



MAX PLANCK INSTITUTE  
FOR DYNAMICS OF COMPLEX  
TECHNICAL SYSTEMS  
MAGDEBURG



COMPUTATIONAL METHODS IN  
SYSTEMS AND CONTROL THEORY

# How (Not) to Compare Model Order Reduction Algorithms

Christian Himpe

Computational Methods in Systems and Control Theory Group  
Max Planck Institute Magdeburg

Virtual CSC Retreat  
2020-06-02

Supported by:



Federal Ministry  
for Economic Affairs  
and Energy

## Alternative titles:

- Model Reduction Worst Practices
- How to Obfuscate your Model Reduction Results
- Why I Don't Believe Your Model Reduction Numerics
- X Ways to Fool the MOR Community
- X Dubious Ways of Testing Model Reduction Algorithms

## Before writing the code:

- (Don't) compare to standard methods,
- (Don't) use [additionally] common benchmarks,
- (Don't) describe free parameter choices,
- (Don't) describe training or test data,
- (Don't) avoid model reduction crime [cf. inverse crime].

## While writing the code:

- (Don't) use state-of-the art implementations,
- (Don't) avoid pessimizing competing methods,
- (Don't) justify selected reduced orders,
- (Don't) use adequate error quantification,
- (Don't) explain or justify comparison.

## When running the code:

- (Dis)regard CPU time vs wall time,
- (Dis)regard implicit / explicit parallelization,
- (Dis)regard efficient memory access [i.e. NUMA architecture],
- (Dis)regard overhead [in relation to compute-time],
- (Dis)regard statistics [i.e. means of repeated runs].

## If in doubt:

- (Don't) read up on anything you did not understand,
- (Don't) be mindful of your compute environment,
- (Don't) aim to be fair,
- (Don't) aim to be FAIR,
- (Don't) focus on **A**[vailability].



Do use **MORscore**<sup>1</sup>:

C. Himpe. **Comparing (Empirical-Gramian-Based) Model Order Reduction Algorithms**. ModRed Proceedings, Accepted, 2020.  
[arXiv:2002.12226](https://arxiv.org/abs/2002.12226)

---

<sup>1</sup>Remember last year's CSC retreat? "[MORscore – Comparability of Model Reduction Algorithms](#)"