot and Warren [154]. In the case of the direct simulation Monte Carlo modelling, the gas was considered to be a hard sphere or variable hard sphere argon flowing in a periodic domain with a time varying gravitational acceleration. All simulations were performed with the dsmcFOAM solver in OpenFOAM or, for simpler calculations, an in-house DSMC code [155]. Results obtained with DSMC are presented in SI units.

5.1 Analysis of steady-state nanofluid flows

For stationary data, simple averaging is the most natural approach to obtain the desired solution. This averaging can be a poor choice, however, when resolution is low or a high level of statistical noise is present. To overcome this limitation, large samples and long averaging periods are required, which can result in bottlenecks, particularly if there is significant intra-scale communication or computationally expensive calculations. Therefore, there is a need for a systematic approach which is able to provide smooth data that can be analysed faster than basic averaging is able to achieve. Employing de-noising techniques can result in a significant reduction of the computational load, particularly in modelling non-stationary flows, as discussed in Sec. 5.2. However, improvements in

[1] `www.ccp5.ac.uk/software`

feature extraction have also been observed when more sophisticated filtering methods are applied to the averaged stationary fields. Provided that the coherent structures are present in the ensemble, i.e. the system is fully developed, applying additional noise removal to the statistical mean of the observables can enhance the quality of the numerical solutions.

In the following analysis we show how utilising the methods discussed in Chap. 3 can be beneficial in steady-state modelling. The first simulation considered was Poiseuille flow of liquid argon in a krypton nanochannel, which was modelled with MD. A periodic domain was specified for the system with dimensions: $25 \times 50 \times 10$ ($x \times y \times z$), with the thickness of the wall set to 5 in reduced units. A simple reflective wall boundary model was constructed by defining a solid structure with *frozen* molecules, not interacting with each other, but having fluid-solid interactions. The L-J parameters describing the argon-krypton interaction were: $\sigma_{Ar-Kr} = 1.02\sigma_{r,Ar}$, and $\epsilon_{Ar-Kr} = 1.18\epsilon_{r,Ar}$, taken from Sofos *et al.* [156] and Gotoh [157]. For all the MD simulations presented in this chapter, the motion of fluid particles was weakly coupled to a thermal reservoir, set at the target temperature, via the Berendsen thermostat [34]. This thermostat controls the temperature through molecular velocity scaling. So as to not affect the calculations, the thermostat was applied only to the velocity component perpendicular to the flow direction. Based on the equipartition theorem, such a configuration ensures that each degree of freedom of the system is close to the right temperature [158]. A desired target mass density was obtained with a density controller through molecular insertions and deletions [37]. For convenience, the computational domain was divided into $M = 500$ horizontal bins, including walls (10 bins per unit length), where the sampling of observables took place. Consequently, each velocity profile consisted of about 400 points. After the target values of steady-state had been reached (temperature $T = 1$ and density $\rho = 0.8187$), the density controller was switched off and a constant force, $F_x = 0.6$, was applied to every argon particle in the fill region; the time-step was set to $\Delta t = 0.0025$ in reduced units and data was output every $t_w = 0.25$ in order to ensure statistical independence (see Sec. 2.1.3).

The mean velocity profile, plotted against an instantaneous measurement in Fig. 5.1, was obtained from an ensemble of 100 noisy samples. It should be stated that an

average distribution of comparable quality could have already been recovered from 30 observations (below that value the profile was more noisy). For simulations of simple L-J fluids, the observables tend to be weakly corrupted with noise, and a smooth profile can generally be extracted from a relatively small number of time-steps. We wanted to investigate if the same structure could be obtained from a single measurement; we used the ensemble of 100 instantaneous measurements as a reference *true* signal. To

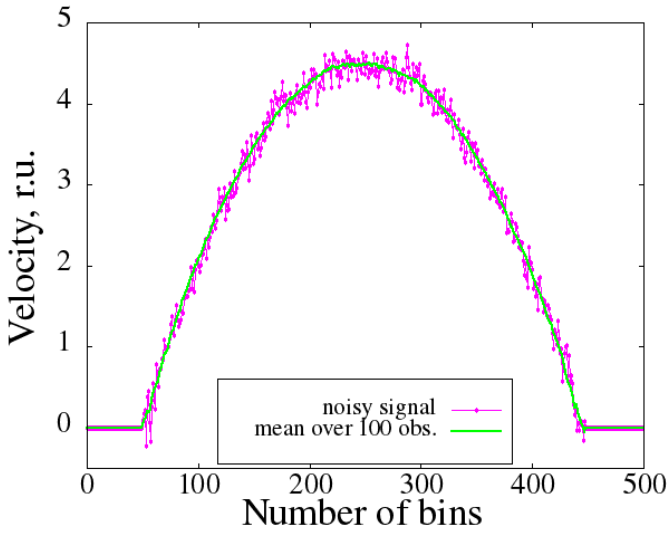
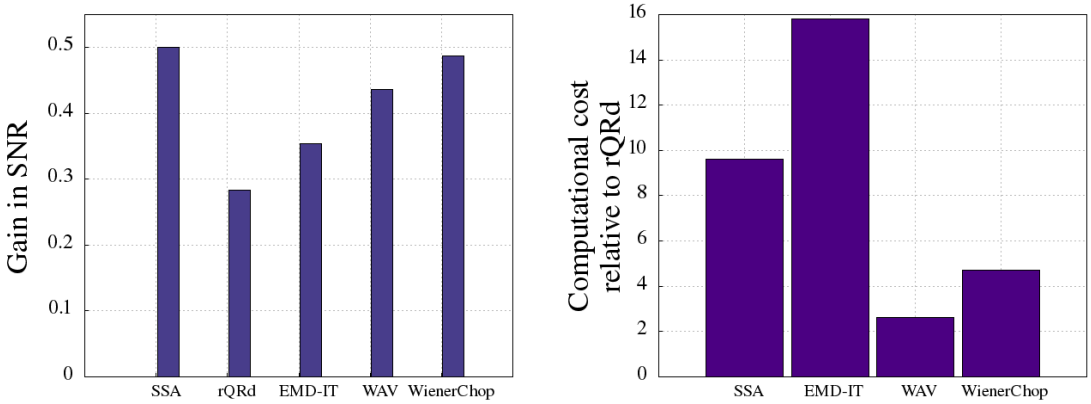


Figure 5.1: Smooth velocity profile in the steady liquid argon flow simulation obtained through averaging over 100 noisy observations.

the velocity data output at the first time-step we applied 1D wavelet thresholding with $sym8$ and five levels of decomposition, and WienerChop with additional $sym4$ and six resolutions for the second transform (note the same parameters were used in Sec. 4.1). Singular spectrum analysis was performed with a window of half the length of the signal, $L = 200$, as recommended for simple dominating trends (see Sec. 3.2.2); three EOFs were chosen because the profile was not an ideal parabola, for which only two eigentriples would need to be recovered. Following the previous study on synthetic data, the oversampling parameter for the rQRd method was set to $p_k = 4$. Interval empirical mode decomposition, EMD-IT, was performed with a constant number of sifting operations, $n = 7$. According to the previous study, EMD-IIT could provide better results as the desired trend was smooth. However, we do not recommend this method as it is too expensive.

All the de-noising techniques were successful in extracting a smooth structure from

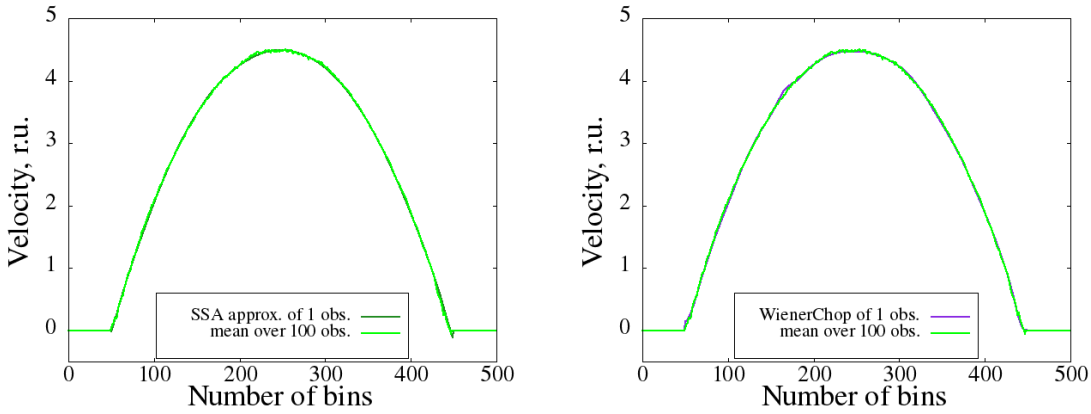
only one instantaneous sample. Gains in SNRs obtained with each method and the corresponding processing times are summarised in Fig. 5.2(a) and Fig. 5.2(b). The best



(a) Gain in SNR. (b) Processing time.

Figure 5.2: Performance summary for $\mathrm{SNR}_{\mathrm{noisy}} = 27.77$ dB; window length for SSA was $L = 200$ and oversampling parameter for rQRd $p_k = 4$; filter $sym8$ was used for 1D-WAV with 5 resolutions, and additional $sym4$ with 6 levels of decomposition for WienerChop.

approximation, i.e. the profile closest to the average of 100 velocity measurements, was obtained with singular spectrum analysis (see Fig. 5.3(a)), which was also the fastest, provided that the number, k, of significant modes was already established. Performing



(a) Profile extracted with SSA. (b) WienerChop de-noising.

Figure 5.3: Ensemble mean and reconstructions obtained from only one observation.

the eigentriple grouping can be done adaptively by, for example, applying the SVHT for unknown noise to the Hankel matrix. Calculating the singular value threshold according to the Eq. (3.29) gave a threshold of 4.442, which resulted in preserving $k = 3$ singular

values. Applying the WienerChop filter also successfully managed to retain the mean shape, as shown in Fig. 5.3(b). Although EMD-IT is the slowest, it is still a valuable method to use; even though essentially no parameters had to be considered before applying the EMD-based interval thresholding (throughout this work the number of sifting operations is kept constant), the retained signal-to-noise ratio was high. In general, all the methods performed well and produced good approximations from one noisy measurement that closely resembled the velocity profile obtained with a minimum of 30 observations. This directly translates to computational savings; after reaching the steady state, the MD simulation was run on 12 processors for a total elapse time of 558 s to provide the ensemble solution of 30 time-steps which could be obtained with the filtering techniques after only one time-step.

Molecular dynamics can be applied to many real-life problems, e.g. for studying water flow in nanostructured membranes. However, such modelling is quite challenging; the system requires small time-steps and the simulations often contain substantial noise which is computationally demanding to reduce. In order to assess how de-noising techniques can improve the analysis of such data, we tested all of the methods for simulation of the Poiseuille flow of water between two rigid planar silicon walls. The size of the computational domain was kept the same as in the previous MD simulations but expressed in the reduced units for water. The water molecules were driven by a constant force, $F_x = 0.6$, and a smaller time-step, $\Delta t = 0.0012$, was used in order to capture all the important dynamics with the write-interval, $t_w = 0.12$ (data was output every 100th Δt). The system's temperature was set to $T = 3.816$, and the water density was $\rho = 1.047$. After reaching a steady-state, an ensemble of 1000 observations, each consisting of 500 spatial points, was generated.

Initially, all the filtering methods were applied to the averaged velocity profile, plotted in Fig. 5.4. This resulted in a smoother distribution than the original averaged solution (see Fig. 5.5(a)). The parameters of each technique were kept the same as for the previous MD simulation. To establish if a similar output would be produced from a smaller ensemble, we applied all the methods to the average over 100 time-steps. For this smaller collection of samples, the velocity profiles obtained with the filtering methods, although smoother, were up to 12% closer to the mean of 1000 samples than

the distribution obtained with simple averaging. Figures 5.5(b)-5.5(c) show the result of applying rQRd to the mean velocity profiles of 1000 and 100 time-steps, respectively. It can be seen that for an increasing number of samples the averaged solution and its filtered approximations converge. However, applying de-noising techniques can provide smoother results faster.

To further confirm this conclusion, we performed the following analysis. Noise reduction techniques were applied to the mean of an increasing ensemble, starting with only 10 observations. We assumed that the average of 1000 profiles was the desired solution, and we measured how much the filtered profiles resembled it in comparison to the statistical mean. The dimensionless gain in SNR was calculated in a similar manner to Eq. (4.2) but with the ensemble mean used instead of $\mathrm{SNR}_{\mathrm{noisy}}$, i.e.

$$\mathrm{Gain} = \frac{\mathrm{SNR}_{\mathrm{approximation}} - \mathrm{SNR}_{\mathrm{average}}}{\mathrm{SNR}_{\mathrm{average}}}. \tag{5.1}$$

Figure 5.6 shows the SNR gain obtained with each technique computed for different ensemble sizes. In other words, the graph shows how much higher was the SNR of each approximation with respect to the SNR of the statistical mean. All the methods, up to 500 samples, extracted velocity distributions closer to the final profile than simple averaging. For larger collections of samples the methods were producing smoother results than the mean of 1000 profiles.

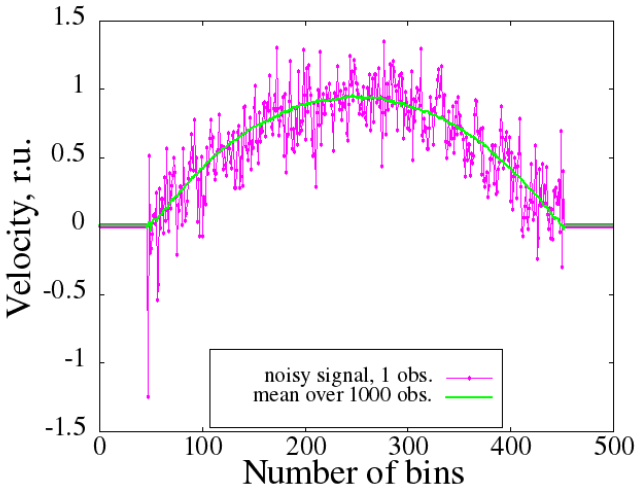
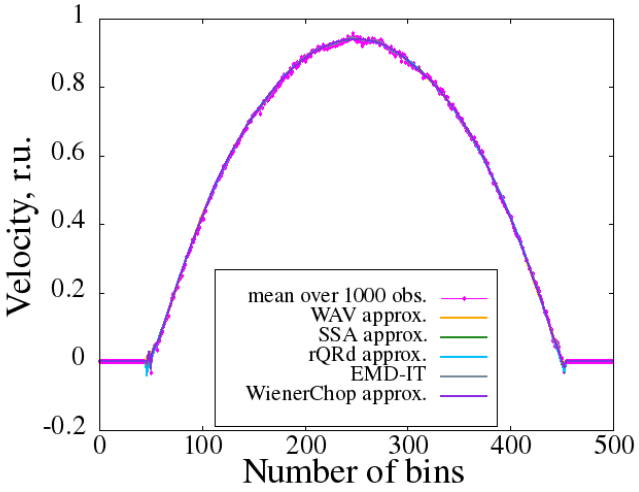
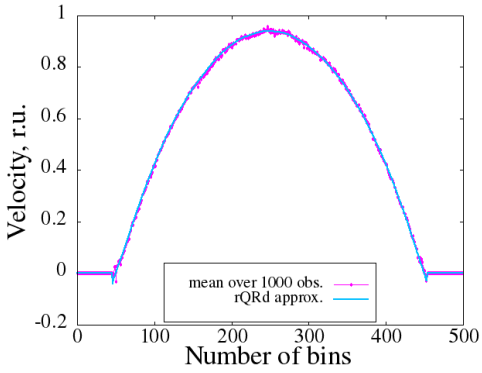


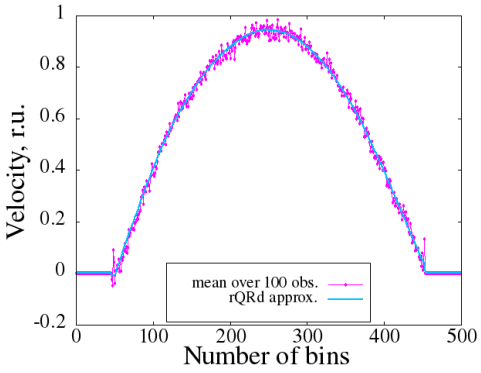
Figure 5.4: Velocity profile from water flow simulation with constant forcing obtained through averaging over 1000 noisy observations.



(a) All the approximations filtered from the mean velocity profile.



(b) Profile extracted with rQRd from the mean of $N = 1000$.



(c) Velocity recovered from $N = 100$.

Figure 5.5: Ensemble mean and its smoother approximations obtained with de-noising methods.

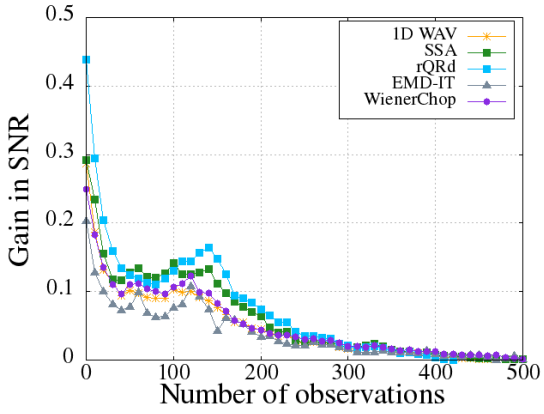


Figure 5.6: Gain in signal-to-noise ratio of each approximation with respect to the mean solution for different ensemble sizes.

The next example is a simulation of shear-driven flow, modelled with DPD. The purpose of this analysis was to show the versatility of the filtering methods. The studied system consisted of a periodic box of $10 \times 10 \times 10$ (in DPD units) filled with 3000 unbonded particles of unit mass. The fluid interaction was defined as a purely repulsive, soft potential with strength $A_C^{(ij)} = 25$ (see Eq. 2.17) and cut-off radius $r_{\text{cut}} = 1$. The Stoyanov-Groot thermostat [159], which is a combination of the Lowe-Andersen and a Galilean-invariant Nosé-Hoover thermostat, was used to control both the fluid viscosity and the temperature. The flow domain was divided into 100 horizontal bins, each of width equal to $0.1r_{\text{cut}}$. The fluid was driven by Lees-Edwards shearing boundaries [160] orthogonal to the x-axis (the flow direction) moving with unit velocity. This way of introducing a shear flow prevents the spatial inhomogeneities induced close to the moving walls. More details on the simulation can be found in [47]. Assuming that the mean of 10000 samples shown in Fig. 5.7 is the required solution, we measured the gain in SNR (according to Eq. (4.2)) obtained with each de-noising method applied to a noisy velocity profile from one time-step. The parameters for each technique were kept the

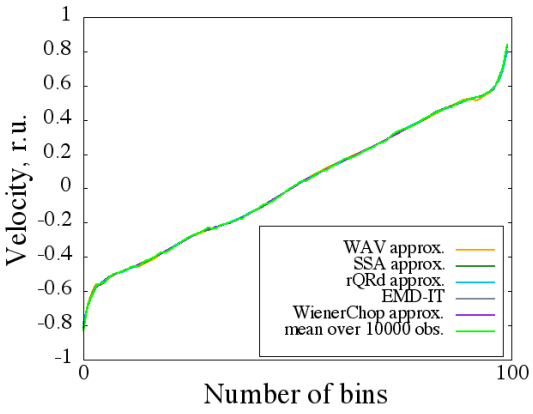
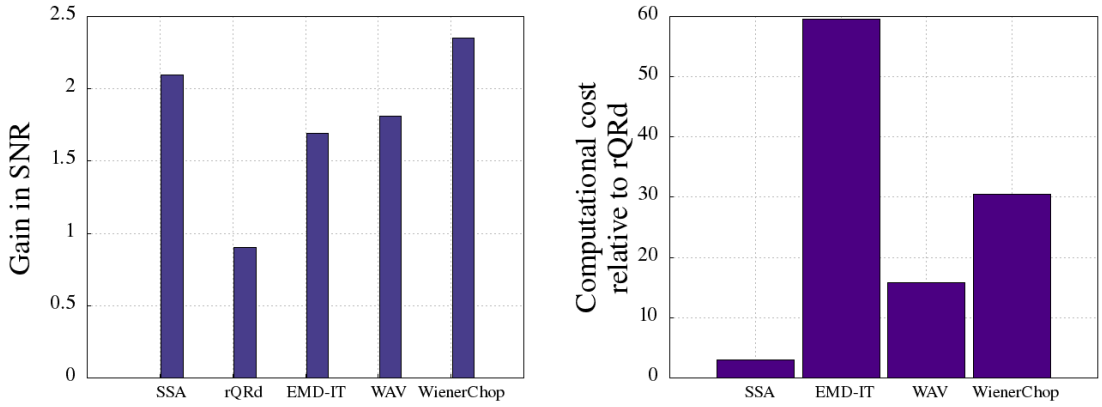


Figure 5.7: Smooth velocity profile and its approximations in a steady shear-driven flow DPD simulation obtained through averaging over 10000 noisy observations.

same as in previous simulations. All the methods significantly improved the quality of the signal, with velocity distributions with up to 234% higher SNR, resembling more the desired solution (see Fig. 5.8). The highest enhancement was obtained with WienerChop and SSA; the SSA and rQRd were the fastest for a pre-determined number of significant eigentriples and oversampling parameter. Figure 5.9 depicts the velocity profile obtained with WienerChop which, unlike the original noisy signal, allows the calculation of more



(a) Gain in SNR. (b) Processing time.

Figure 5.8: Performance summary of employing filtering techniques to shear-driven flow simulation performed with DPD for $\mathrm{SNR}_{\mathrm{noisy}} = 5.55$ dB; window length for SSA was $L = 50$ and oversampling parameter for rQRd $p_k = 4$; filter *sym*8 was used for 1D-WAV with 5 resolutions, and additional *sym*4 with 6 levels of decomposition for WienerChop.

accurate shear rate. Figure 5.10(a) summarises how much SNR gain, according to

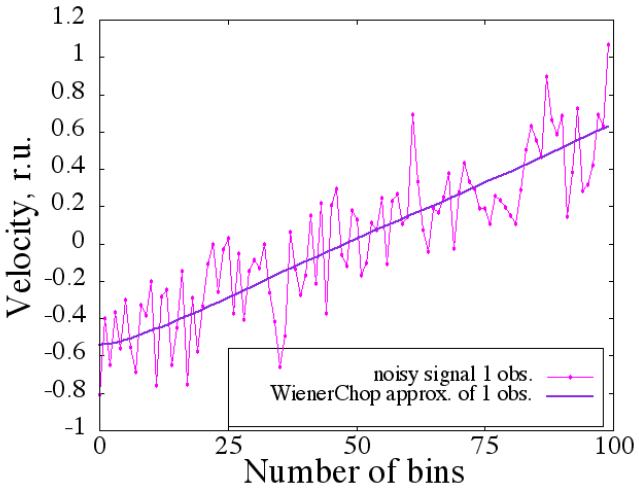
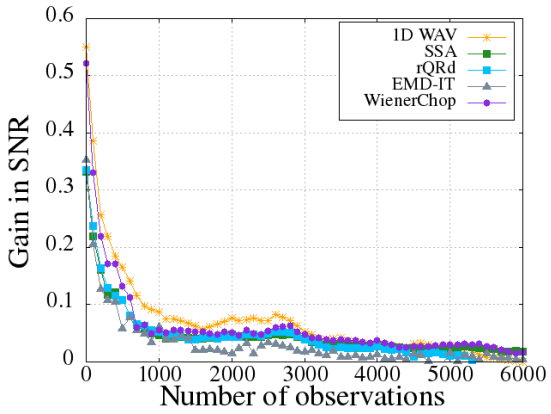


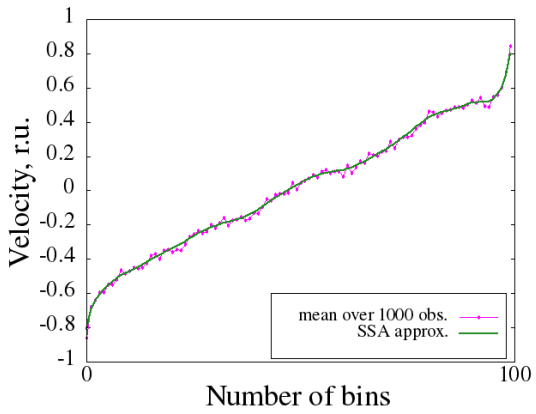
Figure 5.9: Approximation obtained with WienerChop from one noisy observation.

Eq. (5.1), was obtained with each technique for a different number of observations; below 6000 measurements, the approximations resembled the final solution better than the averaged profile, as depicted in Fig. 5.10(b), where an SSA approximation of the averaged velocity of 1000 measurements is plotted against the mean solution.

In conclusion, employing filtering techniques can significantly improve the analysis of stationary flow results, especially in cases where there is only a small number of observables available, insufficient to average out all the unwanted fluctuations, or when



(a) Gain in signal-to-noise ratio of each approximation with respect to the ensemble solution.



(b) Approximation obtained with SSA.

Figure 5.10: Result of applying filtering techniques to 1000 observations of a shear-driven flow simulation performed with DPD.

the computational cost of running a simulation for a long time is too high. Utilising SSA, WienerChop and 1D wavelet thresholding seemed to be the most beneficial. However, all the methods require some *a priori* analysis to be performed, either to determine the number of significant modes, or to choose adequate filters for the wavelet transform. Singular spectrum analysis extracted smooth profiles particularly well, even for very high noise levels, and there is the potential to automate the eigentriple grouping through SVHT. The other procedures, EMD-IT and rQRd, are the least conditioned by predefined parameters, with the latter being also the fastest in data processing. Iterating the EMD-based thresholding improves SNRs for smooth trends, but also increases the time of performing an already computationally intensive method.

5.2 De-noising of time-dependent particle-based simulation data

In the case of modelling stationary phenomena, the averaged solution is estimated over a certain number of independent samples taken sequentially in time. Computing the ensemble mean for non-stationary simulations is more challenging as it is not obvious how to define a time interval over which the data should be averaged [25]. For transient flows the mean distribution can be obtained from an ensemble of realisations. It is also possible to perform phase averaging, if the flow exhibits a limit cycle, and integrate over a large number of repeating periods of oscillation. However, constructing the results based on a number of realisations, N_r, improves the accuracy only by a factor of $\sqrt{N_r}$ [67, 161].

In this section we demonstrate how applying de-noising techniques to non-stationary simulations can improve the information extraction relative to the standard processing approach. To test the performance of POD with time windows, POD+ methods, 2D wavelet thresholding, and 2D WienerChop, we carried out several simulations involving unsteadiness: oscillating shear- and force-driven flows, with and without roughness, and density separation phenomena. Different modelling techniques were employed, including MD, DPD and DSMC, to show how applicable the procedures are. First, we present results from liquid argon and water flow simulations performed with MD in Sec. 5.2.1. We then show how DPD data can be processed in Sec. 5.2.2, followed by DSMC simulation of gaseous argon driven by a time-periodic force.

5.2.1 Results from non-stationary MD simulations

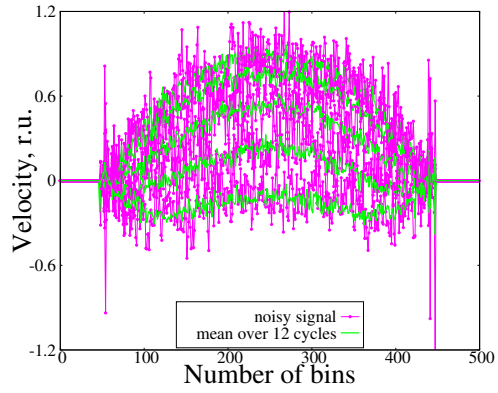
Different types of non-stationary flow of liquid argon in a krypton nanochannel were modelled with MD. For each set-up, the configuration was kept the same as in Sec. 5.1, including the system size and target values for steady-state. For oscillating Poiseuille flow, a periodic force given by $F_x = \Delta P \sin(\omega t)$, where $\Delta P = 0.6$ and $\omega = 2\pi/80$, was applied to every argon particle in the fill region, and for time-dependent Couette flow, the upper wall was set to move with the same period but with an amplitude of $\Delta V_x = 0.5$. The time-step for each simulation was again set to $\Delta t = 0.0025$ in reduced

units, and data was output every $t_w = 0.25$ in order to ensure statistical independence.

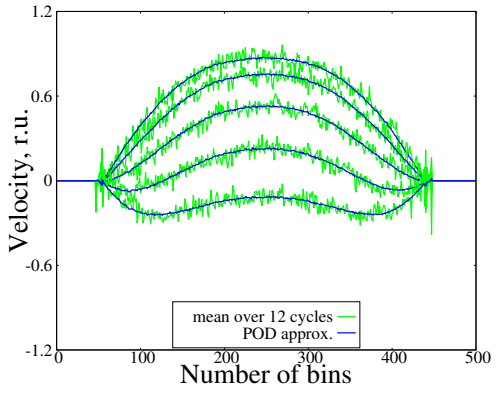
The moving window was utilised in the manner described by Grinberg [25]. While the window moved throughout the simulation, the matrix was being updated every N_{ts} of t_w samples. To allow a straightforward comparison of all the de-noising methods, in most of the cases no averaging was applied prior to data filtering, i.e. $N_{ts} = 1$. We stress that POD and WPOD methods essentially differ only in the implementation. Therefore, the results obtained from POD with a moving window are labelled in figures as POD approximations. All the other procedures were applied to the same T_{POD} ensemble. For all the simulations the wavelet thresholding was performed with the *db*8 filter and 7 decompositions to retain more smoothness, and for the WienerChop procedure additional *db*4 and 8 resolutions were employed; EMD-IT within POD was performed with a constant number of sifting processes, $n = 7$; POD+SSA was utilised with a window size of $L = 50$, unless indicated otherwise, and the oversampling parameter for POD+rQRd was $p = 4$ as in the previous study.

If statistical averaging is to be employed in order to improve the quality of the results, a simulation has to be either performed several times or, in the case of periodic oscillations, left to run for long enough to gather a sufficient number of samples. In the simulation of non-stationary force-driven flow in a smooth channel, a full period was every $t = 80$, which translated to 320 velocity measurements as $\frac{80}{t_w} = 320$. The initial matrix contained $N = N_{\mathrm{POD}} = 4000$ observations and $M = 500$ velocity measurements at each time-step; therefore there were 12 complete oscillations in the ensemble, which were used to obtain the mean solution.

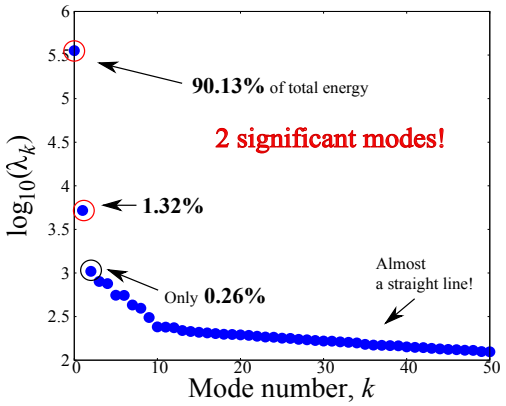
Figure 5.11(a) shows how disturbed the original data was, and Fig. 5.11(b) compares the quality of the cumulative mean (the average of 12 cycles) and our WPOD approximation. The latter clearly extracted a smoother velocity profile for the same number of measurements ($N = 12 \times 320 = 3840$). To obtain a comparable level of de-noising with statistical averaging, much more data would have to be collected, increasing the computational cost. Moreover, WPOD does not require any *a priori* information regarding the nature of oscillations, which is beneficial, e.g. when the frequency of fluctuations changes over time. In this case, only two modes were used to extract the velocity field with WPOD after performing the following analysis on the whole ensemble: examina-



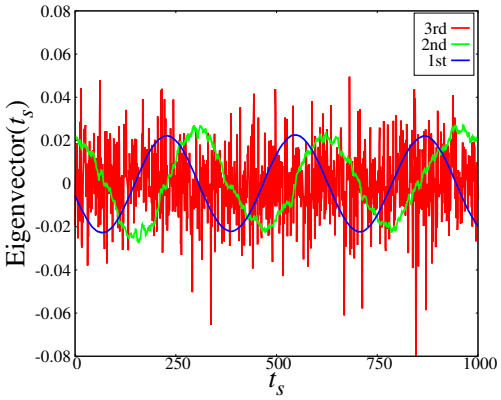
(a) Noisy data and ensemble mean.



(b) De-noised velocity profiles with WPOD.



(c) LEV diagram of eigenspectrum; the first 50 values.



(d) The eigenvectors for 1000 samples.

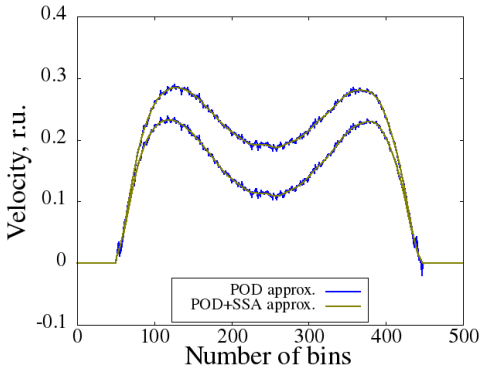
Figure 5.11: Result of applying POD with a moving window to the developed velocity field from an MD simulation of a periodically-pulsating flow in a smooth channel; $N_{ts} = 1$ and $N_{\text{POD}} = 12 \times 320$.

tion of energy level of eigenspectrum, the rate of decay of eigenvalues, and studying the smoothness of eigenvectors. The first two eigenvalues were the most energetic. The semi-log diagram is illustrated in Fig. 5.11(c); it can be seen that $\lambda_{k=1} = s_1^2$ and $\lambda_{k=2} = s_2^2$ were decaying much faster than the eigenvalues, i.e. they corresponded to dominant modes. Eigenvalues with $k > 2$ represented features with a short correlation time (noise). As presented in Fig. 5.11(d), the first and second eigenvectors (or left singular vectors) described the oscillating nature of the data; the remaining modes contained high frequencies. In addition, it should be stressed that the first eigenvector was smoother than the second one, which was slightly disturbed. This observation suggests that noise was not entirely filtered out from the second mode. We also utilised

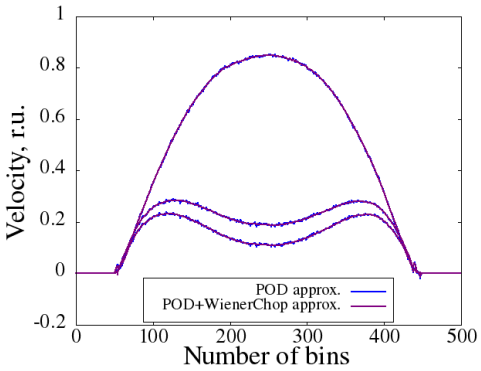
the SVHT for singular values, but the estimated threshold was too low, even when the noise level was determined from wavelet coefficients. This is due to the fact that simulation data often suffers from correlated noise for which, as discussed in Sec. 4.3, the optimal threshold developed by Donoho and Gavish [94] was not designed. Therefore, our modified procedure for estimating the threshold was employed; the square root of the standard deviation calculated from Eq. (3.61) was inserted into Donoho's formula for SVHT with known noise, to confirm the same number, k, of orthogonal functions. In addition, by replacing the median of singular values with the median of eigenvalues, improved thresholds were computed and used to verify the previous result. For all the simulations discussed in this section we utilised both approaches in WPOD calculations. Only when a trajectory (Hankel) matrix in the SSA analysis was considered, the original definition of SVHT was used.

Analysis of the temporal modes suggested that WPOD did not entirely remove noise from the data. This was improved by utilising our POD+ methods which incorporate additional filtering to the SVD in order to separate spatial fluctuations from the ensemble solution. Figure 5.12(a) depicts how the results were further improved by combining WPOD with SSA, and in Fig. 5.12(b) the approximations obtained with POD+WienerChop are plotted; all the POD+ techniques produced comparable output, as shown in Fig. 5.12(c). The wavelet-based techniques, 2D-WAV and 2D-WienerChop, applied directly to the noisy velocity profiles, produced the poorest results (see Fig. 5.13 as the wavelets seemed to *follow* the noise. In addition, when only $N = 400$ observations (10 times fewer) were used for de-noising, POD+ methods were still capable of extracting similar profiles as for the larger number of measurements, while applying the other techniques resulted in artifacts and unwanted frequencies. Figures 5.14(b)-5.14(d) compare WPOD and POD+ methods for $N = 400$, showing that the latter are more efficient in extracting information from the noisy data. Applying the criteria for defining k to the matrix with $N = 400$ managed to successfully identify the number of significant modes, i.e. $k = 2$.

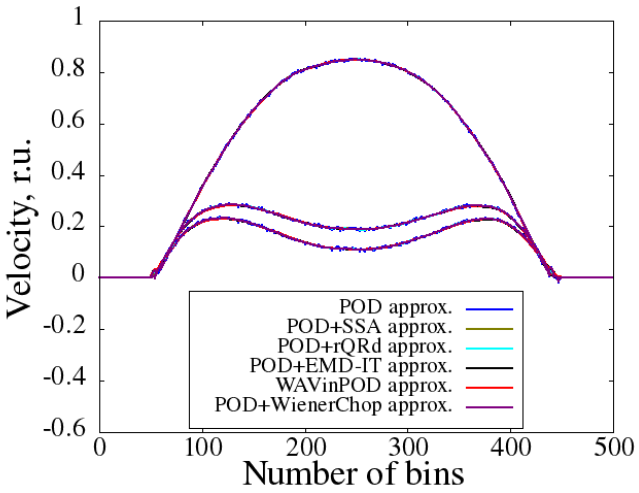
Molecular dynamics simulations are often used to investigate the influence of atomistic scale surface roughness on the slip behaviour in liquid films [156, 162]. In order to show how applying de-noising techniques can improve the study of slip phenomena, we



(a) POD+SSA approximation.



(b) POD+WienerChop approximation.



(c) De-noised velocity profiles obtained with POD+ methods and WPOD; $N_{ts} = 1$ and $N_{\mathrm{POD}} = 4000$.

Figure 5.12: Comparison of WPOD, and POD+ methods in de-noising velocity data from the simulation of oscillating Poiseuille flow performed with MD; db8 and 7 decompositions were used for the WT_1, and db4 and 8 resolutions for WT_2; for $L = 50$ in SSA analysis $k = 5$ EOFs were preserved.

introduced surface roughness to the system described previously. A periodic roughness was applied by placing a cavity with dimensions $5 \times 3 \times 10$ within the lower wall, as presented in Fig. 5.15. All other simulation parameters were kept the same. During one simulation run, $N = N_{\mathrm{POD}} = 10000$ velocity profiles consisting of $M = 500$ points were collected. Figures 5.16(a) and 5.16(a) illustrate how poor were the original instantaneous measurements and the averaged results (over $10000/320 \approx 31$ full cycles) in comparison to the velocity profiles obtained with WPOD; dashed vertical lines in the plot indicate the position of the cavity. For such a large number of measurements, proper orthogonal decomposition managed to extract smooth signals, which did not

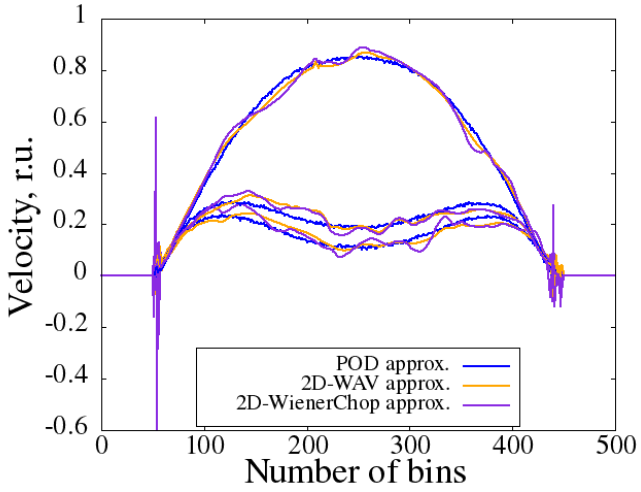
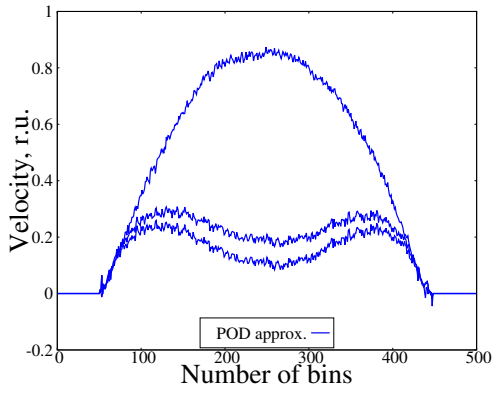


Figure 5.13: Result of applying 2D wavelet thresholding and WienerChop to noisy measurements; db8 and 7 decompositions were used for the WT_1, and db4 and 8 resolutions for WT_2. Note poor de-noising performance relative to POD+ methods.

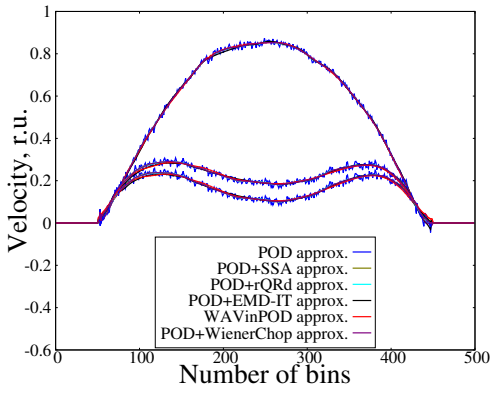
require much improving with POD+ methods (see Fig. 5.17(a)). In order to observe how well POD+ techniques would perform for a smaller number of measurements, we applied all the methods to the ensemble of $N = 1000$, and to only one full cycle, i.e. $N = 320$ snapshots; the results are depicted in Fig. 5.17(b) and Fig. 5.17(c). For a decreasing number of samples, WPOD was less successful in removing noise. Combining it with the other techniques improved the signal-to-noise ratios even for the smallest data-set over which no averaging could be carried out.

Assuming that the WPOD approximation of the largest ensemble shown in Fig. 5.16 is the desired solution, we analysed the gains in SNRs obtained with each technique for different data-sets, and summarise the results in Fig. 5.18(a) and Fig. 5.18(b) for $N = 1000$ and $N = 320$, respectively. It can be seen that the noise reduction of 2D wavelet thresholding and the WienerChop did not improve with different sizes of the data-set. In contrast, for the smallest ensemble, POD+ methods enhanced WPOD's performance, achieving on average 55% greater gain[2], or 1.2× higher SNR, whereas for $N = 1000$ the average SNR was 1.1× higher than for WPOD. Note that the SNR values were computed assuming that WPOD's mean extracted from the large ensemble is the true solution, even though it was not as smooth as the POD+ approximations. Singular spectrum analysis performed on spatial modes, POD+rQRd and POD+EMD-

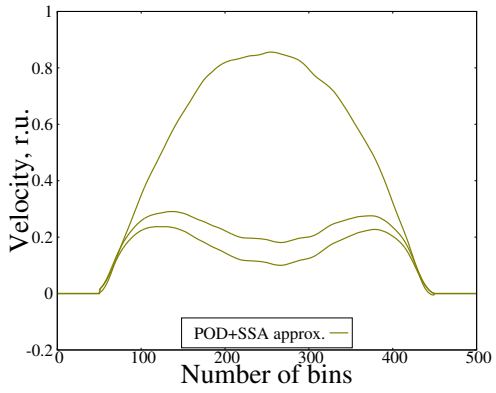
[2]The difference is simply established by subtracting the average gain achieved with POD+ methods from WPOD's gain in SNR.



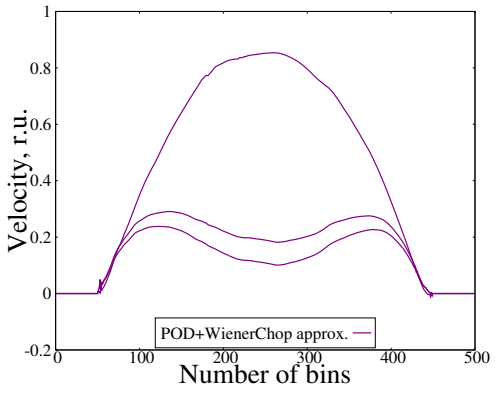
(a) Velocity measurements provided with WPOD.



(b) De-noised velocity profile obtained with POD+ methods and WPOD approximations.



(c) POD+SSA approximation.



(d) POD+WienerChop approximation.

Figure 5.14: De-noising performance of WPOD and POD+ methods for a smaller ensemble of $N = 400$ velocity measurements of oscillating argon flow modelled with MD.

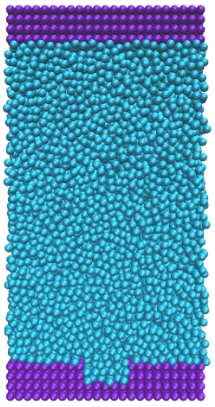
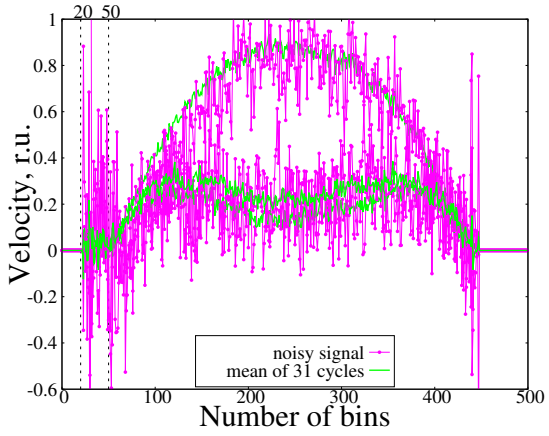
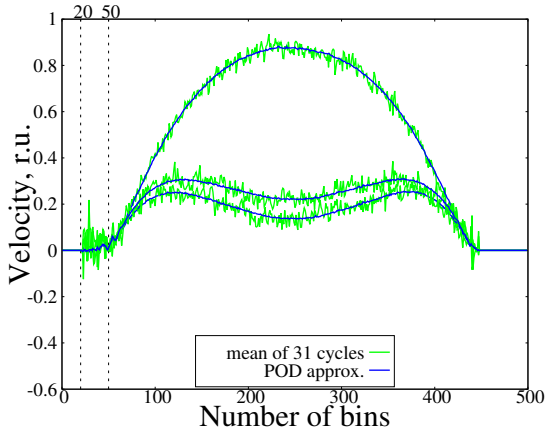


Figure 5.15: Snapshot of the MD simulation with an introduced cavity in the lower wall.

IT provided particularly good results, producing smooth distributions even at the wall with the cavity; some examples are depicted in Fig. 5.19 and Fig. 5.20. It should be



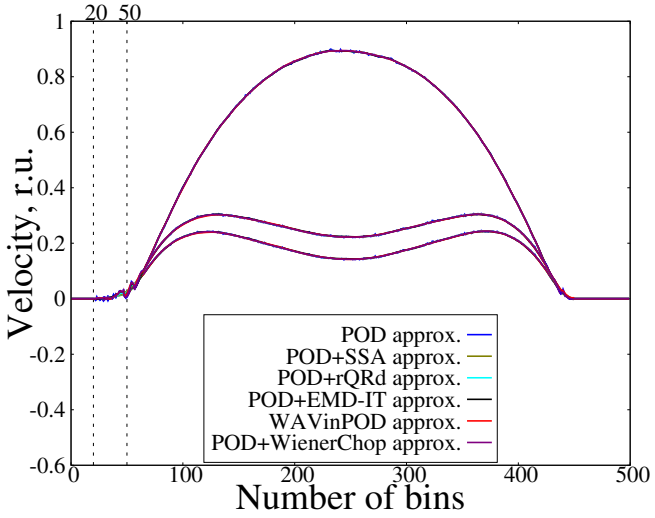
(a) Instantaneous data and the ensemble average.



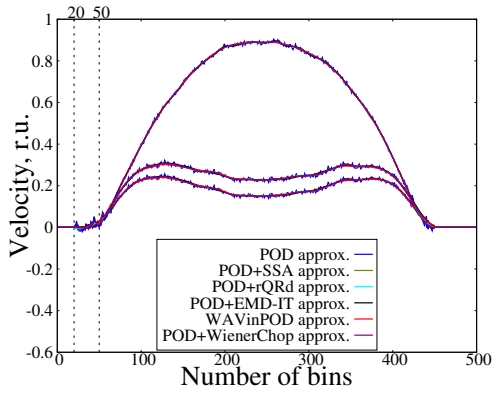
(b) WPOD approximation.

Figure 5.16: Velocity profiles obtained with WPOD and statistical averaging over 31 full cycles containing 320 measurements, $N = 320 \times 31 = 9920$.

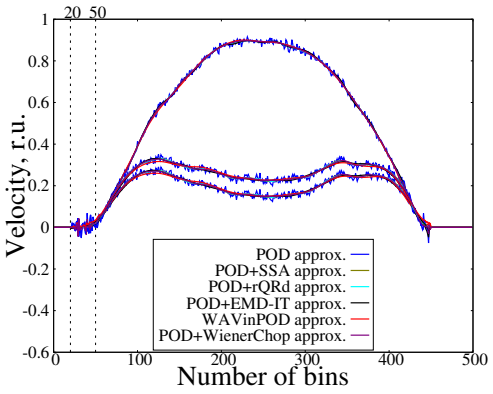
stressed that both WAVinPOD and POD+WienerChop techniques managed to recover high SNRs, and even better de-noising could have been achieved with a different basis. However, for simplicity, the same filters were used in this study for all the simulations. Employing POD+WienerChop or POD+EMD-IT was the most computationally expensive. Both 2D wavelet thresholding and the WienerChop filter applied directly to noisy data did not perform well; the latter offered the poorest enhancement in the data quality as the clean signal estimation and noise variance established in the first transform, WT_1, were not precise due to the high noise level. Additional analysis was carried out to confirm that no benefit was gained from applying de-noising to both- temporal and spatial modes, as in the WAV2inPOD method.



(a) Low-rank approximations obtained with WPOD and POD+ methods; $N_{ts} = 1$ and $N_{\mathrm{POD}} = 10000$.



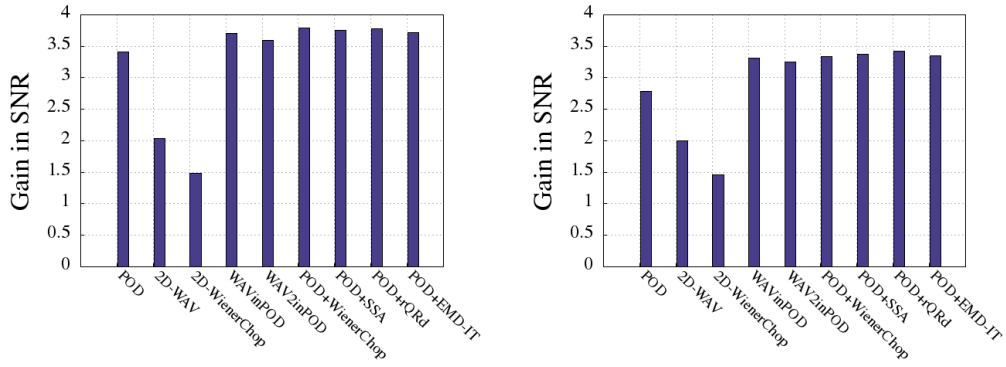
(b) $N = N_{\mathrm{POD}} = 1000$.



(c) $N = N_{\mathrm{POD}} = 320$.

Figure 5.17: Comparison of WPOD and POD+ methods for decreasing number of velocity measurements. All the methods enabled a clearer view of the slip-phenomena. Again POD+ methods outperformed WPOD, extracting smooth profiles even for $N = 320$ (one full oscillation).

The last example with liquid argon was a non-equilibrium steady-state and time-periodic molecular dynamics simulation of Couette flow. The influence of oscillatory shear on the boundary slip is often studied with particle-based simulations, as experimental analysis is challenging [163]. We applied WPOD, POD+ and WAV techniques to a collection of $N = N_{\mathrm{POD}} = 6000$ noisy velocity profiles (consisting again of $M = 500$ bins) obtained from the shear-driven flow induced by an oscillating upper wall ($V_x = 0.5 \sin\left(\frac{2\pi t}{80}\right)$ in reduced units). The noise level in the data was very high, and even though the velocity profiles recovered with WPOD contained fewer high frequen-

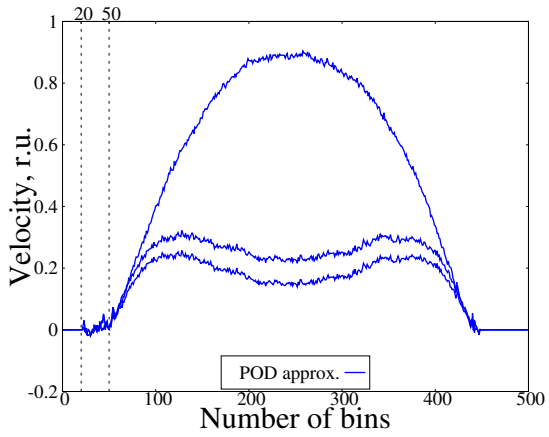


(a) Gain in SNR for signals recovered from ensemble of $N = 1000$ measurements.

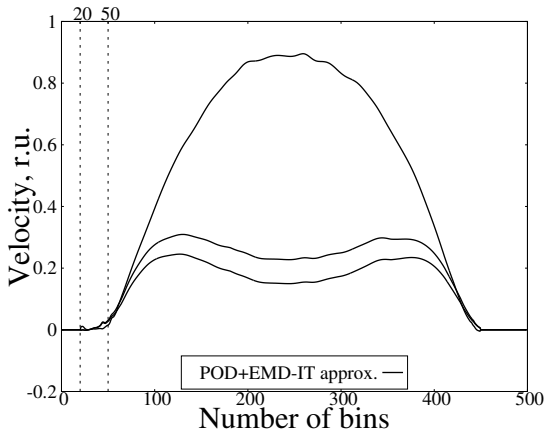
(b) Gain in SNR for $N = 320$.

Figure 5.18: Comparison of de-noising efficiency in processing data-sets of different sizes; $\mathrm{SNR}_{\mathrm{noisy}} = 7.42$ dB for $N = 1000$ and $\mathrm{SNR}_{\mathrm{noisy}} = 7.30$ dB for $N = 320$.

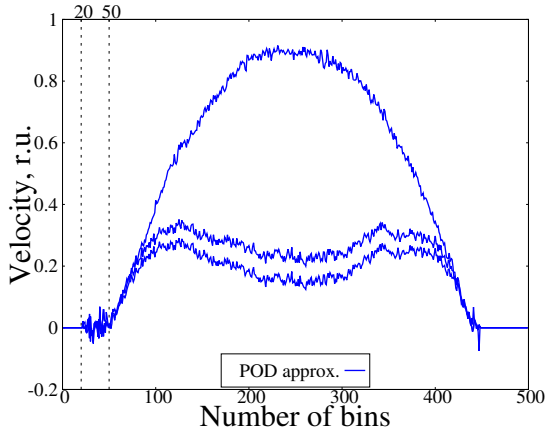
cies than the distributions averaged over 18 cycles shown in Fig. 5.21(a), they were not smooth. The presence of additional disturbances was confirmed with the analysis of the dominant temporal modes plotted in Fig. 5.21(b). Further smoothing of POD spatial modes was carried out by applying POD+ methods; examples of the velocity profiles extracted with POD+WienerChop and POD+EMD-IT are shown in Fig. 5.22(b) and Fig. 5.22(c), respectively. The results were further compared with a smaller data-set containing only one period of oscillation, $N = 320$ measurements, in order to establish whether any of the methods has the ability to reduce the computational time required to extract easier-to-analyse data. For such a small number of samples, no averaging could be performed without loss of information. The signal-to-noise ratios were computed for each approximation obtained from the small ensemble, assuming that the mean distribution obtained with WPOD for $N_{\mathrm{POD}} = 6000$ was the true result, even though the profiles were not entirely smooth. Substantial noise reduction was observed with values of SNRs being even 1560% (or 15.6×) higher than the original noisy signals (see Fig. 5.23(b) and Fig. 5.23(a)); for example, WAVinPOD recovered SNR = 23.36 dB (while the actual MD data had SNR = -1.6 dB) and produced results closer to the desired solution than WPOD, plotted in Fig. 5.24. On average, POD+ methods achieved 200% more gain in SNR than WPOD (they recovered 1.19× higher SNRs), i.e., they produced distributions more closely resembling the approximation extracted from $N = 6000$ samples.



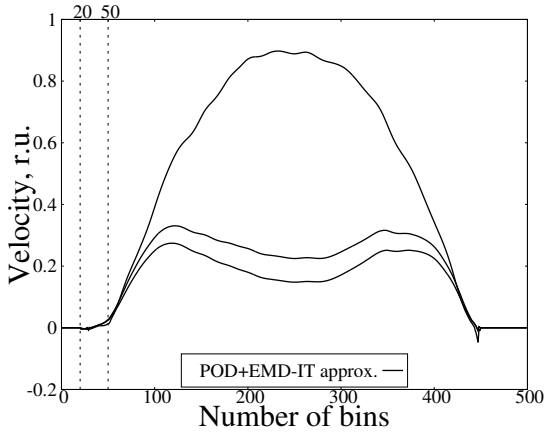
(a) WPOD for $N = 1000$ measurements.



(b) POD+EMD-IT de-noising for $N = 1000$.



(c) WPOD for $N = 320$.

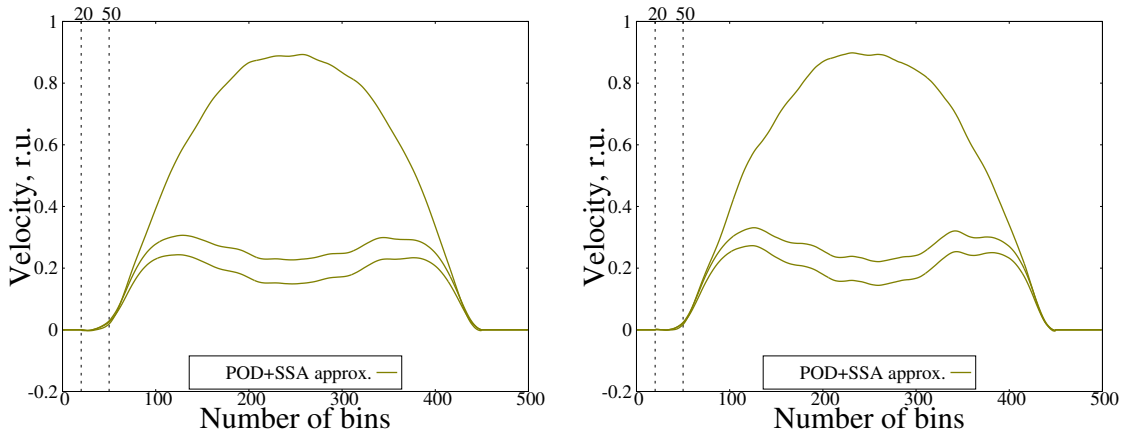


(d) POD+EMD-IT results for $N = 320$.

Figure 5.19: Low-rank approximations recovered with POD+EMD-IT for data-sets of different sizes; soft thresholding with constant number of sifting processes, $n = 7$.

In conclusion, employing a POD+ technique can significantly reduce computational time even for very noisy non-stationary MD data, achieving much better results from just one period of oscillation than statistical averaging from 18×320 samples. For this example, all the parameters apart from the window length in the SSA analysis, were kept the same as in our previous simulations. In the case of shear-driven flow, a larger window size, $L = 250$, had to be applied due to strong contamination; employing $L = 50$ resulted in a decomposition that was too coarse. As a consequence, POD+SSA was not the fastest method to perform.

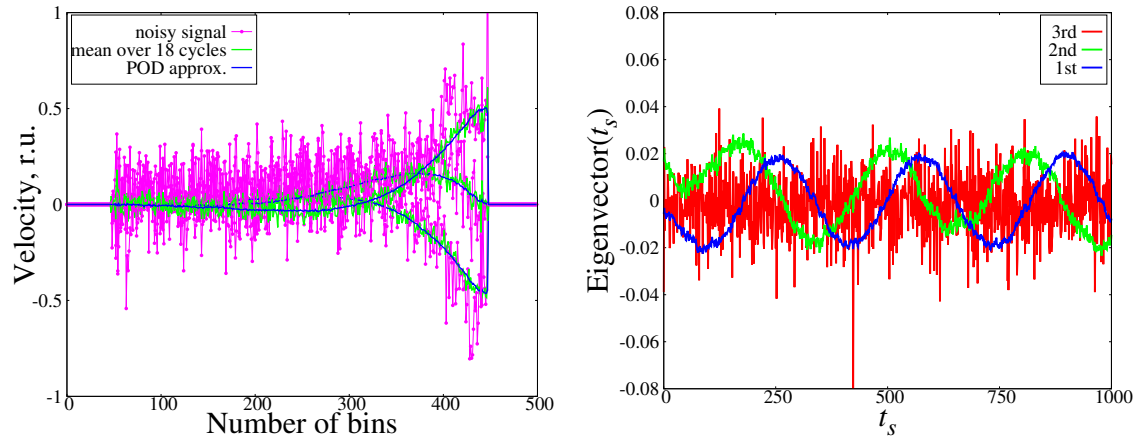
In order to compare the results obtained for simulations of liquid argon with more complex (and noisy) water flow, the system considered in Sec. 5.1 was used with a peri-



(a) Approximations for $N = 1000$ measurements.

(b) Results for $N = 320$.

Figure 5.20: Low-rank approximations recevered with POD+SSA for data-sets of different sizes; window length was set to $L = 50$ and $k = 4$.

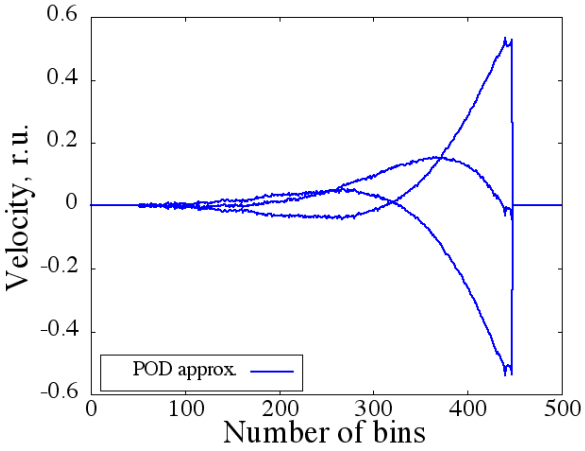


(a) De-noised velocity profiles with WPOD and the statistical mean over 18 cycles.

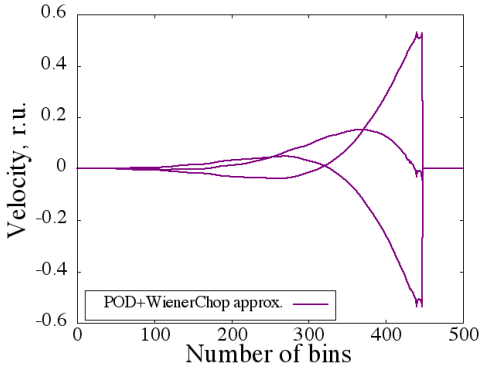
(b) The eigenvectors for the first 1000 samples.

Figure 5.21: Result of applying WPOD to the developed velocity field from an MD simulation of an oscillating Couette; $N_{ts} = 1$ and $N_{\text{POD}} = 6000$, $k = 2$ orthogonal modes were used.

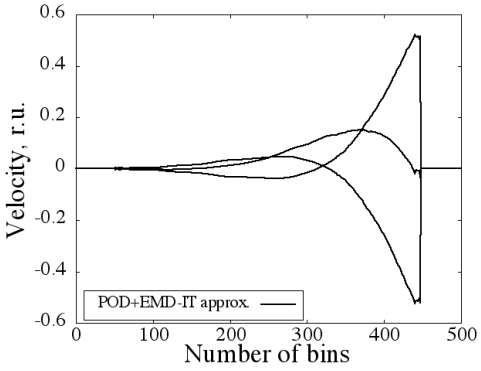
odically oscillating force $F_x = \Delta P \sin(\omega t)$, where $\Delta P = 0.6$ and $\omega = 2\pi/80$ in reduced units for water. The same time-step, $\Delta t = 0.0012$, with a write-interval $t_w = 0.12$ (data was output every 100th Δt) was applied; i.e. one complete oscillation consisted of $80/t_w \approx 666$ observations. Figure 5.25(a) shows how WPOD with $k = 1$ performed in comparison with statistical averaging over full cycles for the same data-set; it can be easily observed that the ensemble mean was still very noisy and more measurements would be required to extract the same velocity profiles as WPOD. Additional smoothing



(a) WPOD.

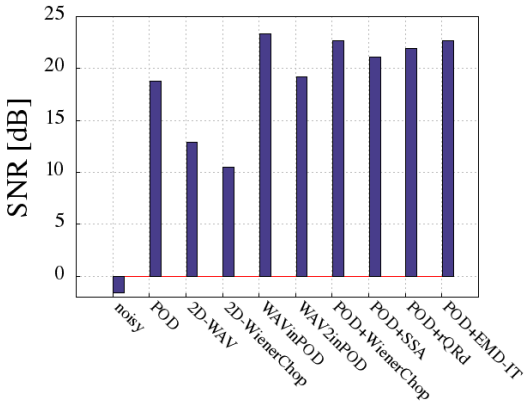


(b) POD+WienerChop.

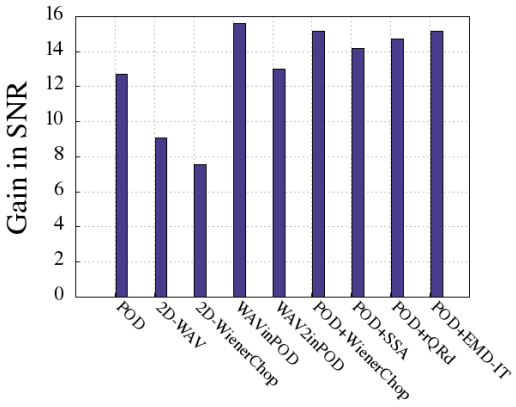


(c) POD+EMD-IT.

Figure 5.22: De-noising of velocity profiles with $N_{ts} = 1$, $N_{\mathrm{POD}} = 6000$, and $k = 2$ using WPOD and POD+ methods for oscillating Couette flow.

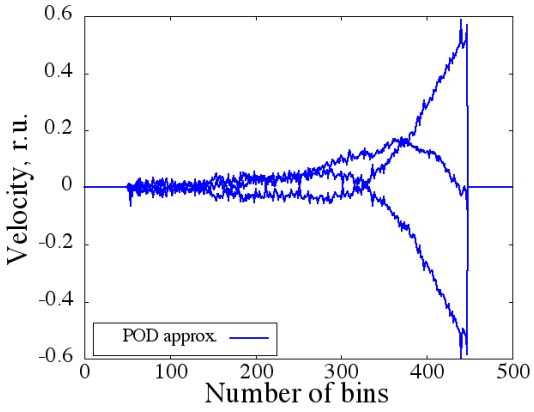


(a) SNRs of noisy signal and the approximations.

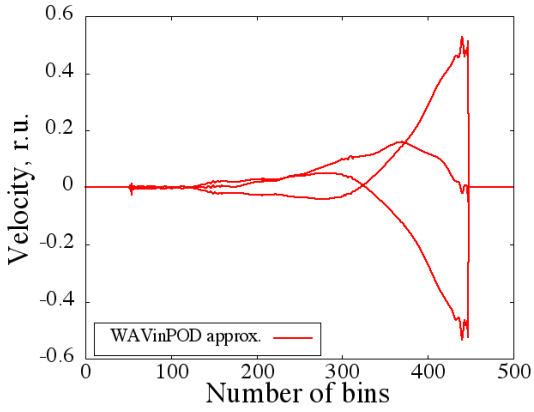


(b) Gain in SNR.

Figure 5.23: Comparison of SNR values for velocity profiles recovered from ensemble of $N = 320$ measurements (one full oscillation) with $\mathrm{SNR}_{\mathrm{noisy}} = -1.6$ dB.

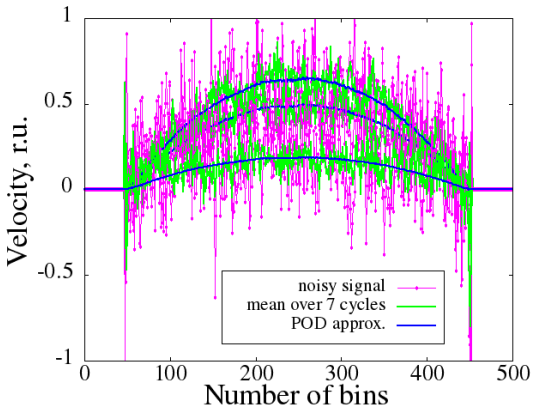


(a) Approximation constructed with WPOD.

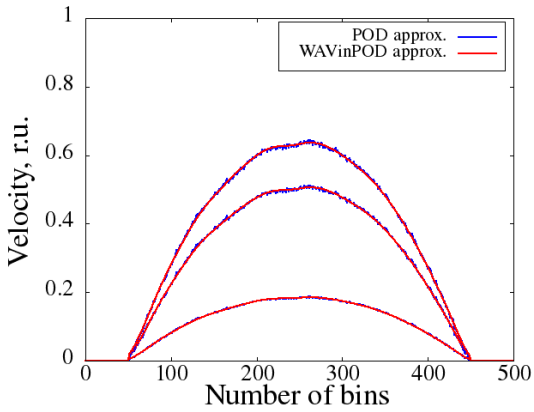


(b) Reconstruction with WAVinPOD.

Figure 5.24: Comparison of results obtained with WPOD and WAVinPOD with $k = 2$ for only one period of oscillation, for $N_{ts} = 1$ and $N = N_{\mathrm{POD}} = 320$.



(a) WPOD for $N_{\mathrm{POD}} = 4662$ and the average of $\frac{4662}{666} = 7$ cycles.

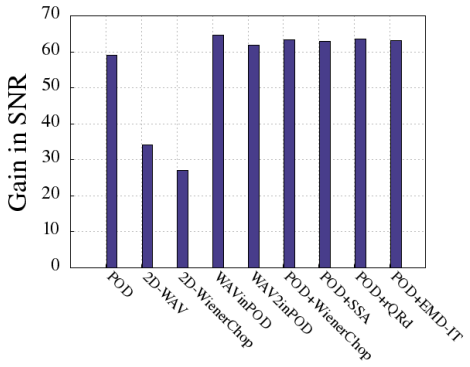


(b) Low-rank approximation for $N_{\mathrm{POD}} = 4662$.

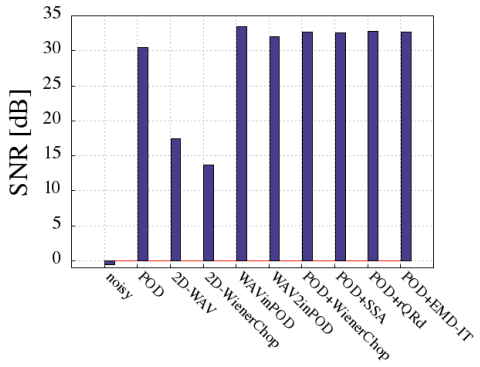
Figure 5.25: Comparison of WPOD and WAVinPOD (with filter *db*8, 7 resolutions and $k = 1$ significant orthogonal mode) in de-noising velocity data from the simulation of unsteady water flow. Both WAVinPOD and WPOD produced better quality profiles than statistical averaging, for the same number of measurements.

was obtained again by employing the POD+ methods; an example of the WAVinPOD approximation is plotted in Fig. 5.25(b). For a smaller system containing only one complete oscillation, $N = 666$, all the filtering methods offered significant enhancement of the data quality; comparison of gains in SNRs obtained with each technique was summarised in Fig. 5.26(a) and Fig. 5.26(b). Only two-dimensional wavelet thresholding and WienerChop directly applied to instantaneous measurements, as in previous cases, did not achieve an efficiency similar to the WPOD and POD+ methods. The

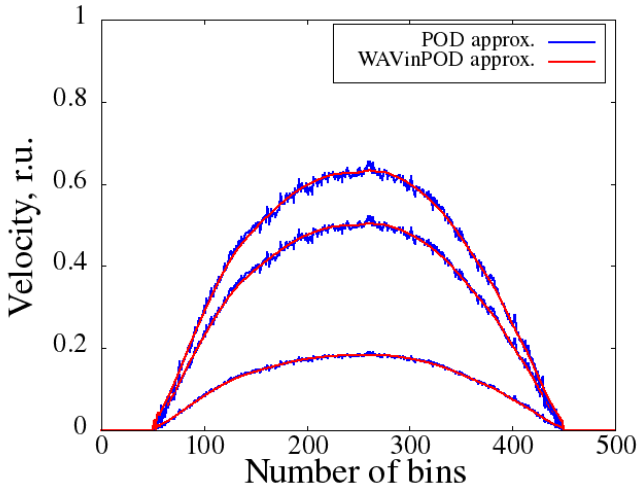
low-rank approximation recovered with WAVinPOD for a small ensemble, presented in Fig. 5.26(c), was comparable to solutions extracted from a 7× larger data-set.



(a) Gains in SNR for each method.



(b) Values of SNRs.



(c) WPOD and WAVinPOD low-rank approximations for a small ensemble.

Figure 5.26: De-noising performance of WPOD and POD+ methods applied to only one full cycle, $N = 666$ with $\mathrm{SNR}_{\mathrm{noisy}} = -0.53$ dB, over which no averaging can be applied without loss of information.

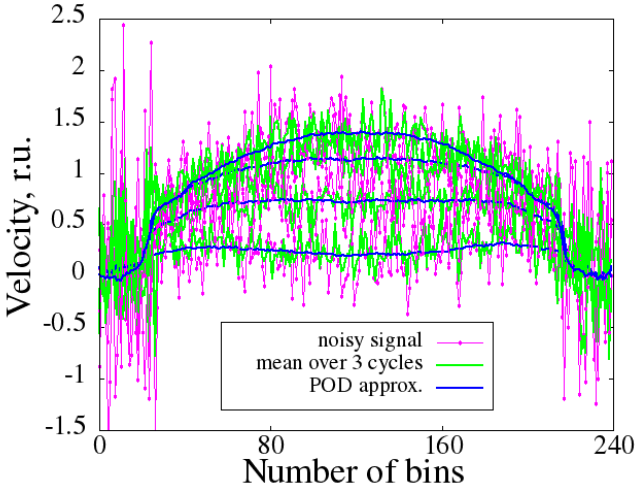
5.2.2 Processing data from DPD modelling

Correlations in noise are expected in MD, e.g. due to temperature control, so we have also tested WPOD and POD+ procedures on an oscillating Poiseuille flow and phase separation phenomena simulated using the DPD mesoscale method. Coarse-graining in DPD reduces the number of degrees of freedom for the particles, neglecting some of the atomistic details that are captured in MD simulations. Modelling a number of

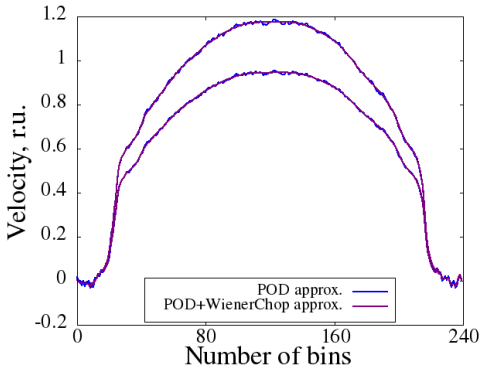
MD particles as one DPD bead results in fewer statistically dependent measurements, which should be easier to process for WPOD. Moreover, in DPD the temperature is controlled locally with the use of random and dissipative forces, which introduce and decrease the energy in the system, respectively. The variable that defines the strength of the random force produces a Gaussian distribution. This stochastic local thermostat plays a role in relaxing the correlations in the data and should enable a more efficient de-noising than in the case of a globally thermostatted system. The following analysis aimed to test this hypothesis. Parameters of POD+ estimators were kept the same as in the MD analysis, unless otherwise specified. The decomposition level of wavelet transforms was dependent on the length of the processed functions but, in general, high resolutions were preferred for removing dominant fluctuations.

All the de-noising techniques were used to process velocity measurements from a non-stationary flow simulated with the DPD mesoscale particle method. The system consisted of 3000 unbounded particles in a box of $10 \times 10 \times 10$ DPD units with walls of frozen particles of unit thickness and a particle density of 3. The planar flow was driven by a time-periodic force, $F_x = \Delta P \sin(\omega \Delta t)$, with $\Delta P = 0.1$, $\omega = 2\pi/80$ and $\Delta t = 0.01$, i.e. one complete cycle consisted of 8000 Δt. The result of applying WPOD to the $N = N_{\mathrm{POD}} = 24000$ and $M = 240$ matrix with streaming-velocity is compared to the ensemble mean (over $24000/8000 = 3$ full periods) in Fig. 5.27(a). Clearly, WPOD produced much smoother velocity data, which statistical averaging could not obtain for the same number of observables. In consequence, employing the POD+ methods provided at least the same quality results as WPOD. Examples of velocity profiles produced with POD+WienerChop and WAVinPOD are depicted in Fig. 5.27(b) and Fig. 5.27(c), respectively.

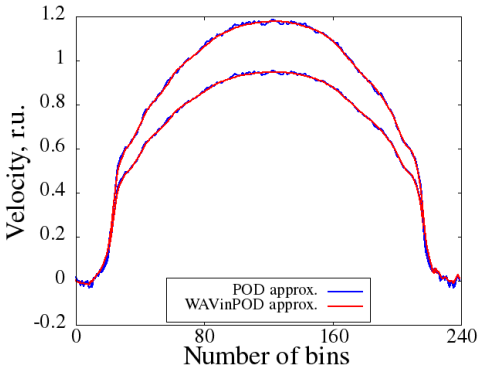
The performance of all the techniques was further compared against the desired output (produced with WPOD with $N_{\mathrm{POD}} = 24000$) for a smaller size of the moving window. For $N_{\mathrm{POD}} = 4000$ observables (half of the complete cycle), the SNR gains for POD+ reconstructions were higher than for WPOD, e.g. the biggest gain, 68% more than WPOD (over 7× higher SNR than original noisy signal), was achieved with POD+EMD-IT, and the lowest with POD+WienerChop, 23%. Less improvement was offered with POD+WienerChop relative to WAVinPOD because the estimated noise



(a) Comparison of WPOD with $k = 2$ and statistical averaging over 3 full cycles.



(b) POD+WienerChop and 2D wavelet thresholding; *db*8 and 6 resolutions for WT_1, *db*4 and 7 decompositions for WT_2.

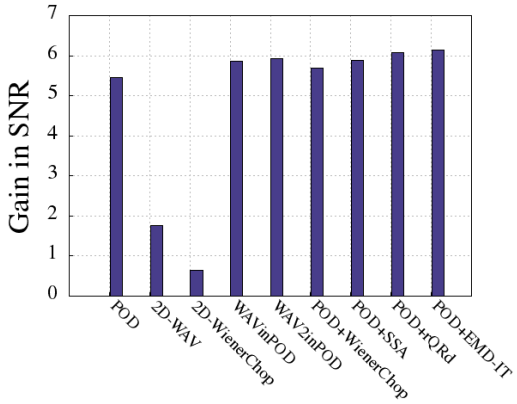


(c) Output of WAVinPOD.

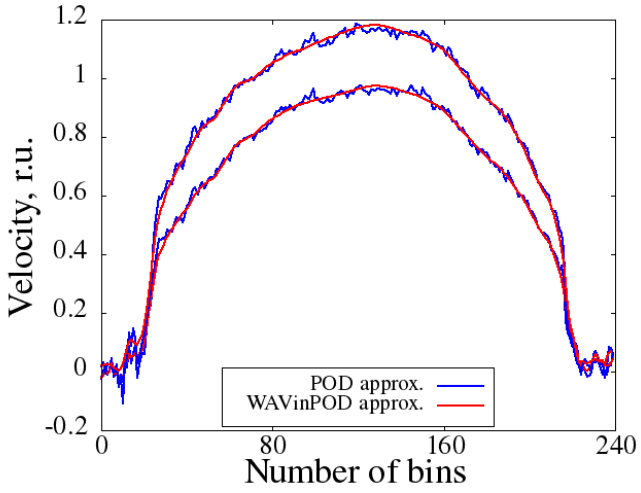
Figure 5.27: Results of applying WPOD, POD+, 2D wavelet thresholding and statistical averaging to an ensemble of $N = 24000$ velocity measurements from oscillating Poiseuille flow simulated with DPD.

from WT_1 for the first (most energetic) mode was negligible, and due to the choice of filters. Additional de-noising of temporal modes in this case appeared to be beneficial as the eigenvectors were strongly contaminated but long enough to preserve their orthogonality after processing. Similar to previous problems, 2D wavelet thresholding and WienerChop filter did not perform well due to the significant noise level. The results are summarised in Fig. 5.28(a), and comparison of approximations recovered with WPOD and WAVinPOD is given in Fig. 5.28(b).

It was mentioned above that DPD simulation results are expected to contain less



(a) Gain in SNRs for all the methods.



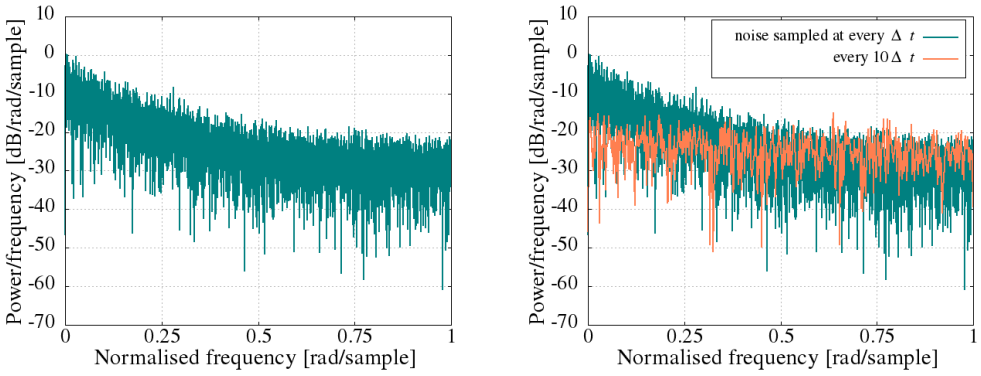
(b) WAVinPOD approximation contructed with *db*8 and 6 resolutions.

Figure 5.28: De-noising performance in processing $N = 4000$ time-steps of $M = 240$ velocity measurements with $\text{SNR}_{\text{noisy}} = 3.45$ dB from oscillating Poiseuille flow simulated with DPD; for SSA $L = 24$ and $k = 4$, and for rQRd $P = 8$ random vectors were employed.

strongly correlated noise. However, when the periodogram of the noise was plotted, obtained from subtracting WPOD's approximation from the original data, it was observed that the fluctuations were correlated. Figure 5.29(a) denotes the energy shift towards lower frequencies. Moreover, the Hurst index, briefly described in Sec. 3.5.2, was determined to be $H_{in} \approx 0.87$ following the procedure discussed by Di Matteo *et al.* [164]. This suggested that the noise added to each spatial bin was a time-series with a long-term positive autocorrelation, making it more difficult to separate from the desired trend for the smaller number of samples. In order to improve the analysis, the statistical inefficiency (introduced in Sec. 2.1.3) can be calculated for the noise;

as shown in Fig. 5.30(a), the *memory* of the time-series was $10\Delta t$ long, which means that if data was sampled every 10 time-steps, the noise should be de-correlated. Figure 5.29(b) compares the frequency distribution for the fluctuations sampled at every Δt and in intervals of $10\Delta t$. In the latter case, the noise appeared to be white, which was further confirmed by calculating the Hurst index, $H_{in} \approx 0.49$. Obviously, to be able to establish this relation it is required to know the true signal *a priori*. However, it can be determined by running a small test-case with the same parameters, before the larger simulation is performed, and will result in some computational savings. The statistical inefficiency can also be directly calculated from the transient simulation but only for an ensemble with N much smaller than the period of oscillations. Figure 5.31 shows the statistical inefficiency distribution computed from $N = 400$ noisy measurements; with this analysis the correlation of $s_{in} = 10$ was confirmed. Processing the data-set with WPOD and POD+ methods consisting of e.g. $N = 400$ measurements taken every $10\Delta t$, or simply setting $N_{ts} = 10$ instead of $N_{ts} = 1$, produced comparable de-noising performance and was about 100× faster. In addition, for the system with less correlated noise, 2D wavelet thresholding and WienerChop produced over 2× higher SNRs than for the data-set consisting of all time-steps. Changing the sampling approach can *whiten* the noise, but not necessarily remove its correlation with the signal which can further disturb the filtering process.

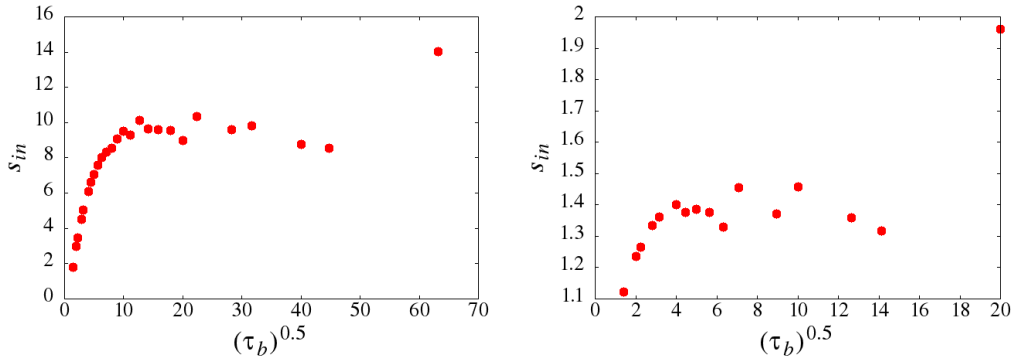
In order to show that WPOD and POD+ methods can be useful in analysing not only



(a) Periodogram for DPD noise sampled at every Δt.

(b) Comparison of distributions.

Figure 5.29: Periodograms of DPD noise for different sampling procuderes.



(a) Noise sampled at every time-step.

(b) Every 10th measurement.

Figure 5.30: The calculation of statistical inefficiency s_{in} with approach to the plateau for noise from DPD simulation of oscillating Poiseuille flow.

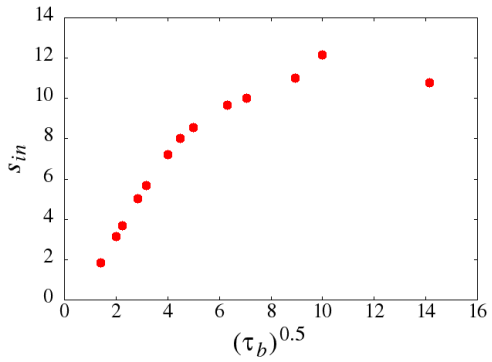


Figure 5.31: The calculation of statistical inefficiency s_{in} for a noisy DPD data-set with $N = 400$.

velocity data, we applied them to density fields from a simulation of phase separation phenomena performed with DPD. The studied system consisted of a periodic box filled with 3000 particles of 2 species (i.e. 1500 each). The particles from both species had the same sizes, and each bead had a mass = 1 in DPD units, but were set to repel each other in order to form two layers as shown in Fig. 5.32. We applied POD+ methods together with WPOD to density profiles obtained from 400 bins spanning the x-direction. The simulation was run for $20000\Delta t$, with $N_{ts} = 10$ and $N_{\mathrm{POD}} = 2000$, which means that averaging (the rolling mean) over 10 time-steps was performed. All the POD+ methods produced comparable results. For clarity, we discuss only wavelet-based approximations.

In Fig. 5.33(a), the noisy original profiles for each species are plotted against WPOD and WAVinPOD approximations. The profiles were extracted by retaining only $k = 2$ modes and using the *db*8 filter with 7 decompositions. Both de-noising techniques

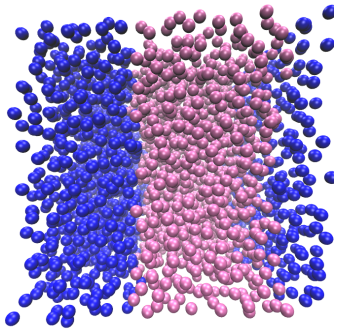
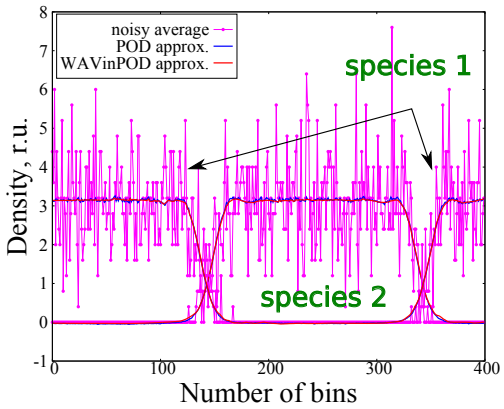
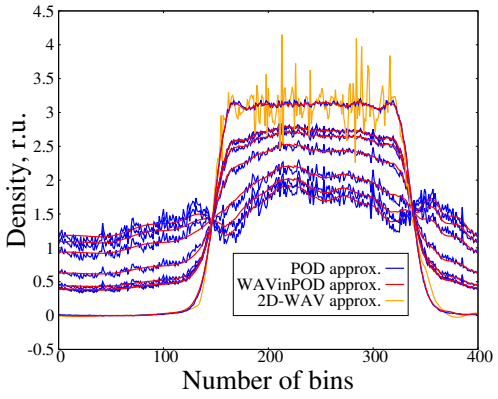


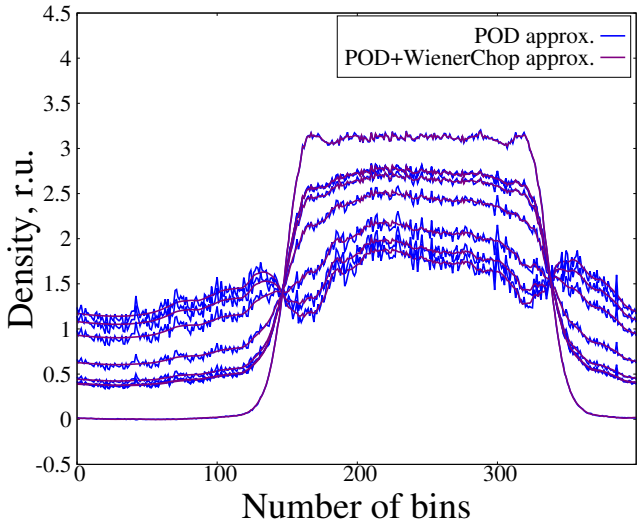
Figure 5.32: Snapshots of the last time-step of a DPD simulation of a phase separation phenomena.



(a) Approximations at the last Δt.



(b) Results obtained with WAVinPOD, 2D-WAV and WPOD.



(c) Density profiles extracted with POD+WienerChop.

Figure 5.33: Comparison of WAVinPOD, WPOD and 2D-WAV applied to the density distribution at different time-steps from a DPD simulation of a phase separation.

seemed to produce similar output. However, when the density profiles were plotted at different instances (see Fig. 5.33(b)), it was observed that WAVinPOD provided smoother estimates. Again, applying 2D wavelet thresholding alone resulted in poor filtering quality. Additional results obtained with POD+WienerChop with *db*4 and 8 resolutions for WT_1 are plotted in Fig. 5.33(c). It should be stressed that no additional averaging could have been performed without losing information on how the system was evolving.

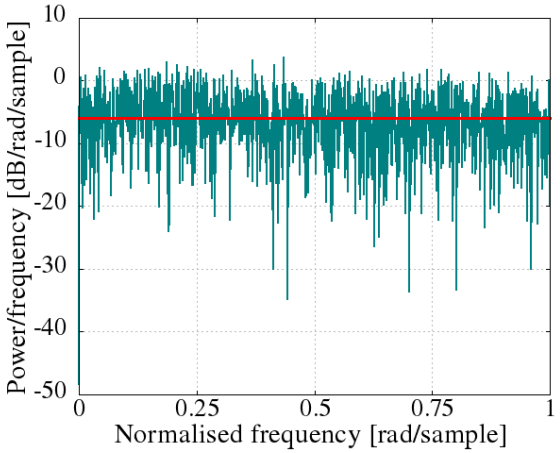


Figure 5.34: Periodogram power spectral density estimate of noise from the DPD simulation of phase separation.

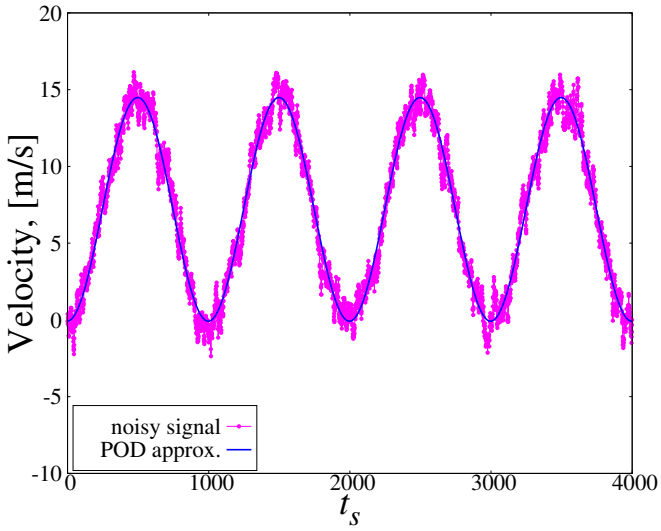
In the simulation of oscillating Poiseuille flow, we observed that the added noise was temporally correlated due to the small time-steps, but sampling every $10\Delta t$ flattened the power spectrum of the fluctuations. In the phase separation modelling, the correlations were relaxed by setting $N_{ts} = 10$. A periodogram of the system's fluctuations from one bin over time is presented in Fig. 5.34; the signal had a uniformly distributed power spectral density indicating that the density profiles were contaminated with white noise. Moreover, the Hurst index was computed to be $H_{in} \approx 0.5$, as expected from the random time-series. Both DPD simulations were thermostatted in the same manner, hence, as expected, the difference in types of noise could not be caused by temperature control. The unwanted noise correlation seemed to be present purely due to over-sampling.

5.2.3 Challenges in filtering DSMC measurements

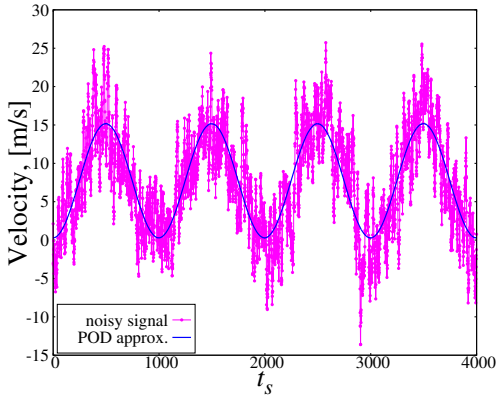
Direct simulation Monte Carlo is considered to be a very efficient algorithm for simulating rarefied gases. As with all particle-based schemes, DSMC requires statistical averaging to measure hydrodynamic values. One of the main challenges associated with DSMC is reducing the statistical scatter in the results. The statistical fluctuations in DSMC increase the amount of sampling that has to be done in order to extract accurate features. In addition, the number of simulator particles is a key parameter that affects the accuracy of DSMC results. When a DSMC simulation particle represents a large number of physical molecules, the variance of fluctuations is magnified relative to the ratio of coarse-graining [165]. For that reason we decided to analyse how WPOD and POD+ techniques would perform in processing data from simulations with different numbers of DSMC particles.

The test problem was a two-dimensional simulation of a gas flowing in a periodic domain with a time varying gravitational acceleration. It was performed with the OpenFOAM software. The flow domain was 532 nm $\times$ 532 nm, with periodic boundaries in the x-direction and specular walls at both y-direction extremities (which simply reflect the incident particles with the perpendicular component of velocity reversed). The working gas was hard sphere argon at a temperature of 292 K and a number density of 1.2564×1026 m^{-3}. An acceleration with an amplitude of 2.28589×10^{10} ms^{-2} and a period of 0.3183 ns was applied to the gas, and the instantaneous velocity profile was measured and recorded at the end of each time-step. The time-step was $\Delta t = 2$ ps and an ensemble of $N = 4000$ profiles consisting of $M = 50$ velocity measurements (collected in bins) was constructed. In order to recover results with different levels of statistical uncertainty, the parameter that controls how many real argon atoms each DSMC particle represented was varied to give different numbers of simulator particles in the domain. The smallest simulation had only 2104 particles, the medium 189170, and the largest (i.e. with a small statistical scatter) had 3783495 particles. For the latter two cases, WPOD recovered very smooth profiles with only one dominant EOF, $k = 1$; the approximations of the velocity varying with time in the middle bin for both systems are presented in Fig. 5.35(a) and Fig. 5.35(b).

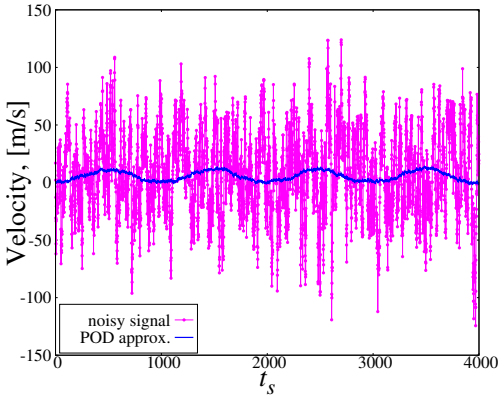
The spatial mode did not contain fluctuations that could be removed with POD+



(a) Large system, 3783495 DSMC particles.



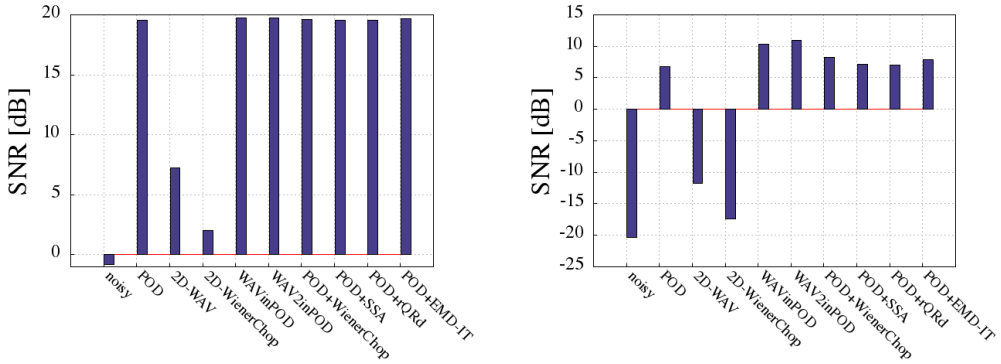
(b) Medium-size simulation, 189170 particles.



(c) Highly coarse-grained system, only 2104 particles.

Figure 5.35: Velocity measurements in a central bin, varying with time, recovered with WPOD from an DSMC ensemble of $N = 4000$ and $M = 50$; different coarse-graining levels were considered.

techniques, hence no enhancement was observed even when all the filtering procedures were applied to only one full cycle. Assuming that the result obtained from the large system was the desired solution, the SNR values were computed for the other, coarser systems and are summarised in Fig. 5.36(a) and Fig. 5.36(b). Even for the extreme case with a substantial noise level, WPOD managed to retain reasonably good velocity profiles as seen in Fig. 5.35(c), but with a different amplitude of oscillations due to the lack of information in the data-set. Additional filtering of spatial modes did not offer much improvement, as SVD extracted mostly low-frequencies contained in the signals. In such cases, applying additional de-noising to temporal modes can be beneficial.



(a) Medium-size simulation, 189170 particles. (b) System with only 2104 DSMC particles.

Figure 5.36: Values of SNRs for noisy data and approximations obtained with each filtering method; for larger number of particles, the data had $\mathrm{SNR}_{\mathrm{noisy}} = -0.78$ dB, and for the most coarse system, $\mathrm{SNR}_{\mathrm{noisy}} = -20.32$ dB.

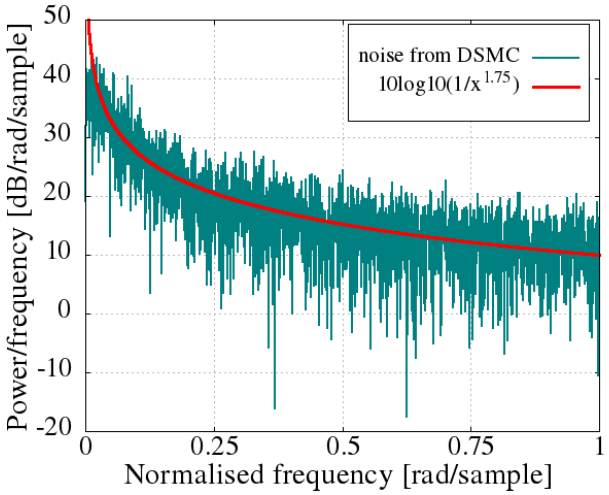


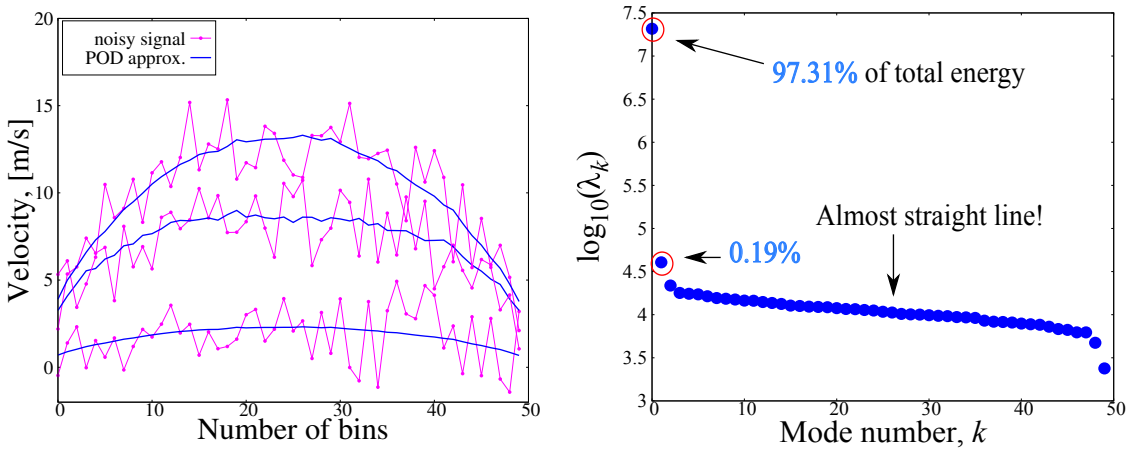
Figure 5.37: Periodogram power spectral density estimate of noise from DSMC oscillating gas flow.

To further explore the reason why filtering methods did not remove more of the unwanted fluctuations in the most coarse set-up, we decided to analyse the type of noise which was affecting DSMC data. We took the velocity measurements from the middle bin over time and subtracted them from the desired output obtained with WPOD for the system with a large number of particles (see Fig. 5.35(a)); the difference is the noise that we wanted to remove. Using MATLAB's periodogram function, an estimate of the power spectral density using a rectangular window was produced. Figure 5.37 indicates how the variance of the data was distributed over the frequency components. It can be seen that the power spectral density was inversely proportional to the frequency of the

signal, following closely the function $y = 1/x^{1.75}$. This observation showed that DSMC data was corrupted with fluctuations of the type $S(f) \propto 1/f^{\alpha}$ where f is a frequency and $0 < \alpha \leq 2$, with exponent $\alpha = 2$ describing the spectral density of Brownian noise (see Sec. 4.3). This means that more energy was concentrated in lower frequencies, making them very difficult to extract. Moreover, the Hurst index of the fluctuations also indicated the long-term memory of the series, $H_{in} \approx 0.93$ ($H_{in} = 1$ denotes a perfect positive correlation).

The correlation of noise contained in DSMC data might be caused by the collisions of particles. Although the system forces are considered as random, the exchange of the momentum gives rise to long hydrodynamic memory due to a particle's past motion. This phenomena translates to thermal forces, which display a coloured noise spectrum [166]. As discussed previously, removing coloured noise is still a challenging task; some information on the buried signal and the nature of noise, with which it can also be correlated, is needed to improve the filtering performance. In addition, with increasing levels of coarse-graining, the energy was spread more evenly among all the eigenvalues; for the coarsest system the first mode contained only 6% of the total variance. This suggests that there was not enough statistics or information in the provided data to extract the same structures as in the system with a large number of DSMC particles.

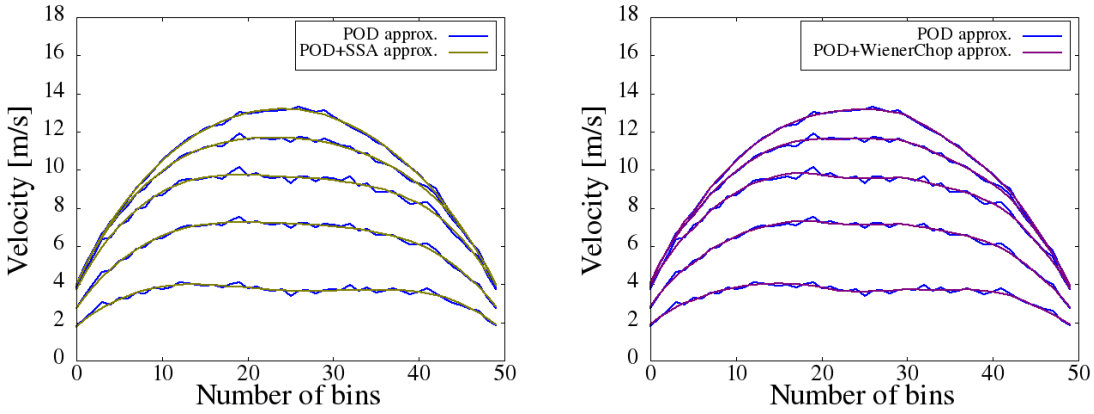
To further investigate the effect of using a different number of simulator particles per cell, another DSMC case was analysed. An oscillating flow of argon particles was simulated with a different DSMC code developed by John *et al.* [155, 167]. In this case, the variable hard sphere model was used. In this problem we applied diffuse reflection at the solid walls to produce a more parabolic velocity profile. The dimensions of the domain were 625 nm $\times$ 625 nm, divided into 50 bins. Gaseous argon at 273 K and atmospheric pressure was set to move with an acceleration of amplitude 5×10^9 ms^{-2} and an angular frequency of 1.5×10^{10} rads^{-1}. An ensemble of $N = 4000$ instantaneous velocity profiles was built with $\Delta t = 7.4$ ps. Two cases were considered: with 10000 particles per cell, and only 1000 DSMC particles per cell. For the more dense distribution, WPOD recovered a much improved signal with $k = 2$ as shown in Fig. 5.38(a). The number of dominant modes was suggested, among other criteria, through examination of the LEV diagram in Fig. 5.38(b). Additional smoothing was obtained



(a) Approximation obtained with WPOD, $k = 2$. (b) LEV plot.

Figure 5.38: Reconstruction of velocity profiles obtained with WPOD for simulation of argon channel flow; 10000 DSMC simulator particles per cell were used.

by utilising POD+ methods; the example of POD+SSA with $L = 25$ and 3 EOFs is compared with WPOD in Fig. 5.39(a), and POD+WienerChop output, obtained with *db*8 and 7 decompositions for WT_1, and *db*4 and 8 decompositions[3] for WT_2, is plotted in Fig. 5.39(b). When the number of particles per cell was decreased by a factor of 10,



(a) Output of POD+SSA. (b) De-noising with POD+WienerChop.

Figure 5.39: Additional smoothing provided with POD+SSA and POD+WienerChop for the measurements performed with DSMC (10000 simulator particles per cell); for SSA $L = 25$ and 3 EOFs were used, and for WienerChop, *db*8 and 7 decompositions for WT_1 and *db*4 and 8 resolutions for WT_2.

the noise level increased substantially and the small profile changes, that were present

[3]Higher decomposition than expected from the signal's length can be performed by extending the input vector through some extrapolation [168].

in the signal, were lost. Figure 5.40 compares LEV diagrams for both cases. The

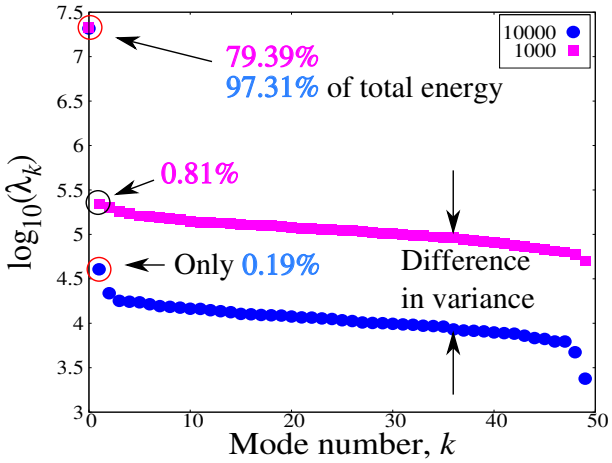
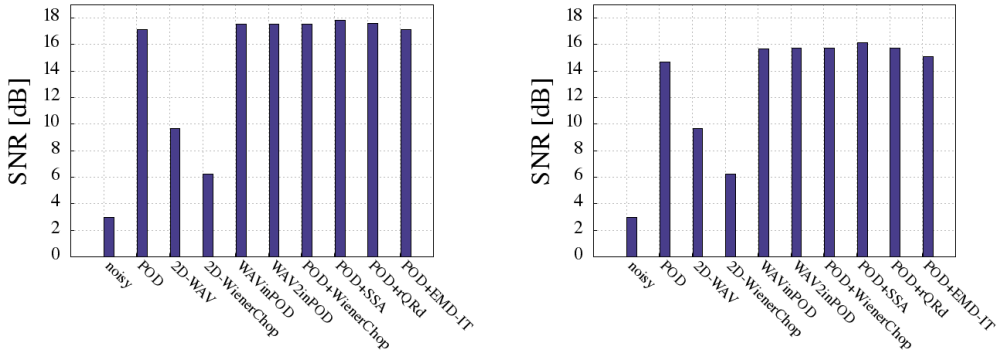


Figure 5.40: Comparison of LEV diagrams and energy distribution for two systems with 10000 (blue) and 1000 (purple) particles per bin.

first eigenvalue was not affected much, but the energy of the eigenspectrum was spread over the higher modes. As a result the information represented by the second mode was no longer distinguishable from the noise, which was either more energetic, or the statistics were not even present in the data. If we assume that WPOD's approximation from the previous case was the true solution, higher SNR values were recovered with WPOD and POD+ for $k = 1$ than for $k = 2$ modes (see Fig. 5.41). Matrices of rank



(a) Reconstruction with $k = 1$. (b) Approximations with $k = 2$.

Figure 5.41: Values of SNRs for noisy data and approximations obtained from DSMC simulation of oscillating argon flow with 1000 particles per cell and $\mathrm{SNR}_{\mathrm{noisy}} = 2.97$ dB. Note that better performance was achieved for reconstructions of rank $k = 1$; for wavelet thresholding different filters were used: $db6$ with 3 decompositions for WT_1 and $db5$ with 4 decompositions for WT_2.

$k = 2$ provided with POD+ methods had higher SNRs than WPOD because additional filtering separated some noise from the second mode. Different sets of filters were applied

for wavelet transforms in order to retain fewer oscillations in the signals. In conclusion, it is not possible to recover the same structures from coarse systems as from a densely populated DSMC domain. Applying WPOD or POD+ improves the data quality but cannot extract information that is not present, or is significantly less energetic than the noise.

5.3 Summary

All the de-noising techniques presented in this thesis have the potential to alleviate the problem of statistical noise in particle-based simulations, and consequently reduce the computational cost of multi-scale modelling. In the study of stationary data, we listed the wavelet-based methods as the most universal. In addition, we showed that very good results can be obtained with SSA for less noisy, dominant trends. The major challenge in processing data from particle-based simulations is ensuring that the unwanted correlations in the ensemble are relaxed. Although the filtering methods are capable of removing higher frequencies, the approximations obtained after only a few time-steps might not exactly match the trends extracted from large data-sets simply due to a lack of statistics.

A clear benefit was observed in the case of modelling non-stationary processes. Extracting useful information from the data required significantly smaller ensembles and less computing time when using WPOD and POD+ techniques than for standard averaging. We recommend employing the POD+ methods as they can only improve the quality of signals reconstructed with WPOD. Each POD+ procedure offers different benefits and it is difficult to label any of them as the best solution. If the processed simulation results consist of very long univariate signals, with low-rank underlying structure, the POD+rQRD (or urQRd) method should be employed. In cases where we expect less complex shapes to be recovered, such as parabolic profiles, POD+SSA can offer the best quality after a relatively straightforward analysis. The combination of POD and wavelet thresholding is the most flexible and can be very successful in capturing all the information contained in the measurements, unless we are not clear about what we want to see. Choosing a proper wavelet basis for the transform can be tricky as each decomposition permits the analysis of the data at different resolutions and smoothness

level. In contrast, wavelet thresholding is one of the fastest algorithms.

Applying the WienerChop filter in the SVD domain can offer additional improvement in signal-to-noise ratio relative to WAVinPOD, at the cost of computational complexity, but mostly for cases with random disturbances. It might be utilised when wavelet thresholding is not capable of removing all the unwanted frequencies and the noise level can be fairly well established from the wavelet coefficients. Noise reduction based on empirical mode decomposition was shown to be the easiest to employ; we used the method in an essentially parameter-free manner by fixing all the variables (including the number of sifting processes). However, it is the most expensive technique, and so not suitable for multi-scale modelling.

In addition to trend recovery, all the POD+ methods ensure low-dimensionality of the results, which reduces the storage requirements and complexity of performing other data-dependent tasks. Furthermore, the singular value decomposition (or EVD) can be used to determine whether the simulation has reached a steady-state by observing the energy shift towards lower modes. The general rule is that if there is a clear breaking point in the eigenspectra and the first few modes contain together about $70\% - 95\%$ of the total variance, there are enough statistics to extract the ensemble solution.

Chapter 6

Conclusions and Future Work

"One never notices what has been done; one can only see what remains to be done."

Maria Skłodowska-Curie, (1867-1934).

The main goal of this thesis was to review, develop, and evaluate new methodologies for solving statistical inverse problems in computational nanofluidics. The fluid flows under consideration were modelled separately via particle-based simulations using MD, its meso-scale counterpart DPD, and the DSMC method. The filtering tools were comprehensively studied and applied to a wide range of numerical results for both synthetic test-cases and real simulation data. Furthermore, novel procedures were proposed and shown to outperform the other techniques in extracting significant information from raw particle data. This research has therefore provided additional insight into the challenge of improving the communication in multi-scale modelling that couples a molecular domain to continuum fluid dynamics.

The thesis started with a brief introduction to particle-based simulation methods. This was followed by an overview of the mathematical procedures with the main focus on their ability to remove the errors from noisy numerical data. In the subsequent chapters, the de-noising performance of the algorithms was initially analysed with synthetically generated signals that were corrupted with Gaussian noise and then measurements obtained from particle flow simulations. At first, proper orthogonal decomposition was discussed, because it has been widely utilised in the fluid dynamics and turbulence

community to extract coherent structures. Also known under different names, including Karhunen-Loève decomposition or principal component analysis, POD provides an orthonormal basis for representing given multidimensional data in a least squares sense. This dimensionality reduction method has only recently been applied to molecular simulations; its extension employs time windows, enabling us to process sampled measurements more efficiently and to directly link the outputs with continuum solvers. In addition, even without any pre-averaging within the POD windows, the technique is shown to require substantially less computing time and memory than the standard calculation of an ensemble mean for sets (matrices) with time-dependent signals. However, it was stressed that the choice of significant modes used for the low-rank approximation of measurements is often quite challenging. Different criteria for selecting the number of eigentriples have been introduced. Particularly interesting is determination of the dominant modes through an optimal singular value hard threshold which, under certain conditions, can provide a strategy for automating the whole de-noising process. To the best of the author's knowledge, this is the first time that SVHT analysis has been utilised in filtering raw molecular data. One significant disadvantage of POD is its poor performance when applied to small data-sets or collections of stationary signals. In the latter case, POD extracts an approximation which is comparable in quality to the result of statistical averaging. This conclusion led to another technique, also based on truncated SVD or EVD factorisation, known as singular spectrum analysis.

It was shown that SSA is an alternative to POD in cases where only one signal is to be processed. To be able to apply SVD factorisation, a data array is transformed into a trajectory (e.g. Hankel) matrix; after truncation of the singular values, which can be done with the use of SVHT, the signal is reconstructed through diagonal averaging of the low-rank approximation of the data-set. The only parameter that needs to be considered *a priori* is the window length, L, determining the structure of the matrix, which is basically a collection of lagged vectors. Unfortunately, an inadequate choice of L can make the noise separation difficult to carry out, i.e. it can result in a data decomposition that is too fine or too coarse. Certain guidelines for defining the dimensions of the trajectory matrix were provided. However, the accurate selection of L is still an open research subject. Variations of the method were also listed, including a

two-dimensional extension. However, the application of 2D-SSA appeared to be computationally intensive, making it not very useful for the large data-sets obtained from real simulations.

A promising new method, referred to as random QR de-noising, has recently been introduced to tackle the issue of processing very long data arrays. This procedure draws from SSA the concept of building a matrix from one-dimensional data, but instead of SVD it utilises random sub-sampling of the system. In our analysis with synthetically generated signals, it was found that rQRd does not achieve as high signal-to-noise ratio as SSA, it does however offer more flexibility: for a relatively wide range of oversampling parameters, which define the size of the random space, comparably good approximations were obtained. In the case of SSA, an inadequate choice of EOFs generally resulted in unwanted artifacts, either missing information or added noise. A further improvement of the algorithm, called uncoiled random QR de-noising was also discussed, which further reduces the processing time of long signals by performing a fast matrix-vector multiplication based on FFT. However, it was shown that for *short* arrays, urQRd was not more efficient than the original algorithm, rQRd.

The best basis of SVD might be difficult to determine and manipulate numerically. Furthermore, in some practical applications, signals contain certain irregularities, and its these local features that are of particular interest. In such cases, transforming the signal to a different, pre-determined domain, e.g. the wavelet domain, can be more effective. For this reason, the benefit of thresholding the wavelet coefficients for noise cancellation was investigated. A wavelet transform is very fast, and is able to precisely locate the high frequency components and estimate the trend of the multi-dimensional data. The main difficulty comes from the selection of the appropriate basis for a particular application from a very rich class of discrete wavelet transforms. Depending on whether, for example, the regularity or frequency resolution is important, a different set of quadrature mirror filters should be applied. On the other hand, this can be regarded as an advantage, if the flexibility of the methodology is crucial. In the problems that were considered in this thesis, the filters were limited to Daubechies and, in the case of univariate data, Symlet wavelets, with a good *trade-off* between smoothness and computing time. There is therefore the potential to use the same

wavelet class for different simulation flow measurements with satisfactory results in the final SNRs. If required, filters can be custom-made, constructed with the desired properties for more complex and unique trends. The discussion on wavelet de-noising was extended by describing the empirical Wiener filter, WienerChop, which incorporates two wavelet transforms; the first transform, WT_1, produces estimates of the desired data and noise. The approximations are then used to design the filter, which de-noises the signal in the second wavelet domain, WT_2. It was shown that this method can outperform wavelet thresholding carried out with the same QMFs. However, in contrast to what was stated in the original published work, it was observed in this thesis that WienerChop is quite sensitive to the choice of filters for particular transforms. In addition, it is slower than wavelet thresholding as the entire signal is decomposed into the approximation and detail coefficients twice.

The last method explored in this research was empirical mode decomposition. Unlike the wavelet transform it is entirely data-adaptive, and decomposes the signal into a set of intrinsic mode functions with a decreasing number of zero-crossings. The natural oscillatory modes of the signal are represented by these IMFs, which serve as the basis determined by the signal itself, rather than pre-defined kernels. One of the shortcomings of the method lies in the fact that intrinsic functions are not strictly orthogonal. Moreover, EMD often suffers from mode-mixing, which can be improved by performing noise-assisted EMD but results in very intensive computations. In general, the algorithm lacks a strong theoretical foundation. Direct noise separation through partial reconstruction is troublesome, and there are currently no well-established criteria to define which IMFs contain noise and should therefore be eliminated. In this thesis, an interval thresholding EMD was utilised as an example of a noise reduction method based on analysis of IMFs. This was proposed in the literature and inspired by the concept of translation invariant wavelet de-noising with a universal threshold. Although this methodology was found to be the most computationally expensive out of all the techniques discussed in this thesis, EMD-IT is the least conditioned by user-defined parameters; this is a major benefit in making the procedure simple to implement for an application to data processing.

All the noise-reduction methods offer a balance between different properties. It is hard to say that one algorithm can always outperform others for any case. Rather than choosing a universal approach, this work aimed to give a comparative overview and guidelines on how to benefit from each procedure. Furthermore, to improve certain common weaknesses, e.g. computational intensity, all the techniques have been combined with POD. In our proposed POD+ methods, additional de-noising is performed on the SVD's dominant spectral modes to enable more efficient filtering of unwanted frequencies, which would not be possible with POD alone for the same number of observations. Each coupling provides different benefits; the common feature is that the POD+ approaches are fast, and more successful in recovering signals buried in noise, particularly white random fluctuations, than when the techniques are applied separately. For example, as mentioned before, SSA and EMD-IT are too computationally intensive to process large data-sets. Applying them in the SVD domain tackles that issue, at the same time resulting in higher SNRs than POD can achieve alone. For clarity, the strengths and weaknesses of all the methods investigated in this thesis are summarised in Table 6.1.

This thesis shown that applying sophisticated de-noising tools to particle-based simulations can reduce the computational time and memory required to obtain acceptable ensemble solutions. The main challenge is coloured noise, which can be correlated with itself or an underlying signal. Although the introduced techniques are not as efficient in removing such coherent fluctuations, they still manage to improve the quality of the final output, which is far more difficult to achieve using the standard averaging approach. It is recommended to employ POD+ methods as they offer more efficient de-noising than the other estimators we considered. If POD alone is capable of extracting smooth trends without any high frequencies, the POD+ processing will always return the same result; it will never produce lower SNRs.

Techniques	**Strengths**	**Weaknesses**
POD **WPOD**	+ Data-adaptive basis, + No parameters needed, + The most optimal approximation obtained for k.	- Large amount of data needed, - Computationally intensive for large cases, - Difficult determination of significant EOFs.
SSA	+ Data-adaptive basis, + Can be applied to 1D data, + Provides the most optimal solution in L_2 norm.	- Window size defined prior to processing, - Not applicable for large datasets, - Difficulties in determination of number k.
rQRd **urQRd**	+ Very fast in processing large matrices, + Higher flexibility in the choice of EOFs.	- Less optimal solution than SVD.
Wavelet thresholding	+ Can recover high SNRs, + Offers good resolution in frequency, + Fast procedure, + Applicable to large matrices and data arrays.	- *A priori* basis, - Conditioned by many parameters.

EMD-IT	+ Solely data-dependent, + Simple algorithm, + No parameters needed.	- Can cause mode mixing and costly iterative methods need to be applied, - Does not preserve sharp edges.
WienerChop	+ Retains higher SNR than WAV for strongly disturbed data, + Can be applied to signals and large matrices.	- Dependent on number of parameters, - Pre-determined basis, - Slower than 1D- or 2D-WAV.
POD+ methods		
WAVinPOD	+ Less dependent on the choice of basis, + Higher SNR than POD or WAV alone for additive white noise, + Preserves SVD's dimensionality reduction.	- Choice of the filter and number k, - Slower than WAV or POD.
POD+SSA	+ Allows for applying SSA (or MSSA) to larger data-sets, + Higher SNR than POD alone, + No *a priori* basis needed.	- Computationally more intensive than POD or wavelet thresholding, - Multiple determination of EOFs.

POD+ EMD-IT	+ Higher SNR than for WPOD for the same number of observations, + Enables multivariate EMD-based de-noising.	- The most expensive combination.
POD+ Wiener-Chop	+ Slightly enhanced SNR compared to WAVinPOD for matrices, + Recovers the highest SNR for studied problems.	- More expensive than WAVinPOD, - Basis is defined twice.

Table 6.1: Comparison of all the noise reduction techniques for improvement of the data quality.

Future work

This research has opened up a number of potential avenues for future investigation:

1. As mentioned before, the biggest challenge in processing simulation data is the treatment of energetic coherent noise. Our study showed that there is potential in utilising modified thresholding techniques, particularly within the wavelet domain. As the wavelet basis functions are well localised in time or space (in contrast to e.g. the Fourier transform), they are ideal candidates for analysing non-stationary signals suffering from discrepancies with long-memory. There is a clear need for more work to be focused on designing an orthogonal wavelet basis with useful properties in treating simulation data, or using other existing transforms, e.g. wavelet packet decomposition [169]. There is also scope for different thresholding approaches to be applied, e.g. modified level-dependent estimators [152].

2. Employing singular value hard thresholding in any SVD-based processing is an im-

portant step towards making the methodology fully adaptive and data-dependent. It was shown, though, that more research needs to be conducted for improving the determination of adequate thresholds in the presence of coloured noise. In this thesis, certain suggestions were given on how to enhance the computation of SVHT allowing it to adapt better to unknown rank in the case of correlated fluctuations in particle data. However, a sound analytical analysis is required to confirm that this solution is universally applicable for treatment of data affected by low-frequency noise.

3. Various methods of calibration that can be used in practice to improve the accuracy of reduced models based on SVD should be explored, e.g. Tikhonov regularisation [170].

4. An application of univariate EMD as a noise reduction tool is quite a recent concept and there is still a lot to learn about the method's capabilities. In future, it would be interesting to investigate whether the interval thresholding idea could easily be extended to multivariate versions of EMD that are currently of great interest to the signal processing community [171]. Different signal de-noising schemes based on EMD could also be tested.

5. There is an emerging topic in multi-scale simulations in which the compounding of information over disparate length and time scales is entirely performed with the use of wavelet-based techniques. While the use of multiresolution analysis has been mentioned in this context, there is still no rigorous methodology that would allow for efficient application of wavelets in a hybrid model. Surprisingly, limited attention has been given to the actual development of a wavelet-based tools for bridging different domains (apart from the noise reduction discussed in this thesis), and only a structure referred to as Compound Wavelet Matrix has been proposed in the literature [172, 173]. The general procedure decomposes the data from different modelling techniques into wavelet coefficients, establishes overlapping scales, copies and thresholds fine- and coarse-scale features, and, depending on the application, compares the statistics or performs an inverse transform. While the concept and some preliminary results are encouraging, a substantial valida-

tion needs to be carried out, and a number of shortcomings, some mentioned by Mirchandani and Evans [174], require investigation.

6. There is potential in utilising dynamic mode decomposition to extract important information from disturbed simulation data. Apart from ensuring orthogonality of the eigenvectors, an additional step could be to perform partial de-noising in the early stage of the singular value decomposition. Moreover, another method, referred to as independent component analysis [175], can be applied for separating a multivariate signal into additive components based on statistical independence. It would be interesting to analyse this approach to see if it can perform better than POD in removing noise from particle data.

We are currently living at a time when big data is everywhere, from smart cities to crowd and security control, through to large data sets generated by modelling and simulation. The common theme linking these disparate topics is how to remove unwanted information, i.e. noise. The methods introduced and compared in this thesis have the potential to make a significant impact in a wide range of subjects, and could prove critical for reconstructing key information within coupling techniques multi-scale modelling.

References

[1] R. L. Parker. *Geophysical Inverse Theory*. Princeton University Press, 1994.

[2] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*, volume 93. Springer Science & Business Media, 2012.

[3] Y. L. You, W. Xu, A. Tannenbaum, and M. Kaveh. Behavioral analysis of anisotropic diffusion in image processing. *IEEE Transactions on Image Processing*, 5(11):1539–1553, 1996.

[4] E. J. Candes and D. L. Donoho. Recovering edges in ill-posed inverse problems: Optimality of curvelet frames. *Annals of Statistics*, 30(3):784–842, 2002.

[5] A. V. Goncharskii, A. M. Cherepashchuk, and A. G. Iagola. Ill-posed problems of astrophysics. *Moscow Izdatel Nauka*, 1, 1985.

[6] J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13(49-52):28, 1902.

[7] P. C. Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, volume 4. SIAM, 1998.

[8] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. CRC Press, 1998.

[9] G. Karniadakis, A. Beskok, and N. Aluru. *Microflows and Nanoflows: Fundamentals and Simulation*. Springer Science & Business Media, 2006.

[10] A. Alexiadis and S. Kassinos. Molecular simulation of water in carbon nanotubes. *Chemical Reviews*, 108(12):5014–5034, 2008.

[11] W. D. Nicholls, M. K. Borg, D. A. Lockerby, and J. M. Reese. Water transport through (7, 7) carbon nanotubes of different lengths using molecular dynamics. *Microfluidics and Nanofluidics*, 12(1-4):257–264, 2012.

[12] K. M. Mohamed and A. A. Mohamad. A review of the development of hybrid atomistic–continuum methods for dense fluids. *Microfluidics and Nanofluidics*, 8 (3):283–302, 2010.

[13] S. T. O'Connell and P. A. Thompson. Molecular dynamics–continuum hybrid computations: a tool for studying complex fluid flows. *Physical Review E*, 52(6): R5792, 1995.

[14] D. A. Fedosov, B. Caswell, and G. E. Karniadakis. A multiscale red blood cell model with accurate mechanics, rheology, and dynamics. *Biophysical Journal*, 98 (10):2215–2225, 2010.

[15] S. S. Collis. Monitoring unresolved scales in multiscale turbulence modeling. *Physics of Fluids*, 13(6):1800–1806, 2001.

[16] A. F. Tuck. From molecules to meteorology via turbulent scale invariance. *Quarterly Journal of the Royal Meteorological Society*, 136(650):1125–1144, 2010.

[17] P. A Kollman, I. Massova, C. Reyes, B. Kuhn, S. Huo, L. Chong, M. Lee, T. Lee, Y. Duan, W. Wang, et al. Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models. *Accounts of Chemical Research*, 33(12):889–897, 2000.

[18] N. Asproulis, M. Kalweit, and D. Drikakis. A hybrid molecular continuum method using point wise coupling. *Advances in Engineering Software*, 46(1):85–92, 2012.

[19] N. Bellomo and C. Dogbe. On the modeling of traffic and crowds: A survey of models, speculations, and perspectives. *SIAM Review*, 53(3):409–463, 2011.

[20] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall Englewood Cliffs, 1999.

[21] S. V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, 2008.

[22] M. Dimian and P. Andrei. *Noise-Driven Phenomena in Hysteretic Systems*. Springer, 2014.

[23] N. J. Kasdin. Discrete simulation of colored noise and stochastic processes and $1/f^{\alpha}$ power law noise generation. *Proceedings of the IEEE*, 83(5):802–827, 1995.

[24] T. Sanghi and N. R. Aluru. Thermal noise in confined fluids. *The Journal of Chemical Physics*, 141(17):174707, 2014.

[25] L. Grinberg. Proper orthogonal decomposition of atomistic flow simulations. *Journal of Computational Physics*, 231(16):5542–5556, 2012.

[26] J. Habasaki, K. L. Ngai, and Y. Hiwatari. Time series analysis of ion dynamics in glassy ionic conductors obtained by a molecular dynamics simulation. *The Journal of Chemical Physics*, 122(5):054507, 2005.

[27] J. Habasaki and K. L. Ngai. Heterogeneous dynamics of ionic liquids from molecular dynamics simulations. *The Journal of Chemical Physics*, 129(19):194501, 2008.

[28] Z. Li, A. Borrmann, and C. C. Martens. Wavelet analysis of condensed phase molecular dynamics. *Chemical Physics Letters*, 214(3):362–366, 1993.

[29] A. C. To, Y. Fu, and W. K. Liu. Denoising methods for thermomechanical decomposition for quasi-equilibrium molecular dynamics simulations. *Computer Methods in Applied Mechanics and Engineering*, 200(23):1979–1992, 2011.

[30] Y. Kopsinis and S. McLaughlin. Development of EMD-based denoising methods inspired by wavelet thresholding. *IEEE Transactions on Signal Processing*, 57(4): 1351–1362, 2009.

[31] G. Rilling and P. Flandrin. One or two frequencies? The empirical mode decomposition answers. *IEEE Transactions on Signal Processing*, 56(1):85–95, 2008.

[32] R. P. Feynman. There's plenty of room at the bottom. *Engineering and Science*, 23(5):22–36, 1960.

[33] P. Gravesen, J. Branebjerg, and O. S. Jensen. Microfluidics-a review. *Journal of Micromechanics and Microengineering*, 3(4):168, 1993.

[34] M. P. Allen and D. J. Tildesley. Computer simulation of liquids. 1989.

[35] K. Binder, J. Horbach, W. Kob, W. Paul, and F. Varnik. Molecular dynamics simulations. *Journal of Physics: Condensed Matter*, 16(5):S429, 2004.

[36] D. C. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, 2004.

[37] M. K. Borg. *Hybrid Molecular-Continuum Modelling of Nano-Scale Flows*. PhD thesis, University of Strathclyde, 2011.

[38] B. J. Alder and T. E. Wainwright. Phase transition for a hard sphere system. *The Journal of Chemical Physics*, 27(5):1208–1209, 1957.

[39] W. D. Nicholls. *Molecular Dynamics Simulations of Liquid Flow in and around Carbon Nanotubes*. PhD thesis, University of Strathclyde, 2012.

[40] R. Friedberg and J. E. Cameron. Test of the Monte Carlo method: fast simulation of a small Ising lattice. *The Journal of Chemical Physics*, 52(12):6049–6058, 1970.

[41] P. J. Hoogerbrugge and J. M. V. A. Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *EPL (Europhysics Letters)*, 19(3):155, 1992.

[42] N. Goga, A. J. Rzepiela, A. H. de Vries, S. J. Marrink, and H. J. C. Berendsen. Efficient algorithms for Langevin and DPD dynamics. *Journal of Chemical Theory and Computation*, 8(10):3637–3649, 2012.

[43] P. Espanol and P. Warren. Statistical mechanics of dissipative particle dynamics. *EPL (Europhysics Letters)*, 30(4):191, 1995.

[44] P. Espanol. Dissipative particle dynamics. In *Handbook of Materials Modeling*, pages 2503–2512. Springer, 2005.

[45] A. Ghoufi, J. Emile, and P. Malfreyt. Recent advances in Many Body Dissipative Particles Dynamics simulations of liquid-vapor interfaces. *The European Physical Journal E*, 36(1):1–12, 2013.

[46] P. B. Warren. Vapour-liquid coexistence in many-body dissipative particle dynamics. *ArXiv version: cond-mat/0306027*, 2003.

[47] M. A. Seaton and W. Smith. DL MESO USER MANUAL. URL `http://www.stfc.ac.uk/SCD/resources/CSE/pdf/USRMAN.pdf`.

[48] M. A Seaton, R. L. Anderson, S. Metz, and W. Smith. DL MESO: highly scalable mesoscale simulations. *Molecular Simulation*, 39(10):796–821, 2013.

[49] A. A. Louis, P. G. Bolhuis, J. P. Hansen, and E. J. Meijer. Can polymer coils be modeled as "soft colloids"? *Physical Review Letters*, 85(12):2522–2525, 2000.

[50] C. Chen, L. Zhuang, X. Li, J. Dong, and J. Lu. A many-body dissipative particle dynamics study of forced water-oil displacement in capillary. *Langmuir*, 28(2): 1330–1336, 2011.

[51] I. Pagonabarraga and D. Frenkel. Dissipative particle dynamics for interacting systems. *The Journal of Chemical Physics*, 115:5015, 2001.

[52] S. Y. Trofimov, E. L. F. Nies, and M. A. J. Michels. Thermodynamic consistency in dissipative particle dynamics simulations of strongly nonideal liquids and liquid mixtures. *The Journal of Chemical Physics*, 117:9383, 2002.

[53] S. Y. Trofimov, E. L. F. Nies, and M. A. J. Michels. Constant-pressure simulations with dissipative particle dynamics. *The Journal of Chemical Physics*, 123:144102, 2005.

[54] A. Ghoufi and P. Malfreyt. Mesoscale modeling of the water liquid-vapor interface: A surface tension calculation. *Physical Review E*, 83(5):051601, 2011.

[55] P. B. Warren. No-go theorem in many-body dissipative particle dynamics. *Physical Review E*, 87:045303, 2013.

[56] T.-H. Lin, W.-P. Shih, C.-S. Chen, and Y.-T. Chiu. Simulation and analysis of interfacial wettability by dissipative particle dynamics. In *Nano/Micro Engineered and Molecular Systems, 2006. NEMS'06. 1st IEEE International Conference on*, pages 571–574. IEEE, 2006.

[57] C. Chen, C. Gao, L. Zhuang, X. Li, P. Wu, J. Dong, and J. Lu. A many-body dissipative particle dynamics study of spontaneous capillary imbibition and drainage. *Langmuir*, 26(12):9533–9538, 2010.

[58] Y. Zhao, Y. Yue, X. Zhang, S. Li, and A. Sajjanhar. Many-body dissipative particle dynamics simulation of wetting phenomena. *Frontiers of Chemical Engineering in China*, 4(3):280–282, 2010.

[59] J. Meng, L. Wu, J. M. Reese, and Y. Zhang. Assessment of the ellipsoidal-statistical Bhatnagar-Gross-Krook model for force-driven Poiseuille flows. *Journal of Computational Physics*, 251(0):383 – 395, 2013.

[60] G. A. Bird. Approach to translational equilibrium in a rigid sphere gas. *Physics of Fluids (1958-1988)*, 6(10):1518–1519, 1963.

[61] G. A. Bird. Molecular Gas Dynamics and the Direct Simulation of Gas Flows. 1994.

[62] F. J. Alexander and A. L. Garcia. The direct simulation Monte Carlo method. *Computers in Physics*, 11(6):588, 1997.

[63] A. Garcia and W. Wagner. Time step truncation error in direct simulation Monte Carlo. *Physics of Fluids*, 12(10):2621, 2000.

[64] N. G. Hadjiconstantinou. Analysis of discretization in the direct simulation Monte Carlo. *Physics of Fluids*, 12(10):2634–2638, 2000.

[65] R. C. Palharini. *Atmospheric Reentry Modelling Using an Open-Source DSMC Code*. PhD thesis, University of Strathclyde, 2014.

[66] K. C. Kannenberg. *Computational methods for the direct simulation Monte Carlo technique with application to plume impingement*. PhD thesis, Cornell University, 1998.

[67] N. G. Hadjiconstantinou, A. L. Garcia, M. Z. Bazant, and G. He. Statistical error in particle simulations of hydrodynamic phenomena. *Journal of Computational Physics*, 187(1):274–297, 2003.

[68] J. L. Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178, 1967.

[69] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

[70] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[71] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[72] K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37. 1947. 3–79 pp.

[73] M. Loève. Fonctions Aléatoires du Second Ordre. *Jacques Gabay*, 1948. Supplement to P. Levy, Processus Stochastic et Mouvement Brownien.

[74] J.-C. Pinoli. *Mathematical Foundations of Image Processing and Analysis*. John Wiley & Sons, 2014.

[75] D. D. Kosambi. Statistics in function space. *Journal of Indian Mathematical Society*, 7(1):76–88, 1943.

[76] A. Alvera-Azcárate, A. Barth, M. Rixen, and J.-M. Beckers. Reconstruction of incomplete oceanographic data sets using empirical orthogonal functions: application to the adriatic sea surface temperature. *Ocean Modelling*, 9(4):325–346, 2005.

[77] E. N. Lorenz. Empirical orthogonal functions and statistical weather prediction. *Statistical Forecatsing Project*, 1956.

[78] C. Spearman. " General intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904.

[79] R. P. McDonald. *Factor Analysis and Related Methods*. Psychology Press, 2014.

[80] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1998.

[81] P. Berkooz, G.and Holmes and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993.

[82] G. Kerschen, J.-C. Golinval, A. F. Vakakis, and L. A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear Dynamics*, 41(1-3):147–169, 2005.

[83] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.

[84] P. C. Hansen. Numerical tools for analysis and solution of fredholm integral equations of the first kind. *Inverse Problems*, 8(6):849, 1992.

[85] L. Sirovich. Turbulence and the dynamics of coherent structures. I-Coherent structures. II-Symmetries and transformations. III-Dynamics and scaling. *Quarterly of Applied Mathematics*, 45:561–571, 1987.

[86] M. A. Rakha. On the Moore-Penrose generalized inverse matrix. *Applied Mathematics and Computation*, 158(1):185–200, 2004.

[87] D. Kalman. A singularly valuable decomposition: the SVD of a matrix. *The College Mathematics Journal*, 27(1):2–23, 1996.

[88] E. S. Baker and R. D. DeGroat. Evaluating EVD and SVD errors in signal processing environments. In *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, volume 2, pages 1027–1032. IEEE, 1998.

[89] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[90] H. Chen, D. L. Reuss, and V. Sick. On the use and interpretation of proper orthogonal decomposition of in-cylinder engine flows. *Measurement Science and Technology*, 23(8):085302, 2012.

[91] H. F. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 1960.

[92] I. T. Jolliffe. Discarding variables in a principal component analysis. I: Artificial data. *Applied Statistics*, pages 160–173, 1972.

[93] R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966.

[94] D. L. Donoho and M. Gavish. The Optimal Hard Threshold for Singular Values is 4/sqrt (3). *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[95] J. M. Craddock and C. R. Flood. Eigenvectors for representing the 500 mb geopotential surface over the Northern Hemisphere. *Quarterly Journal of the Royal Meteorological Society*, 95(405):576–593, 1969.

[96] D. S. Wilks. *Statistical Methods in the Atmospheric Sciences*, volume 100. Academic Press, 2011.

[97] R. Renaut and S. Zhu. Application of Fredholm integral equations inverse theory to the radial basis function approximation problem. Technical report, University of Oxford, Mathematical Institute, 2012.

[98] S. Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2014.

[99] R. Cangelosi and A. Goriely. Component retention in principal component analysis with application to cDNA microarray data. *Biology Direct*, 2(2):1–21, 2007.

[100] D. S. Broomhead and G. P. King. Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, 20(2):217–236, 1986.

[101] D. S. Broomhead and G. P. King. On the qualitative analysis of experimental dynamical systems. *Nonlinear Phenomena and Chaos*, 113:114, 1986.

[102] J. Gillard. Cadzow's basic algorithm, alternating projections and singular spectrum analysis. *Statistics and Its Interface*, 3(3):333–343, 2010.

[103] J. A. Cadzow. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(1):49–62, 1988.

[104] N. Golyandina, V. Nekrutkin, and A. Zhigljavsky. *Analysis of Time Series Structure: SSA and Related Techniques*. CRC Press, 2010.

[105] N. Golyandina and A. Zhigljavsky. *Singular Spectrum Analysis for Time Series*. Springer, 2013.

[106] M. Ghil, M. R. Allen, M. D. Dettinger, K. Ide, D. Kondrashov, M. E. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, et al. Advanced spectral methods for climatic time series. *Reviews of Geophysics*, 40(1):3–1, 2002.

[107] M. R. Allen and L. A. Smith. Monte Carlo SSA: Detecting irregular oscillations in the presence of colored noise. *Journal of Climate*, 9(12):3373–3404, 1996.

[108] N. Golyandina. On the choice of parameters in singular spectrum analysis and related subspace-based methods. *ArXiv version:1005.4374*, 2010.

[109] H. D. Simon. The Lanczos algorithm with partial reorthogonalization. *Mathematics of Computation*, 42(165):115–142, 1984.

[110] G. Beliakov. On fast matrix-vector multiplication with a hankel matrix in multiprecision arithmetics. *ArXiv version:1402.5287*, 2014.

[111] L. Chiron, M. A. van Agthoven, B. Kieffer, C. Rolando, and M.-A. Delsuc. Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry. *Proceedings of the National Academy of Sciences*, 111(4):1385–1390, 2014.

[112] A. Korobeynikov. Computation- and Space-Efficient Implementation of SSA. *ArXiv version:0911.4498v2*, 2010.

[113] N. Golyandina, A. Korobeynikov, A. Shlemov, and K. Usevich. Multivariate and 2D Extensions of Singular Spectrum Analysis with the Rssa Package. *Journal of Statistical Software*, 2014.

[114] H. Hassani. Singular spectrum analysis: methodology and comparison. *Journal of Data Science*, 5(2):239–257, 2007.

[115] F. J. Alonso, J. M. Del Castillo, and P. Pintado. Application of singular spectrum analysis to the smoothing of raw kinematic signals. *Journal of Biomechanics*, 38 (5):1085–1092, 2005.

[116] R. Mahmoudvand and M. Zokaei. On the singular values of the Hankel matrix with application in singular spectrum analysis. *Chilean Journal of Statistics*, 3 (1):43–56, 2012.

[117] M. R. Allen and L. A. Smith. Optimal filtering in singular spectrum analysis. *Physics Letters A*, 234(6):419–428, 1997.

[118] D. Danilov and A. Zhigljavsky. Principal components of time series: the "Caterpillar" method. *St. Petersburg: University of St. Petersburg*, pages 1–307, 1997. in Russian.

[119] N. E. Golyandina and K. D. Usevich. 2D-extension of Singular Spectrum Analysis: algorithm and elements of theory. *Matrix Methods: Theory, Algorithms, Applications World Scientific*, pages 449–473, 2010.

[120] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250. ACM, 2001.

[121] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[122] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized

algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.

[123] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[124] B. B. Hubbard. *The World According to Wavelets The Story of a Mathematical Technique in the Making.* Universities Press, 1998.

[125] M. Farge and K. Schneider. Analysing and computing turbulent flows using wavelets. In *New Trends in Turbulence Turbulence: nouveaux aspects*, volume 74, pages 449–504. Springer, 2001.

[126] S. G. Mallat. *A wavelet Tour of Signal Processing.* Academic Press, 1999.

[127] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*, volume 87. Prentice Hall PTR Englewood Cliffs, New Jersey, 1995.

[128] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[129] Y. Meyer. Orthonormal wavelets. In *Wavelets*, pages 21–37. Springer, 1989.

[130] I. Daubechies et al. *Ten Lectures on Wavelets*, volume 61. SIAM, 1992.

[131] R. Polikar. The wavelet tutorial. 1996. URL `http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html`.

[132] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[133] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, volume 2. MIT press Cambridge, MA, 1949.

[134] S. P. Ghael, A. M. Sayeed, and R. G. Baraniuk. Improved wavelet denoising via empirical Wiener filtering. In *Optical Science, Engineering and Instrumentation'97*, pages 389–399. International Society for Optics and Photonics, 1997.

[135] H. Choi and R. Baraniuk. Analysis of wavelet-domain Wiener filters. In *Time-Frequency and Time-Scale Analysis, 1998. Proceedings of the IEEE-SP International Symposium on*, pages 613–616. IEEE, 1998.

[136] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998.

[137] Z. Wu and N. E. Huang. A study of the characteristics of white noise using the empirical mode decomposition method. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2046): 1597–1611, 2004.

[138] P. Flandrin, G. Rilling, and P. Goncalves. Empirical mode decomposition as a filter bank. *Signal Processing Letters*, 11(2):112–114, 2004.

[139] P. Flandrin, P. Gonçalves, G. Rilling, et al. EMD equivalent filter banks, from interpretation to applications. *Hilbert-Huang Transform and its Applications*, pages 57–74, 2005.

[140] Z. Wu and N. E. Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in Adaptive Data Analysis*, 1(01):1–41, 2009.

[141] Z. Wu and N. E. Huang. On the filtering properties of the empirical mode decomposition. *Advances in Adaptive Data Analysis*, 2(04):397–414, 2010.

[142] R. R. Coifman and D. L. Donoho. *Translation-Invariant De-noising*. Springer, 1995.

[143] B. R. Bakshi. Multiscale PCA with application to multivariate statistical process monitoring. *AIChE Journal*, 44(7):1596–1610, 1998.

[144] M. Aminghafari, N. Cheze, and J.-M. Poggi. Multivariate denoising using wavelets and principal component analysis. *Computational Statistics & Data Analysis*, 50 (9):2381–2398, 2006.

[145] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.

[146] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[147] G. Tissot, L. Cordier, N. Benard, and B. R. Noack. Model reduction using Dynamic Mode Decomposition. *Comptes Rendus Mécanique*, 342(6):410–416, 2014.

[148] F. Guéniat, L. Mathelin, and L. R. Pastur. A dynamic mode decomposition approach for large and arbitrarily sampled systems. *Physics of Fluids*, 27(2): 025113, 2015.

[149] D. Duke, J. Soria, and D. Honnery. An error analysis of the dynamic mode decomposition. *Experiments in Fluids*, 52(2):529–542, 2012.

[150] T. J. Ross. *Fuzzy Logic with Engineering Applications*. John Wiley & Sons, 2009.

[151] D. Achlioptas and F. Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2):9, 2007.

[152] I. M Johnstone and B. W. Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(2):319–351, 1997.

[153] J. L. F. Abascal and C. Vega. A general purpose model for the condensed phases of water: TIP4P/2005. *The Journal of Chemical Physics*, 123(23):234505, 2005.

[154] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *Journal of Chemical Physics*, 107 (11):4423, 1997.

[155] B. John, X.-J. Gu, and D. R. Emerson. Investigation of heat and mass transfer in a lid-driven cavity under nonequilibrium flow conditions. *Numerical Heat Transfer, Part B: Fundamentals*, 58(5):287–303, 2010.

[156] F. D. Sofos, T. E. Karakasidis, and A. Liakopoulos. Effects of wall roughness on flow in nanochannels. *Physical Review E*, 79(2):026305, 2009.

[157] K. Gotoh. Liquid structure and Lennard-Jones (6, 12) pair potential parameters. *Nature*, 239(96):154–156, 1972.

[158] A. I. Khinchin. *Mathematical Foundations of Statistical Mechanics*. Courier Corporation, 1949.

[159] S. D. Stoyanov and R. D. Groot. From molecular dynamics to hydrodynamics: A novel Galilean invariant thermostat. *The Journal of Chemical Physics*, 122(11): 114112, 2005.

[160] A. W. Lees and S. F. Edwards. The computer study of transport processes under extreme conditions. *Journal of Physics C: Solid State Physics*, 5(15):1921, 1972.

[161] L. Grinberg, V. Morozov, D. Fedosov, J. A Insley, M. E. Papka, K. Kumaran, and G. E. Karniadakis. A new computational paradigm in multiscale simulations: Application to brain blood flow. In *High Performance Computing, Networking, Storage and Analysis (SC), 2011 International Conference for*, pages 1–12. IEEE, 2011.

[162] N. V. Priezjev. Effect of surface roughness on rate-dependent slip in simple fluids. *The Journal of Chemical Physics*, 127(14):144708, 2007.

[163] N. V. Priezjev. Molecular dynamics simulations of oscillatory Couette flows with slip boundary conditions. *Microfluidics and Nanofluidics*, 14(1-2):225–233, 2013.

[164] T. Di Matteo, T. Aste, and M. M. Dacorogna. Long-term memories of developed and emerging markets: Using the scaling analysis to characterize their stage of development. *Journal of Banking & Finance*, 29(4):827–851, 2005.

[165] A. L. Garcia. Direct Simulation Monte Carlo: Theory, Methods, and Open Challenges. Technical report, DTIC Document, 2011.

[166] T. Franosch, M. Grimm, M. Belushkin, F. M. Mor, G. Foffi, L. Forró, and S. Jeney. Resonances arising from hydrodynamic memory in Brownian motion. *Nature*, 478 (7367):85–88, 2011.

[167] B. John, X.-J. Gu, and D. R. Emerson. Nonequilibrium gaseous heat transfer in pressure-driven plane poiseuille flow. *Physical Review E*, 88(1):013018, 2013.

[168] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. SIAM, 1996.

[169] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1186–1191, 1993.

[170] C. W. Groetsch. *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*. Pitman, 1984.

[171] N. Rehman and D. P. Mandic. Multivariate empirical mode decomposition. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 466, pages 1291–1302. The Royal Society, 2009.

[172] G. Frantziskonis and P. A. Deymier. Wavelet methods for analysing and bridging simulations at complementary scales-the compound wavelet matrix and application to microstructure evolution. *Modelling and Simulation in Materials Science and Engineering*, 8(5):649, 2000.

[173] K. Muralidharan, S. K. Mishra, G. Frantziskonis, P. A. Deymier, P. Nukala, S. Simunovic, and S. Pannala. Dynamic compound wavelet matrix method for multiphysics and multiscale problems. *Physical Review E*, 77(2):026714, 2008.

[174] G. Mirchandani and J. T. Evans. Multiresolution in Multiscale: A New Role for Wavelets. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, pages 669–674. IEEE, 2009.

[175] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

Appendix A

Example of Computing SVD

Singular value decomposition breaks down a rectangular $N \times M$ matrix A into the product of three matrices: an orthogonal matrix U, a diagonal matrix Σ, and the transpose of an orthogonal matrix V. For small data sets a manual computation can be easily performed. The procedure for determining $A = U\Sigma V^{\dagger}$ is summarised as follows:

1. Depending on the matrix size, compute $A^{\dagger}A$ if $N > M$, or $AA^{\dagger}$ otherwise.
2. Determine the singular values, $s_j = \sqrt{\lambda_j},\ j = 1, \ldots, M$, and corresponding singular vectors v_j by finding the eigenvalues and orthonormalised eigenvectors of $A^{\dagger}A$.
3. Compute the first M columns of U via $u_j = s_j^{-1} A v_j,\ j = 1, \ldots, M$.
4. The remaining columns of U are chosen such that U is unitary.

The example presented here explains how SVD of a small 3×2 matrix can be computed.

- *Step 1*: In order to find the SVD for

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{bmatrix}, \tag{A.1}$$

we need to first establish

$$A^\dagger A = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 8 \\ 8 & 9 \end{bmatrix}. \tag{A.2}$$

- *Step 2*: To compute the eigenvalues, we begin with solving the eigenvalue problem

$$\begin{aligned} A^\dagger A V &= \Lambda V, && \text{(A.3)} \\ A^\dagger A V - \Lambda V &= 0, && \text{(A.4)} \\ \left(A^\dagger A - \Lambda I\right) V &= 0, && \text{(A.5)} \end{aligned}$$

where I is an identity matrix. As the determinant $\left|A^\dagger A - \Lambda I\right|$ has to be zero, the following is obtained

$$\left|A^\dagger A - \Lambda I\right| = \left| \begin{pmatrix} 9 & 8 \\ 8 & 9 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right| = \begin{vmatrix} 9-\lambda & 8 \\ 8 & 9-\lambda \end{vmatrix} = 0. \tag{A.6}$$

Deriving the formula for the determinant

$$\begin{aligned} \begin{vmatrix} 9-\lambda & 8 \\ 8 & 9-\lambda \end{vmatrix} = (9-\lambda)(9-\lambda) - 8 \cdot 8 = 81 - 2 \cdot 9 \cdot \lambda + \lambda^2 - 64 &= 0 && \text{(A.7)} \\ \lambda^2 - 18\lambda + 17 &= 0 && \text{(A.8)} \end{aligned}$$

Solving the quadratic equation leads to the eigenvalues (in decreasing order) $\lambda_1 = 17$ and $\lambda_2 = 1$, and singular values $s_1 = \sqrt{\lambda_1} = \sqrt{17}$ and $s_2 = \sqrt{\lambda_2} = \sqrt{1} = 1$. Thus,

$$\Sigma = \begin{bmatrix} \sqrt{17} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{A.9}$$

To find the eigenvectors we substitute a general vector $v_1 = \begin{bmatrix} v_1(1) \\ v_1(2) \end{bmatrix}$ and λ_1 into

the Eq. A.3:

$$\begin{bmatrix} 9 & 8 \\ 8 & 9 \end{bmatrix} \begin{bmatrix} v_1(1) \\ v_1(2) \end{bmatrix} = 17 \begin{bmatrix} v_1(1) \\ v_1(2) \end{bmatrix}. \tag{A.10}$$

The calculation leads to unsolvable equations:

$$\begin{cases} -8v_1(1) + 8v_1(2) &= 0, \\ 8v_1(1) - 8v_1(2) &= 0. \end{cases} \tag{A.11}$$

However, based on Eq. A.11, a relationship between two vectors can be established

$$8v_1(1) = 8v_1(2) \Rightarrow v_1(1) = v_1(2), \tag{A.12}$$

$$V = \begin{bmatrix} v_1(1) \\ v_1(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{A.13}$$

In general, it is common to set $v_1(1) = 1$. Following the same procedure for the second eigenvalue, $\lambda_2 = 1$, the second vector can be constructed $v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. This leads to

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} v_1(1) & v_2(1) \\ v_1(2) & v_2(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{A.14}$$

After normalisation (dividing the vector by its length), the 2×2 matrix, V, takes the form

$$V = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}. \tag{A.15}$$

- *Step 3*: The first two columns of U can be computed as

$$u_1 = s_1^{-1} A v_1 = \frac{1}{\sqrt{17}} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{A.16}$$

$$= \frac{1}{\sqrt{34}} \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}, \tag{A.17}$$

and

$$u_2 = s_2^{-1} A v_2 = \frac{1}{\sqrt{1}} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \tag{A.18}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}. \tag{A.19}$$

- *Step 4*: In order to determine the 3rd vector we have

$$U = \begin{bmatrix} \frac{3}{\sqrt{34}} & \frac{-1}{\sqrt{2}} & u_3(1) \\ \frac{4}{\sqrt{34}} & 0 & u_3(2) \\ \frac{3}{\sqrt{34}} & \frac{1}{\sqrt{2}} & u_3(3) \end{bmatrix}, \tag{A.20}$$

the condition $u_j^\dagger u_3 = \delta_{j3},\ j = 1, 2, 3$ needs to be satisfied. The following choice fulfils the requirement:

$$u_3 = \frac{1}{\sqrt{17}} \begin{bmatrix} 2 \\ -3 \\ 2 \end{bmatrix}. \tag{A.21}$$

The final result is given by

$$A = U\Sigma V^\dagger = \begin{bmatrix} \frac{3}{\sqrt{34}} & \frac{-1}{\sqrt{2}} & \frac{2}{\sqrt{17}} \\ \frac{4}{\sqrt{34}} & 0 & \frac{-3}{\sqrt{17}} \\ \frac{3}{\sqrt{34}} & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{17}} \end{bmatrix} \begin{bmatrix} \sqrt{17} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}. \tag{A.22}$$

It can be noticed that in Eq. A.22, due to the last row in the diagonal matrix having only zeros, the last column of the U matrix, u_3, is for this case redundant. The reduced SVD is expressed as

$$A = U_r \Sigma_r V^\dagger = \begin{bmatrix} \frac{3}{\sqrt{34}} & \frac{-1}{\sqrt{2}} \\ \frac{4}{\sqrt{34}} & 0 \\ \frac{3}{\sqrt{34}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{17} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}. \tag{A.23}$$

Appendix B

Calculation of SSA

Having a signal $X = [0, 0.5, 1, 1.5, 2, 2, 2, 2, 2.5, 2.5, 3, 3]$ and window length $L = 4$, the following steps are performed:

- *Step 1*: Embedding - build a trajectory matrix with L rows and $K = M - L + 1 = 12 - 4 + 1 = 9$ columns

$$Y = \begin{bmatrix} 0.00 & 0.50 & 1.00 & 1.50 & 2.00 & 2.00 & 2.00 & 2.00 & 2.50 \\ 0.50 & 1.00 & 1.50 & 2.00 & 2.00 & 2.00 & 2.00 & 2.50 & 2.50 \\ 1.00 & 1.50 & 2.00 & 2.00 & 2.00 & 2.00 & 2.50 & 2.50 & 3.00 \\ 1.50 & 2.00 & 2.00 & 2.00 & 2.00 & 2.50 & 2.50 & 3.00 & 3.00 \end{bmatrix} \tag{B.1}$$

- *Step 2*: Decomposition - perform economical SVD

$$\text{SVD}(Y) = U_{(L\times K)}\Sigma_{(K\times K)}V^{\dagger}_{(M\times K)}, \tag{B.2}$$

where

$$U_{(L\times K)} = \begin{bmatrix} -0.4117 & 0.7572 & -0.1342 & -0.4890 \\ -0.4682 & 0.2502 & -0.1741 & 0.8294 \\ -0.5290 & -0.1861 & 0.8251 & -0.0693 \\ -0.5757 & -0.5739 & -0.5205 & -0.2611 \end{bmatrix}, \tag{B.3}$$

$$\Sigma_{(K\times K)} = \begin{bmatrix} 12.0365 & 0 & 0 & 0 \\ 0 & 1.4181 & 0 & 0 \\ 0 & 0 & 0.4552 & 0 \\ 0 & 0 & 0 & 0.3942 \end{bmatrix}, \tag{B.4}$$

$$V^{\dagger}_{(M\times K)} = \begin{bmatrix} -0.1351 & -0.6501 & -0.0939 & -0.1175 \\ -0.2176 & -0.5629 & -0.0980 & -0.1049 \\ -0.2761 & -0.2733 & 0.4696 & 0.2389 \\ -0.3127 & 0.0819 & 0.1310 & 0.6707 \\ -0.3298 & 0.3488 & -0.0165 & 0.0503 \\ -0.3537 & 0.1465 & -0.5882 & -0.2809 \\ -0.3757 & 0.0808 & 0.3181 & -0.3688 \\ -0.4190 & -0.0333 & -0.4449 & 0.3520 \\ -0.4581 & 0.1680 & 0.3139 & -0.3562 \end{bmatrix}. \tag{B.5}$$

- *Step 3*: Grouping - determine k-dimensional subspace for $k = 1$

$$\begin{aligned} \widetilde{Y} &= U_{(L\times k)}\Sigma_{(k\times k)}V^{\dagger}_{(M\times k)} = \\ &= \begin{bmatrix} 0.6697 & 1.0782 & 1.3683 & 1.5494 & 1.6341 & 1.7527 & 1.8615 & 2.0764 & 2.2701 \\ 0.7616 & 1.2263 & 1.5561 & 1.7621 & 1.8585 & 1.9933 & 2.1171 & 2.3615 & 2.5817 \\ 0.8605 & 1.3854 & 1.7580 & 1.9907 & 2.0996 & 2.2519 & 2.3918 & 2.6679 & 2.9167 \\ 0.9365 & 1.5078 & 1.9134 & 2.1667 & 2.2852 & 2.4509 & 2.6032 & 2.9037 & 3.1745 \end{bmatrix}. \end{aligned} \tag{B.6}$$

- *Step 4*: Reconstruction - average over the diagonals, e.g. $(1.0782 + 0.7616)/2 = 0.9199$. New signal is

$$\widetilde{X} = [0.6697, 0.9199, 1.1517, 1.3568, 1.6655, 1.8788, 2.0303, 2.1827, 2.3686, 2.6176, 2.9102, 3.1745]. \tag{B.7}$$