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

Citation for published version:
Zimon, MJ, Prosser, R, Emerson, DR, Borg, M, David, B, Grinberg, L & Reese, J 2016, 'An evaluation of noise reduction algorithms for particle-based fluid simulations in multi-scale applications' Journal of Computational Physics, vol. 325, pp. 380–394. DOI: 10.1016/j.jcp.2016.08.021

Digital Object Identifier (DOI):
10.1016/j.jcp.2016.08.021

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Journal of Computational Physics

Journal of Computational Physics 325 (2016) 380–394

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

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

M.J. Zimoń [a,b,*], R. Prosser [a], D.R. Emerson [b], M.K. Borg [c], D.J. Bray [b], L. Grinberg [d], J.M. Reese [c]

[a] School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Manchester M13 9PL, UK
[b] Scientific Computing Department, STFC Daresbury Laboratory, Warrington WA4 4AD, UK
[c] School of Engineering, The University of Edinburgh, Edinburgh EH9 3FB, UK
[d] IBM T.J. Watson Research Center, 1 Rogers St, Cambridge, MA 02142, United States

ARTICLE INFO

Article history:
Received 18 May 2016
Accepted 21 August 2016
Available online 26 August 2016

Keywords:
Noise reduction
Molecular dynamics
Windowed proper orthogonal decomposition
Wavelet thresholding
Singular spectrum analysis
Empirical mode decomposition

ABSTRACT

Filtering of particle-based simulation data can lead to reduced computational costs and enable more efficient information transfer in multi-scale modelling. This paper compares the effectiveness of various signal processing methods to reduce numerical noise and capture the structures of nano-flow systems. In addition, a novel combination of these algorithms is introduced, showing the potential of hybrid strategies to improve further the de-noising performance for time-dependent measurements. The methods were tested on velocity and density fields, obtained from simulations performed with molecular dynamics and dissipative particle dynamics. Comparisons between the algorithms are given in terms of performance, quality of the results and sensitivity to the choice of input parameters. The results provide useful insights on strategies for the analysis of particle-based data and the reduction of computational costs in obtaining ensemble solutions.

1. Introduction

Numerical simulation is an essential tool 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) [1], Monte Carlo methods, e.g. direct simulation Monte Carlo (DSMC) [2] or dissipative particle dynamics (DPD) [3]; a comprehensive summary of all the modelling strategies can be found in Karniadakis et al. [4]. These procedures can be used to resolve accurately the dynamics at atomistic, meso- and micro-scales and are widely used to simulate nano/micro flows confined in channels such as carbon nanotubes [5,6]. In addition, information obtained from molecular simulations forms the basis of new and emerging hybrid multi-scale modelling methods for physical and biological applications (see [7] for a review). Examples demonstrating the ubiquity of multi-scale, multi-physics applications include the dynamics of complex fluid flows [8], the classical turbulence problem [9], meteorological predictions [10], chemical and biological reactions [11]. Moreover, there is significant potential to apply multi-scale techniques to sociological problems, such as crowd and traffic flow [12].

[*] Corresponding author at: Scientific Computing Department, STFC Daresbury Laboratory, Warrington WA4 4AD, UK. Current address: IBM Research, Daresbury Laboratory, Warrington WA4 4AD, UK.
E-mail address: malgorzata.zimon@uk.ibm.com (M.J. Zimoń).

http://dx.doi.org/10.1016/j.jcp.2016.08.021

The central problems with all particle-based and multi-scale modelling are the computational cost of the simulations and the accurate measurement and transfer of information across disparate length and time scales; there currently exist many sources of uncertainty and noise disturbing this intra-scale transfer, with an associated loss of simulation fidelity. Circumventing this problem often requires large samples and long averaging periods, resulting in computationally expensive calculations.

The objective of this paper is to investigate the capabilities of a number of mathematical algorithms introduced in the literature to assist noise reduction in particle-based modelling. A number of benchmark fluid flow problems, performed with molecular dynamics and dissipative particle dynamics, are used to investigate the usefulness of the considered methods and provide guidelines on how the algorithms can be successfully applied. The main focus of this paper is on novel procedures that provide rapid, adaptive, noise-free coarse-graining of micro-scale phenomena, and can further be employed in molecular-continuum simulations.

In this paper, new algorithms are proposed that combine the strengths of proper orthogonal decomposition (POD) with other techniques, hereafter referred to as POD+ methods, in order to achieve better efficiency in processing time-dependent fields. This work directly tackles the important challenge of extracting information from the data without significant additional computational cost.

The paper is organised as follows: the basic theory for the methods is described in Sec. 2. A comparison of the performance of each technique in de-noising particle-based simulations is presented in Sec. 3, followed by remarks and recommendations for the use of the methods under investigation.

2. Theoretical background

In the following section we briefly review the numerical methods employed. First, we discuss algorithms based on singular value decomposition (or eigenvalue decomposition) and QR decomposition. The second part of the review focuses on strategies using wavelet transforms, wavelet thresholding and the WienerChop filter [13]. We also discuss the application of empirical mode decomposition to noise reduction. Novel couplings of proper orthogonal decomposition to these algorithms are introduced at the end of the Section.

2.1. Noise filtering with singular value decomposition and QR factorisation

2.1.1. Proper orthogonal decomposition

Define an element $A(\tau^s, x)$ of the real $N \times M$ matrix as a measurement from the x-th probe taken at the τ^s-th time instant. Proper orthogonal decomposition can be done either by eigenvalue decomposition (EVD) of the symmetric matrix $C = AA^\dagger$ ($A^\dagger A$ if $N > M$),[1] or by singular value decomposition of A:

$$A = U \Sigma V^\dagger, \tag{1}$$

where, in the 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 diagonal entries of Σ, called singular values, are the square roots of eigenvalues, $s_n = \sqrt{\lambda_n}$ for $s_1 \geq s_2 \geq \ldots \geq s_n \geq 0$, where λ_n is the n-th eigenvalue of the diagonal matrix.

If A is a collection of measurements corrupted by additive noise, it can be represented in the form $A = A_t + B$, where A_t is a matrix that contains the *true* signals, and B denotes the noise. Given the decomposition in Eq. (1), the rank-k approximation of A can be written in vector form as

$$A_k(\tau^s, x) = \sum_{n=1}^{k} s_n u_n v_n^\dagger, \tag{2}$$

where $1 \leq k \leq min(N, M)$, u_n and v_n are the orthonormal (temporal or spatial) POD modes corresponding to the n-th columns of the matrices U and V, respectively. The key property of POD is that A_k is optimal in the sense that $min\left\{\|A_t - A_k\|_F^2\right\} = \sum_{n=k+1}^{min(N,M)} s_n^2$, where $\|.\|_F$ is the Frobenius norm. In this paper, the rank k is referred to as the number of *dominant* modes. A practical implementation of POD based on time-windows (WPOD) described by Grinberg [14], can be used for particle-based simulations.

The main challenge in estimation of A_t is the choice of the truncation parameter. One possible approach for defining the rank k is to select a cumulative percentage of total variation which modes should contribute. Unfortunately, such cut-off is often insufficient, as important aspects of the observables can be present in the direction of low variance modes. Therefore, in addition to studying the energy content of eigenvalues, most of the reported data is analysed here with the log-eigenvalue diagram (LEV), $\log_{10}(\lambda_n = s_n^2)$, based on the assumption that if higher singular vectors represent uncorrelated noise, then the corresponding λ_n should decay exponentially with increasing n [15]. We also considered the smoothness of the temporal

[1] The superscript † indicates matrix transpose since $A \in \mathbb{R}$.

modes [14] as the ordering of eigenvalues also corresponds to an ordering of the vectors u_n (for $A(\tau^s, x)$) from low to high frequency. In some cases, the choice of k is verified with a recently proposed singular value hard threshold (SVHT) [16]; the principal algorithm was evaluated based on the asymptotic mean square error for matrix sizes larger than the underlying rank.

2.1.2. Singular spectrum analysis (SSA)

While proper orthogonal decomposition can successfully extract the signal from its noise-contaminated measurements, when applied to data that is temporally (or spatially) steady, it appears to be no more efficient in separating unwanted components from the mean ensemble than statistical averaging. In such cases, the SSA of Broomhead and King [17,18] can be applied. The method has been widely used in the filtering and forecasting of climatic, meteorological, geophysical, and electrical data series.

The basic scheme of SSA is described in [19]. The algorithm consists of four main steps (two for the decomposition stage, and two for the reconstruction): *embedding*, SVD, *eigentriple grouping*, and *diagonal averaging*. Instead of the matrix $A \in \mathbb{R}^{N\times M}$ we consider one of the vectors forming its column, $X \in \mathbb{R}^M$. In the embedding stage, the series is broken into a sequence of *lagged vectors* of size L by sliding $X_i = (x_i, \dots, x_{i+L-1})^\dagger$, $(1 \le i \le K)$, where $K = M - L + 1$. As a result, a *trajectory* matrix $Y \in \mathbb{R}^{L\times K}$ of series X is formed:

$$Y = [X_1, \dots, X_K] = (x_{ij})_{i,j=1}^{L,K}. \tag{3}$$

In the second key step of the decomposition process, the trajectory matrix is subject to SVD forming a collection of elementary matrices, *eigentriples*, $Y = s_1 u_1 v_1^\dagger + \dots + s_r u_r v_r^\dagger$, where r is the rank of Y. In the following step, a low-dimensional approximation, $\widetilde{Y} = \sum_{n\in I_n} s_n u_n v_n^\dagger$ is formed, where I_n denotes *grouped* subsets, e.g. $I_n = \{2, \dots, 5, 10\}$. For simplicity, in this work, SSA approximation is formed for $I_n = \{1, \dots, k\}$, similar to POD analysis. The reconstructed matrix does not exhibit the same elements along its descending diagonals. The subsequent averaging over its anti-diagonals replaces these differing entries and yields a new series $\widetilde{X} = (\tilde{x}_1, \dots, \tilde{x}_M)$ with reduced noise.

The window length, L, is the only parameter to be determined prior to the SSA. Its choice can result in a weaker (or stronger) separability between information and noise, influencing the effectiveness of the signal reconstruction process. There is no universal rule regarding the optimal value of L, but several principles have been described by Golyandina and Zhigljavsky [19]. In general, smaller windows lengths are recommended for detailed trends, while relatively large values of L are preferred for simple profiles.[2] In addition, the window should not be greater than half of the length of the analysed series, $L \le M/2$ [20]. If there is a known periodic component in the processed data, then L should be chosen to be proportional to that period. For long signals and large values of L, the method is computationally very intensive.

2.1.3. Random QR de-noising (u/rQRd)

Random sampling has received considerable attention as an alternative dimensionality reduction tool which is significantly less expensive than SVD or EVD [21–23]. In the recent paper by Chiron et al. [24], a new method based on random sampling has been described which, similar to SSA, seeks a low-rank approximation of the trajectory matrix.

In random QR de-noising, a matrix Y_Ω, containing most of the significant information of the trajectory matrix $Y \in \mathbb{R}^{L\times K}$, is obtained by calculating the product of Y and a set of random vectors stored in a matrix $\Omega \in \mathbb{R}^{K\times Z}$:

$$Y_{\Omega(L\times Z)} = Y_{(L\times K)} \times \Omega_{(K\times Z)}. \tag{4}$$

As the number of vectors $Z \le L$, and $L \le K$, the matrix Y_Ω is smaller in size than the trajectory matrix. A factorisation $Y_\Omega = QR$ is performed in order to construct a projection of Y onto the reduced rank orthonormal basis Q, $\widetilde{Y} = QQ^\dagger Y$, where $\widetilde{Y}$ has a rank equal to Z. A de-noised time series $\widetilde{X}$ is then obtained by diagonal averaging. An approximation error for such a procedure satisfies $\|Y - \widetilde{Y}\|_2 \le \left[1 + 9\sqrt{Z}\cdot\sqrt{L}\right] s_{k+1}$, with probability of at least $(1 - 3\zeta^{-\zeta})$, where $\zeta = Z - k$ is the oversampling parameter, s_{k+1} is the $(k+1)$ greatest singular value of Y [22], and $\|.\|_2$ denotes the L_2 operator norm. The recommended action is to have a small ζ; however, in real situations, the number of components k is unknown *a priori*. The solution given by random QR de-noising, rQRd, may be less optimal than the approximation constructed with SVD, but is obtained much faster, particularly for large data-sets [24]. An improved algorithm for very long signals, called uncoiled random QR de-noising (urQRd) [24], based on fast Fourier transforms (FFT) is also used in this article.

2.2. Signal processing with wavelet transforms

The orthogonal wavelet transform uses a shifted and scaled *mother wavelet*, ψ, to form a basis (*children*) of $L^2(\mathbb{R})$ defined as

[2] By notion of *simplicity* we describe the trend that is concentrated in a narrow subdomain of low frequencies, or that X is well approximated by a series with finite and small rank k.

$$\psi_{j,m}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t - m2^j}{2^j}\right), \tag{5}$$

where $j, m \in \mathbb{Z}$ represent resolution and translation, respectively, and ψ satisfies the *admissibility condition*, $C_\psi = \int_0^{+\infty} \frac{|\widehat{\psi}(\omega)|^2}{\omega} d\omega < +\infty$, where $\widehat{\psi}(\omega) = \int_{-\infty}^{+\infty} \psi(t)e^{-i\omega t}dt$ [25,26]. To ensure that the integral is finite, $\widehat{\psi}(0) = \int_{-\infty}^{+\infty} \psi(t)dt = 0$. The wavelet is continuously differentiable, i.e. ψ has sufficient time decay, which implies smoothness, $\int_{-\infty}^{+\infty} (1 + |t|)\,|\psi(t)|\,dt < +\infty$. By enforcing the vanishing of higher order moments, $\int_{-\infty}^{+\infty} t^b\psi(t)dt = 0$, polynomials up to degree b are reproduced. A complement in $L^2(\mathbb{R})$ of the linear space spanned by the wavelets is determined by the translations and dilations of a scaling function, $\phi_{j,m}(t) = \frac{1}{\sqrt{2^j}}\phi\left(\frac{t-m2^j}{2^j}\right)$. Decomposition of a signal $f(t)$ can be written as

$$f(t) = \sum_{m\in\mathbb{Z}} c_{j_0,m}\phi_{j_0,m}(t) + \sum_{j,m\in\mathbb{Z}} d_{j,m}\psi_{j,m}(t), \tag{6}$$

where $c_{j_0,m} = \langle f, \phi_{j_0,m}\rangle$ is a smoothed (coarse) approximation, $d_{j,m} = \langle f, \psi_{j,m}\rangle$ are the fine scale details at different resolutions $-J \le j < 0$ and $0 \le m < 2^{-j}$, where J is the maximum number of decompositions and $j \le j_0$.

Through *multiresolution theory*, Mallat [27] and Meyer [28] established a link between an orthogonal wavelet basis and the conjugate mirror filters that describe the interaction between wavelet and scaling function. This allows for the fast implementation of a discrete wavelet transform (DWT), where the signal is convolved with the low-pass and high-pass filters, followed by a dyadic decimation, after which the procedure is recursively repeated on the decimated low-pass output.

To overcome the lack of translation-invariance of the DWT, the stationary wavelet transform (SWT) was introduced [29]. In this strategy, an undecimated transform is performed with upsampled filter coefficients at each level of the decomposition. The scheme contains redundancy which is seen as advantageous in de-noising, but increases the computational complexity from $O(2^J)$ to $O(J2^J)$ operations. We note that the SWT is closely related to cycle-spinning [30].

2.2.1. Wavelet thresholding

Motivated by the idea that a wavelet transform provides a sparse representation of data, *estimate functions* are obtained by inverting thresholded coefficients whose magnitude exceed an estimated noise level. Donoho and Johnstone [31] have proposed simple procedures for recovering information from noisy data, namely hard and soft thresholding. A non-linear hard threshold estimator is defined by $T_H(d_{j,m}) = d_{j,m}\mathbb{1}_{\{|d_{j,m}|>T_u\}}$, while soft thresholding (wavelet shrinkage) is given as $T_S(d_{j,m}) = \mathrm{sgn}(d_{j,m})(|d_{j,m}| - T_u)\mathbb{1}_{\{|d_{j,m}|>T_u\}}$, where $\mathbb{1}$ and sgn are the indicator and sign functions, respectively, and T_u is a threshold value. In this paper, to avoid over-smoothing [25], the first approach is applied together with a universal threshold:

$$T_u = \sigma_n\sqrt{2\log(M)}. \tag{7}$$

The white noise level estimate is defined as $\sigma_n = MAD/0.6745$, with *MAD* being the median absolute deviation of the fine scale wavelet coefficients [32]. As most of the simulation results reported later are corrupted by correlated noise, a level-dependent variance estimation introduced by Johnstone and Silverman [33] is used in our paper.

2.2.2. WienerChop

Wiener [34] formulated an optimal estimation analysis of time series which has been used in a range of applications, such as signal detection and noise reduction [35]. However, the main practical problem in the implementation of a Wiener filter is that a desired signal needs to be known *a priori*. Ghael et al. [13] proposed a straightforward estimate of the signal and noise by using wavelet transforms. Wavelet-based empirical Wiener filtering (referred to as WienerShrink or WienerChop) performs two wavelet transforms in order to estimate the filter parameters and perform de-noising. In the wavelet domain, a noisy signal $f(t) = f_t(t) + f_n(t)$ is defined as a set of coefficients $f_w = f_{t_w} + f_{n_w}$, consisting of an underlying true structure, $f_{t_w} = WT(f_t(t))$, and noisy details, $f_{n_w} = WT(f_n(t))$, where WT denotes a wavelet transform. The goal is to estimate the true signal wavelet coefficients, f_{t_w}, from the noisy observation, f_w. An approximation, $\tilde{f}_{t_{w_1}}$, of the signal's coefficients is obtained in the wavelet space by thresholding the wavelet coefficients, $T_H(f_{w_1}) = T_H(WT_1(f(t)))$. The noise level, σ_n, is calculated from the finest details [32], and an inverse transform is applied to the data, $\tilde{f}_t(t) = IWT_1(\tilde{f}_{t_{w_1}})$. 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:

$$W_{\text{Chop}} = \frac{\tilde{f}_{t_{w_{21}}}}{\tilde{f}_{t_{w_{21}}} + \sigma_n^2}. \tag{8}$$

In Eq. (8), the subscript w_{21} indicates that WT_2 was applied to the signal estimate obtained from WT_1, $\tilde{f}_{t_{w_{21}}} = WT_2(\tilde{f}_t(t))$. The WT_2 of the original signal, $f_{w_2} = WT_2(f(t))$, allows to filter the coefficients with WienerChop. After de-noising, the inverse transform is applied to recover the final estimation, $\tilde{\tilde{f}}_t(t)$. The rationale behind applying Wiener filtering to wavelet coefficients arises from the fact that the wavelet transform decorrelates data [36]. Assuming perfect decorrelation of noisy coefficients, the filter is optimal in the sense of minimising the mean squared error.

2.3. Empirical mode decomposition

Huang et al. [37] pioneered a nonlinear technique, referred to as empirical mode decomposition, for adaptively decomposing an unsteady signal into a finite sum of zero-mean amplitude modulated and frequency modulated elements, called *intrinsic mode functions* (IMFs). The method is adaptive, with the basis of the decomposition purely data-driven. Empirical mode decomposition looks at the evolution of a signal, $f(t)$, between two consecutive extrema, e.g. two minima occurring at t_-^1 and t_-^2, and defines a local high-frequency part (detail), $h(t)$, and a low-frequency element (trend), $r(t)$ [38,39], i.e. $f(t) = r(t) + h(t)$ for $t_-^1 \le t \le t_-^2$. The algorithm of EMD can be summarised in the following steps. *Step 1*: Identify all minima (t_-) and maxima (t_+) of $f(t)$. *Step 2*: Obtain the envelopes, $e_{\min}(t)$ and $e_{\max}(t)$, by interpolating between t_- and t_+, respectively. *Step 3*: Compute the mean of the two envelopes, $m_1^1(t) = (e_{\min}(t) + e_{\max}(t))/2$. *Step 4*: Extract the detail, $h_1^1(t) = f(t) - m_1^1(t)$. Examine whether the residual, $h_1^1(t)$, satisfies the definition of IMF according to a stopping criterion. If not, repeat n_s-times *Step 2* → *Step 5* until the conditions are met (*sifting* process). Then: The first IMF is found, $\mathrm{IMF}_1 = h_1^{n_s}(t)$. *Step 6*: Iterate on the residual, $f(t) - \mathrm{IMF}_1 = r_1$.

The necessary conditions for existence of intrinsic mode functions (in *Step 5*) 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. Flandrin et al. [40] proposed that the average number of zero-crossings in a mode is a meaningful way of defining its mean frequency. The original signal can be recovered by $f(t) = \sum_i^I \mathrm{IMF}_i(t) + r_I(t)$. Multi-dimensional versions of the empirical mode decomposition is still an active research area [41,42].

In this work, a wavelet-inspired thresholding of IMFs proposed by Kopsinis and McLaughlin [43] is used. In the procedure referred to as EMD interval thresholding (EMD-IT), the intervals between zero-crossings in each function are analysed to determine whether they are noise- or signal-dominant, based on the single extrema that corresponds to this interval. In the paper, multiples of the universal threshold are applied, i.e. $T_{\mathrm{multi}} \times T_u$ (see Eq. 7 in [43]) with the noise estimation obtained from the energy of the first IMF.

2.4. POD+

We propose an improvement in processing non-stationary fields by combining POD with all the algorithms described in this paper. Filtering procedures, e.g. wavelet thresholding, are applied within the domain of SVD to decrease the number of observations required for clean data recovery, while preserving dimensionality reduction. These hybrid strategies are called POD+ techniques, such as POD + SSA; POD + wavelet thresholding is referred to as WAVinPOD (or SWAVinPOD if based on SWT) and has been first introduced by Zimoń et al. [44]. In cases where $N \ll M$, the left singular vectors arising in the POD+ techniques are left unchanged to not weaken the orthogonality property of the shorter modes. However, for less *thin* matrices, modifying both sets of vectors can add to the performance. For large data sizes, the results obtained with POD and POD+ methods converge and improved filtering can be obtained by processing temporal and spatial modes [44]; this is shown in Fig. 1, where error in L_2 norm, $\delta_2 = \|A_t - A_k\|_2 \,/\, \|A_t\|_2$ of different approximations is plotted for synthetically generated signals.

The following steps summarise all POD+ procedures. *Step 1*: Perform SVD on the matrix of data A. *Step 2*: Define the number k and set $s_n = 0$ for $n > k$. *Step 3*: Perform additional filtering with e.g. wavelet thresholding of the retained k modes. *Step 4*: Reconstruct the de-noised data according to Eq. (1). The methods can be applied during simulations run by following the strategy described by Grinberg [14].

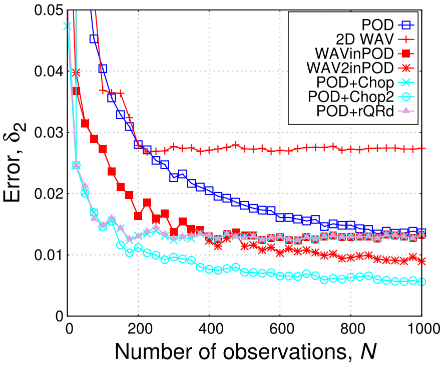


Fig. 1. Reconstruction error of POD and selected POD+ methods for increasing $N = 1, \ldots, 1000$. Signals of length $M = 1000$ were corrupted with white noise; the original data-set was generated as $A_t(\tau^s, x) = v_1(\cos(2\pi\tau^s/N)) + v_2(\cos(0.5\pi\tau^s/N))$, where $v_1 = \cos(4\pi x/M)$, $v_2 = 0.25\cos(20\pi x/M)$, $x = 1, \ldots, M$, and $\tau^s = 1, \ldots, N$. Wavelet thresholding and WienerChop filtering (WT_1) were performed with *sym2* for and $J = 6$ levels of decomposition and, in POD + Chop, additional *sym4* and $J = 6 + 1$ were used in WT_2. Parameters for rQRd were: $Z = 20$ random vectors and window length $L = 100$. Note that for larger N performing filtering on both singular vectors, in POD + Chop2 and WAV2inPOD, produced the best results.

3. Simulation results

In this section, we report the results of applying the filtering methods to data from several typical particle-based flow problems. The aim is to investigate the benefits of using de-noising techniques for filtering simulation measurements. Performance of POD+ is compared with POD for extracting information from molecular simulations [14] and, where applicable, with widely used statistical averaging. Unless specified, the following input parameters were used for the filtering methods; the oversampling coefficient for the rQRd procedure was kept constant, $\zeta = Z - k = 2$, while the number of dominant modes, k, for the SSA and POD analysis was established based on the tests discussed in Sec. 2.1.1. De-noising with empirical mode decomposition was performed with $n_s = 7$ sifting procedures and the hard thresholding approach. Following Donoho and Johnstone [31], wavelet de-noising was performed using the Symlets filter; for both wavelet transforms *sym*2 was employed. For all MD results, the wavelet-based algorithms used hard thresholding with level-dependent variance estimation [33] for coloured noise[3] (as opposed to a single value established from the finest details); only in the case of DPD simulations, due to coarse-graining and the use of local random thermostats, we assumed that the data was corrupted with white noise rather than correlated fluctuations. For estimating the coefficients of WienerChop, we performed a decimated wavelet transform with the same parameters. The second filter for WT_2 in the WienerChop algorithm contained double the number of vanishing moments of the WT_1, i.e. *sym*4, and the frequency resolution was one level higher to aid the filtering. For MD data containing correlated noise, we introduced our modification of WienerChop algorithm, by employing level-dependent thresholding.

The quality of the extracted information was verified with an averaged signal-to-noise ratio (SNR); for a de-noised matrix $A_k \in \mathbb{R}^{N,M}$ and a set of true signals $A_t \in \mathbb{R}^{N,M}$, SNR is calculated as

$$\mathrm{SNR} = \sum_{\tau^s=1}^{N} 10 \cdot \log_{10} \frac{\sum_{x=1}^{M} \left(A_t(\tau^s, x)\right)^2}{\sum_{x=1}^{M} \left(A_t(\tau^s, x) - A_k(\tau^s, x)\right)^2}. \tag{9}$$

In order to relate the results to an initial noise level, we introduce the overall dimensionless gain in signal-to-noise ratio:

$$\mathrm{Gain} = \frac{\mathrm{SNR_{approx}} - \mathrm{SNR_{noisy}}}{\mathrm{SNR_{noisy}}}, \tag{10}$$

where $\mathrm{SNR_{noisy}}$ and $\mathrm{SNR_{approx}}$ are the SNR values of the original corrupted signal and de-noised data, respectively. In addition, the processing time required to obtain a de-noised solution was assessed by recording the internal time at execution of each procedure. We note that this measure was biased as it depended on the implementation of the algorithms and the operational system used. For clarity, we report the most successful application of POD+ for each case (based on SNR values or visual quality) with recommendations for treating particle data resembling the analysed problems. In many cases, we limited the use of wavelet transform only to a decimated strategy, DWT, which is computationally the least expensive.

3.1. Simulation description

The molecular dynamics simulations were carried out using the open-source mdFOAM solver, built in OpenFOAM [45–47]. Modelling with DPD was performed with DL_MESO [48]. As a first test case, we considered molecular dynamics simulations of unsteady argon flow through a converging–diverging nanochannel, as shown in Fig. 2. The channel had a length $L_x = 68$ nm in the flow direction, a depth of $L_z = 5.44$ nm, and channel height varying between 3.4 nm and 3.04 nm as shown in Fig. 2(a). More details on this case can be found in Borg et al. [47]. We applied three types of body force to the argon molecules with the same magnitude $F = 0.487$ pN to achieve various time-scale separated flow conditions (Fig. 2(b)–(d)), to which statistical averaging would be difficult:

1. *Harmonically pulsating flow*. An oscillation force with time period $t = 108$ ps was used.
2. *A start-up flow problem*. An instantaneous application of F.
3. *Mixed period flow*. An oscillating gravity force with gradually increasing period: $0.22 \rightarrow 10.8$ ns.

In addition to liquid argon, a water flow case was also considered:

4. *Water flow through a carbon nanotube (CNT)*. Each water molecule inside the single-wall CNT (see Fig. 3) was subject to a harmonically oscillating force with an amplitude of $F = 0.034$ pN and period of $t = 0.199$ ns. We used the rigid TIP4P/2005 water model described by Abascal and Vega [49]. The Lennard–Jones parameters of the water–carbon interactions for the CNT were taken from Werder et al. [50].

[3] Coloured noise can be defined as a signal with a power spectral density per unit of bandwidth proportional to $1/f^{\alpha}$, where f is frequency and $0 < \alpha < 2$; noise with $\alpha = 1$ is referred to as *pink*.

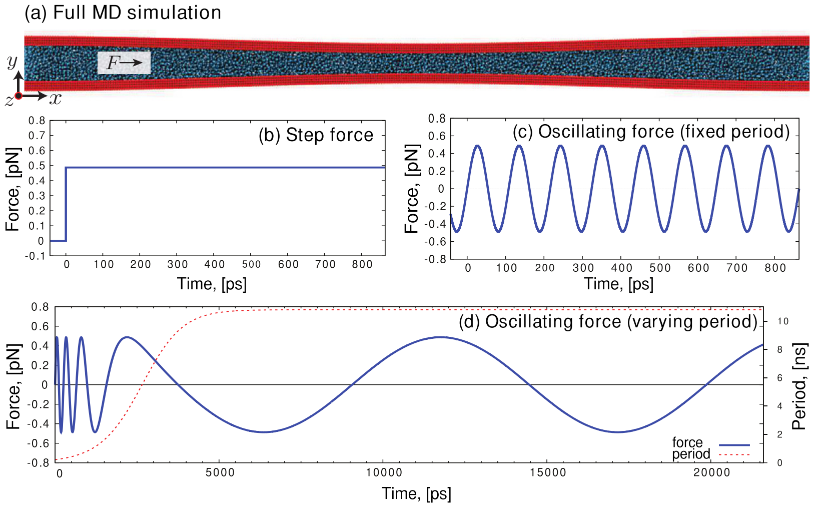


Fig. 2. (a) Molecular dynamics simulation of a converging–diverging channel, with flow generated by the application of (b) a step forcing, (c) an oscillating force with fixed period, or (d) with varying period.

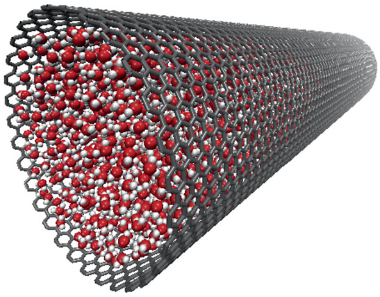


Fig. 3. Simulation snapshot of water flowing through a CNT.

A DPD simulation was also performed to show versatility of the considered algorithms:

5. *Phase separation phenomenon*. A DPD simulation of a solvent mixture of oil, water and surfactant was undertaken. The system consisted of 192000 DPD beads, made up of three bead types: water, hydrophobic oil group and hydrophilic head group. The oil molecules were made of three bonded oil beads, the surfactant molecules were of three bonded beads, consisting of two oil beads followed by a head bead, and each water molecule was represented by a single non-bonded bead. For both oil and surfactant molecules the bonds consisted of harmonic potential on the bond length. A snapshot of the simulation system is presented in Fig. 4.

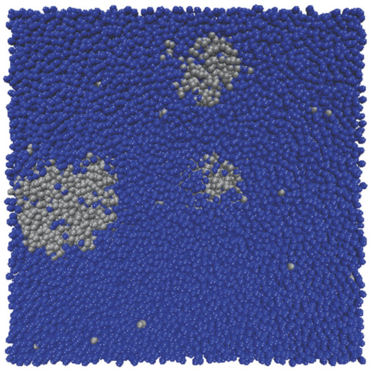


Fig. 4. Last time-step of phase separation phenomenon simulated with DPD.

In all the problems considered, the spatial distribution of the observables was calculated using the binning method [51].

3.2. Results

3.2.1. Converging–diverging channel

The first three simulations produced sets of velocity profiles varying in both time and space. Such measurements impose restrictions on the length of time and the number of spatial micro-elements (bins) over which the atomistic data can be averaged. One way of enhancing the processing of non-stationary fields is to replicate the computational domain with identical flow conditions which is an inherently expensive solution. 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 $\sqrt{N_r}$ [52]. To overcome this difficulty, Grinberg [14] introduced the window proper orthogonal decomposition which was also employed by Borg et al. [47] to extract smoother distributions (see Fig. 7 in [47]). Therefore, we compared the performance of POD+ with the results obtained by applying POD alone.

Harmonically pulsating flow. The velocity profiles generated with simulation no. 1 formed a data-set consisting of 200 spatial measurements over 160000 time samples; this was equivalent to 864 ps of real time. Calculations were performed on the matrix $A(x, \tau^s)$ with $N = 200$ signals of length $M = 160000$. Proper orthogonal decomposition over the entire set recovered mostly low frequency signals which described well the dynamics of the system. Fig. 5 shows how the POD approximation with $k = 3$ compared with the original noisy signal. The previously described tests were applied to recover the optimal rank of the matrix. As the velocity was changing over time and across the domain, statistical averaging could not be applied without loss of information. We assumed that the POD solution for $N = 200$ and $M = 160000$ was the targeted result and used it as A_t (Eq. (9)) for further analysis. To reduce the cost of modelling the flow in the channel, we examined if similar velocity profiles could have been obtained but with only $N = 20$ micro-elements spread across the domain (10 times fewer); we also analysed how POD and POD+ techniques performed with smaller time-windows over which the filtering was performed. The shorter the time after which an optimal approximation can be obtained, reduces the communication bottleneck in multi-scale modelling.

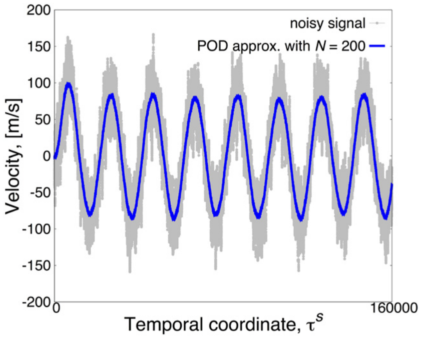
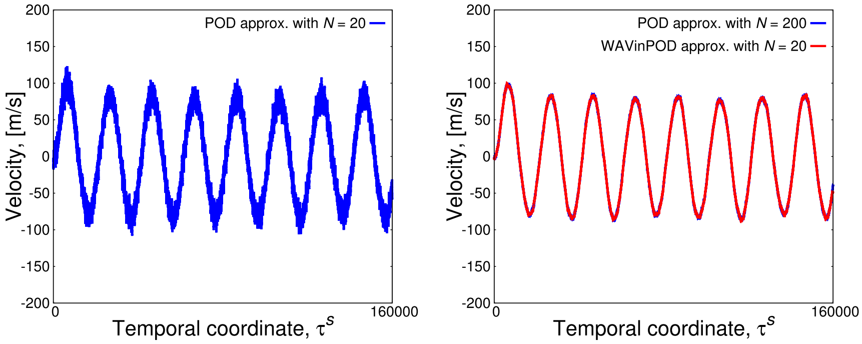


Fig. 5. Result of applying POD to the velocity field from the MD simulation of a periodically-pulsating flow; noisy signal SNR = 8.56 dB, $N = 200$ and $k = 3$.

Figs. 6(a) and 6(b) compare POD and WAVinPOD for $N = 20$, showing that the latter was more efficient in extracting information from a smaller noisy data-set with an average SNR = 9.24 dB. The reconstructed profile from the first bin obtained with WAVinPOD had SNR = 34.07 dB and Gain ≈ 3, almost twice as much as POD, for which SNR = 17.44 dB. It

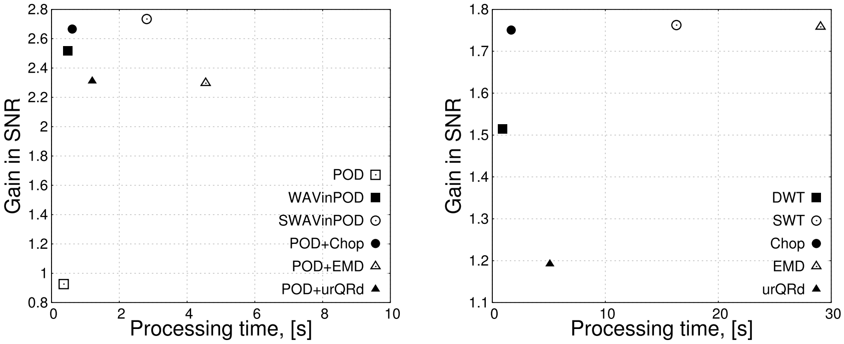


(a) POD, SNR = 17.91 dB.

(b) WAVinPOD, SNR = 34.07 dB.

Fig. 6. Result of applying POD and WAVinPOD to the velocity field from the MD simulation of a periodically-pulsating flow with $N = 20$ bins and $k = 3$; *sym2* and $J = 10$ levels of decomposition.

took WAVinPOD $t \approx 0.48$ s to produce comparable results to information extracted with POD from the 10× larger system in $t \approx 7.82$ s. Having a well predicted velocity distribution from a smaller number of bins or micro-elements also results in a less expensive simulation. Applying wavelet thresholding alone takes double the time of WAVinPOD and provides only an average Gain = 1.5. Comparably good approximations to WAVinPOD were obtained with other POD+ methods. However, as expected POD+EMD-IT and POD+SSA were the most expensive. In addition, for the empirical mode decomposition a larger multiple of the universal threshold T_{multi} had to be applied. As there is no general rule for determining an optimal value of T_{multi}, the threshold was chosen based on experience. All the measures are summarised in the Fig. 7(a), where the overall Gain in SNR is plotted against the processing time. To show how much improvement is achieved by combining the algorithms with POD, we report the results of applying the methods alone in Fig. 7(b).



(a) Results for POD and POD+ methods.

(b) Filtering without POD.

Fig. 7. Comparison of all the methods applied to noisy velocity data from the simulation of a periodically oscillating flow. Note that SSA is not listed as the processing time was too high.

Additional analysis examined how much improvement could be obtained by applying the WAVinPOD method in comparison to POD for smaller moving time-windows during the simulation run (similar to WPOD [14]). Each method was applied to a velocity matrix formed from $N = 20$ spatial measurements and data length divided into $M \in \{50, 100, 200, 400\}$ blocks, i.e. the shortest time-length consisted of $M = 160000/400 = 400$ instances. The window moved until the profiles of the full length were reconstructed. Fig. 8 compares the averaged gain in SNR recovered with WAVinPOD and POD for the same input parameters as used before[4] and $k = 2$ (smaller number of modes was needed for decreased data size). The hybrid method outperformed POD allowing for a much higher gain in SNR with only $t = 2$ s of processing time for the biggest window size.

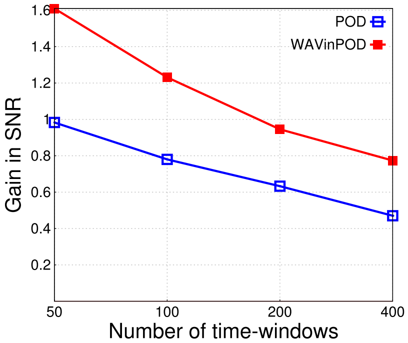
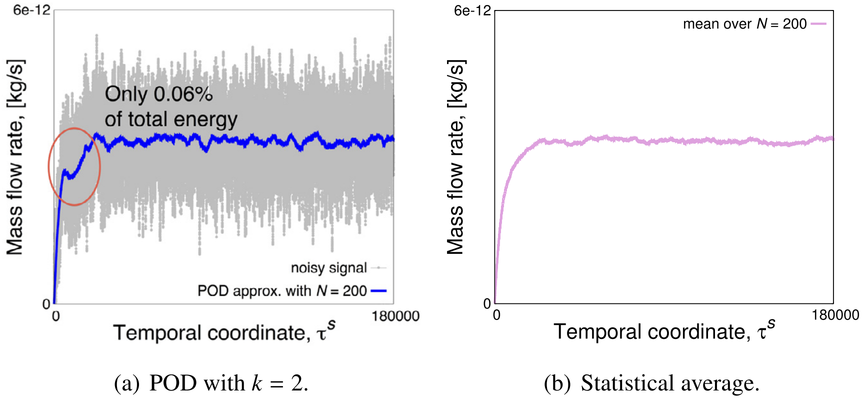


Fig. 8. Gains in SNR recovered with POD and WAVinPOD from noisy data-sets with SNR = 9.24 dB, $N = 20$ and $M = 160000/i$, where $i = 50, 100, 200, 400$.

A start-up flow problem. Figs. 9(a) and 9(b) present the results for the stream-wise distribution of mass flow rate from the first micro-domain. Both POD and mean approximation were obtained from a matrix of size $N = 200$ and $M = 180000$ (duration of $t = 972$ ps). The POD method picked up an acoustic response at the beginning of the flow. Statistical averaging over all the bins resulted in unwanted over-smoothing, losing the step-change present in the instantaneous data (see Fig. 9(b)). Small, abrupt changes in the time-dependent flow state can be much less energetic than the long-time trend and might not be represented by the first POD eigenmode (for non mean-centred data). The change in the mass flow rate in Fig. 9(a) had

[4] In this work, when the signal length is not a power of 2, the symmetric boundary value replication is applied.

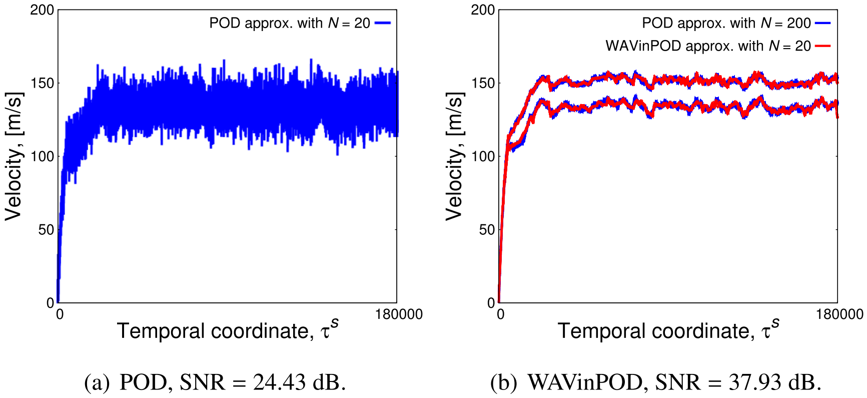


(a) POD with $k = 2$.

(b) Statistical average.

Fig. 9. POD analysis and standard averaging of the time-varying mass flow rate from an MD simulation of a start-up flow, $N = 200$ spatial bins.

only 0.06% of the energy, while the first eigenvalue representing the mean contained over 90%. Therefore, applying POD analysis can outperform averaging in terms of information extraction. In the case of unsteady simulations, further improvement in filtering high frequencies is provided with POD+ methods. The techniques were applied to velocity measurements with amplitude varying along the channel. For the large system, all methods generated comparable outputs (only visually smoother). However, when less spatial sampling was available, POD clearly started recovering more noise, burying small scale dynamics. In order to extract the acoustic response from the velocity measurements, $k = 10$ modes were needed from a smaller ensemble with $N = 20$. Figs. 10(a) and 10(b) compare the velocity profiles from the first and 5th bin filtered with POD and WAVinPOD with the same input parameters as in the previous simulation. Clearly, the latter performed better producing an approximation with SNR = 37.93 dB (the whole data-set had Gain = 1.04), while POD obtained SNR = 24.43 dB with the overall Gain = 0.29. This example showed that POD+ methods allow for retaining small changes in the flow, which can otherwise be lost in POD analysis or statistical averaging.

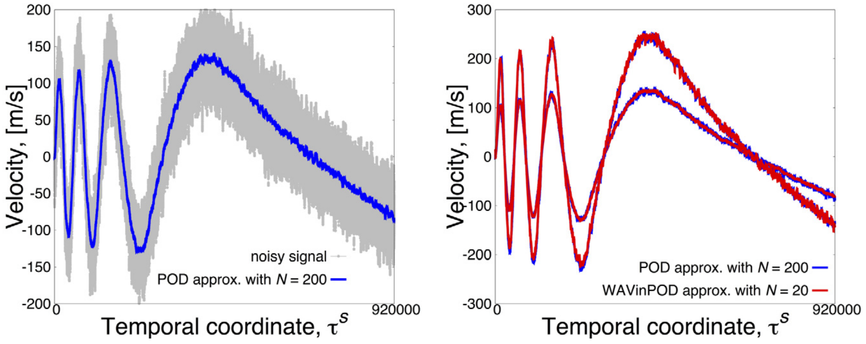


(a) POD, SNR = 24.43 dB.

(b) WAVinPOD, SNR = 37.93 dB.

Fig. 10. POD and WAVinPOD approximations of the velocity from the start-up flow simulation; $N = 20$ bins and $k = 10$; *sym2* and $J = 10$. In the plot (b), the velocity measurements were obtained from the first and 5th bin (corresponding to 41st bin in matrix $N = 200$); for clarity, the plot (a) shows only the result from the 5th bin.

Mixed period flow. The last simulation of the converging/diverging channel generated $M = 920000$ time samples equivalent to $t = 4969$ ps. No averaging was applied prior to POD. The de-noised profile, reconstructed with only $k = 3$ modes, is plotted with the original velocity distribution in Fig. 11(a). Again POD successfully separated unwanted fluctuations from the correlated flow of particles. It should be stressed that POD does not require any *a priori* information regarding the nature of oscillations. All the methods were tested on the original signal length but with only $N = 20$ bins. Apart from the window size for SSA and urQRd, $L = M/2000 = 460$, all the other input parameters were kept the same as in previous simulations. For calculation of SNRs, the POD approximation obtained from the full matrix was used as A_t. It should be noted that it was a biased estimation as POD+ algorithms, even for smaller systems, were returning smoother profiles than POD. Fig. 11(b) presents how well the dynamics of the system were recovered with WAVinPOD; the two velocity profiles show acceleration of the flow in the first half of the channel. Recovered SNR = 29.16 dB, translating to Gain ≈ 1.4, was higher than for POD, SNR = 21.64 dB (Gain = 0.75); processing time of WAVinPOD, although 1.5× slower than POD, was only $t = 2.72$ s. Again WAVinPOD obtained similar information for smaller data-set as POD with $N = 200$; processing the full size matrix took POD 20× longer. The results are summarised in Fig. 12. In addition, Fig. 13 shows that 1.5× higher Gain was achieved with WAVinPOD than POD constructed from shorter signal portions.



(a) POD and noisy signal, $N = 200$. (b) WAVinPOD, SNR = 29.16 dB, $N = 20$.

Fig. 11. Outcome of applying POD and WAVinPOD to the velocity field from the MD simulation of a pulsating flow with a varying period of oscillations; parameters used: filter *sym2* with $J = 10$ levels of decomposition, $k = 3$ modes. For $N = 20$, the noisy data-set had SNR = 12.22 dB. In the plot (b), the measurements were obtained from the first and 12th bin.

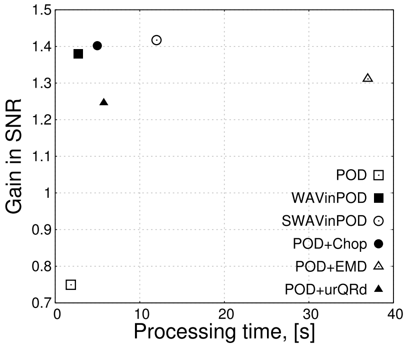


Fig. 12. Filtering velocity data from the simulation of a periodically oscillating flow. Note that SSA is not shown due to high processing time, $t = 188$ s; POD + SSA obtained Gain = 1.36 with $k = 1$ for all the modes.

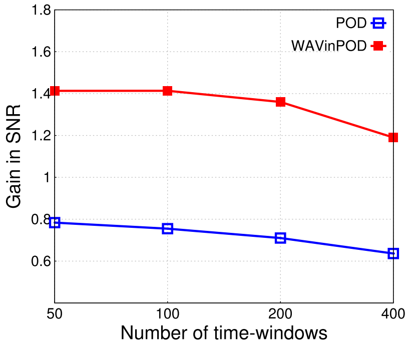
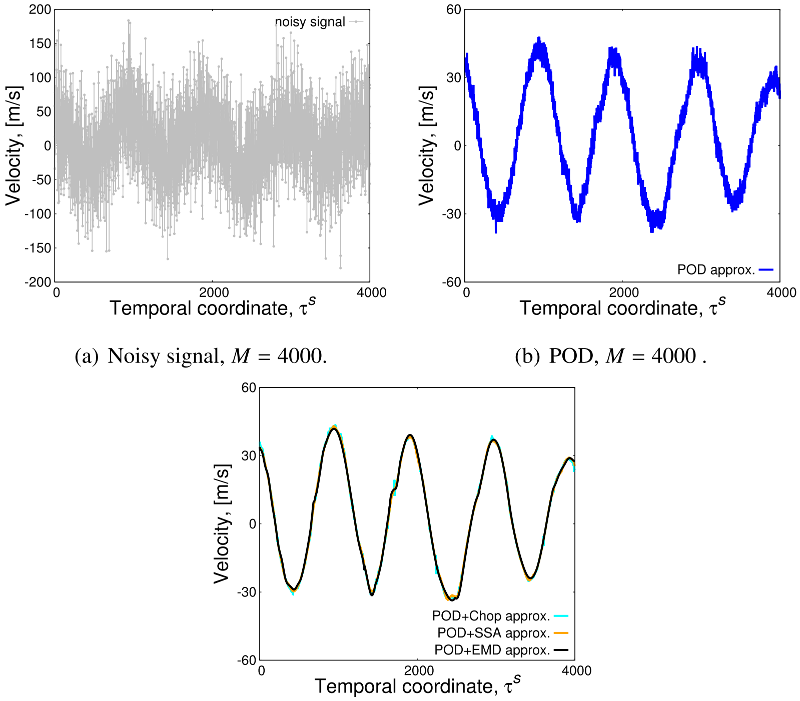


Fig. 13. Gains in SNR recovered with POD and WAVinPOD from noisy data-sets with SNR = 12.22 dB, $N = 20$ and $M = 920000/i$, where $i = \{50, 100, 200, 400\}$. Approximations were constructed with rank $k = 2$.

3.2.2. Water flow through a carbon nanotube

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 performed an additional study on velocity measurements from oscillating water flow through a CNT. The initial matrix contained $M = 4000$ observations and $N = 512$ velocity measurements at each time-step. There were 4 complete oscillations in the ensemble, which were used to obtain the mean solution. Noisy velocity distribution from the 10th bin is shown in Fig. 14(a), while Fig. 14(b) and 14(c) compare the quality of POD and POD+ results. The latter clearly extracted smoother profiles. Calculating a cumulative mean (the average of 4 cycles, $M = 4000$) did not reduce the noise as well as POD from only $M = 1000$ velocity measurements (see Fig. 15(a)). Further improvement was achieved with POD+ techniques which removed the remaining fluctuations as shown in Fig. 15(c). To obtain a comparable

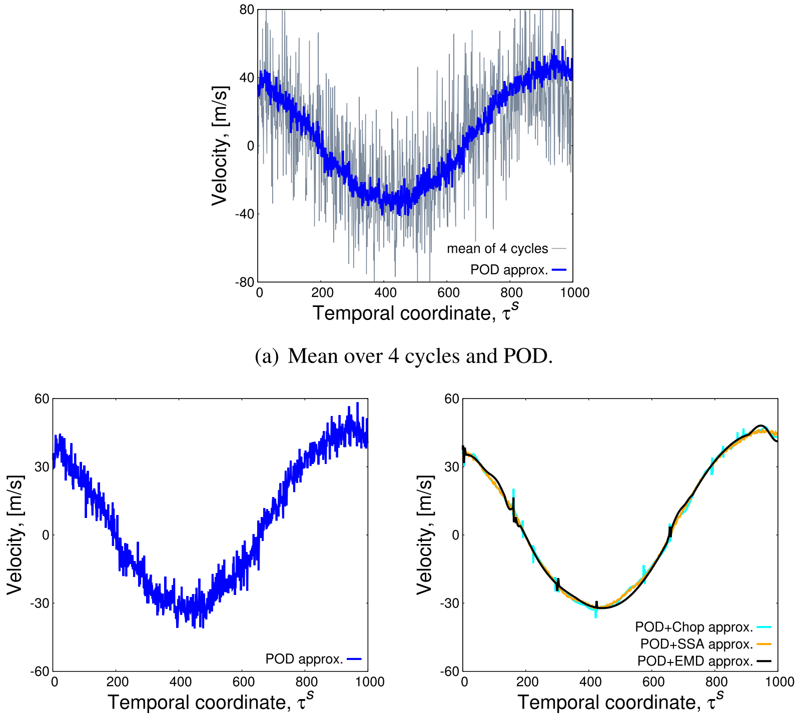


(a) Noisy signal, $M = 4000$.

(b) POD, $M = 4000$.

(c) POD+, $M = 4000$.

Fig. 14. Velocity approximations recovered with POD and POD+ from the MD simulation of an oscillating water flow through CNT; $k = 7$ modes were extracted for POD; POD + WienerChop: *sym2* and $J = 6$ levels of decomposition for WT_1; EMD-IT: $T_{\text{multi}} = 1$; $k = 1$ and $L = 50$ for SSA.



(a) Mean over 4 cycles and POD.

(b) POD, $M = 1000$.

(c) POD+, $M = 1000$.

Fig. 15. Comparison of approximations obtained from statistical averaging over 4 cycles, POD and POD+ applied to only 1 full oscillation, i.e. $M = 1000$ of velocity data from the CNT simulation; $k = 7$ modes were extracted for POD. POD + WienerChop: *sym2* and $J = 6$ levels of decomposition for WT_1; EMD-IT: $T_{\text{multi}} = 1$; $k = 1$ and $L = 50$ for SSA.

level of de-noising with statistical averaging, much more data would have to be collected, increasing the computational cost. In this case, $k = 7$ modes were used to extract the velocity field after performing previously introduced analysis.

3.2.3. Phase separation phenomenon

In the last simulation, as the system was evolving from a mixed state until visible separation was achieved, statistical averaging could not be used to analyse how the particle distribution was changing over time. At first, we applied POD to 3 density fields of size 128×128 at instances $\tau^S = \{1, 50, 100\}$ to extract density contour plots. Visible improvement in smoothness was achieved with POD+ methods; the contours of approximations obtained with SWAVinPOD are compared with POD in Fig. 16, and additional results generated with POD + SSA, POD + EMD-IT and POD + rQRd are shown in Fig. 17 for completeness. In this case, filtering was performed on both left and right singular vectors as no significant aliases were expected for $N = M$. Data size was small, therefore only $J = 4$ levels of decomposition were performed for SWAVinPOD. In the case of EMD-IT, the threshold multiplier was similar to the CNT simulation, $T_{\text{multi}} = 1$. Dominant modes for POD and SSA were automatically established using previously described SVHT analysis as the noise present in DPD simulations was assumed to be less correlated than in MD modelling. 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 POD and POD+ methods. Moreover, in DPD the temperature is controlled locally (unlike commonly adopted Berendsen

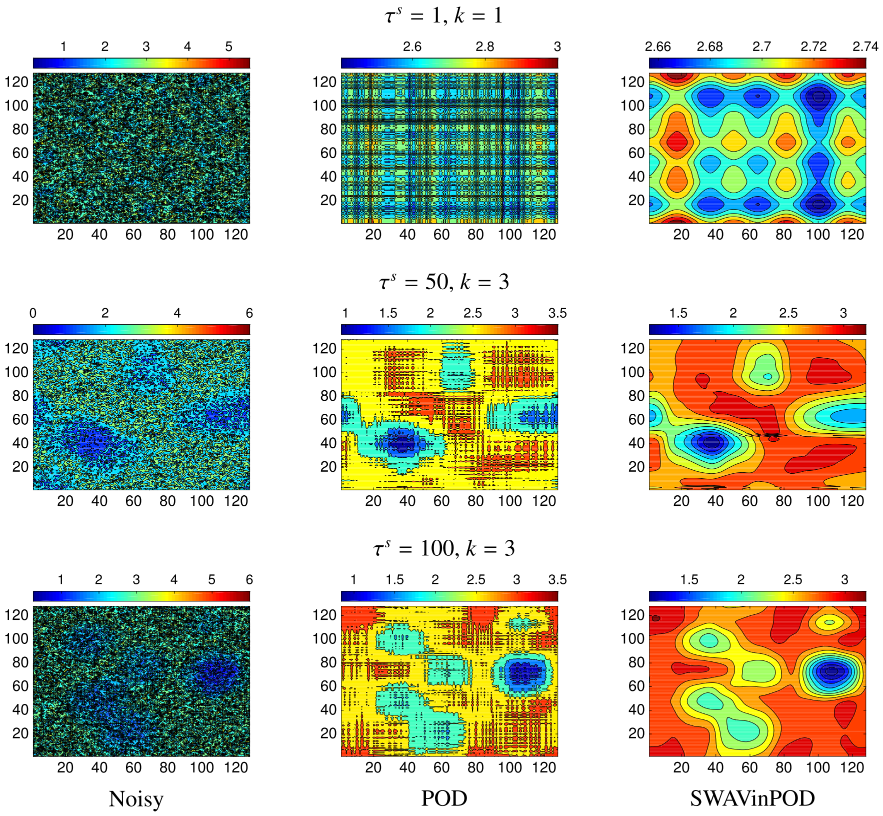


Fig. 16. Density contours of water from DPD simulation. De-noising was performed at 3 instances; *sym2* with $J = 4$ was used for SWAVinPOD.

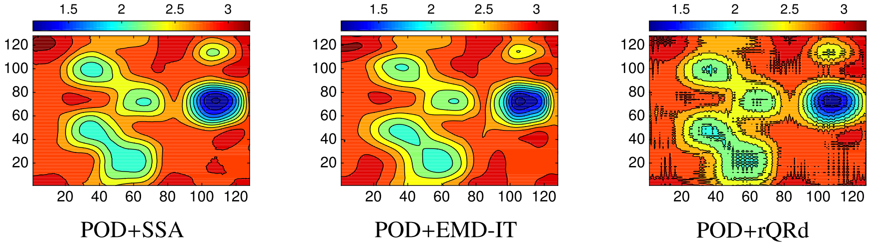


Fig. 17. Density contours of water at the last time-step, $\tau^s = 100$, obtained with POD + SSA, POD + rQRd and POD + EMD-IT. Optimal number of modes for SSA with $L = 16$ was established with SVHT, $k = 2$, and $\zeta = 2$ for rQRd; $T_{\text{multi}} = 1$ for EMD-IT.

thermostat used in MD simulations) with the use of random and dissipative forces. 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 enables a more efficient de-noising than in the case of a globally thermostatted system.

4. Conclusions and remarks

The main goal of this paper was to review, develop, and evaluate new methods for solving the problem of noise reduction in computational nanofluidics. The filtering tools were studied and applied to a wide range of numerical results. The fluid flows under consideration were modelled separately via particle-based simulations using MD and DPD. Novel procedures were also proposed that outperform the other techniques in extracting significant information from instantaneous measurements. This paper has shown that applying sophisticated de-noising tools to particle-based simulations can reduce the computational time and memory required to obtain acceptable ensemble solutions.

All the noise-reduction methods offer a balance between different properties. Rather than choosing a universal approach, this paper 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 of the algorithms have been combined with POD. In our proposed POD+ methods, additional de-noising is performed on the dominant modes of SVD 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 than when the techniques are applied separately. The hybrid of POD and wavelet-based thresholding appeared to be the most flexible and efficient.

Acknowledgements

The research leading to these results was performed at the University of Manchester and STFC Daresbury Laboratory. This work is financially supported by the UK's Engineering and Physical Sciences Research Council (EPSRC) under grants EP/I011927/1, EP/N016602/1, EP/K038621/1, EP/K038427/1, EP/K038664/1, and EPSRC Doctoral Prize Award. Results were obtained using ARCHER, the UK's national supercomputing service, supported by an ARCHER Leadership grant. Supporting data are available open access at http://dx.doi.org/10.7488/ds/1467. Other information can be obtained from the corresponding author.

References

[1] D.C. Rapaport, The Art of Molecular Dynamics Simulation, Cambridge University Press, 2004.
[2] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Oxford University Press, 1994.
[3] P. Espanol, P. Warren, Statistical mechanics of dissipative particle dynamics, Europhys. Lett. 30 (4) (1995) 191.
[4] G. Karniadakis, A. Beskok, N. Aluru, Microflows and Nanoflows: Fundamentals and Simulation, Springer Science & Business, Media, 2006.
[5] A. Alexiadis, S. Kassinos, Molecular simulation of water in carbon nanotubes, Chem. Rev. 108 (12) (2008) 5014–5034.
[6] W.D. Nicholls, M.K. Borg, D.A. Lockerby, J.M. Reese, Water transport through (7,7) carbon nanotubes of different lengths using molecular dynamics, Microfluid. Nanofluid. 12 (1–4) (2012) 257–264.
[7] K.M. Mohamed, A.A. Mohamad, A review of the development of hybrid atomistic–continuum methods for dense fluids, Microfluid. Nanofluid. 8 (3) (2010) 283–302.
[8] D.A. Fedosov, B. Caswell, G.E. Karniadakis, A multiscale red blood cell model with accurate mechanics, rheology, and dynamics, Biophys. J. 98 (10) (2010) 2215–2225.
[9] S.S. Collis, Monitoring unresolved scales in multiscale turbulence modeling, Phys. Fluids 13 (6) (2001) 1800–1806.
[10] A.F. Tuck, From molecules to meteorology via turbulent scale invariance, Q. J. R. Meteorol. Soc. 136 (650) (2010) 1125–1144.
[11] P.A. Kollman, I. Massova, C. Reyes, B. Kuhn, S. Huo, L. Chong, M. Lee, T. Lee, Y. Duan, W. Wang, O. Donini, P. Cieplak, J. Srinivasan, D. Case, T.E. Cheatham, Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models, Acc. Chem. Res. 33 (12) (2000) 889–897.
[12] N. Bellomo, C. Dogbe, On the modeling of traffic and crowds: a survey of models, speculations, and perspectives, SIAM Rev. 53 (3) (2011) 409–463.
[13] S.P. Ghael, A.M. Sayeed, R.G. Baraniuk, Improved wavelet denoising via empirical Wiener filtering, in: Optical Science, Engineering and Instrumentation '97, International Society for Optics and Photonics, 1997, pp. 389–399.
[14] L. Grinberg, Proper orthogonal decomposition of atomistic flow simulations, J. Comput. Phys. 231 (16) (2012) 5542–5556.
[15] J.M. Craddock, C.R. Flood, Eigenvectors for representing the 500 mb geopotential surface over the Northern Hemisphere, Q. J. R. Meteorol. Soc. 95 (405) (1969) 576–593.
[16] D.L. Donoho, M. Gavish, The optimal hard threshold for singular values is 4/sqrt (3), IEEE Trans. Inf. Theory 60 (8) (2014) 5040–5053.
[17] D.S. Broomhead, G.P. King, Extracting qualitative dynamics from experimental data, Physica D 20 (2) (1986) 217–236.
[18] D.S. Broomhead, G.P. King, On the qualitative analysis of experimental dynamical systems, Nonlinear Phenom. Chaos (1986) 113–114.
[19] N. Golyandina, A. Zhigljavsky, Singular Spectrum Analysis for Time Series, Springer, 2013.
[20] H. Hassani, Singular spectrum analysis: methodology and comparison, J. Data Sci. 5 (2) (2007) 239–257.
[21] E. Bingham, 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, ACM, 2001, pp. 245–250.
[22] N. Halko, P.-G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288.
[23] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, M. Tygert, Randomized algorithms for the low-rank approximation of matrices, Proc. Natl. Acad. Sci. USA 104 (51) (2007) 20167–20172.
[24] L. Chiron, M.A. van Agthoven, B. Kieffer, C. Rolando, M.-A. Delsuc, Efficient denoising algorithms for large experimental datasets and their applications in Fourier transform ion cyclotron resonance mass spectrometry, Proc. Natl. Acad. Sci. USA 111 (4) (2014) 1385–1390.

[25] S.G. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1999.

[26] I. Daubechies, Ten Lectures on Wavelets, vol. 61, SIAM, 1992.

[27] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 674–693.

[28] Y. Meyer, Wavelets and Operators, vol. 1, Cambridge University Press, 1995.

[29] G.P. Nason, B.W. Silverman, The stationary wavelet transform and some statistical applications, in: Lecture Notes in Statistics, vol. 103, Springer-Verlag, New York, 1995, pp. 281–299.

[30] R.R. Coifman, D.L. Donoho, Translation-Invariant De-Noising, Springer, 1995.

[31] D.L. Donoho, J.M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (3) (1994) 425–455.

[32] D.L. Donoho, De-noising by soft-thresholding, IEEE Trans. Inf. Theory 41 (3) (1995) 613–627.

[33] I.M. Johnstone, B.W. Silverman, Wavelet threshold estimators for data with correlated noise, J. R. Stat. Soc., Ser. B, Stat. Methodol. 59 (2) (1997) 319–351.

[34] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series, vol. 2, MIT Press, Cambridge, MA, 1949.

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

[36] H. Choi, R. Baraniuk, Analysis of wavelet-domain Wiener filters, in: Proceedings of the IEEE-SP International Symposium on Time–Frequency and Time–Scale Analysis, IEEE, 1998, pp. 613–616.

[37] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc., Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995.

[38] Z. Wu, N.E. Huang, A study of the characteristics of white noise using the empirical mode decomposition method, Proc. R. Soc., Math. Phys. Eng. Sci. 460 (2046) (2004) 1597–1611.

[39] G. Rilling, P. Flandrin, One or two frequencies? The empirical mode decomposition answers, IEEE Trans. Signal Process. 56 (1) (2008) 85–95.

[40] P. Flandrin, P. Gonçalves, G. Rilling, EMD equivalent filter banks, from interpretation to applications, in: Hilbert–Huang Transform and Its Applications, 2005, pp. 57–74.

[41] Z. Wu, N.E. Huang, X. Chen, The multi-dimensional ensemble empirical mode decomposition method, Adv. Adapt. Data Anal. 1 (3) (2009) 339–372.

[42] G. Huang, Y. Su, A. Kareem, H. Liao, Time–frequency analysis of nonstationary process based on multivariate empirical mode decomposition, J. Eng. Mech. 142 (1) (2015) 04015065.

[43] Y. Kopsinis, S. McLaughlin, Development of EMD-based denoising methods inspired by wavelet thresholding, IEEE Trans. Signal Process. 57 (4) (2009) 1351–1362.

[44] M.J. Zimoń, J.M. Reese, D.R. Emerson, A novel coupling of noise reduction algorithms for particle flow simulations, J. Comput. Phys. 321 (1) (2016) 169–190, http://dx.doi.org/10.1016/j.jcp.2016.05.049.

[45] G.B. Macpherson, J.M. Reese, Molecular dynamics in arbitrary geometries: parallel evaluation of pair forces, Mol. Simul. 34 (1) (2008) 97–115.

[46] K. Ritos, N. Dongari, M.K. Borg, Y. Zhang, J.M. Reese, Dynamics of nanoscale droplets on moving surfaces, Langmuir 29 (23) (2013) 6936–6943.

[47] M.K. Borg, D.A. Lockerby, J.M. Reese, A hybrid molecular–continuum method for unsteady compressible multiscale flows, J. Fluid Mech. 768 (2015) 388–414.

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

[49] J.L.F. Abascal, C. Vega, A general purpose model for the condensed phases of water: TIP4P/2005, J. Chem. Phys. 123 (23) (2005) 234505.

[50] T. Werder, J. Walther, R. Jaffe, T. Halicioglu, P. Koumoutsakos, On the water–carbon interaction for use in molecular dynamics simulations of graphite and carbon nanotubes, J. Phys. Chem. B 107 (6) (2003) 1345–1352.

[51] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Oxford University Press, 1989.

[52] N.G. Hadjiconstantinou, A.L. Garcia, M.Z. Bazant, G. He, Statistical error in particle simulations of hydrodynamic phenomena, J. Comput. Phys. 187 (1) (2003) 274–297.