Application of Incremental Proper Orthogonal Decomposition for the Reduction of Very Large Transient Flow Field Data

Daiki Matsumoto 1) Marco Kiewat 1) Christoph A. Niedermeier 2) Thomas Indinger 1)
1) Technical University of Munich, Chair of Aerodynamics and Fluid Mechanics, Boltzmannstrasse 15, D-85748, Garching bei Muenchen, Germany (E-mail: daiki.matsumoto@tum.de)
2) Fluidyna GmbH, Edisonstrasse 3, D-85716, Unterschleissheim, Germany

Received on August 23, 2018

ABSTRACT: With the increase of available computer performance, unsteady Computational Fluid Dynamics (CFD) is now widely used for industrial applications. For the analysis of unsteady vehicle aerodynamics, massive data storage is required for saving time series of spatially highly resolved flow fields. The size of these transient datasets can be significantly reduced using the Incremental Proper Orthogonal Decomposition (POD) by computing POD modes in parallel to the CFD. In this paper, we present a successful approximation of the transient flow field using a reduced number of modes computed by Incremental POD.

KEY WORDS: heat · fluid, computational fluid dynamics, Reduced Order Modeling, Proper Orthogonal Decomposition [D1]

1. Introduction

In recent years, the improvement of high-performance computing hardware enabled the utilization of unsteady Computational Fluid Dynamics (CFD), such as Large Eddy Simulation (LES) and Detached Eddy Simulation (DES), also for industrial applications. Unsteady CFD can not only provide highly accurate steady-state results, but also helps to understand the transient phenomena that exist in the flow field. One advantage of unsteady CFD especially lies in its application for road vehicle aerodynamics, since the strong transient flow fluctuations around a vehicle can severely affect its driving stability and ride comfort. One prominent example for such a negative influence are the low frequency fluctuations created by gusty crosswinds. 1)-(2)

A difficulty in the analysis of the transient flow field by CFD is that the time series of flow field data typically needs to be saved to disk during and after a simulation. This often requires massive storage space, as the transient flow field data around a vehicle is spatially highly resolved to capture complex flow structures consisting of various time and length scales. It is very costly to perform even a single unsteady aerodynamics simulation with respect to not just computational power requirements but also the demands for disk storage space. In addition, such a high demand for storage space makes it very cumbersome to archive the transient flow field data of multiple unsteady CFD simulations for a longer period of time.

One possible solution to reduce the total amount of data is to approximate a transient flow field in reduced order. Proper Orthogonal Decomposition (POD) is a well-known data-driven modal analysis method that is often used for reduced-order modeling of the flow field. 3)-(4) POD decomposes the transient flow field into the orthogonal basis, which is often used for seeking energetic coherent flow structures since its first proposal by Lumley 3). For example, Manhart et al. applied POD on the flow field around the surface-mounted cube in the channel for the analysis of the dominant flow structures. 4), while Muld et al. applied POD on the aerodynamics around a high-speed train to identify the coherent structures causing slipstream. 5)

However, one can not only use POD for these analyses of the transient flow structures, but also to approximate and subsequently archive the transient flow field data in reduced form. Nevertheless, the conventional POD algorithm still consumes massive computational resources. Firstly, the full set of the transient flow field data needs to be saved to disk during the CFD simulation, which often requires huge amounts of disk space due to the highly resolved flow field regarding both time and space. Secondly, all the saved data needs to be loaded into the main memory (RAM) at the same time to compute the POD modes, which results in very huge RAM requirements. In contrast to that, Incremental Proper Orthogonal Decomposition (IPOD) or Incremental Singular Value Decomposition) updates modes incrementally when new snapshot data is available. Therefore, it is not necessary to save all the snapshots of the transient flow field to disk during a CFD simulation. Additionally, IPOD occupies much less RAM than conventional POD, since only truncated POD modes have to be loaded into the RAM during the IPOD computation.

Another issue is that any data-driven method of modal analysis is often very sensitive to different parameter setups. This sensitivity depends on the complexity of the target datasets. In other words, if the target flow field is very complex, a modal analysis algorithm without proper parameters can easily produce irrelevant results. Additionally, incremental algorithms of modal analysis are even more sensitive to different parameter setups and as already mentioned above the simulated transient flow field around a road vehicle is very complex, because it consists of various time and length scales. As a consequence, the application of IPOD on vehicle aerodynamics is very challenging in terms of the parameter setup.

In this paper, we show that IPOD modes can approximate the complex transient flow field around a road vehicle model using a proper parameter setup, which results in a successful data reduction of transient datasets. The validation is conducted using the DrivAer
2. Proper Orthogonal Decomposition Algorithms

Proper Orthogonal Decomposition\(^{(3)-(5)}\) is one of the well-known data-driven modal analysis methods, it decomposes the datasets into the orthogonal basis. In this section, we introduce the algorithm of conventional POD. Then, the relatively new incremental algorithm of POD is introduced. Finally, the post-processing strategy of POD modes is also explained.

2.1. Proper Orthogonal Decomposition

Some modal analysis methods in the area of fluid mechanics approximate the discrete temporal-spatial field values sampled in the domain by the finite combination of linear modes with coefficients as shown in the following.

\[ z(x, t) = \sum_{k=0}^{m} a_k(t) u_k(x) \]  

(1)

The choice of spatial modes \( u_k(x) \) and corresponding time coefficients \( a_k(t) \) shown above is not unique. Proper Orthogonal Decomposition (POD) chooses the orthogonal basis for the spatial modes \( \mathbf{U} = [u_0(x), ..., u_m(x)] \) for the expansion. The orthogonal basis extracts the flow structures hierarchically with respect to the coherence through the matrix diagonalization. Therefore, the computed POD mode matrix \( \mathbf{U} \) becomes orthogonal.

\[ \mathbf{U}^* \mathbf{U} = \mathbf{U} \mathbf{U}^* = \mathbf{I} \]  

(2)

where \( \mathbf{I} \) represents an identity matrix and \( \mathbf{U}^* \) represents the transpose of the matrix \( \mathbf{U} \). These POD modes are computed by the eigendecomposition of the auto-covariance of the snapshot matrix \( \mathbf{Z} = [z_0(x), ..., z_{m-1}(x)] \in \mathbb{R}^{n \times m} \) consisting of \( m \) snapshots of the \( n \)-dimensional field.

\[ \mathbf{Z}^* \mathbf{Z} \mathbf{U} = \mathbf{U} \mathbf{\Sigma}^2 \]  

\[ \mathbf{\Sigma} \in \mathbb{R}^{m \times m} \]  

(3)

However, it is often impossible to compute the above auto-covariance of the snapshot matrix, since the simulated complex flow field consists of enormous \( n \)-dimensional components in space. For this reason, the method of snapshot\(^{(7)}\) is adopted to compute the POD modes in practice. This computes POD modes by the eigendecomposition of the auto-covariance matrix in reversed direction.

\[ \mathbf{Z}^* \mathbf{Z} \mathbf{W} = \mathbf{W} \mathbf{\Sigma}^2 \]  

\[ \mathbf{\Sigma} \in \mathbb{R}^{m \times m} \]  

(4)

Then, the POD modes are computed as follows.

\[ \mathbf{U} = \mathbf{Z} \mathbf{W} \mathbf{\Sigma}^{-1} \]  

(5)

The method of snapshot requires much less computational resources than the original POD algorithm. Instead, only the first \( m \) modes out of \( n \) modes are computed at maximum, which results in the bi-orthogonal POD mode matrix.

Considering the analogy of the method of snapshot to the Singular Value Decomposition (SVD), the left singular vector of the snapshot matrix \( \mathbf{Z} \) can be interpreted as POD modes.

\[ \mathbf{Z} = \mathbf{U} \mathbf{\Sigma} \mathbf{W}^* \]  

(6)

Both the method of snapshot and the SVD were often adopted for POD analysis so far, however, they both consume huge amounts of RAM, as the processed snapshot matrix must be loaded into the RAM at once. In addition, the full set of the processed datasets must be saved to disk during the CFD simulation. This again shows that processing highly complex transient flow field data is not an easy task.

2.2. Incremental Proper Orthogonal Decomposition

The most severe part of the POD application in practice is the huge consumption of RAM to load the full set of the flow field data. An incremental update and a truncation of the POD modes can both reduce the memory consumption for the mode computation. This algorithm is called Incremental Singular Value Decomposition (ISVD) or Incremental Proper Orthogonal Decomposition (IPOD)\(^{(6)-(7)}\). This algorithm computes the singular vectors of datasets incrementally when new data is available. Please note that left singular vectors basically have the same meaning as POD modes as mentioned above.

We adopted the ISVD algorithm proposed by Oxberry et al.\(^{(7)}\), which is based on the algorithm by Brand\(^{(6)}\). This paper adopts the one-step update version of ISVD, while there is another multi-time step update version which is faster but consumes more memory. Here, the updating processes of singular values and vectors in the IPOD are briefly presented. Please refer to (6) and (7) for a more detailed description of the algorithm and its mathematical backgrounds.

The existing \( r \)-rank SVD matrices shown as Eq. (6) are extended when the new snapshot vector \( \mathbf{z}_i \) is obtained.

\[ \begin{bmatrix} \mathbf{Z} & \mathbf{z}_i \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{q} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{l} \end{bmatrix} \begin{bmatrix} \mathbf{W} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{0}^T \end{bmatrix} \]  

where \( p = \| (I - \mathbf{U} \mathbf{U}^*) \mathbf{z}_i \|. \mathbf{l} = (I - \mathbf{U} \mathbf{U}^*)\mathbf{z}_i / p \) and \( \mathbf{I} = \mathbf{U}^* \mathbf{z}_i \).

The middle matrix is diagonalized by the additional SVD, as shown in the following.

\[ \begin{bmatrix} \mathbf{\Sigma} & \mathbf{l} \\ \mathbf{0} & \mathbf{p} \end{bmatrix} = \mathbf{U}_s \mathbf{\Sigma}_s \mathbf{W}_s \]  

(7)

Then, the singular vectors and values are updated as follows.

\[ \begin{bmatrix} \mathbf{U}_{\text{new}} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{q} \end{bmatrix} \mathbf{U}_s \]  

(8)

\[ \mathbf{\Sigma}_{\text{new}} = \mathbf{\Sigma}_s \]  

(9)

\[ \mathbf{W}_{\text{new}} = \begin{bmatrix} \mathbf{W} & 0 \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{W}_s \]  

(10)

The above processes are repeated until all vectors in the datasets are processed. When the rank of modes reaches the pre-defined maximum rank, the truncation of vectors is performed, which results in less memory consumption.

2.3. Reconstruction from POD modes

The original transient flow field is reconstructed from POD modes by the inverse process of SVD, which is the matrix multiplication of singular values and vectors shown in Eq. (6). One important point in practice is that the reconstruction of the field from the mode element is spatially not interdependent. Therefore, the reconstruction is not necessary to be performed for the entire space and time. In other words, the time series of the partial
components of the flow field, such as velocity vectors measured on the cutting plane or on probes, can be reconstructed from preliminarily measured partial POD modes there. This also helps to reduce the memory consumption for the reconstruction. If the target partial domain for reconstruction is small enough, it may be possible to perform this process on a small machine such as a laptop.

3. Numerical flow simulation

POD is applied on the transient flow field around a DrivAer model, this flow field is numerically simulated using a GPU-based LBM solver. In this section, the computational setup and the results of the simulation are presented, the numerical flow simulation is the same as the one presented at the 2nd Shanghai-Stuttgart-Symposium.\(^{(18)}\)

3.1. DrivAer model

In order to reproduce a realistic application case for transient CFD, the flow field is numerically simulated around the DrivAer model which is a realistic and generic car model developed basically as a combination of an Audi A4 and a BMW 3 series at the Technical University of Munich.\(^{(19)}\) This model may help to close the gap between relatively low-detailed geometries such as the Ahmed body adopted at research level and complex geometries investigated at production level but which have limited access from academic side. This geometry is publicly available from the website of the Chair of Aerodynamics and Fluid Mechanics.\(^{(10)}\) The DrivAer model was recently extended to reproduce engine bay flows to simulate more realistic operating conditions.\(^{(12)}\) In this paper, we present a numerical simulation of the latest DrivAer notch-back model with closed rims and with engine bay flow according to the setup by Collin et al.\(^{(15)}\) In addition, the wheel house to the engine bay is opened. Experimental and numerical simulations for this setup are presented in (15), these are used for the validation of the simulation here.

3.2. Numerical methods

The numerical flow simulation is performed with a GPU-based LBM solver, ultraFluidX, developed by Fluidyna GmbH, Germany.\(^{(19)}\) This solver is mainly designed for the external aerodynamics simulation of road vehicles and optimized for massively parallel computation on NVIDIA GPUs. The standard Smagorinsky model is used to perform Large Eddy Simulation (LES) to compute the transient phenomena in the flow field with high accuracy, while a wall model based on the Turbulent Boundary Layer Equations (TBLE) ensures consistent near-wall treatment at high Reynolds numbers.

3.3. Computational setups

As mentioned above, the 40% scale notch-back DrivAer model with engine bay (Fig. 1 (a)) is simulated. The geometry of the wind tunnel such as the nozzle, the top sting attached to the model and the wheel struts are not reproduced in the numerical simulation. The center of the coordinate system in the domain is located in the middle of the wheel base with respect to the x- and z-coordinates and the center line of the car with respect to the y-coordinate.

The inflow velocity is \( u = 44 \text{ m/s} \) (158.4km/h), which gives a Reynolds number of \( 3.0 \times 10^6 \) with respect to the reference length of 1.11m (wheel base). The wheel rotation and the moving ground are reproduced by a velocity boundary condition to mimic the wind tunnel experiment. The size of the moving ground in the simulation is the same as in the experiment.

The radiator model installed in the engine bay causes a pressure drop of the incoming flow. Collin et al.\(^{(15)}\) reproduced the different pressure drop conditions in the experiment by changing the openness rate on the perforated plate installed just behind the front grill of the car. In the simulation, the effect on the incoming flow in the radiator is modeled by a pressure loss according to the Darcy-Forchheimer equation. The coefficients are determined based on the measured pressure loss in the case of an experimental openness rate of \( Ao/Ac = 50.3\% \).\(^{(15)}\)

The computational grid is automatically generated by the solver based on the input geometry with multiple volumetric refinement regions defined by the user. In total, 272 million voxels are generated with a finest voxel size of 1mm close to the surface of the vehicle. An overview of the grid around the vehicle is shown in Fig. 1(b)-c).

![Image](a) DrivAer model. (b) Computational grid around the engine bay. (c) Computational grid around the vehicle at \( y = 0.0\text{m} \)

Fig. 1 DrivAer model and surrounding computational grids.

3.4. Results of the flow simulation

In this subsection, the results of the numerical simulation are discussed. The transient flow field was numerically simulated on 10 GPUs (NVIDIA P100) on a single computer server for 3.2 sec of physical time.

The time-averaged aerodynamic forces acting on the vehicle are compared with the measurement in the wind tunnel presented in (15). The force coefficients of drag \( C_D \), front lift \( C_{LF} \) and rear lift \( C_{LR} \) are presented in Fig. 2 in comparison to experimental results for the same setup. As a reference, the experimental values of the mock-up model are also shown in Fig. 2. Please note that the values for \( C_{LF} \) and \( C_{LR} \) don’t include forces acting on the tires and rims, while \( C_D \) includes them. According to Fig. 2, the \( C_D \) value from our simulation is in very good agreement with the experimental results, especially considering the inexistence of detailed wind tunnel geometries in the simulation. We observe a slight underestimation for both \( C_L \) values from our simulation compared to the experiment.

Copyright © 2019 Society of Automotive Engineers of Japan, Inc. All rights reserved
4. Results of the POD computation

In this section, the results of the POD on the simulated transient flow field are discussed. IPOD is applied on the velocity vector field sampled in the pre-defined subvolume region close to the vehicle which includes about 215 million voxels.

4.1. Computational setup for POD

As explained before, the IPOD computation is often very sensitive to different parameter setups and improper settings often result in irrelevant POD modes. Therefore, preprocessing of the dataset before computing the POD modes is one of the most important factors to compute relevant modes, especially when the dataset is very complex or noisy. In this paper, a temporal low-pass filter is applied on the dataset before IPOD processing, because IPOD without any filtering often results in irrelevant modes or modes outside of the target range, especially when the dataset is very complex as mentioned above and only relatively few modes are computed regarding the complexity. Another important parameter for IPOD is the pre-defined maximum rank of the computed IPOD modes, because the more IPOD modes remain, the more computational time and memory are required. On the other hand, the rank cannot be extended after finishing the IPOD computation. Therefore, the proper maximum rank for the POD should be pre-defined depending on the processed flow properties considering the available resources.

In this paper, a total of 1,001 snapshots of the velocity vector field are sampled with the constant interval of 2.691 × 10^{-3} sec for IPOD processing. Sampling of the field starts after 1.345 sec from the beginning of the simulation, where the flow fluctuations already become stable. Those velocity fields are low-pass filtered at 20 Hz cutoff frequency by using a 2nd order on-the-fly Butterworth filter. This filtering process is integrated in the IPOD algorithm and filters a snapshot just before updating the singular vectors and value in each iteration process. One issue in one-directional analog filter is that a signal at the beginning is not effectively filtered due to the filter delay. Therefore, the first 100 snapshots are used just for computing the filter delay and the remaining 901 filtered snapshots are processed by the IPOD in this paper.

The number of computed POD modes has been conventionally determined according to the variance contribution rate of eigenvalues, which indicates the contribution rate of the extracted mode with respect to the total energy of the reconstructed field. However, it is difficult to derive the variance contribution of eigenvalues by IPOD, because the number of modes must already be chosen before the actual IPOD computation. Based on our experience with using the conventional POD on the temporally filtered flow field around a vehicle with similar complexity as shown in this work, using 100 POD modes from around 900 snapshots extracts more than 99% of the variance contribution rate. Therefore, we finally computed 100 modes by IPOD from the filtered snapshot matrix of the velocity vector.

Here, the IPOD computation is performed as a separate post-processing step after the actual CFD simulation, for which it finally occupies 2.2 TB RAM, and takes 20 hours with 480 cores (Intel Xeon E7-8870 10C) to compute 100 modes from 901 snapshots. The memory consumption is estimated to be roughly less than 20% of the conventional POD algorithm. However, if the IPOD would be performed in parallel to the CFD simulation, the required resources could be even smaller. One reason is that the number of vector elements processed in this paper, which is 215 million cells, is more than necessary to capture the interesting phenomena at low frequencies targeted in this paper. The second reason is that the data loading and data distribution are major parts of the overall time consumption. Roughly more than 60% of the computational time is data loading from disk storage and data distribution to each processor. Therefore, the computational time could be much shorter if the dataset directly stayed in the RAM during the CFD simulation, and the data distribution processes in the code could be more sophisticated.

Another benefit would be that only the computed POD modes would be saved to disk instead of the full data set of the transient flow field, and this would require less time for the writing process. Yet, the actual IPOD computation of course also takes some time, therefore, it will depend on the ratio between additional time for the computation and saved time for the writing if there would be an overall benefit or penalty by using the IPOD in parallel to the CFD simulation. However, this question is beyond the scope of this paper, because our current IPOD implementation has not been optimized for maximum performance and for running in parallel with the CFD simulation yet, and both the compute performance and the I/O speed of the hardware would be crucial for this
investigation and require benchmarking on different systems for comprehensive results.

Using the IPOD as a post-processing tool in this work, the amount of disk-saved data is reduced to about 11% compared to the full original dataset of the transient flow field, by remaining 100 IPOD modes out of 901 snapshots of the velocity vector field.

4.2. Probe Velocity

The time series of velocity components measured at some probes are reconstructed from the computed IPOD modes. The probe velocity is reconstructed from modes which are preliminarily sampled at the target probe. This process saves memory and computational time for the reconstruction. Fig. 4 shows the locations of the sampling probes, where Probe.A is located at \((x, y, z) = (1.4m, 0.2m, 0.2m)\) and Probe.B is located at \((1.0m, 0.2m, 0.0m)\) in the computational domain.

First, sequences of the streamwise velocity are reconstructed from IPOD modes at the probe A. One sequence is reconstructed from 50 modes (red line in Fig. 5) and another sequence is reconstructed from 100 modes (blue line in Fig. 5) out of 100 resultant IPOD modes. Those reconstructed velocity sequences are compared with the original but temporally filtered streamwise velocity as processed before IPOD (black dashed line in Fig. 5). According to Fig. 5, some peaks of relatively high frequency fluctuations are not perfectly captured in the reconstructed sequence by 100 modes, such as observed in between 1.9 sec and 2.4 sec. However, the reconstructed sequence by 100 modes captures most of the characteristic fluctuations compared to the original sequence. The sequence by 50 modes seems to capture many characteristic fluctuations, however, amplitudes of more peaks are underestimated compared to the sequence by 100 modes.

Additionally, spectra of these streamwise velocity sequences are shown in Fig. 6, which are computed by a Fast Fourier Transform (FFT) with the Hanning window function. According to Fig. 6, the spectrum of the reconstructed sequence by 100 modes is mostly in agreement with the original sequence. The reconstructed sequence by 50 modes also shows the similarity, but an underestimation of the amplitude around 6-8Hz and an overestimation at 8Hz and 16Hz can be observed. Therefore, 50 modes are not enough to reproduce the characteristic fluctuations, while 100 modes seem to be sufficient.

For further discussion, the lateral (y-component of velocity) and vertical (z-component of velocity) are reconstructed at Probe.B where flow fluctuations seem to be more complex than at Probe.A due to the vortex shedding from the body. When the fluctuations of the flow include more frequency components, more POD modes seem to be required to approximate the flow field. In the same manner as the above discussion about the streamwise velocity at Probe.A, sequences of lateral and vertical velocity at Probe.B are reconstructed from 50 and 100 IPOD modes, respectively (Fig. 7 and Fig. 9). In addition, spectra of these sequences are computed by FFT with windowing and shown in Fig. 8 and Fig. 10.

According to Fig. 7, the sequence of the reconstructed lateral velocity from 100 IPOD modes doesn’t reproduce some peaks of high-frequency fluctuations existing in the original sequence, especially after 2 sec. This discrepancy seems to appear as the underestimation of Power Spectral Density (PSD) around 13Hz-15Hz and 19Hz in Fig. 8. In addition, some overestimated PSD in the reconstructed sequence by 100 modes are observed around 18Hz in Fig. 8, which is probably caused by the fluctuations around 0.8sec and 1.2sec in Fig. 7. However, it seems that still most of the characteristic fluctuations are well approximated by 100 modes. On the other hand, 50 modes are again not enough to reproduce the characteristics of these relatively complex fluctuations, since PSD at most frequencies are too underestimated as shown in Fig. 8.

Regarding the vertical velocity shown in Fig. 9, the reconstructed sequence by 100 modes doesn’t reproduce some peaks observed in the original sequence, which is a similar tendency as for the lateral velocity discussed above. These discrepancies observed in between 8Hz and 10Hz and over 12Hz...
are the over- and underestimations of PSD in Fig. 10. However, it seems that still most of the characteristic fluctuations of the vertical velocity are again well approximated by 100 modes, while 50 modes are not enough.

In summary, the reconstructed velocity field by 100 IPOD modes mostly reproduces the transient characteristics of the original field. In contrast, 50 modes are not enough to reproduce the transient characteristics, which appears as large discrepancies in PSD especially in the complex fluctuations.

**Fig. 7** Sequences of lateral velocity at Probe.B in the original field (dashed black line) and in the reconstructed field from 50 modes (solid red line) and 100 modes (solid blue line).

**Fig. 8** Spectra of the lateral velocity at Probe.B in the original field (black stick) and in the reconstructed field from 50 modes (red bar) and 100 modes (blue bar).

**Fig. 9** Sequences of vertical velocity at Probe.B in the original field (dashed black line) and in the reconstructed field from 50 modes (solid red line) and 100 modes (solid blue line).

**Fig. 10** Spectra of the vertical velocity at Probe.B in the original field (black stick) and in the reconstructed field from 50 modes (red bar) and 100 modes (blue bar).

4.3. Cutting Plane

The transient characteristics at the probes are mostly captured in the reconstructed field by using 100 IPOD modes, as discussed above. For a further discussion of the spatial structure of the reconstructed field, the distribution of the velocity components is reconstructed at some planes and compared with the original field.

The distribution of the velocity components is reconstructed from preliminarily sampled 100 IPOD modes on the target plane, just as the reconstruction at the probes. Fig. 11, Fig. 12 and Fig. 13 show the reconstructed streamwise velocity at \( y = 0.2 \) m, vertical velocity at \( y = 0.2 \) m, and lateral velocity at \( z = 0.2 \) m, respectively. Figs. 11-13 also show the same component of the original velocity as reconstructed field, but those are already filtered as processed by the IPOD. Here, only 2 snapshots at different time steps are presented for each component of the velocity.

With respect to the streamwise velocity at \( y = 0.2 \) m shown in Fig. 11, the distribution of the reconstructed field is very similar to the original field across the entire plane. One small difference between the original and the reconstructed field is the shape of the negative velocity region just behind the vehicle, at both time steps of 0.000sec and 1.6121sec.

The vertical velocity distribution at \( y = 0.2 \) m is more complex, especially just behind the vehicle due to the vortex shedding from the body. According to Fig.12, the reconstructed vertical velocity is slightly overestimated compared to the original just behind the vehicle at 0.0000sec. In addition, the shape of the velocity distribution at 0.0000sec at the region one vehicle length behind is slightly different between the reconstructed and the original field. The reconstructed velocity far behind the vehicle at 1.6121sec is slightly underestimated. However, the velocity distributions in most regions are well approximated by 100 IPOD modes, even though slight differences in the magnitude and shape can still be observed.

The lateral velocity distribution at \( z = 0.0 \) m is even more complicated as shown in Fig. 13. Therefore, overestimation and underestimation of the velocity can be observed in the reconstructed field at both 0.0000sec and 1.6121sec. Nevertheless, the flow structures are mostly well approximated by IPOD modes.

Overall, the combination 100 IPOD modes approximates almost all components of the original velocity field with respect to the transient flow structures in space.
In this paper, the Incremental Proper Orthogonal Decomposition (IPOD) is applied on the transient velocity field around the DrivAer model simulated by a GPU-based LBM solver. The computed IPOD modes successfully approximate the complex transient flow field around the vehicle with respect to both the transient characteristics and the instantaneous flow structures. The amount of transient flow field data is reduced to 11% of the original size with the presented setups. This IPOD computation occupies roughly 80% less memory than the conventional POD algorithm. Unlike conventional POD, the memory consumption of the IPOD is unchanged as long as the maximum rank of the modes is the same, even if more snapshots of the flow field are processed.

However, there are still some issues regarding the application of the IPOD in practice. One issue is that the IPOD still requires pre-processing such as temporal filtering to approximate the field by a limited number of modes. This is because of both the difficulty to capture high-frequency fluctuations by just a few POD modes in general and also the sensitivity of the IPOD algorithm to different parameter setups. Using more POD modes to capture higher frequencies would be possible, but this would thwart the goal of reducing the size of the transient flow field data. Another issue is that this IPOD still occupies a huge amount of RAM for computing only 100 modes out of 900 snapshots, which is still difficult to be performed on a small cluster in parallel to a CFD simulation. This issue is related to the limitation of the approximation by POD modes. To solve this issue, it would be necessary to find a different way of expansion of the complex flow field to be approximated with fewer modes than the orthogonal modes used in this paper.
This paper is written based on a proceeding presented at JSAE 2018 Annual Congress (Spring).

References

(1) Theissen, P., Wojciak, J., Heuler, K., Demuth, R., Indinger, T., and Adams, N., Experimental Investigation of Unsteady Vehicle Aerodynamics under Time-Dependent Flow Conditions - Part 1, SAE Technical Paper No.2011-01-0177. (2011).
(2) Wojciak, J., Theissen, P., Heuler, K., Indinger, T., Adams, N., and Demuth, R., Experimental Investigation of Unsteady Vehicle Aerodynamics under Time-Dependent Flow Conditions - Part 2, SAE Technical Paper No.2011-01-0164. (2011).
(3) Lumley, J. L. The structure of inhomogeneous turbulent flows. Atmospheric turbulence and radio wave propagation. (1967)
(4) Manhart, M., & Wengle, H. A spatiotemporal decomposition of a fully inhomogeneous turbulent flow field. Theoretical and computational fluid dynamics, Vol.5, pp.223-242. (1993).
(5) Muld, T. W., Efraimsson, G., & Henningson, D. S., Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. Computers & Fluids, Vol.57, pp.87-97. (2012).
(6) Brand, M., Incremental singular value decomposition of uncertain data with missing values. In European Conference on Computer Vision, pp. 707-720. Springer Berlin Heidelberg. (2002, May).
(7) Oxberry, G. M., Kostova - Vassilevska, T., Arrighi, W., & Chand, K., Limited memory adaptive snapshot selection for proper orthogonal decomposition. International Journal for Numerical Methods in Engineering, Vol.109, No.2, pp.198-217. (2017).
(8) Heft, A. I., Indinger, T., & Adams, N. A., Introduction of a new realistic generic car model for aerodynamic investigations, SAE Technical Paper No. 2012-01-0168. (2012).
(9) Mack, S., Indinger, T., Adams, N. A., Blume, S., & Unterlechner, P., The interior design of a 40% scaled DrivAer body and first experimental results. In ASME Fluids Engineering Division Summer, Puerto Rico, USA, FEDSM2012-72371. (2012).
(10) Heft, A. I., Indinger, T., and Adams, N. A., Experimental and Numerical Investigation of the DrivAer model, In ASME 2012 Fluids Engineering Summer Meeting, Puerto Rico, USA, FEDSM2012-72272. (2012)
(11) Collin, C., Mack, S., Indinger, T., & Mueller, J., A Numerical and Experimental Evaluation of Open Jet Wind Tunnel Interferences using the DrivAer Reference Model, SAE Technical Paper No2016-01-1597. (2016).
(12) Wittmeier, F. and Kuthada, T., Open Grille DrivAer Model - First Results, SAE Int. J. Passeng Cars – Mech. Syst, Vol.8, No.1, 2015-01-1553, pp. 252-260. (2015)
(13) Matsumoto, D., Haag, L., & Indinger, T., Investigation of the unsteady external and underhood air flow of the DrivAer model by Dynamic Mode Decomposition methods, International Journal of Automotive Engineering, Vol.8, Issue 2, pp.56-62. (2017).
(14) Haag, L., Kiewat, M., Indinger, T., & Blacha, T., "Numerical and Experimental Investigations of Rotating Passenger Car Wheel Aerodynamics on the DrivAer Model with Engine Bay Flow," Proceedings accepted in FEDSM2017. (2017).
(15) Collin, C., Mueller, J., Islam, M., & Indinger, T., On the Influence of Underhood Flow on External Aerodynamics of the DrivAer Model, Proceedings in the 12th FKFS-Conference Progress in Vehicle Aerodynamics and Thermal Management, Stuttgart, Germany, (2017).
(16) Chair of Aerodynamics and Fluid mechanics: DrivAer Model, Technical University of Munich, http://www.drievaer.com. (Accessed April 2, 2018).
(17) Sirovich, L. Turbulence and the dynamics of coherent structures. I. Coherent structures. Quarterly of applied mathematics, Vol.45, No.3, pp.561-571, (1987).
(18) Matsumoto, D., Kiewat, M., Niedermeier, C., and Indinger, T., Reduced Order Modeling for Unsteady CFD, Presentation in Shanghai-Stuttgart-Symposium, Shanghai, China, (2017)
(19) FluiDyna GmbH: ultraFluidX, FluiDyna GmbH, http://www.fluidyna.com/content/ultrafluidx. (Accessed April 2, 2018).