1 Introduction

pyMOR is a free software library for building model order reduction (MOR) applications in the Python programming language. All of its algorithms are based on abstract interfaces which allow seamless integration with external partial differential equation (PDE) solver packages (see [1]). Its initial focus was on reduced basis methods for linear and nonlinear parameterized PDEs. We present here recent developments in system-theoretic MOR methods and related algorithms.

In Section 2, we give a brief overview of linear systems and balanced truncation (BT). Next, we describe pyMOR’s abstract interfaces and the implementation of the methods in Section 3. We demonstrate BT applied to a PDE model discretized using FEniCS.
4 Numerical example

As an example, we consider the heat equation over a cross section of a heat sink (see [2] for the problem description). Finite element discretization, using FEniCS, leads to a system of the form (1) with $n = 12\,296$, $m = 1$, and $p = 1$. The top-middle plot in Figure 1 shows the state of the full-order model at $t = 10$ when the input is $u(t) = \sin\left(\frac{\pi}{3}t\right)^2$. The average run time is about 8 seconds on a laptop (model: Vaio VJS132C11L; processor: Intel® Core™ i7-8550U CPU @ 1.80 GHz, 4 cores, 8 logical processors, hyper-threading activated; RAM: 8 GB; cache: 8 MB).

We applied BT (using pyMOR 0.5) to get a ROM of order 10, with a run time of about 7 seconds. Based on Hankel singular values, we find the relative $H_\infty$-error lies in the interval $[7.27 \times 10^{-5}, 1.23 \times 10^{-4}]$, shown also in the left plot of Figure 1. The relative $H_2$-error can be computed and its value is $7.37 \times 10^{-3}$. In the bottom row of Figure 1, we see the state and output errors of the ROM. As expected, we see that the output is approximated better than the state. The average run time of the simulation is about 15 milliseconds, which is roughly 500 times faster then for the full-order model.

The entire script to generate these results can be found in [2].

5 Conclusion and outlook

We demonstrated BT applied to PDE models discretized in the PDE solver package FEniCS. This is possible due to the LR-ADI method’s implementation using pyMOR’s abstract interfaces.

The example used FEniCS, but other solver packages are also possible. Currently also deal.II, DUNE, and NGSolve are supported. Custom (domain specific) solvers can be easily integrated with pyMOR.

In pyMOR 0.5, further variants of BT for first-order systems are available, together with a few methods for second-order systems. Furthermore, interpolatory methods for first-order and second-order systems are also included.

Some PDE solver packages do not support complex numbers (e.g., FEniCS). In the example, we considered a system with symmetric $E$ and $A$, where real shifts $p_i$ were sufficient. For general systems, we plan to add support to pyMOR for solving linear systems with complex shifts based on real linear algebra to further enable applicability of system-theoretic methods to models defined with such PDE solver packages.

Acknowledgements This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2044 - 390685587, Mathematics Münster:Dynamics - Geometry - Structure, the German Federal Ministry of Education and Research (BMBF) under contract 05M18PMA, and DFG “pyMOR - Sustainable Software for Model Order Reduction” project (RA 3055/1-1, SA 3477/1-1).

References

[1] R. Milk, S. Rave, and F. Schindler, SIAM J. Sci. Comput. 38(5), pp. S194–S216 (2016).
[2] P. Mlinarić and S. Rave, pyMOR demo for balanced truncation applied to a heat sink model in FEniCS, Zenodo (2019), http://doi.org/10.5281/zenodo.3232900.