

# HAPOD – Fast, Simple and Reliable Distributed POD Computation<sup>\*</sup>

Christian Himpe<sup>\*</sup> Tobias Leibner<sup>\*\*</sup> Stephan Rave<sup>\*\*</sup>

<sup>\*</sup> *Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany (e-mail: himpe@mpi-magdeburg.mpg.de)*

<sup>\*\*</sup> *Applied Mathematics, University of Münster, Germany (e-mail: {stephan.rave, tobias.leibner}@uni-muenster.de)*

*Keywords:* Model reduction, Proper Orthogonal Decomposition, Singular Value Decomposition

## 1. INTRODUCTION

Proper Orthogonal Decomposition (POD) is a widely used technique for constructing low-order approximation spaces from high-dimensional input data. Apart from numerous applications in the data sciences, POD is also a fundamental tool for the basis generation in projection-based reduced order modelling methods. In these methods, POD is used to construct low-dimensional state spaces that capture with high accuracy the relevant dynamics of a given high-dimensional discrete model. A quickly computable low-dimensional surrogate model is then obtained by projection of the governing equations of the original model onto the POD approximation space.

The POD space is obtained from a given set of ‘snapshot’ vectors  $\mathcal{S}$  by writing the elements of  $\mathcal{S}$  as a matrix of column-vectors, of which a truncated singular value decomposition (SVD) is computed. The left-singular values of this decomposition then form a basis (POD modes) of desired POD space (cf. Sirovich (1987)).

For large-scale applications with an increasing amount of input data vectors, however, computing the POD quickly becomes prohibitively expensive, in particular when the generated data is so large that the snapshot set  $\mathcal{S}$  cannot be stored entirely in memory.

In this contribution we introduce a generic, easy to implement approach to compute an approximate POD based on arbitrary tree hierarchies of worker nodes, where each worker computes a POD of only a small amount of snapshot vectors  $s \in \mathcal{S}$ . The tree hierarchy can be freely adapted to optimally suit the available computational resources. In particular, this hierarchical approximate POD (HAPOD) allows for both, simple parallelization with low communication overhead, as well as live sequential POD computation under restricted memory capacities. We present rigorous error estimates and numerical examples which underline the performance and reliability of our approach.

<sup>\*</sup> Supported by the Deutsche Forschungsgemeinschaft, DFG EXC 1003 Cells in Motion – Cluster of Excellence, Münster, Germany, by the Center for Developing Mathematics in Interaction, DEMAIN, Münster, Germany, by Cells in Motion (CiM) Cluster of Excellence in flexible funds project FF-2015-07, and by the German Federal Ministry of Education and Research (BMBF) under contract number 05M13PMA.

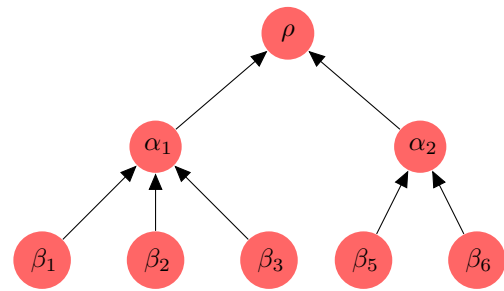


Fig. 1. Tree structure for HAPOD computation with leaf (input) nodes  $\beta_i$ , intermediate nodes  $\alpha_i$  and root (output) node  $\rho$ .

## 2. HIERARCHICAL PROPER ORTHOGONAL DECOMPOSITION

The HAPOD algorithm is based on an abstract tree structure (cf. Fig 1) where each node of the tree corresponds to a worker node performing computations on only a small set of snapshot/POD data. The HAPOD computation consists of the following steps:

- (1) Distribute snapshot vectors  $s \in \mathcal{S}$  among workers at leaf nodes  $\beta_i$ .
- (2) At each node:
  - (a) compute POD of input vectors for given local error tolerance.
  - (b) scale POD modes by corresponding singular values.
  - (c) communicate scaled POD modes to parent node.
- (3) Return the POD modes at root node  $\rho$  as output.

Note that communication is only performed upwards in the worker hierarchy from child to parent nodes. Local PODs can be computed in parallel and asynchronously as soon as all required input data is available. These properties make the HAPOD ideally suited for complex heterogeneous compute architectures and cloud computing, where frequent, synchronous communication of high-dimensional data is not an option. The HAPOD can be combined with any available algorithm for the local POD computation, possibly taking advantage of (parallelized) high-performance implementations specifically adapted to the given problem and computing environment.

## 2.1 Theoretical Analysis

POD is optimal in the sense that the POD space spanned by the first  $N$  modes minimizes the mean square approximation error

$$\sum_{s \in \mathcal{S}} \|s - P_{V_N} s\|^2, \quad (1)$$

where  $P_{V_N}$  is the orthogonal projection onto a given  $N$ -dimensional subspace  $V_N$ . In Himpe, Leibner, and Rave (2016) we show that for any given tree structure and snapshot set  $\mathcal{S}$ , we can define local truncation error tolerances for the mean error (1) such that for the resulting HAPOD space  $V_\rho$  the bound

$$\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \|s - P_{V_\rho}(s)\|^2 \leq \varepsilon^{*2} \quad (2)$$

holds for any prescribed target error  $\varepsilon^*$ . Moreover, the number of resulting HAPOD modes is bounded by

$$|\text{HAPOD}[\mathcal{S}, \varepsilon_{\mathcal{T}}](\rho)| \leq |\text{POD}(\mathcal{S}, \omega \cdot \varepsilon^*)|, \quad (3)$$

where  $|\text{POD}(\mathcal{S}, \omega \cdot \varepsilon^*)|$  denotes number of modes for a POD of  $\mathcal{S}$  with a target error of  $\omega \cdot \varepsilon^*$  with arbitrary  $\omega \in [0, 1]$ . At the same time, the number of HAPOD modes at the intermediate nodes  $\alpha$  is bounded by

$$\begin{aligned} & |\text{HAPOD}[\mathcal{S}, \varepsilon_{\mathcal{T}}](\alpha)| \\ & \leq |\text{POD}(\tilde{\mathcal{S}}_\alpha, (L-1)^{-1/2} \cdot \sqrt{1-\omega^2} \cdot \varepsilon^*)|, \end{aligned} \quad (4)$$

with  $L$  being the depth of the considered tree and  $\tilde{\mathcal{S}}_\alpha$  the set of snapshot vectors assigned to the leaves below  $\alpha$ .

Thus, while guaranteeing a prescribed approximation error  $\varepsilon^*$ , the parameter  $\omega$  allows us to control the trade off between an optimal approximation space of minimal dimension ( $\omega = 1$ ) and reduction of computational effort, i.e. smaller intermediate PODs ( $\omega = 0$ ).

## 2.2 Numerical Evaluation

In Fig 2 we show timing results for HAPOD and POD computation on 2D solution trajectories of a  $P_{15}$  moment closure/finite volume approximation of the Boltzmann equation for neutron transport, considering the checkerboard test case from Brunner and Holloway (2005). The trajectories were computed on uniform  $k \times k$ -grids with linearly increasing numbers of timesteps for 125 choices of scattering and absorption coefficients, yielding for  $k = 200$  about 2.5 terabytes of snapshot data.

The trajectories were computed in parallel on eleven compute nodes of a distributed memory computer cluster<sup>1</sup> utilizing one processor core for each trajectory.

In all cases the computation time for the trajectories was negligible in comparison to the required time for the POD/HAPOD computation. Already for  $k \geq 60$ , the POD could no longer be computed due to memory limitations. In addition, the HAPOD was twice as fast for  $k = 200$  as a standard POD for  $k = 40$ , even though the amount of data that needed to be processed increased by a factor of 125.

<sup>1</sup> Each node encloses two Intel Xeon Westmere X5650 CPUs ( $2 \times 6$  cores) and 48GB RAM.

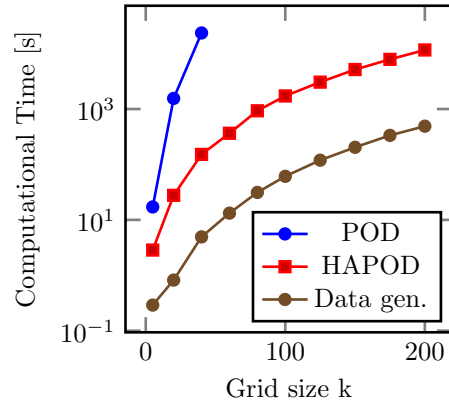


Fig. 2. Computational wall time for POD and HAPOD as well as time for snapshot generation for increasing state-space dimension and number of snapshots ( $\varepsilon^* = 10^{-4}$ ,  $\omega = 0.95$ ).

For full details and further numerical experiments we refer to Himpe, Leibner, and Rave (2016). A reference implementation of the HAPOD algorithm can be found under <https://git.io/hapod>.

## REFERENCES

- Brunner, T. and Holloway, J. (2005). Two-dimensional time dependent Riemann solvers for neutron transport. *Journal of Computational Physics*, 210(1), 386–399.
- Himpe, C., Leibner, T., and Rave, S. (2016). Hierarchical Approximate Proper Orthogonal Decomposition. *submitted*. URL <https://arxiv.org/abs/1607.05210>.
- Sirovich, L. (1987). Turbulence and the Dynamics of Coherent Structures Part I: Coherent Structures. *Quarterly of Applied Mathematics*, 45(3), 561–571.