Doctoral Thesis

MÜLLER Michael

michaelm@fel.zcu.cz

2023-07-31

Errors and omissions excepted

Suggestions and discussions welcome, just leave a message

1 Why Model Order Reduction?

[[[Sch08, p. 8]; [Che99, p. 11]; [Ben+21, p. 2]]]

Real world problems are too complex to be solved analytically. In classes one deals with simplifications like the projectile motions neglecting air resistance or

> ϟ alert
> !! second example needed

and representing electrons as point charges with no volume in space. However, if one applies the underlying equations to reality things are not so easy anymore. Partial differential equations (PDE) are usually the way to describe the laws of Physics. It turns out that by trying to solve these equations there is only in the rarest cases an algebraic expression as a solution. In all other cases equations must be solved numerically and are therefore only approximations to the problem. One approach to compute such approximations is using the finite element method (FEM). This computational technique discretizes the domain of interest into finite cells (mesh) and with given boundary conditions (BC) the field variable of the governing PDE is determined. The more accurate the solution should be the more cells and more grid points that define a cell are needed. Even for simple geometries the degrees of freedom (DOFs), i. e. the numbers of unknown can be in orders of millions and billions. Using brute force (simply adding abundant computational power with RAM and fastest processors) is not advisable in terms of time and energy. Instead of dealing with the original large scale system model simplifications reduces complexity. The trick here is to get rid of all superfluous and unnecessary details that provide little to no contribution to the solution. This is where model order reduction (MOR) got its name. The following picture gives a rough explanation of this.

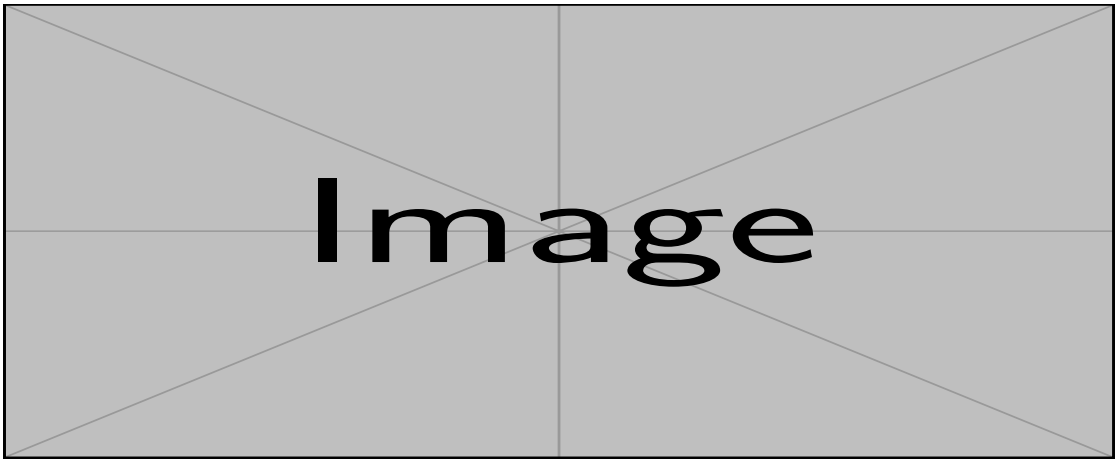


Fig. 1.1: [Ben+21, p. 2]. It seems that MOR is all about reducing the number of cells and grid points but that is not the essenece. The picture is given to demonstrate what reducing complexity but still holding the basic structure looks like.

revise
++ add more background from schilders
?? more background on FEM (examples where it is used today)
?? basic explaining equations for MWE

under construction
++ further explanation what MOR stands for

2 Setup/Theory asynchronous machine

[Sah17, p722] [Wil06, p263]

[Kas+19] kaska

alert
!! pagenumber is missing in Kaska

[Got97, p62+]

3 POD explanation principle

[BM+21, p595++] [Rib+21, p4+]

3.1 what can be done with COMSOL Agros2D, example

4 SVD explanation principle

some background information/history proof of svd Singular Value Decomposition as Simply as Possible SVD example/explanation

under construction
X: matrix from COMSOL; each row of the matrix consists of a sample solution taken at a specific value of time, and the number of rows in the matrix is the number of samples taken at evenly spaced values in time.

$$X = U\Sigma V^* \qquad \text{(SVD factorization)}$$

$$X \in \mathbb{C}^{m\times n} \qquad U \in \mathbb{C}^{m\times m} \qquad \Sigma \in \mathbb{R}^{m\times n} \qquad V \in \mathbb{C}^{n\times n}$$

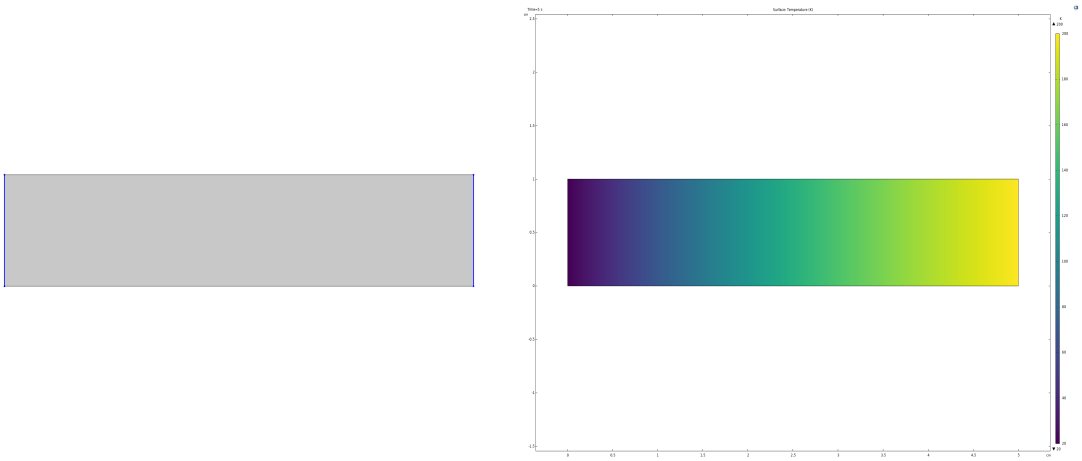
Σ diagonal matrix, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$
$\sigma_i = \sqrt{\lambda_i(XX^*)}$ X^* is conjugate transpose (in $\mathbb{C}$) or simple transposed (in $\mathbb{R}$).
pod paper definition, section2 slide 3

Why are the concepts of existence and uniqueness important? Differential equations frequently model real-world systems as a function of time. Knowing that a solution exists means that the system has predictable future states[[Adkins 85]]

5 Setup and results

5.1 I-shape

Before diving into a real and complex setup I decided it is best to start with basic examples where calculations can be done by hand and prototyping yields to faster results. For this the most simplest setup was taken, a simple 2d rectangular setup with fixed temperature $T_0 = 20\,\mathrm{K}$ on the left and $T_1 = 200\,\mathrm{K}$ on the right. The ambient temperature T_{amb} was given by $T_{\mathrm{amb}} = 293.15\,\mathrm{K}$. The geometry is modelled as copper and only the thermal conductivity κ was used as material property. This property and value, however, does not play a major role in the benchmark setup. Same goes with mesh quality. For quick prototyping reasons coarse meshes were accepted.



(a) Geometry setup with fixed temperature on the left T_0 and right T_1 (highlighted in blue). **(b)** Temperature distribution after 5 s.

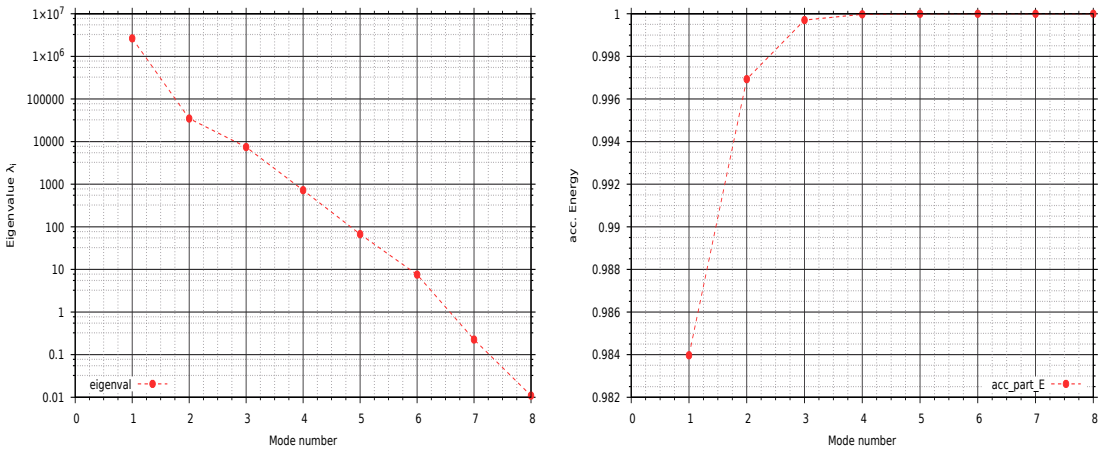
Fig. 5.1

POD	eigenval	acc eigenval	part E	acc part E
1	2.641702e+06	2.642e+06	9.840e-01	0.983972

2	3.478282e+04	2.676e+06	1.296e-02	0.996928
3	7.451407e+03	2.684e+06	2.775e-03	0.999703
4	7.218385e+02	2.685e+06	2.689e-04	0.999972
5	6.723362e+01	2.685e+06	2.504e-05	0.999997
6	7.572878e+00	2.685e+06	2.821e-06	1.000000
7	2.260862e-01	2.685e+06	8.421e-08	1.000000
8	1.093760e-02	2.685e+06	4.074e-09	1.000000

Tab. 5.1: First 8 modes are listed. *eigenval* column listed the eigenvalues in descending order (ordered from most to least important so to speak), *acc eigenval* the sum of all eigenvalues up to this row, *part E* columns represents the percentual energy each eigenvalue contributes to the system and the last column *acc part E* the accummulated partial energy up to this row.

In this setup the energy stored is $E_{\text{stored}} = 2.685 \times 10^6$ and the first mode is contributing more than 98 % of this value. That means that the first mode dominates the system. Having a look on the following left graph demonstrates this fact.

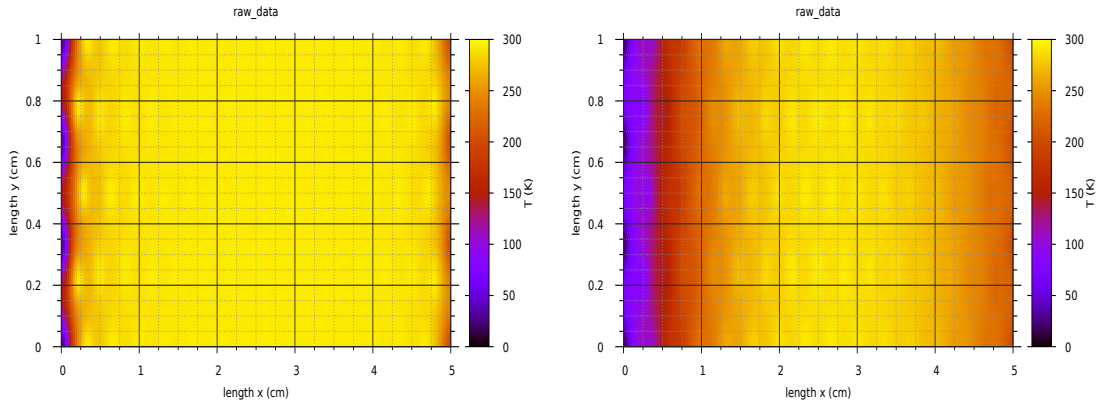


(a) Decay of the eigenvalues by a logarithmic scale on the y-axis. Each next mode is approximately one order of magnitude smaller.

(b) The first 8 modes are plotted against the accummulated partial energy. The more modes are taken into account the more they must add up to 1. Here, only 2 modes are sufficient to capture more than 99 % of the systen.

Fig. 5.2

With this reasonable data equipped one can make now the test if these modes do actually yield the input data.

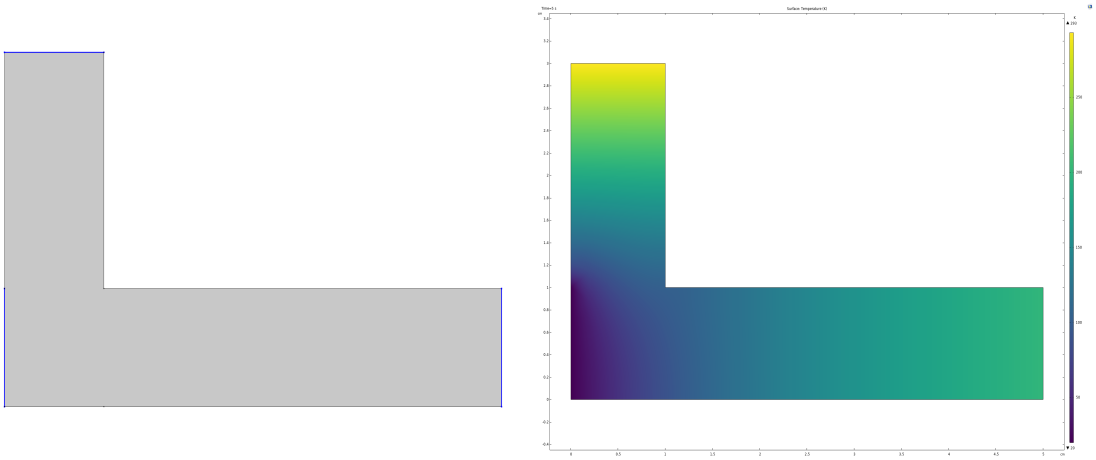


(a) Temperature distribution after 1 time step. **(b)** Temperature distribution after 5 time steps.

Fig. 5.3: The temperature distribution is gradually smoothing over the geometry.

5.2 L-shape

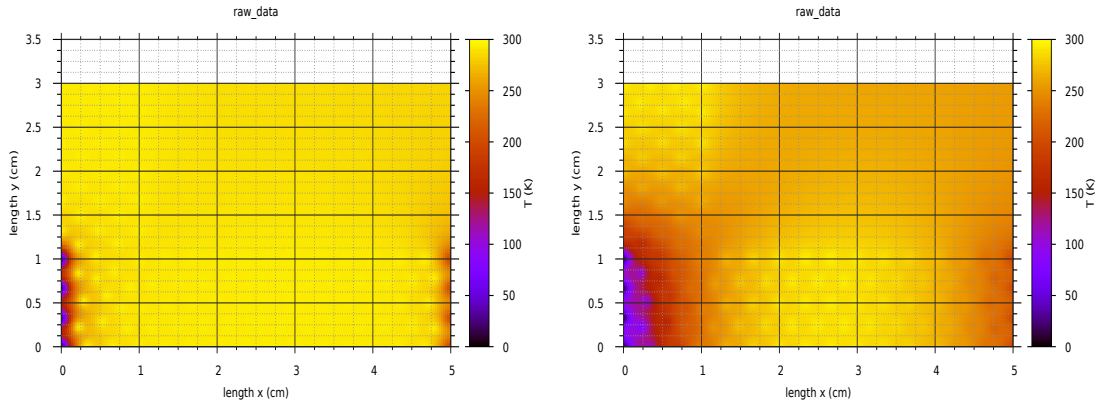
To apply the code on a simple more complex geometry a L was chosen with three different borders held constant. To generate a more diverse temperature distribution the fixed temperature values are $T_0 = 20\,\mathrm{K}$, $T_1 = 123\,\mathrm{K}$ and $T_2 = 200\,\mathrm{K}$. As in the previous case, ambient temperature was set to $T_{\mathrm{amb}} = 293.15\,\mathrm{K}$.



(a) Geometry setup with fixed temperature on the left T_0, on the right T_1 and on the top T_1. **(b)** Temperature distribution after 5 s.

Fig. 5.4

As in the case with the I-shape the decay was of similar nature, i. e. only few eigenvalues are needed to take the total energy stored in the system. The code didn't need any modification at all, only different snapshot file from COMSOL was loaded. As in the previous case



(a) Temperature distribution after 1 time step.

(b) Temperature distribution after 5 time steps.

Fig. 5.5: The temperature distribution is slowly spreading through the geometry but it cannot be correct. As visible in the geometry the point $P(4,3)$ is obviously not part of the geometry but there is associated temperature to this point. That means that there is an error in postprocessing the data and remapping of the data values to the geometry must be done with a different approach. Mesh artefacts are tolerated at this time.

Next approach is not only export the data (nodes of mesh+values) but also the geometry from COMSOL. The nodes are not sufficient as artifacts will be created (here interpolation between nodes are performed and there is no naive way to "force" interpolation only on provided nodes). The interpolator has no way to understand if a connection line of nodes lies inside or outside the geometry and is therefore doomed to fail. So another approach has to be taken. My next trials is to outsource the geometry and the mesh-generation (probably with Gmsh, generate snapshot matrix (probably with FreeFEM++), generate POD marix (already done) and then to the interpolation of this matrix on the original geometry+mesh (not sure how to do). First trials were successful, i. e. geometry could be meshed and stationary Poisson-PDE could be solved (switching to time-dependent situation is next step but no fundamental problem) with FreeFEM++. However, snapshot matrix needs to be extracted but should be possible.

6 Mathematical background of POD and SVD

6.1 The SVD

Among all matrix decompositions like LU decomposition or Cholesky decomposition in linear algebra the singular value decomposition of a matrix A into $\mathsf{U}\Sigma\mathsf{V}^T$ has assumed a special role. One reasons is numerical stability: small perturbations in A correspons to small perturbations in Σ and vice versa. (source)
As a general statement, each matrix $\mathsf{A} \in \mathbb{R}^{m\times n}$ can be factorized into orthogonal matrices $\mathsf{U} \in \mathbb{R}^{m\times m}$, $\mathsf{V} \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\Sigma := \operatorname{diag}(\sigma_1, \ldots, \sigma_d) \in \mathbb{R}^{m\times n}$ with $d = \min(m,n)$ and in decreasing order values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq \sigma_{d+1} = 0$ such that

$$\mathsf{U} = \mathsf{L}\Sigma\mathsf{R}^T. \qquad [6.1]$$

The values σ_i are called the singular values of U.(source) In its core the SVD is a procedure that will diagonalize any rectangular matrix, whereas eigenvalue decomposition only

works for square matrices.

Any matrix $\mathsf{U} \in \mathbb{R}^{m \times n}$, the *snapshot matrix*, can be factored into
L is an m x m orthogonal matrix, Σ is an m x n rectangular matrix and R is an n x n orthogonal matrix. The matrix Σ can be written in the form

$$\Sigma = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \qquad [6.2]$$

with $D = \operatorname{diag} \sigma_1, \ldots, \sigma_d$ and . To compute the singular calue decomposition of U is somewhere between computationally expensive and practically infeasible. The reason is that U is m x n matrix with at least m or n very large. To solve this issue snapshot POD was developed by SIROVICH to speed up computation. (source) (source)

6.2 The proper orthogonal decomposition

under construction
(source) (source) proper orthogonal decomposition (POD) was introduced by Lumley in 1967 as an attempt to decompose the random vector field of turbulent flow into a set of deterministic functions. With these limited number of deterministic functions, the POD modes, future prediction are possible. All forms of POD are based on statistical methods and treat the dynamic system as random. The system itself is by no means random but appears to be chaotic, so the most general approach is it to view as random. POD aims to obtain low-dimensional approximations from high-dimensional processes. Its applications comprises among other turbulent fluid flows, structural vibrations, image processing or data analysis. (source). If we have a vector-valued function $u(\boldsymbol{x}, t)$ over some domain of interest and time, we can express this with the standard eigenfunction expansion

$$\boldsymbol{u}'(\boldsymbol{x}, t) = \sum_{k=1}^{\infty} a_k(t) \Phi_k(\boldsymbol{x}) \approx \qquad [6.3]$$

As one can see there is no fundamental difference between the variables t and $\boldsymbol{x}$ and we usually denote them as the temporal coordinate/variable t and spatial coordinate/variable $\boldsymbol{x}$.
Such a separation of variables in [6.3] is not unique. Common functions for $\Phi_k(\boldsymbol{x})$ are Fourier-series, Legendre-polynomials or Chebyshev-polynomials. For each such a set of functions $a_k(t)$ are different.
explain what the letters in POD stands for
$\Phi_k(\boldsymbol{x})$ are an orthogonal set of eigenfunctions
we can approximate it with a finite sum in variables-separated form Note that Phi is not unique

$$\boldsymbol{u}'(\boldsymbol{x}, t) = \sum_{k=1}^{\infty} a_k(t) \Phi_k(\boldsymbol{x}) \qquad [6.4]$$

$\Phi_k(\boldsymbol{x})$ are POD spatial mode and $a_k(t)$ their time coefficients. (source)
temporal variable t and sparial variable $\boldsymbol{x}$. There are two ways to read POD:

- a decomposition in deterministic spatial modes and random time coefficients (direct method)
- a decomposition in deterministic temporal modes with random spatial coefficients (snapshot method)

For both decompositions the eigenvalues are the same.

6.3 The connection between POD and SVD

The connection between POD and SVD can be understood best when one calculates the matrices C_1 and C_2:

$$\mathsf{C}_1 = \frac{1}{m-1}\left(\mathsf{U}^T\mathsf{U}\right) \quad [6.5]$$

$$= \frac{1}{m-1}\left(\left(\mathsf{L}\Sigma\mathsf{R}^T\right)^T\left(\mathsf{L}\Sigma\mathsf{R}^T\right)\right) \quad [6.6]$$

$$= \frac{1}{m-1}\left(\mathsf{R}\Sigma^T\mathsf{L}^T\mathsf{L}\Sigma\mathsf{R}^T\right) \quad [6.7]$$

$$= \frac{1}{m-1}(\mathsf{R}(\Sigma^T\Sigma)\mathsf{R}^T) \quad [6.8]$$

and

$$\mathsf{C}_2 = \frac{1}{m-1}\left(\mathsf{U}\mathsf{U}^T\right) \quad [6.9]$$

$$= \frac{1}{m-1}\left(\left(\mathsf{L}\Sigma\mathsf{R}^T\right)\left(\mathsf{L}\Sigma\mathsf{R}^T\right)^T\right) \quad [6.10]$$

$$= \frac{1}{m-1}\left(\mathsf{L}\Sigma^T\mathsf{R}^T\mathsf{R}\Sigma\mathsf{L}^T\right) \quad [6.11]$$

$$= \frac{1}{m-1}(\mathsf{L}(\Sigma^T\Sigma)\mathsf{L}^T) \quad [6.12]$$

m number of degrees of freedom in spatial discretization of PDE and much larger than number of time steps n.

(source) (source) (source)