Chapter 7

Nonlinear Model Order Reduction

In the previous chapters, linear systems were considered, e.g., linear elasticity expressed with a linear finite element approach of the form

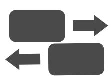
$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{f}.$$

Using a Petrov-Galerkin projection the reduced representation of the system could be obtained as

$$\underbrace{\underset{n\times N}{\boldsymbol{W}^T}\,\underset{N\times N}{\boldsymbol{M}}\,\underset{N\times n}{\boldsymbol{V}}}_{\bar{\boldsymbol{M}}\in\mathbb{R}^{n\times n}}\ddot{\bar{\boldsymbol{q}}} + \underbrace{\underset{n\times N}{\boldsymbol{W}^T}\,\underset{N\times N}{\boldsymbol{D}}\,\underset{N\times n}{\boldsymbol{V}}}_{\bar{\boldsymbol{D}}\in\mathbb{R}^{n\times n}}\dot{\bar{\boldsymbol{q}}} + \underbrace{\underset{n\times N}{\boldsymbol{W}^T}\,\underset{N\times N}{\boldsymbol{K}}\,\underset{N\times n}{\boldsymbol{V}}}_{\bar{\boldsymbol{K}}\in\mathbb{R}^{n\times n}}\bar{\boldsymbol{q}} = \underbrace{\underset{n\times N}{\boldsymbol{W}^T}\,\underset{N\times p}{\boldsymbol{f}}}_{\bar{\boldsymbol{f}}\in\mathbb{R}^{n\times p}}.$$

However, linearity is a strong assumption which is insufficient in many cases.

Origins of nonlinearities for mechanical systems:

- nonlinear boundary conditions 
- geometric/physical nonlinearities, e.g., buckling 
- nonlinear material behaviour 
- ...

In the following a a second-order nonlinear structural dynamical system of the form

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \underbrace{\boldsymbol{f}^{int}(\boldsymbol{q},\dot{\boldsymbol{q}},t)}_{\text{internal forces and moments}} = \underbrace{\boldsymbol{f}^{ext}(\boldsymbol{q},\dot{\boldsymbol{q}},t)}_{\text{externally applied loads}} \tag{7.1}$$

is considered.

Goal: Find optimal basis for representing nonlinear ODE in low-dimensional subspace

Problem: We can't simply apply formerly presented reduction approaches for a nonlinear System

Idea: Find basis for representing the time evolution of the system dynamics (state trajectories)

7.1 Proper Orthogonal Decomposition

(also known as Principle Component Analysis)

Collect snapshots of the (high-fidelity) dynamics at time steps

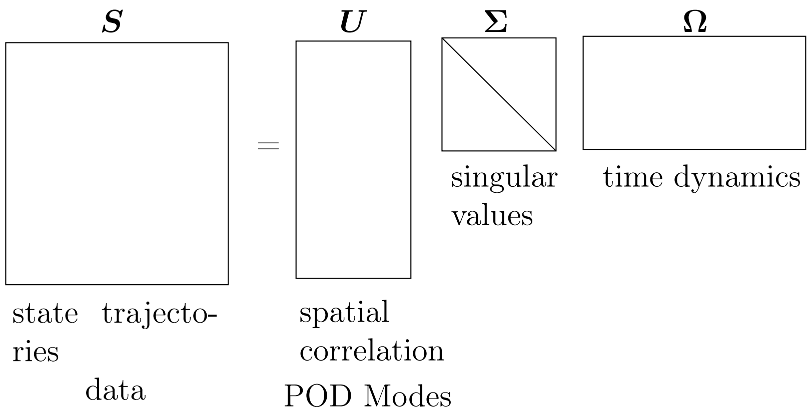
$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{q}(t_1) & \boldsymbol{q}(t_2) & \dots & \boldsymbol{q}(t_\eta) \end{bmatrix}$$

Question: Where does this data live? In what kind of space does it live in?

We already know how we can find the H_2 optimal r-dimensional approximation of a matrix $\rightarrow$ singular value decomposition

Recap SVD: Matrix can be decomposed into

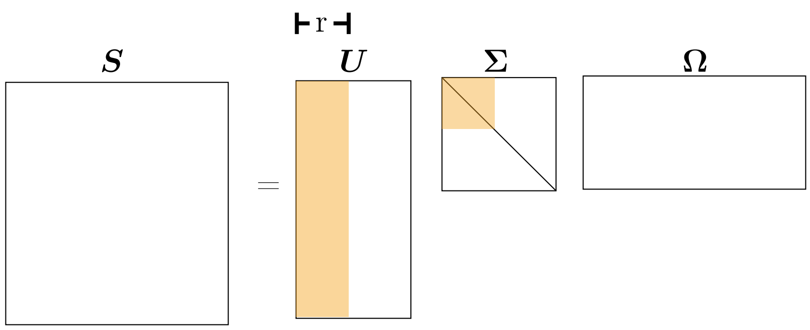
$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Omega}$$



The SVD gives us an orthogonal set of vectors (columns of $\boldsymbol{U}$) in which the data is embedded and indicates how important each direction is via its correlating singular values.

A low-dimensional subspace can be found by simply truncating $\boldsymbol{U}$

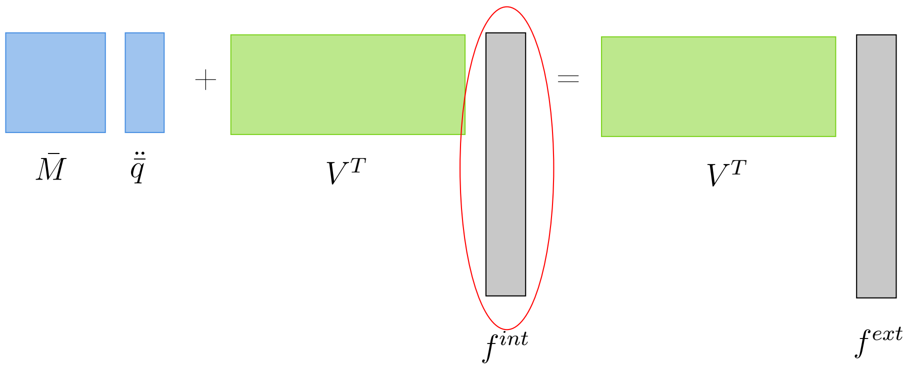
$$\boldsymbol{V} = \boldsymbol{U}_r \in \mathbb{R}^{N \times r} = \boldsymbol{U}_r(:, 1:r)$$



7.2 Nonlinear Reduction

Definition 7.1. *A low-rank approximation of system (7.1) can be obtained with a Galerkin projection as*

$$\underbrace{\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}}_{\bar{\boldsymbol{M}}}\ddot{\bar{\boldsymbol{q}}} + \boldsymbol{V}^T\boldsymbol{f}^{int}(\boldsymbol{V}\bar{\boldsymbol{q}}, \boldsymbol{V}\dot{\bar{\boldsymbol{q}}}, t) = \boldsymbol{V}^T\boldsymbol{f}^{ext}(\boldsymbol{V}\bar{\boldsymbol{q}}, \boldsymbol{V}\dot{\bar{\boldsymbol{q}}}, t), \tag{7.2}$$



Example 7.1 (challenges with nonlinearity)**.** .

$\boldsymbol{f}^{int}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}, t) = \tilde{\boldsymbol{q}}^3$

Two mode expansion

$\tilde{\boldsymbol{q}}(t) = a_1(t)\boldsymbol{V}_1 + a_2(t)\boldsymbol{V}_2$

Inner products are problematic

$\tilde{\boldsymbol{q}}(t)^3 = a_1^3\boldsymbol{V}_1^3 + 3a_1^2a_2\boldsymbol{V}_1^2\boldsymbol{V}_2 + 3a_1a_2^2\boldsymbol{V}_1\boldsymbol{V}_2^2 + a_2^3\boldsymbol{V}_2^3$

Problem: Computation of nonlinear forces and moment terms needs to be performed in physical (high-dimensional) space and must be updated for each time step for every element of the model

7.3 Hyper Reduction

Idea: Evaluate nonlinear forces not for all elements but only a few selected ones

Goal: Approximate the projected forces and moments up to a certain tolerance with as few elements as possible. This is usually done with greedy algorithms which iteratively add elements to the reduced mesh until a tolerance criterion is met

Methods:

- Gappy POD and collocation-based methods
  - use a small number of evaluations of nonlinear terms to find the solution which fits the observations within the low-dimensional subspace spanned by the basis vectors best
- Optimized global cubature methods
  - specifically developed for second-order nonlinear dyn. systems and structural dynamics
  - approximate projected force and moment vectors instread of nonlinear force and moment vectors directly
  - popular approach: Energy-Conserving Sampling and Weighting Method (ECSW)

7.3.1 ECSW

ECSW was developed in applications for computer graphics and animations originally.

One of the most expensive steps in evaluating $\boldsymbol{f}$ is the assembly of the internal forces and moments from the individual element contribution

$$\boldsymbol{V}^T \boldsymbol{f}^{int} = \boldsymbol{V}^T \sum_{\underbrace{e}_{\text{single element}} \in \underbrace{\Omega}_{\text{complete FE mesh}}} \underbrace{\boldsymbol{L}_e^T}_{\text{element connectivity matrix}} \underbrace{\boldsymbol{f}_e}_{\text{vector of associated element forces}} \tag{7.3}$$

Goal: Approximate sum by only evaluating a small subset of the elements $e \in \Omega_r \subset \Omega$. This means Ω_r represents the reduced domain of FEms???. In a first step weighting factors $\boldsymbol{\xi}_e^*$ are introduced in (7.3) yielding

$$\boldsymbol{V}^T \boldsymbol{f}^{int} = \boldsymbol{V}^T \sum_{e \in \Omega} \boldsymbol{\xi}_e^* \boldsymbol{L}_e^T \boldsymbol{f}_e. \tag{7.4}$$

0	0	0	0	0
0	10	0	0	0
0	0	0	0	0
0	0	0	4	1

During training the weighting factors $\boldsymbol{\xi}_e^*$ are selected such that as many of them as poosible are zero but still a high approximation accuracy is reached $\rightarrow$ omitt those ones with $\boldsymbol{\xi}_e = 0$

$$\boldsymbol{V}^T \boldsymbol{f}^{int} \approx \boldsymbol{V}^T \sum_{e \in \underbrace{\Omega_r}_{\text{only elements with } \boldsymbol{\xi}_e^* > 0}} \boldsymbol{\xi}_e^* \boldsymbol{L}_e^T \boldsymbol{f}_e. \tag{7.5}$$

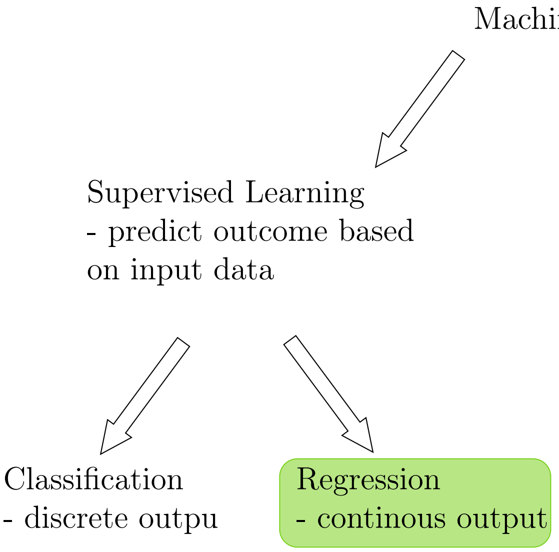
While hyper-reduction methods are able to produce high-quality approximations, they possess one big disadvantage: They are very intrusive, i.e., they require deep manipulations of the used simulation code, e.g., evaluation of the forces at a reduced mesh. Another approach for nonlinear MOR are non-intrusive methods which are often data-based

7.4 MOR and ML

The basic idea is to learn system dynamics based on its behavior, that means based on high-fidelity simulation data. One powerfull tool to extract knowledge out of data is machine learning (ML)

7.4.1 Machine Learning

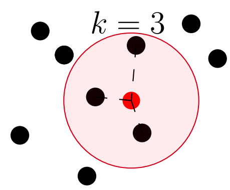
Basic task of ML: approximation of the original distribution $\boldsymbol{\phi}(\boldsymbol{x})$ of a model by fitting a function $\tilde{\boldsymbol{\phi}}(\boldsymbol{x})$ to the given data $\boldsymbol{D} = (x_i, y_i)_{i=1}^n$. In this context $\boldsymbol{x}$ can be an arbitrary data sample (input data) and $\boldsymbol{y}$ the corresponding observation (output data).



k-Nearest Neighbors

For a given input $\boldsymbol{x}$ the k ‘nearest‘ points of the training data set $\boldsymbol{D}$ are located. The associated output is then predicted based on their averaged outputs

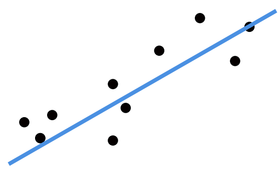
$$\hat{\phi}(\boldsymbol{x}) = \frac{1}{k} \sum_{x_i \in \underbrace{N_k(\boldsymbol{x})}_{\text{neighborhood of } \boldsymbol{x}}} y_i,$$



Linear Regression

Linear combination of the input variables weighted by parameters $\boldsymbol{\beta}$, which are optimized to fit the data in the best manner

$$\hat{\phi}(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j = \begin{bmatrix} 1 & \boldsymbol{x} \end{bmatrix}^T \boldsymbol{\beta} \tag{7.6}$$



...

7.4.2 Approximate Dynamics

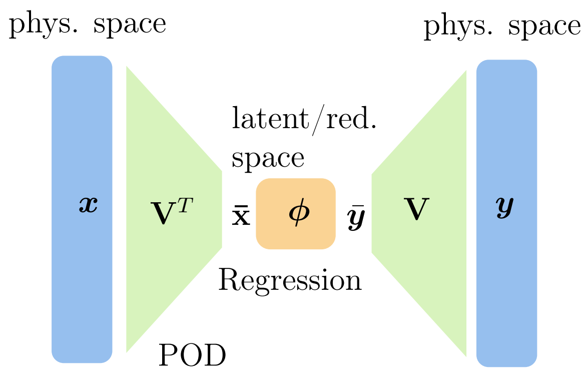
Learn mapping from suitable input parameters $\boldsymbol{X}$ to discrete system dynamics

$$\boldsymbol{\phi} : \boldsymbol{X} \to \begin{bmatrix} \boldsymbol{q}(t_1) & \boldsymbol{q}(t_2) & \dots & \boldsymbol{q}(t_\eta) \end{bmatrix}$$

This is not very efficient since the approximation is computed within the high-dimensional original space. Thus, MOR (e.g., POD) can be used to find low-dimensional approximation

$$\boldsymbol{\phi} : \boldsymbol{X} \to \begin{bmatrix} \bar{\boldsymbol{q}}(t_1) & \bar{\boldsymbol{q}}(t_2) & \dots & \bar{\boldsymbol{q}}(t_\eta) \end{bmatrix}$$

Thus, an approximation of the full time dynamics can be obtained $\tilde{\boldsymbol{q}} = \boldsymbol{V}\boldsymbol{\phi}(\boldsymbol{x})$



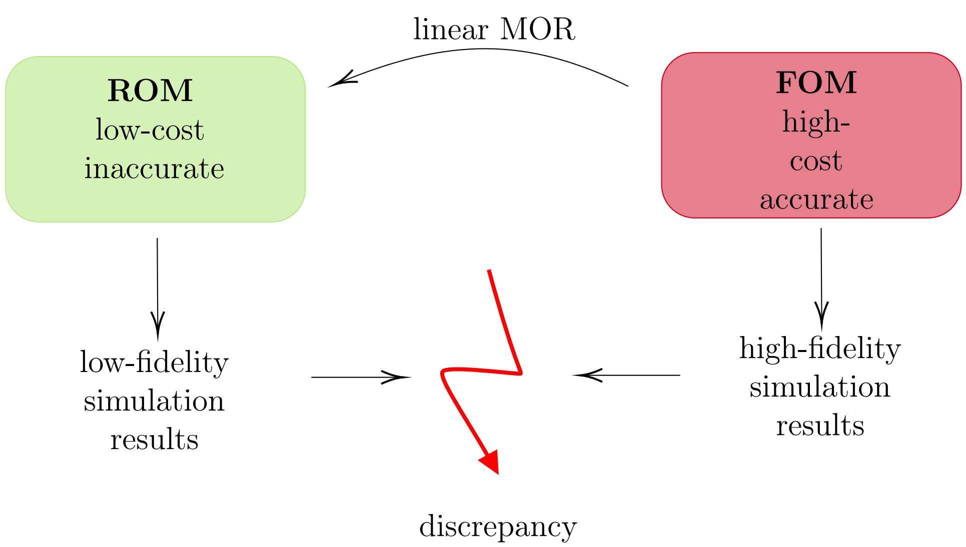
Example 7.2 (Learn state transition). *Given:*

outputs $\boldsymbol{q}(t) = \psi(t, \boldsymbol{p}, \boldsymbol{q}_0)$ of a 'black-box' model (usually a FE-discretized PDE + solver evaluated for certain initial conditions $\boldsymbol{x}_0$ and boundary conditions $\boldsymbol{b} \in \boldsymbol{p}$)

Learn a time-discrete Model approximating the system output

1. *Use POD to find low-dimensional representation of system states $\bar{\boldsymbol{q}} = \boldsymbol{V}^t\boldsymbol{q}$*
2. *learn state transition $\bar{\boldsymbol{q}}(t_{i+1}) = \boldsymbol{\phi}(\bar{\boldsymbol{q}}(t_i), \boldsymbol{p})$*
3. *evaluate $\boldsymbol{\phi}$ for all time steps starting at initial conditions $\bar{\boldsymbol{q}}(t_1)$*

7.4.3 Hybrid Models



ROM is not able to capture the dynamics of the FOM for nonlinear domains.
Idea: learn the 'gap' between the low-fidelity and high-fidelity model

$$\boldsymbol{\phi} : \boldsymbol{X} \rightarrow \boldsymbol{e} = \boldsymbol{q}(t) - \tilde{\boldsymbol{q}}(t),$$

with $\tilde{\boldsymbol{q}} = \boldsymbol{V}\bar{\boldsymbol{q}}$. Enriching the physical-based ROM with the so-obtained data-based model yields the so-called hybrid model.

Example 7.3 (Enrich linear ROM with nonlinear inner forces)**.** *Given:*

FOM: $\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{f}^{int} = \boldsymbol{f}^{ext}$
ROM: $\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{q}}} + \bar{\boldsymbol{D}}\dot{\bar{\boldsymbol{q}}} + \bar{\boldsymbol{K}}\bar{\boldsymbol{q}} = \bar{\boldsymbol{f}}^{ext}$
Gap: $\boldsymbol{e} \approx \boldsymbol{f}^{int} = \boldsymbol{f}^{ext} - \boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{D}\dot{\boldsymbol{q}} - \boldsymbol{K}\boldsymbol{q}$

Learn the low-dimensional representation of the nonlinear inner forces

1. *learn* $\bar{\boldsymbol{f}}^{int} = \boldsymbol{V}^T\boldsymbol{f}^{int}$ *with regression algorithms* $\boldsymbol{\phi} \approx \bar{\boldsymbol{f}}^{int}$
2. *combine ROM with* $\boldsymbol{\phi}$ *to receive* $\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{q}}} + \bar{\boldsymbol{D}}\dot{\bar{\boldsymbol{q}}} + \bar{\boldsymbol{K}}\bar{\boldsymbol{q}} + \boldsymbol{\phi} = \bar{\boldsymbol{f}}^{ext}$
3. *integrate hybrid model to obtain approximated solution* $\bar{\boldsymbol{q}}$
4. *backprojection in original (physical) space to interpret solution*

.