

# PyDMD: Python Dynamic Mode Decomposition

Nicola Demo<sup>1</sup>, Marco Tezzele<sup>1</sup>, and Gianluigi Rozza<sup>1</sup>

<sup>1</sup> Internation School of Advanced Studies, SISSA, Trieste, Italy

DOI: [10.21105/joss.00530](https://doi.org/10.21105/joss.00530)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

**Submitted:** 05 January 2018

**Published:** 29 January 2018

## Licence

Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

Dynamic mode decomposition (DMD) is a model reduction algorithm developed by Schmid (Schmid 2010). Since then has emerged as a powerful tool for analyzing the dynamics of nonlinear systems. It is used for a data-driven model simplification based on spatiotemporal coherent structures. DMD relies only on the high-fidelity measurements, like experimental data and numerical simulations, so it is an equation-free algorithm. Its popularity is also due to the fact that it does not make any assumptions about the underlying system. See (Kutz et al. 2016) for a comprehensive overview of the algorithm and its connections to the Koopman-operator analysis, initiated in (Koopman 1931), along with examples in computational fluid dynamics.

In the last years many variants arose, such as multiresolution DMD, compressed DMD, forward backward DMD, and higher order DMD among others, in order to deal with noisy data, big dataset, or spurious data for example.

In the PyDMD package (“PyDMD: Python Dynamic Mode Decomposition. Available at: <https://github.com/mathLab/PyDMD>” n.d.) we implemented in Python the majority of the variants mentioned above with a user friendly interface. We also provide many tutorials that show all the characteristics of the software, ranging from the basic use case to the most sophisticated one allowed by the package.

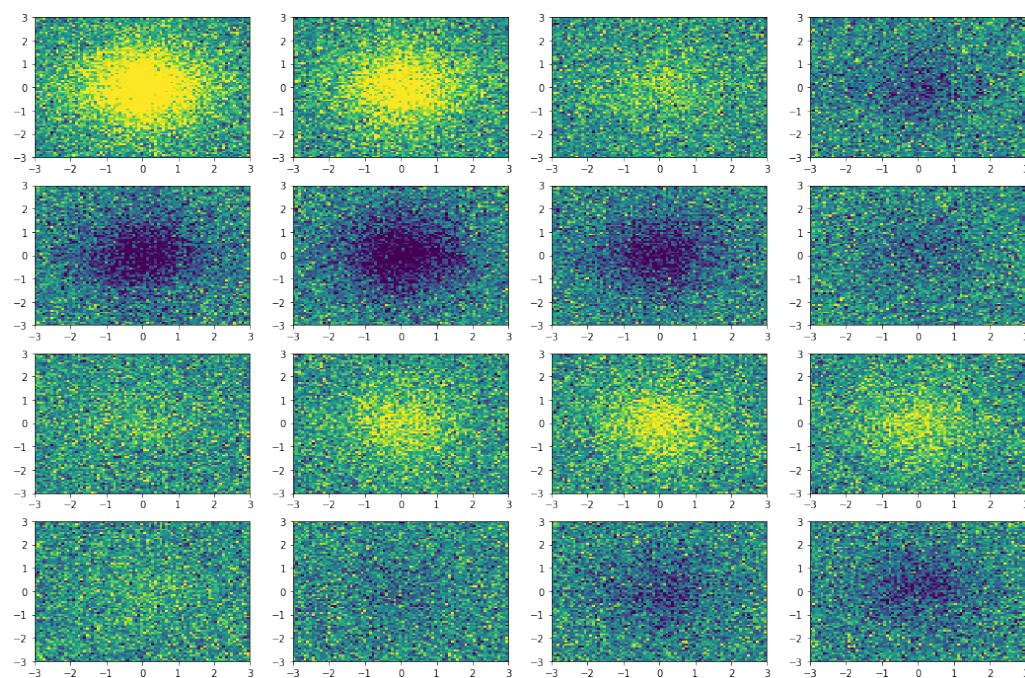
The research in the field is growing both in computational fluid dynamic and in structural mechanics, due to the equation-free nature of the model.

As an example, we show below few snapshots collected from a toy system with some noise. The DMD is able to reconstruct the entire system evolution, filtering the noise. It is also possible to predict the evolution of the system in the future with respect to the available data.

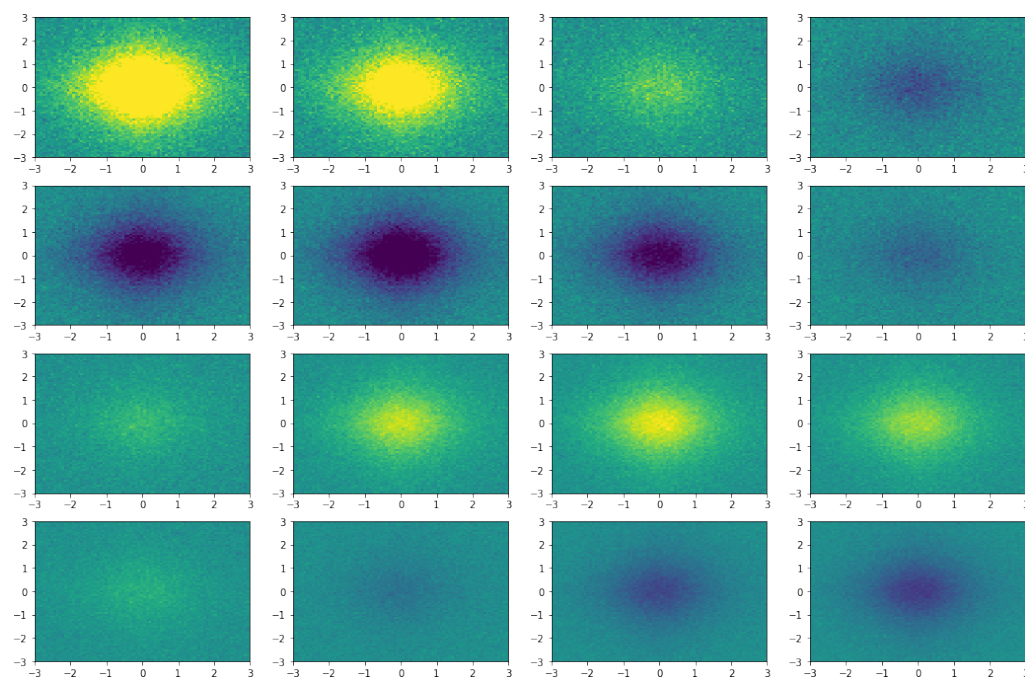
Here we have the reconstruction of the dynamical system. You can observe the sensible reduction of the noise.

## Acknowledgements

This work was partially supported by European Union Funding for Research and Innovation — Horizon 2020 Program — in the framework of European Research Council Executive Agency: H2020 ERC CoG 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics” P.I. Gianluigi Rozza.



**Figure 1: Snapshots**



**Figure 2: Reconstruction**

## References

- Koopman, Bernard O. 1931. “Hamiltonian Systems and Transformation in Hilbert Space.” *Proceedings of the National Academy of Sciences* 17 (5). National Acad Sciences:315–18.
- Kutz, J Nathan, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. 2016. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM. <https://doi.org/10.1137/1.9781611974508>.
- “PyDMD: Python Dynamic Mode Decomposition. Available at: <https://github.com/mathLab/PyDMD>.” n.d. Accessed January 2018.
- Schmid, Peter J. 2010. “Dynamic Mode Decomposition of Numerical and Experimental Data.” *Journal of Fluid Mechanics* 656. Cambridge University Press:5–28. <https://doi.org/10.1017/S0022112010001217>.