MaMiCo: Parallel Noise Reduction for Multi-instance Molecular-Continuum Flow Simulation

Piet Jarmatz[1,3](✉) and Philipp Neumann[2]

[1] Department of Informatics, Technical University of Munich, Garching, Germany
[2] Department of Informatics, University of Hamburg, Hamburg, Germany
[3] Faculty of Mechanical Engineering, Helmut Schmidt University, Hamburg, Germany
jarmatz@hsu-hh.de

Abstract. Transient molecular-continuum coupled flow simulations often suffer from high thermal noise, created by fluctuating hydrodynamics within the molecular dynamics (MD) simulation. Multi-instance MD computations are an approach to extract smooth flow field quantities on rather short time scales, but they require a huge amount of computational resources. Filtering particle data using signal processing methods to reduce numerical noise can significantly reduce the number of instances necessary. This leads to improved stability and reduced computational cost in the molecular-continuum setting.

We extend the Macro-Micro-Coupling tool (MaMiCo) – a software to couple arbitrary continuum and MD solvers – by a new parallel interface for universal MD data analytics and post-processing, especially for noise reduction. It is designed modularly and compatible with multi-instance sampling. We present a Proper Orthogonal Decomposition (POD) implementation of the interface, capable of massively parallel noise filtering. The resulting coupled simulation is validated using a three-dimensional Couette flow scenario. We quantify the denoising, conduct performance benchmarks and scaling tests on a supercomputing platform. We thus demonstrate that the new interface enables massively parallel data analytics and post-processing in conjunction with any MD solver coupled to MaMiCo.

Keywords: Multiscale · Fluid dynamics · HPC · Noise filter · Parallel · Transient · Coupling · Software design · Molecular dynamics · Simulation · Data analytics · Multi-instance · POD · Molecular-continuum

1 Introduction

Multiscale methods in computational fluid dynamics, in particular coupled molecular-continuum simulations [6,7,12,22], allow to go beyond the limitations imposed by modelling accuracy or computational feasibility of a particular

J. M. F. Rodrigues et al. (Eds.): ICCS 2019, LNCS 11539, pp. 451–464, 2019.
https://doi.org/10.1007/978-3-030-22747-0_34

single-scale method. They are frequently applied for instance for nanostructure investigation in chemistry, especially for research in lithium-ion batteries [15].

In the molecular-continuum context, MD regions and continuum flow regions are coupled to extend the simulation capability over multiple temporal and spatial scales. This is useful for many challenging applications involving e.g. nanomembranes [20] or polymer physics [1,18].

Due to applications that often require large domains, long time spans, or fine resolution, parallelism and scalability constitute important aspects of molecular-continuum methods and software. Several codes for massively parallel execution of molecular-continuum simulations on high performance computing (HPC) systems exist, for instance the CPL library [22], MaMiCo [14] or HACPar [19].

Many coupling schemes for molecular-continuum simulations have been investigated [2,8,12,16,24], based amongst others on the internal-flow multiscale method [2] or the heterogeneous multiscale method [8]. In the steady state case, time-averaging of hydrodynamic quantities sampled from MD is sufficient for the coupling to a continuum solver [7]. But a transient simulation with short coupling time intervals can easily become unstable due to fluctuating MD flow field quantities.

One approach to tackle this is multi-instance sampling [13], where averaged information comes from an ensemble of MD systems. This approach, however, is computationally very expensive.

It has been shown that noise removal techniques, i.e. filters, are another effective approach to reduce thermal fluctuations in the molecular-continuum setting. Grinberg used proper orthogonal decomposition (POD) of MD flow field data, demonstrating also HPC applicability of their method by running a coupled atomistic-continuum simulation on up to 294,912 compute cores [11]. Zimoń et al. [26] have investigated and compared different kinds of noise filters for several particle based flow simulations, such as harmonically pulsating flow or water flow through a carbon nanotube. They pointed out that the combination of POD with one of various other noise filtering algorithms yields significant improvements in denoising quality.

One of the main challenges in the field of coupled multiscale flow simulation is the software design and implementation, since goals such as flexibility, interchangeability of solvers and modularity on an algorithmic level often contradict to hardware and high performance requirements. Some very generic solutions exist, such as MUI [23] or the MUSCLE 2 [3] software, which can be used for various multiscale settings. We recently presented the MaMiCo coupling framework [13,14]. It is system-specific for molecular-continuum coupling, but independent of actual MD and CFD solvers. MaMiCo provides interface definitions for arbitrary flow simulation software, hides coupling algorithmics from the solvers and supports 2D as well as 3D simulations.

In this paper, we present extensions of the MaMiCo tool for massively parallel particle simulation data analytics and noise reduction. We introduce a new interface that is intended primarily for MD flow quantity noise filtering, but in the same way usable for any kind of MD data post-processing for the molecular-continuum coupling. We also present an implementation of a HPC compatible and scalable POD noise filter for MaMiCo. Our interface is compatible with multi-instance MD computations in a natural way, thus it enables a novel combination of noise filtering and multi-instance sampling: A small ensemble of separate MD simulations delivers information about the behaviour of the simulated system on a molecular level, while the filtering efficiently extracts a smooth signal for the continuum solver. The goal of this paper is to investigate this new combination and to discuss the related software design issues.

In Sect. 2 we introduce the relevant theoretical background on MD (Sect. 2.1), the considered molecular-continuum coupling (Sect. 2.2) and POD (Sect. 2.3). Section 3 focuses on the implementation of the MaMiCo data analytics and filtering extension. In Sect. 4, we quantify the denoising quality of the filtering. We further analyse signal-to-noise ratios (SNR) for a three-dimensional oscillating Couette flow with synthetic noisy data (Sect. 4.1). To validate the resulting transient coupled molecular-continuum simulation, we use a Couette flow setup and examine its startup (Sect. 4.2). We also conduct performance measurements and scaling tests (Sect. 4.3) of our new coupled flow simulation on the *LRZ Linux Cluster*[1] platforms *CoolMUC-2* and *CoolMUC-3*. Finally, we give a summary and provide an outlook to future work in Sect. 5.

2 Theoretical Background

2.1 Molecular Dynamics (MD)

For the sake of simplicity we restrict our considerations to a set of Lennard-Jones molecules here, without loosing generality since the coupling and noise reduction methodology and software is compatible with other particle systems such as dissipative particle dynamics [9] systems in the same way.

For a number N of molecules with positions $\boldsymbol{x}_i$, velocities $\boldsymbol{v}_i$, mass m and interacting forces $\boldsymbol{F}_i$, the behaviour of the system is determined by a Verlet-type numerical integration, using a particle system time step width dt_{P}, of Newton's equation of motion

$$\frac{d}{dt}\boldsymbol{v}_i = \frac{1}{m}\boldsymbol{F}_i, \qquad \frac{d}{dt}\boldsymbol{x}_i = \boldsymbol{v}_i. \tag{1}$$

$\boldsymbol{F}_i$ is defined by Lennard-Jones parameters ϵ, σ and a cut-off-radius r_c, see [13] for details.

The method is implemented using linked cells [10]. The implementation is MPI-parallelized in the simulation software SimpleMD [14] which is part of MaMiCo; note that this solver choice is for simplicity only as it has been shown that MaMiCo interfaces various MD packages, such as LAMMPS, ls1 mardyn, and ESPResSo/ESPResSo++.

[1] https://www.lrz.de/services/compute/linux-cluster/overview/.

2.2 Coupling and Quantity Transfer

The method used here to couple continuum and MD solver is based on [13,16].

The simulation setup and overlapping domain decomposition is shown in Fig. 1. Nested time stepping is used, i.e. the particle system is advanced over n time steps during every time step of the continuum solver, $dt_{\mathrm{C}} := n \cdot dt_{\mathrm{P}}$.

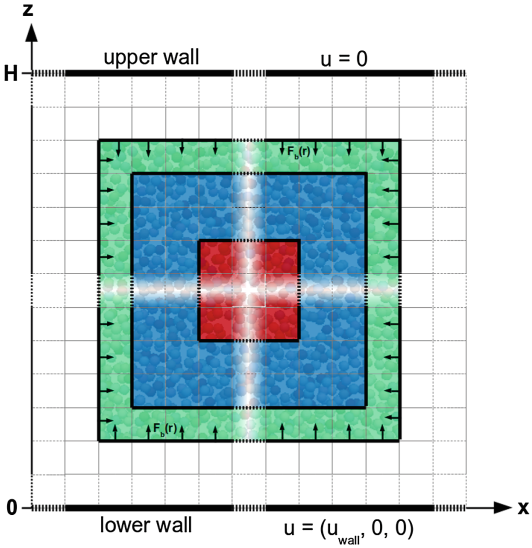


Fig. 1. Coupled Couette simulation transfer region setup. 2D slice through 3D domain, H is the wall distance. Here, only the four outer cell layers close to the MD boundary are shown. At the outer MD domain boundary, an additional boundary force $F_b(r)$ is applied to the molecules. In the green outer MD cells, mass fluxes are transferred from continuum to MD. On the two blue cell layers, velocity values are imposed to MD. Finally, inside the red cells (arbitrarily large region), velocity and density values are sampled, post-processed, and sent back to the continuum solver. (Color figure online)

Particle → continuum data transfer: Continuum quantities for the coupling, such as average velocity $\boldsymbol{u}$, are sampled cell-wise over a time interval dt_{C}:

$$\boldsymbol{u} = \frac{dt_{\mathrm{P}}}{N dt_{\mathrm{C}}} \sum_{k=0}^{\frac{dt_{\mathrm{C}}}{dt_{\mathrm{P}}}-1} \sum_{i=1}^{N} \boldsymbol{v}_i(t_0 + k dt_{\mathrm{P}}) \tag{2}$$

Afterwards, these quantities are optionally post-processed (separately for every MD instance), accumulated from all instances together, and represent the molecular flow field at continuum scale.

Continuum → particle data transfer: Velocities coming from overlapping cells of the continuum flow solver are used to accelerate the molecules in the corresponding grid cells, they are applied via an additional forcing term with shifted time intervals, see [13] for details. Mass flux from the continuum into the particle simulation is realised by particle deletion and insertion with the USHER algorithm [5]. As boundary conditions for the MD-continuum boundary reflecting boundaries are used. We add the boundary force $F_b(r)$ proposed by Zhou et al. [25]. It is given as an analytic fitting formula and estimates missing intermolecular force contributions over a wide range of densities and temperatures.

2.3 Noise Reduction: POD

The proper orthogonal decomposition (POD), also known as principal component analysis, is a general statistical method for high-dimensional data analysis of dynamic processes and a main technique for data dimensionality reduction. It was already used and investigated for analysis of velocity fields from atomistic simulations [11,26]. Here we employ POD as a fast standard method and basic example to demonstrate the capabilities of our data analytics and noise reduction interface. However, note that various algorithms are known to yield significantly stronger denoising, such as the POD+ methods proposed by Zimoń et al. [26], e.g. POD together with wavelet transform or Wiener filtering. Also many multi-dimensional image processing filters like non-local means, anisotropic filtering or total variation denoising [4,17] are promising for CFD data. They can be applied with the noise reduction interface in the same way as POD.

Based on the method of snapshots [21] to analyse a space-time window of fluctuating data, consider a window of N discrete time snapshots $t \in \{t_1, ..., t_N\} \subset \mathbb{R}$. A finite set $\varOmega$ of discrete sampling points $\boldsymbol{x} \in \varOmega \subset \mathbb{R}^3$ defines a set of signal sources $u(\boldsymbol{x}, t)$. POD describes the function $u(\boldsymbol{x}, t)$ as a finite sum:

$$u(\boldsymbol{x}, t) \approx \sum_{k=1}^{k_{max}} \phi_k(\boldsymbol{x}) a_k(t) \tag{3}$$

where the orthonormal basis functions (modes) $\phi_k(\boldsymbol{x})$ and $a_k(t)$ represent the temporal and spatial components of the data, such that the first k_{max} modes are the best approximation of $u(\boldsymbol{x}, t)$. Choosing a sufficiently small k_{max} will approximate only the dominating components of the signal u and exclude noise. The temporal modes $a_k(t)$ can be obtained by an eigenvalue decomposition of the temporal auto-correlation covariance matrix C,

$$C_{ij} = \sum_{\boldsymbol{x} \in \varOmega} u(\boldsymbol{x}, t_i) u(\boldsymbol{x}, t_j) \quad i, j = 1, 2, ..., N. \tag{4}$$

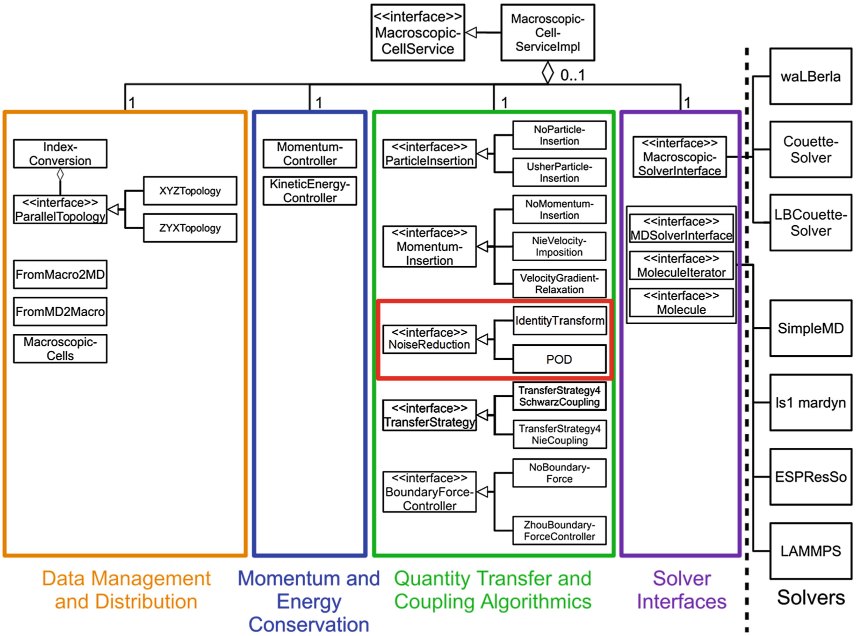
They in turn can be used to compute the spatial modes $\phi_k(\boldsymbol{x})$ with orthogonality relations.

In case of parallel execution, i.e. when subsets of $\varOmega$ are distributed over several processors, Eq. (4) can be evaluated by communicating local versions of C in a global reduction operation. Although this is computationally expensive compared to purely local, independent POD executions, it is helpful to prevent inconsistencies between subdomains and enforce smoothness, especially in a coupled molecular-continuum setting. Every other step of the algorithm can be performed locally.

3 Implementation: MaMiCo Design and Extension

The MaMiCo tool [13,14] is a C++ framework designed to couple arbitrary massively parallel (i.e., MPI-parallel) continuum and particle solvers in a modular and flexible way, employing a Cartesian grid of so-called macroscopic cells to enable the data exchange. It also provides interfaces to coupling algorithms and data exchange routines. In this paper we employ only the built-in MD simulation, called *SimpleMD* on the microscopic side. On the macroscopic side, a simple Lattice Boltzmann (LB) implementation, the *LBCouetteSolver* and an analytical *CouetteSolver* are used. The necessary communication between *SimpleMD* ranks and the ranks of the *LBCouetteSolver*, where the respective cells are located, is performed by MaMiCo.

The extended system design of MaMiCo is shown in Fig. 2, where the latest developments have been marked in the central red box: A newly introduced interface *NoiseReduction* (Listing 1) is part of the quantity transfer and coupling algorithmics bundle. It is primarily intended for noise filtering, but able to manage arbitrary particle data analytics or post-processing tasks during the coupling, independently of the coupling scheme, actual particle and continuum solver in use. It is designed to be compatible with multi-instance MD computations in a natural way, as separate noise filters are instantiated for each MD system automatically. This yields a strong subsystem separation rather than algorithmic interdependency. However, explicit cross-instance communication is still possible if necessary.



<<interface>> MacroscopicCellService
MacroscopicCellServiceImpl
0..1
1
1
1
1
IndexConversion
<<interface>> ParallelTopology
XYZTopology
ZYXTopology
FromMacro2MD
FromMD2Macro
MacroscopicCells
MomentumController
KineticEnergyController
<<interface>> ParticleInsertion
NoParticleInsertion
UsherParticleInsertion
<<interface>> MomentumInsertion
NoMomentumInsertion
NieVelocityImposition
VelocityGradientRelaxation
<<interface>> NoiseReduction
IdentityTransform
POD
<<interface>> TransferStrategy
TransferStrategy4SchwarzCoupling
TransferStrategy4NieCoupling
<<interface>> BoundaryForceController
NoBoundaryForce
ZhouBoundaryForceController
<<interface>> MacroscopicSolverInterface
<<interface>> MDSolverInterface
<<interface>> MoleculeIterator
<<interface>> Molecule
waLBerla
CouetteSolver
LBCouetteSolver
SimpleMD
ls1 mardyn
ESPResSo
LAMMPS
Data Management and Distribution
Momentum and Energy Conservation
Quantity Transfer and Coupling Algorithmics
Solver Interfaces
Solvers

Fig. 2. Extended MaMiCo system design. (Color figure online)

We provide a dummy implementation, *IdentityTransform*, that performs no filtering at all, of the *NoiseReduction* interface, as well as our own massively parallel implementation of a particle data noise filter using POD. The POD implementation employs Eigen[2] to perform linear algebra tasks such as the eigenvalue decomposition. It involves a single invocation of a global (per MD instance) MPI reduction operation to enable detection of supra-process flow data correlation and its separation from thermal noise. It is fully configurable using XML configuration files, where you can specify k_{max} and N.

```
template<unsigned int dim> class
↪  coupling::noisereduction::NoiseReduction {
  public:  [...]
    /** Is called for every macroscopic cell right before sending the
    ↪  macroscopicMass and -Momentum data to the macroscopic solver,
    ↪  but after the call to TransferStrategy::[...] */
    virtual void processInnerMacroscopicCell(
      coupling::datastructures::MacroscopicCell<dim> &cell, const
      ↪  unsigned int &index ){}
    virtual void beginProcessInnerMacroscopicCells(){}
    virtual void endProcessInnerMacroscopicCells(){}
    [...]
}; // Excerpts from main coupling loop:
for (int cycles = 0; cycles < couplingCycles; cycles++){
  couetteSolver->advance(dt_C); // Run one continuum step.
  // Extract data from couette solver and send them to MD.
  fillSendBuffer(*couetteSolver,sendBuffer,...);
  multiMDCellService.sendFromMacro2MD(sendBuffer,...);
  for (int i = 0; i < mdInstances; i++) // Run MD instances.
    simpleMD[i]->simulateTimesteps(dt_P,...);
  // Send back data from MD instances and merge it into recvBuffer of
  ↪  this rank. This automatically calls the noise filter.
  multiMDCellService.sendFromMD2Macro(recvBuffer,...);   }
```

Listing 1: Code snippets from C++ noise reduction interface and main coupling loop, outlining its callback methodology, structure and basic usage.

4 Analysis of Simulation Results

Our test scenario, the Couette flow, consists of flow between two infinite parallel plates. A cubic simulation domain with periodic boundaries in x- and y-direction is used. The upper wall at a distance of $z = H$ is at rest, while the lower wall at $z = 0$ moves in x-direction with constant speed $u_{\text{wall}} = 0.5$. The analytical flow solution for the start-up from unit density and zero velocity everywhere can be derived from the Navier–Stokes equations and is given by:

[2] http://eigen.tuxfamily.org.

$$u_x(z,t) = u_{\text{wall}} \left(1 - \frac{z}{H}\right) - \frac{2u_{\text{wall}}}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} \sin\left(k\pi \frac{z}{H}\right) e^{-k^2\pi^2\nu t/H^2} \tag{5}$$

where ν is the kinematic viscosity of the fluid and $u_y = 0, u_z = 0$.

We refer to the smallest scenario that we frequently use as *MD-30*, because it has a MD domain size of $30 \times 30 \times 30$, embedded in a continuum simulation domain with $H = 50$. We always use a continuum (and MD) cell size of 2.5 and time steps of $dt_{\mathrm{C}} = 0.5$, $dt_{\mathrm{P}} = 0.005$. The Lennard-Jones parameters and the molecule mass are set to $m = \sigma = \epsilon = 1.0$. The MD domain is filled with $28 \times 28 \times 28$ particles, this yields a density value $\rho \approx 0.813$ and a kinematic viscosity $\nu \approx 2.63$. *MD-60* and *MD-120* scenarios are defined by doubling or quadrupling domain sizes and particle numbers in all spatial directions, keeping everything else constant.

For a fluctuating x-velocity signal $u(\boldsymbol{x},t)$ we define the signal-to-noise ratio

$$\text{SNR} = 10\ log_{10} \left(\frac{\sum_{\forall \boldsymbol{x},t} \hat{u}(\boldsymbol{x},t)^2}{\sum_{\forall \boldsymbol{x},t} (\hat{u}(\boldsymbol{x},t) - u(\boldsymbol{x},t))^2} \right) \text{dB} \tag{6}$$

with the analytical noiseless x-velocity $\hat{u}$. As SNR is expressed using a logarithmic scale (and can take values equal to or less than zero - if noise level is equal to or greater than signal), absolute differences in SNR correspond to relative changes of squared signal amplitude ratios, so we define the gain of a noise reduction method as

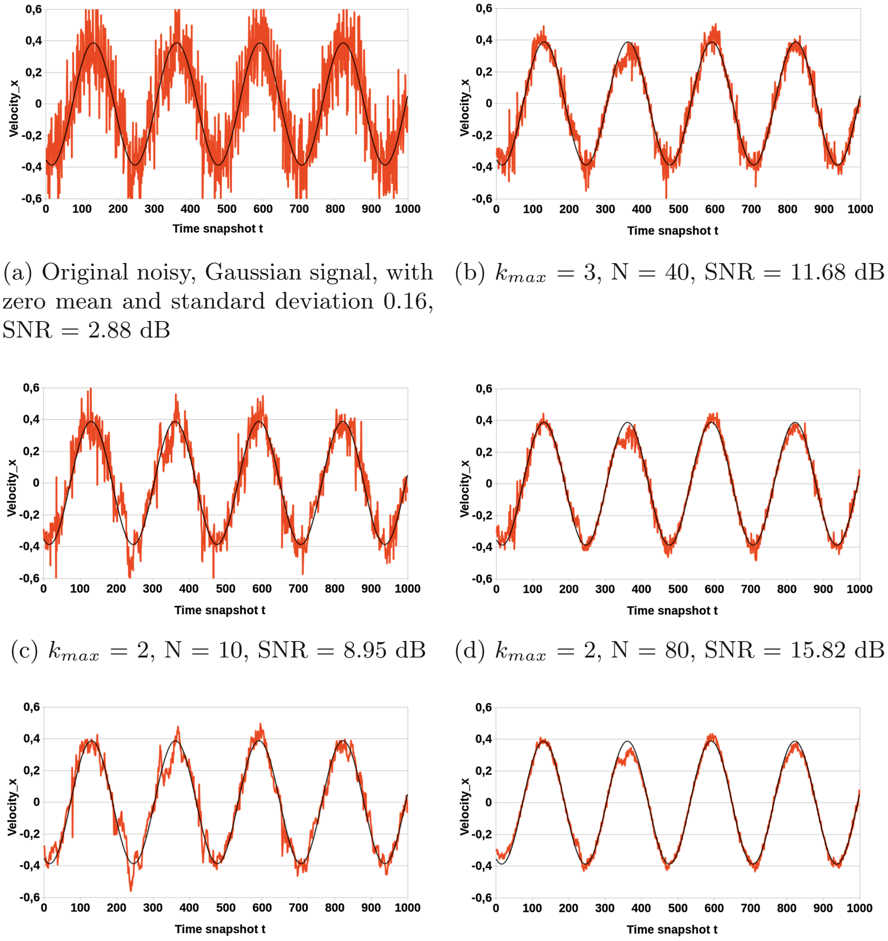
$$\text{gain} = \text{SNR}_{\text{OUT}} - \text{SNR}_{\text{IN}}, \tag{7}$$

with SNR_{OUT} and SNR_{IN} denoting the signal-to-noise ratios of denoised data and original fluctuating signal, respectively.

4.1 Denoising Harmonically Oscillating Flow

To point out and quantify the filtering quality of MaMiCo's new POD noise filter component, we investigate a harmonically oscillating 3D Couette flow. We use the MD-30 scenario and set u_{wall} to a time dependent sine signal. Since we want to investigate only the noise filter here but not the coupled simulation, we use synthetic (analytical) MD data with additive Gaussian noise, without running a real MD simulation. This is not a physically valid flow profile as it disregards viscous shear forces caused by the time-dependent oscillating acceleration, but it is eligible to examine and demonstrate noise filtering performance. We show the influence of varying the POD parameters k_{max} and N in Fig. 3, where the x-component of velocity, for clarity in only one of the cells in the MD domain, is plotted over time; the SNR value is computed over all cells.

The number k_{max} of POD modes used for filtering is considered to be a fixed simulation parameter in this paper, however note that the optimal value of k_{max} depends on the flow features and several methods to choose k_{max} adaptively at runtime by analysing the eigenspectra have been proposed [11,26].



(a) Original noisy, Gaussian signal, with zero mean and standard deviation 0.16, SNR = 2.88 dB

(b) $k_{max} = 3$, N = 40, SNR = 11.68 dB

(c) $k_{max} = 2$, N = 10, SNR = 8.95 dB

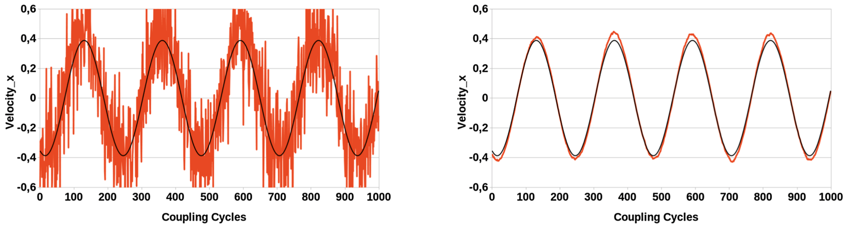
(d) $k_{max} = 2$, N = 80, SNR = 15.82 dB

(e) $k_{max} = 1$, N = 10, SNR = 12.28 dB

(f) $k_{max} = 1$, N = 80, SNR = 19.93 dB

Fig. 3. SNR gain of our POD implementation in MaMiCo, for oscillating 3D Couette flow, using synthetic MD data. Maximum $u_{\text{wall}} = 0.5$. Red: x-component of velocity; Black: True noiseless signal for this cell (Color figure online)

Figure 3f shows the best reduction of fluctuations with a SNR gain of 17.05 dB compared to the input signal Fig. 3a. Only one POD mode is sufficient here, since the sine frequency is low and the flow close to a steady-state, so that higher eigenvalues already describe noise components of the input.



(a) Original noisy signal, SNR = 1.26 dB (b) 8 instances, k_{max} = 1, N = 80, SNR = 21.10 dB

Fig. 4. Oscillating 3D Couette, coupled one-way into a real MD simulation instead of using synthetic data.

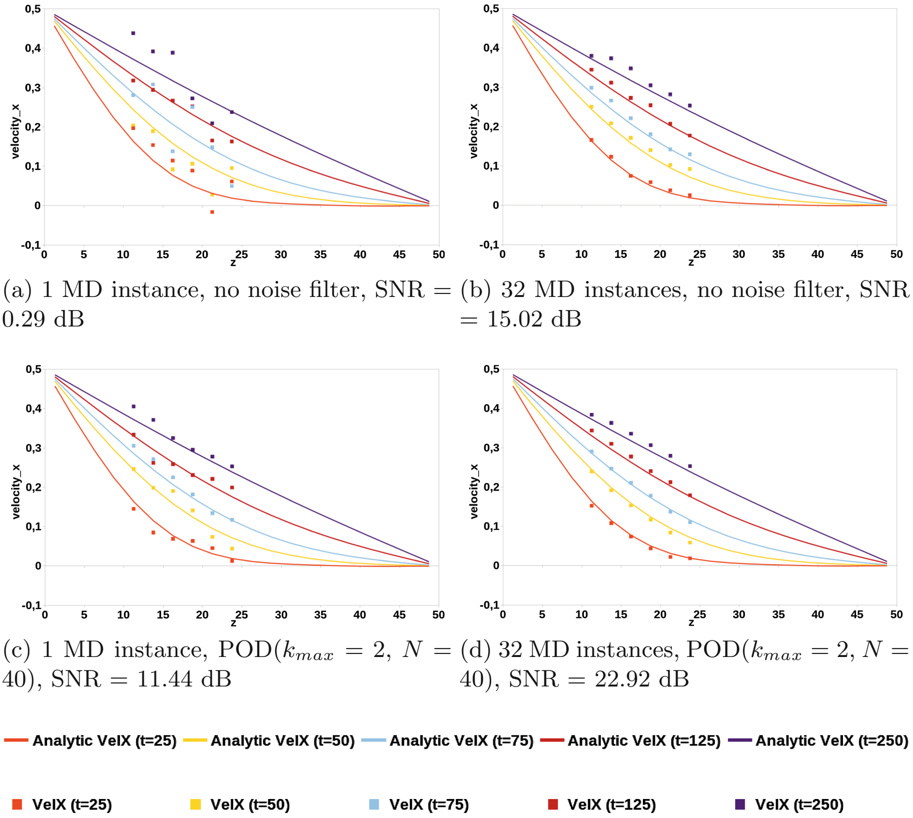
We validate the synthetic MD data test series by repeating the same experiment with a real one-way coupled molecular-continuum simulation. Figure 4a shows the noisy x-velocity signal in one of the cells. In Fig. 4b we additionally enable multi-instance sampling using 8 MD instances and observe a smooth sine output with a SNR gain of 19.84 dB.

4.2 Start-Up of Coupled Couette Flow

We employ a one-way coupled *LBCouetteSolver* → *SimpleMD* simulation running on 64 cores in a MD-30 scenario. The MD quantities that would be returned to the *CouetteSolver* are collected on the corresponding *CouetteSolver* rank. They are compared to the analytical solution in Fig. 5.

Figure 5a shows a very high fluctuation level, because $u_{\text{wall}} = 0.5$ is relatively low compared to thermal fluctuations of MD particles. Figures 5b and c compare multi-instance sampling and noise filtering. The 32 instance simulation is computationally more expensive, but yields a strong gain of 14.73 dB. Theoretically one would expect from multi-instance MD sampling with $I = 32$ instances a reduction of the thermal noise standard deviation by factor $\sqrt{I}$, and a reduction of squared noise amplitude by factor I, so the expected gain is $10\ log_{10}(I)$ dB = 15.05 dB, which is in good compliance with our experimental result. The simulation with POD is using two modes here, which is necessary as data in a singe mode does not capture the fast flow start-up. A smaller comparable gain of 11.15 dB is obtained here, using much less computational resources (see Sect. 4.3). The best result is achieved using a combination of multi-instance sampling and noise filtering shown in Fig. 5d. This novel approach yields a signal-to-noise gain of 22.63 dB for this test scenario, so that the experimentally produced velocity values closely match the analytical solution.

Thus the new combined coupling approach features benefits for both performance and precision. Its ability to extract considerably smoother flow field quantities permits coupling on shorter time scales.



(a) 1 MD instance, no noise filter, SNR = 0.29 dB

(b) 32 MD instances, no noise filter, SNR = 15.02 dB

(c) 1 MD instance, POD(k_{max} = 2, N = 40), SNR = 11.44 dB

(d) 32 MD instances, POD(k_{max} = 2, N = 40), SNR = 22.92 dB

Fig. 5. Couette startup flow profiles, one-way LB → SimpleMD coupling, multi-instance MD versus noise filtering (Color figure online)

4.3 Performance and Scaling Tests

In Table 1 we investigate the performance of the noise filtering subsystem compared to the other components of the coupled simulation. POD always runs with same number of cores as MD. The MaMiCo time includes only efforts for coupling communication – particle insertion and velocity imposition is counted as MD time. The table entries are sampled over 100 coupling cycles each, excluding initializations, using a Couette flow simulation on the LRZ Linux Cluster.

The filter parameter N (time window size) strongly influences the POD runtime, as $C \in \mathbb{R}^{N \times N}$ – and communication and eigenvalue decomposition is performed on C. This leads in practice to a complexity in the order of $\mathcal{O}(N^3)$. Thus, choosing a sufficiently small N is important to limit the computational demand of the method.

Table 1. Impact of noise filter on overall coupled simulation performance. Ratios in the form $\begin{bmatrix} \text{MD} & \text{LB} \\ \textbf{POD} & \text{MaMiCo} \end{bmatrix}$ – given as percentages of total runtime spent in the respective simulation component.

domain size		# cores		N		
LB	**MD**	**LB**	**MD**	**20**	**40**	**80**
100x100x100	**30x30x30**	**1**	**1**	$\begin{bmatrix} 98.90 & 0.68 \\ \mathbf{0.13} & 0.29 \end{bmatrix}$	$\begin{bmatrix} 98.69 & 0.67 \\ \mathbf{0.35} & 0.29 \end{bmatrix}$	$\begin{bmatrix} 97.86 & 0.66 \\ \mathbf{1.19} & 0.29 \end{bmatrix}$
100x100x100	**60x60x60**	**1**	**8**	$\begin{bmatrix} 98.00 & 0.46 \\ \mathbf{0.17} & 1.36 \end{bmatrix}$	$\begin{bmatrix} 97.84 & 0.46 \\ \mathbf{0.33} & 1.36 \end{bmatrix}$	$\begin{bmatrix} 97.23 & 0.46 \\ \mathbf{0.96} & 1.35 \end{bmatrix}$
200x200x200	**120x120x120**	**8**	**64**	$\begin{bmatrix} 69.23 & 0.17 \\ \mathbf{0.37} & 30.24 \end{bmatrix}$	$\begin{bmatrix} 69.15 & 0.17 \\ \mathbf{0.46} & 30.22 \end{bmatrix}$	$\begin{bmatrix} 68.95 & 0.17 \\ \mathbf{0.81} & 30.07 \end{bmatrix}$
400x400x400	**240x240x240**	**512**	**512**	$\begin{bmatrix} 95.67 & 2.62 \\ \mathbf{1.01} & 0.70 \end{bmatrix}$	$\begin{bmatrix} 95.30 & 2.70 \\ \mathbf{1.32} & 0.68 \end{bmatrix}$	$\begin{bmatrix} 94.81 & 2.78 \\ \mathbf{1.71} & 0.70 \end{bmatrix}$

However, the noise reduction runtime is always very low compared to the other simulation components. Particularly with regard to the high gain in signal-to-noise ratio that is reached with relatively little computational effort here, this is a very good result.

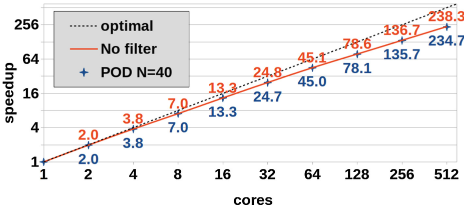


Fig. 6. Strong scaling of coupled simulation, MD-120, including noise filtering. All cores are used for LB, MD and POD, respectively.

The scalability of the coupled simulation, including the noise filter, is evaluated on LRZ CoolMUC-3. The strong scaling tests in Fig. 6 are performed in a fixed MD-120 domain with up to 512 cores used. It is found that our new POD implementation does not significantly impede the scaling, even though it employs a MPI reduction operation. This operation is executed only once per coupling cycle, i.e. every 100 MD time steps, and is restricted to a single MD instance. These results demonstrate the eligibility of our MD data post-processing approach and POD implementation for high-performance computing applications.

5 Conclusions

We have introduced a new noise filtering subsystem into MaMiCo that enables massively parallel particle data post-processing in the context of transient molecular-continuum coupling. Thanks to MaMiCo's modular design, it is compatible with any particle and continuum solver and can be utilized in conjunction with MD multi-instance sampling. Experiments with flow start-up profiles validate our coupled simulation. SNR considerations for oscillating flow demonstrate the filtering quality of our POD implementation. The noise filtering interface and implementation are very scalable and have only a minimal impact on the overall coupled simulation performance.

We observed SNR gains from the POD filter ranging roughly from 11 dB to 17 dB. This corresponds to a potential simulation performance increase by MD instance reduction by a factor of 10 to 50. However, we point out that a combination of both, POD and multi-instance sampling, yields even more smooth quantities and thus enables coupling on shorter time scales, while yielding a higher level of parallelism than a POD-only approach.

As our interface is very generic and flexible, a possible future work suggesting itself would be to conduct experiments with more noise filtering algorithms. Especially the area of image processing offers many promising methods which could be applied to CFD instead of image data. Besides, further data analytics and particle analysis tasks may be tackled in the future, such as flow profile extraction, pressure wave and density gradient detection, or specific molecular data collection modules, e.g. for determination of radial distribution functions. Resilient MD computations by a fault-tolerant multi-instance system with the ability to recover from errors are also conceivable, as well as a machine learning based extension extracting more smooth quantities by learning and detection of prevalent flow features.

Acknowledgements. The provision of compute resources through the Leibniz Supercomputing Centre is acknowledged. The authors further acknowledge financial support by the Federal Ministry of Education and Research, Germany, project *"Task-based load balancing and auto-tuning in particle simulations"* (TaLPas), grant number 01IH16008B.

References

1. Barsky, S., Delgado-Buscalioni, R., Coveney, P.V.: Comparison of molecular dynamics with hybrid continuum-molecular dynamics for a single tethered polymer in a solvent. J. Chem. Phys. **121**(5), 2403–2411 (2004)
2. Borg, M.K., Lockerby, D.A., Reese, J.M.: A hybrid molecular-continuum method for unsteady compressible multiscale flows. J. Fluid Mech. **768**, 388–414 (2015). https://doi.org/10.1017/jfm.2015.83
3. Borgdorff, J., et al.: Distributed multiscale computing with MUSCLE 2, the multiscale coupling library and environment. J. Comput. Sci. **5**(5), 719–731 (2014)
4. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005, CVPR 2005, vol. 2, pp. 60–65 (2005)
5. Delgado-Buscalioni, R., Coveney, P.: USHER: an algorithm for particle insertion in dense fluids. J. Chem. Phys. **119**(2), 978–987 (2003)
6. Delgado-Buscalioni, R., Coveney, P.V., Riley, G.D., Ford, R.W.: Hybrid molecular-continuum fluid models: implementation within a general coupling framework. Philos. Trans. Roy. Soc. Lond. A Math. Phys. Eng. Sci. **363**(1833), 1975–1985 (2005)

7. Dupuis, A., Kotsalis, E., Koumoutsakos, P.: Coupling lattice Boltzmann and molecular dynamics models for dense fluids. Phys. Rev. E **75**(4), 046704 (2007)
8. Weinan, E., Engquist, B., Li, X., Ren, W., Vanden-Eijnden, E.: Heterogeneous multiscale methods: a review. Commun. Comput. Phys. **2**(3), 367–450 (2007)
9. Español, P.: Hydrodynamics from dissipative particle dynamics. Phys. Rev. E **52**, 1734–1742 (1995)
10. Gonnet, P.: A simple algorithm to accelerate the computation of non-bonded interactions in cell-based molecular dynamics simulations. J. Comput. Chem. **28**(2), 570–573 (2007)
11. Grinberg, L.: Proper orthogonal decomposition of atomistic flow simulations. J. Comput. Phys. **231**(16), 5542–5556 (2012)
12. Koumoutsakos, P.: Multiscale flow simulations using particles. Annu. Rev. Fluid Mech. **37**, 457–487 (2005)
13. Neumann, P., Bian, X.: MaMiCo: transient multi-instance molecular-continuum flow simulation on supercomputers. Comput. Phys. Commun. **220**, 390–402 (2017)
14. Neumann, P., Tchipev, N.: A coupling tool for parallel molecular dynamics-continuum simulations. In: 2012 11th International Symposium on Parallel and Distributed Computing (ISPDC), pp. 111–118 (2012)
15. Ngandjong, A.C., Rucci, A., Maiza, M., Shukla, G., Vazquez-Arenas, J., Franco, A.A.: Multiscale simulation platform linking lithium ion battery electrode fabrication process with performance at the cell level. J. Phys. Chem. Lett. **8**(23), 5966–5972 (2017)
16. Nie, X., Chen, S., Robbins, M.O., et al.: A continuum and molecular dynamics hybrid method for micro-and nano-fluid flow. J. Fluid Mech. **500**, 55–64 (2004)
17. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
18. Praprotnik, M., Site, L.D., Kremer, K.: Multiscale simulation of soft matter: from scale bridging to adaptive resolution. Annu. Rev. Phys. Chem. **59**(1), 545–571 (2008)
19. Ren, X.G., Wang, Q., Xu, L.Y., Yang, W.J., Xu, X.H.: HACPar: an efficient parallel multiscale framework for hybrid atomistic-continuum simulation at the micro-and nanoscale. Adv. Mech. Eng. **9**(8), 1–13 (2017)
20. Ritos, K., Borg, M.K., Lockerby, D.A., Emerson, D.R., Reese, J.M.: Hybrid molecular-continuum simulations of water flow through carbon nanotube membranes of realistic thickness. Microfluid. Nanofluid. **19**, 997–1010 (2015)
21. Sirovich, L.: Turbulence and the dynamics of coherent structures. I. Coherent structures. Q. Appl. Math. **45**(3), 561–571 (1987)
22. Smith, E.: On the coupling of molecular dynamics to continuum computational fluid dynamics (2013)
23. Tang, Y.H., Kudo, S., Bian, X., Li, Z., Karniadakis, G.E.: Multiscale universal interface: a concurrent framework for coupling heterogeneous solvers. J. Comput. Phys. **297**, 13–31 (2015)
24. Werder, T., Walther, J.H., Koumoutsakos, P.: Hybrid atomistic-continuum method for the simulation of dense fluid flows. J. Comput. Phys. **205**(1), 373–390 (2005)
25. Zhou, W., Luan, H., He, Y., Sun, J., Tao, W.: A study on boundary force model used in multiscale simulations with non-periodic boundary condition. Microfluid. Nanofluid. **16**(3), 1 (2014)
26. Zimoń, M., et al.: An evaluation of noise reduction algorithms for particle-based fluid simulations in multi-scale applications. J. Comput. Phys. **325**, 380–394 (2016)