Moment matching by projection on Krylov subspace

Overview articles concerning moment matching through Krylov-subspace reduction are e.g. [1] or [2].

Arnoldi Algorithm

If we use a Galerkin projections, $\boldsymbol{V} = \boldsymbol{W}$, only one projection matrix is needed and can be calculated with an Arnoldi Algorithm.

- The following recursion is satisfied:

$$\boldsymbol{M} \cdot \boldsymbol{V}_j = \boldsymbol{V}_j \cdot \boldsymbol{H}_j + \boldsymbol{f}_j \cdot \boldsymbol{e}_j^{\mathrm{H}} \tag{1}$$

- $\boldsymbol{H}_j = \boldsymbol{V}_j^{\mathrm{H}} \cdot \boldsymbol{M} \cdot \boldsymbol{V}_j$ an upper Hessenberg matrix
- $\boldsymbol{V}_j \cdot \boldsymbol{V}_j^{\mathrm{H}} = \boldsymbol{I}_j$ and $\boldsymbol{V}_j \cdot \boldsymbol{f}_j = 0$.

Algorithm 1 Arnoldi Algorithm

```
Input: $\boldsymbol{M} \in \mathbb{R}^{2N \times 2N}, \boldsymbol{r} \in \mathbb{R}^{2N}$
Output: Orthogonal Basis $\boldsymbol{V_j} \in \mathbb{R}^{2N \times j}$ for $\mathcal{K}_j(\boldsymbol{M}, \boldsymbol{r})$
$\boldsymbol{v_1} = \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|}$;
$\boldsymbol{w} = \boldsymbol{M} \cdot \boldsymbol{v_1}$; $\alpha_1 = \boldsymbol{v_1^H} \cdot \boldsymbol{w}$;
$\boldsymbol{f_1} = \boldsymbol{w} - \alpha_1 \boldsymbol{v_1}$;
$\boldsymbol{V_1} = [\boldsymbol{v_1}]$ ; $\boldsymbol{H} = [\alpha_1]$;
for i= 1 ,.., j-1 do
    $\beta_i = \|\boldsymbol{f_i}\|$ ; $\boldsymbol{v_{i+1}} = \boldsymbol{f_i}/\beta_i$
    $\boldsymbol{V_{i+1}} = [\boldsymbol{V_i}, v_{i+1}]$
    $\hat{\boldsymbol{H}}_{\boldsymbol{i}} = \begin{bmatrix} \boldsymbol{H_i} \\ \beta_i \cdot \boldsymbol{e_i^H} \end{bmatrix}$
    $\boldsymbol{w} = \boldsymbol{M} \cdot \boldsymbol{v_{i+1}}$;
    $\boldsymbol{h} = \boldsymbol{V}_{\boldsymbol{i+1}}^{\mathrm{H}} \cdot \boldsymbol{w}$; $\boldsymbol{f_{i+1}} = \boldsymbol{w} - \boldsymbol{V_{i+1}} \cdot \boldsymbol{h}$;
    $\boldsymbol{H_{i+1}} = [\hat{\boldsymbol{H}}_{\boldsymbol{i}}, \boldsymbol{h}]$;
end for
```

Remarks:

- Krylov-subspace based methods are used for the solution of generalized eigenproblems in most of the FE solvers, see Example 01. They have a high convergence rate. If you want to calculate eigenfrequencies within a specific range you create a Krylov subspace with expansion point within this eigenfrequency range. Afterwards the eigenvalues and up-projected eigenmodes of the reduced model match with the eigenvalues of the original system.

- Krylov methods have simple derivation and algorithms.

- Originally, both the Arnoldi and the Lanczos algorithms were developed for SISO systems.

- Krylov methods can be applied to MIMO systems where $\boldsymbol{B} \in \mathbb{R}^{N\times p}$, $\boldsymbol{C} \in \mathbb{R}^{r\times N}$, p and/or r are bigger than one, at the expense of increased bookkeeping. For MIMO systems the matrix Padé via Lanczos (MPVL) [3] respectively the symmetric PVL (SymPVL) [4] for symmetric systems or block Arnoldi [5, 6] algorithms can be used.

- Only *matrix-vector* multiplications are required – no matrix factorizations or inversions. There is no need to compute the transformed nth-order model and then truncate. This reduces the *ill-conditioning* that arises in SVD methods.

- For Petrov-Galerkin (oblique) projections two projection matrices are required which can be calculated simultaneous with the Two-sided Lanczos Algorithm, see e.g. [7, Chapter 10.4.2] which calculates two biorthogonal bases $\boldsymbol{V}_j \in \mathbb{R}^{N\times J}$ and $\boldsymbol{W} \in \mathbb{R}^{N\times J}$ where the following recursions are satisfied:

$$\boldsymbol{M} \cdot \boldsymbol{V}_j = \boldsymbol{V}_j \cdot \boldsymbol{T}_j + \boldsymbol{f}_j \cdot \boldsymbol{e}_j^{\mathrm{H}} \tag{2}$$

$$\boldsymbol{M}^{\mathrm{H}} \cdot \boldsymbol{W}_j = \boldsymbol{W}_j \cdot \boldsymbol{T}_j^{\mathrm{H}} + \boldsymbol{g}_j \cdot \boldsymbol{e}_j^{\mathrm{H}} \tag{3}$$

  where $\boldsymbol{T}_j = \boldsymbol{W}_j^{\mathrm{H}} \cdot \boldsymbol{M} \cdot \boldsymbol{V}_j$ is a tridiagonal matrix, $\boldsymbol{W}_j^{\mathrm{H}} \cdot \boldsymbol{V}_j = \boldsymbol{I}_j$, $\boldsymbol{W}_j^{\mathrm{H}} \boldsymbol{g}_j = 0$ and $\boldsymbol{V}_j^{\mathrm{H}} \cdot \boldsymbol{f}_j = 0$.

Algorithm 2 Two-Sided Lanczos Algorithm

```
Input: $\boldsymbol{M} \in \mathbb{R}^{N\times N}, \boldsymbol{r} \in \mathbb{R}^{N}, \boldsymbol{s^H} \in \mathbb{R}^{N}$
Output: Bi-orthogonal Basis $\boldsymbol{V_j} \in \mathbb{R}^{N\times j}$ and $\boldsymbol{W_j} \in \mathbb{R}^{N\times j}$ for $\mathcal{K}_j(\boldsymbol{M}, \boldsymbol{r})$
and $\mathcal{K}_j(\boldsymbol{M^H}, \boldsymbol{s^H})$ respectively satisfying $\boldsymbol{W_j^T} \cdot \boldsymbol{V_j} = \boldsymbol{I_j}$
$\beta_1 = \sqrt{|\boldsymbol{r^H} \cdot \boldsymbol{s^H}|}, \quad \gamma_1 = \mathrm{sign}(\boldsymbol{r^H} \cdot \boldsymbol{s^H})\beta_1$
$\boldsymbol{v_1} = \frac{\boldsymbol{r}}{\beta_1}, \quad \boldsymbol{w_1} = \frac{\boldsymbol{s^H}}{\gamma_1}$
for i= 1 ,.., j-1 do
    $\alpha_i = \boldsymbol{w_i^H} \cdot \boldsymbol{M} \cdot \boldsymbol{v_i}$;
    $\boldsymbol{p_i} = \boldsymbol{M} \cdot \boldsymbol{v_i} - \alpha_i \cdot \boldsymbol{v_i} - \gamma_i \cdot \boldsymbol{v_{i-1}}$;
    $\boldsymbol{q_i} = \boldsymbol{M^H} \cdot \boldsymbol{w_i} - \alpha_i \cdot \boldsymbol{w_i} - \beta_i \cdot \boldsymbol{w_{i-1}}$;
    $\beta_{i+1} = \sqrt{|\boldsymbol{p_i^H} \cdot \boldsymbol{q_i}|}, \quad \gamma_{i+1} = \mathrm{sign}(\boldsymbol{p_i^H} \cdot \boldsymbol{q_i})\beta_{i+1}$
    $\boldsymbol{v_{i+1}} = \frac{\boldsymbol{p_i}}{\beta_{i+1}}$
    $\boldsymbol{w_{i+1}} = \frac{\boldsymbol{q_i}}{\gamma_{i+1}}$
end for
$\boldsymbol{V} = \left[v_1, \dots, v_j\right], \boldsymbol{W} = \left[w_1, \dots, w_j\right]$
```

- One drawback of Krylov reduction methods is that stability or other system properties like passivity are not necessarily preserved. The topic of stability preservation for Krylov-subspace reduction methods is discussed e.g. in [8, 9]. One possibility to preserve stability is preservation via post-processing (restarting), see e.g. [10, 11, 12].

- Computational efficiency of Krylov Algorithms explained in Landau Notation:

Algorithm 3 rational Arnoldi algorithm

```
Input: matrix $\boldsymbol{M}$, $\boldsymbol{R}, J$
Output: basis $\boldsymbol{V}$
/*Initialize for every expansion point*/
$\boldsymbol{M} = (\boldsymbol{A} - s_k \cdot \boldsymbol{I})^{-1}$, $\boldsymbol{R} = (\boldsymbol{A} - s_k \cdot \boldsymbol{I})^{-1} \cdot \boldsymbol{B}$
Inverse of $\boldsymbol{M}^{-1}$ is not explicitly built instead a $[\boldsymbol{L}, \boldsymbol{U}] = \text{LUDecomposition}(\boldsymbol{M}^{-1})$ is used.
Computational effort LU($\boldsymbol{M}$) is $\mathcal{O}(N^{\beta})$ with $1.1 \leq \beta \leq 1.5$ for sparse matrices.
$\boldsymbol{V} = \tilde{\boldsymbol{V}} = \frac{\boldsymbol{R}}{\|\boldsymbol{R}\|_2}$
for Matching Order do
    $\boldsymbol{R} = \boldsymbol{M} \cdot \tilde{\boldsymbol{V}}$
    Action of $\boldsymbol{M}$ on $\tilde{\boldsymbol{V}}$ use $\boldsymbol{R} = \text{EquationSolve}(\boldsymbol{L}, \boldsymbol{U}, \tilde{\boldsymbol{V}})$
    $\mathcal{O}(N^{\alpha})$ backward/forward substitutions with $1 \leq \alpha \leq 1.2$
    $\tilde{\boldsymbol{V}} = \text{GramSchmidt}(\boldsymbol{R}, \boldsymbol{V})$
    $\mathcal{O}(Np^2)$ Frobenius products where $p$ is number of columns of $\boldsymbol{V}$
    $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V} & \tilde{\boldsymbol{V}} \end{bmatrix}$
end for
```

Krylov-subspaces for Second Order Systems

For the model reduction of mechanical systems, the second order structure of the system is kept if model reduction is done on the second order level without a transformation to a first order system. In addition, for orthogonal reduction $\boldsymbol{W} = \boldsymbol{V}$ and a full rank projection matrix the system properties of the second order matrices are preserved.

The transfer matrix of the second order system is

$$\boldsymbol{H}(s) = \boldsymbol{C}_e \cdot (s^2 \boldsymbol{M}_e + s\boldsymbol{D}_e + \boldsymbol{K}_e)^{-1} \cdot \boldsymbol{B}_e \tag{4}$$

and the power series of the Laplace transform of the elastic coordinate $\boldsymbol{Q}(s)$ around a specific expansion point s_k can be written as

$$\boldsymbol{Q}(s) = \boldsymbol{R}_0^{s_k} + \boldsymbol{R}_1^{s_k}(s - s_k) + \cdots = \sum_{j=0}^{\infty} \boldsymbol{R}_j^{s_k}(s - s_k)^j, \tag{5}$$

where $\boldsymbol{R}_j^{s_k}$ are called second order system moments and the relation between the output moments and the second order system moments

$$\boldsymbol{T}_j^{s_k} = \boldsymbol{C}_e \cdot \boldsymbol{R}_j^{s_k} \tag{6}$$

can be easily identified.

Definition 0.1. *A second order Krylov-subspace is defined as*

$$\mathcal{G}_J(\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{R}_1) = \text{colspan}\{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_{J-1}\} \tag{7}$$

where

$$\begin{cases} \boldsymbol{P}_o = \boldsymbol{R}_1, \quad \boldsymbol{P}_1 = \boldsymbol{M}_1 \cdot \boldsymbol{R}_1 \\ \boldsymbol{P}_i = \boldsymbol{M}_1 \cdot \boldsymbol{P}_{i-1} + \boldsymbol{M}_2 \cdot \boldsymbol{P}_{i-2}, \quad i = 2, 3, \ldots \end{cases} \tag{8}$$

and $\boldsymbol{M}_1$, $\boldsymbol{M}_2 \in \mathbb{R}^{N \times N}$, $\boldsymbol{R}_1 \in \mathbb{R}^{N \times p}$ are constant matrices. The columns of $\boldsymbol{R}_1$ are called the starting vectors and the matrices $\boldsymbol{P}_i$ are called basic blocks.

It can be shown, see e.g. [13, 14, 15] that the second order Krylov-subspace $\mathcal{G}_J$ spans the same space as the upper half of the standard Krylov-subspace $\mathcal{K}_J(\boldsymbol{M}, \boldsymbol{R})$, where $\boldsymbol{M}, \boldsymbol{R}$ are the matrices obtained by rewriting the recursion (8) into a matrix form

$$\underbrace{\begin{bmatrix} \boldsymbol{P}_i \\ \boldsymbol{P}_{i-1} \end{bmatrix}}_{\tilde{\boldsymbol{P}}_i} = \underbrace{\begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_2 \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}}_{\tilde{\boldsymbol{M}}} \cdot \underbrace{\begin{bmatrix} \boldsymbol{P}_{i-1} \\ \boldsymbol{P}_{i-2} \end{bmatrix}}_{\tilde{\boldsymbol{P}}_{i-1}} \tag{9}$$

and $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{bmatrix}$ is used as a starting vector. In a next step, the second order Arnoldi (SOAR) algorithm, compare e.g. [13, 14] is repeated in its simplest form: single input, only one expansion point, no orthogonalization to an additional already given initial basis, real expansion point and no deflation strategy. The SOAR Algorithm produces an orthonormal basis $\boldsymbol{V}_u$ and the upper Hessenberg matrix $\boldsymbol{T}$ of the second order Krylov-subspace $\mathcal{G}_r(\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{r}_1)$.

Algorithm 4 Second Order Arnoldi Algorithm (SOAR)

```
Input: matrix $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{r}_1$, $J$
Output: basis $\boldsymbol{V}_u$ and Hessenberg matrix $\boldsymbol{T}$
/* Initialize */
$\boldsymbol{p}_u = \frac{\boldsymbol{r}_1}{\|\boldsymbol{r}_1\|_2}$
$\boldsymbol{p}_l = \boldsymbol{0}$
$\boldsymbol{V}_{u(:,1)} = \boldsymbol{p}_u$
$\boldsymbol{V}_{l(:,1)} = \boldsymbol{p}_l$
for $j = 1 : J$ do
    $\boldsymbol{p}_u = \boldsymbol{M}_1 \cdot \boldsymbol{V}_{u(:,j)} + \boldsymbol{M}_2 \cdot \boldsymbol{V}_{l(:,j)}$
    $\boldsymbol{p}_l = \boldsymbol{V}_{u(:,j)}$
    for $k = 1 : j$ do
        $\boldsymbol{T}_{(k,j)} = \boldsymbol{V}^T_{u(:,k)} \cdot \boldsymbol{p}_u$
        $\boldsymbol{p}_u = \boldsymbol{p}_u - \boldsymbol{V}_{u(:,k)} \boldsymbol{T}_{(k,j)}$
        $\boldsymbol{p}_l = \boldsymbol{p}_l - \boldsymbol{V}_{l(:,k)} \boldsymbol{T}_{(k,j)}$
    end for
    $\boldsymbol{T}_{(j+1,j)} = \|\boldsymbol{p}_u\|_2$
    $\boldsymbol{V}_{u(:,j+1)} = \frac{\boldsymbol{p}_u}{\boldsymbol{T}_{(j+1,j)}}$
    $\boldsymbol{V}_{l(:,j+1)} = \frac{\boldsymbol{p}_l}{\boldsymbol{T}_{(j+1,j)}}$
end for
```

The basic recurrence of the SOAR algorithm is:

$$\begin{aligned} \boldsymbol{M}_1 \cdot \boldsymbol{V}_{u(:,1:J)} + \boldsymbol{M}_2 \cdot \boldsymbol{V}_{l(:,1:J)} &= \boldsymbol{V}_{u(:,1:J)} \cdot \boldsymbol{T}_{(1:K,1:J)} + \boldsymbol{V}_{u(:,J+1)} \cdot \boldsymbol{e}_J^T \cdot \boldsymbol{T}_{(J+1,J)}, \\ \boldsymbol{V}_{u(:,1:J)} &= \boldsymbol{V}_{l(:,1:J)} \cdot \boldsymbol{T}_{(1:J,1:J)} + \boldsymbol{V}_{l(:,J+1)} \cdot \boldsymbol{e}_J^T \cdot \boldsymbol{T}_{(J+1,J)}. \end{aligned} \tag{10}$$

In addition, the basic recurrence of the SOAR process can be written in compact form

$$\begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_2 \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{V}_{u(:,1:J)} \\ \boldsymbol{V}_{l(:,1:J)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_{u(:,J+1)} \\ \boldsymbol{V}_{l(:,J+1)} \end{bmatrix} \cdot \hat{\boldsymbol{T}}_J, \tag{11}$$

where $\hat{\boldsymbol{T}}_J$ is an $((J+1) \times J$ upper Hessenberg matrix of the form $\hat{\boldsymbol{T}}_J = \begin{bmatrix} \boldsymbol{T}_{(1:K,1:J)} \\ \boldsymbol{e}_J^T \cdot \boldsymbol{T}_{(j+1,j)} \end{bmatrix}$. The essential difference between the SOAR algorithm and the Arnoldi algorithm is the fact that the SOAR algorithm enforces the orthonormality of the vectors $\boldsymbol{V}_u$ of dimension N whereas the Arnoldi process enforces the orthonormality of the vector $\begin{bmatrix} \boldsymbol{V}_u \\ \boldsymbol{V}_l \end{bmatrix}$ of dimension $2N$. That is, the SOAR algorithm ensures only the orthonormality of $\boldsymbol{V}_u$, however, the orthonormality of $\boldsymbol{V}_l$ follows due to the recursion properties of the second order Arnoldi algorithm. As proven by [14] for SISO systems the system moment $\boldsymbol{R}_j^{s_k}$ is analytically related to the output Hessenberg matrix $\boldsymbol{T}$ by the following analytical expression:

$$\begin{bmatrix} \boldsymbol{R}_j^{s_k} \\ \boldsymbol{R}_{j+1}^{s_k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_u \\ \boldsymbol{V}_l \end{bmatrix} \boldsymbol{T}^j_{(1:K,1:J)} \cdot \boldsymbol{e}_1, \quad \text{for } j = 0, 1, \dots, J-1. \tag{12}$$

Another property of the SOAR algorithm is, e.g. proven in [14]: The first unmatched output moment error $\Delta\boldsymbol{T}^{s_k}_{j+1} = \boldsymbol{T}^{s_k}_{j+1} - \bar{\boldsymbol{T}}^{s_k}_{j+1}$ can be calculated as

$$\Delta\boldsymbol{T}^{s_k}_{j+1} = \boldsymbol{T}^{s_k}_{j+1} - \bar{\boldsymbol{T}}^{s_k}_{j+1} = \boldsymbol{C}_e \left(\prod_{i=1}^{j} \boldsymbol{T}_{(i+1,i)} \right). \tag{13}$$

For MIMO Systems a block second order Arnoldi algorithm (BSOAR), explained e.g. in [16, 14] can be used to calculate an orthonormal basis of the second order Krylov-subspace $\mathcal{G}_J$ for MIMO systems. The major difference between the BSOAR algorithm to the second order Arnoldi (SOAR) is the initial step. The orthonormality of the initial matrix is preserved by using the QR decomposed $\boldsymbol{Q} \cdot \boldsymbol{R} = \mathrm{QR}(\boldsymbol{R}_1)$ starting matrix. In this context, in Morembs the dual second order rational Arnoldi algorithm explained in [17] is implemented. The dual second order algorithm used in Morembs is optimized for the usage of model reduction in EMBS. The algorithm has the following properties which are very important to successfully reduce mechanical systems. Without considering these facts wrong reduction results are achieved

- Moment matching can be achieved at multiple arbitrary expansion points s_k. First, an orthogonal basis $\boldsymbol{V}_1$ of the Krylov-subspace

  $$\mathcal{G}^{s_1}_{J_b^{s_1}}(-\check{\boldsymbol{K}}_1^{-1} \cdot \check{\boldsymbol{D}}_1, -\check{\boldsymbol{K}}_1^{-1} \cdot \boldsymbol{M}_e, -\check{\boldsymbol{K}}_1^{-1} \cdot \boldsymbol{B}_e) \tag{14}$$

  is built. In successive steps the i-th orthogonal bases $\boldsymbol{V}_i$ of the Krylov-subspace

  $$\mathcal{G}^{s_i}_{J_b^{s_i}}(-\check{\boldsymbol{K}}_i^{-1} \cdot \check{\boldsymbol{D}}_i, -\check{\boldsymbol{K}}_i^{-1} \cdot \boldsymbol{M}_e, -\check{\boldsymbol{K}}_i^{-1} \cdot \boldsymbol{B}_e) \tag{15}$$

  are calculated, where in the Gram-Schmidt procedure the new basis vectors are also orthogonalized to previous calculated bases

  $$\boldsymbol{V}_i \perp \boldsymbol{V}_{i-1} \perp \cdots \perp \boldsymbol{V}_1. \tag{16}$$

  If both projection matrices $\boldsymbol{V} \neq \boldsymbol{W}$ are needed, the same procedure is done with the output Krylov-subspace, for details see [17].

- In modal model reduction of EMBS usually mass orthogonal eigenmodes $\boldsymbol{\phi}^T \cdot \boldsymbol{M}_e \cdot \boldsymbol{\phi} = \boldsymbol{I}$ are used in the projection process. For model reduction with Krylov-subspaces a mass orthogonal projection matrix is calculated if in the Gram-Schmidt orthogonalization instead of the standard scalar product

  $$< \boldsymbol{u}, \boldsymbol{v} >= \boldsymbol{v}^T \cdot \boldsymbol{u} \tag{17}$$

  the energy scalar product with the regular mass matrix $\boldsymbol{M}_e$

  $$< \boldsymbol{u}, \boldsymbol{v} >_{\boldsymbol{M}_e} = \boldsymbol{v}^T \cdot \boldsymbol{M}_e \cdot \boldsymbol{u} \tag{18}$$

  is used. The calculated projection matrices are then automatically mass orthogonal $\boldsymbol{V}^T \cdot \boldsymbol{M} \cdot \boldsymbol{V} = \boldsymbol{I}$ after the model reduction with the Krylov-subspace. The usage of an energy scalar product (18) is computationally more expensive then the usage of the standard scalar product (17) due to the fact that always an expensive matrix vector product with a matrix of size N is performed.

- Krylov-subspace orthogonal to an initial basis: For model reduction in EMBS it is important that the projection matrices are orthogonal to the mass orthogonal rigid body modes. The mass orthogonal rigid body modes $\boldsymbol{\phi}_{rigid}$ are analytically calculated from the translational $\boldsymbol{S}_t$ and rotational $\boldsymbol{S}_r$ projectors and are given as an initial mass orthogonal basis in the Gram-Schmidt orthogonalization process. Another possibility is the usage of input and output matrices which are mass orthogonal to rigid body modes

$$\boldsymbol{B}_{e\perp} = (\boldsymbol{I} - \boldsymbol{M}_e \cdot \boldsymbol{\phi}_{rigid} \cdot \boldsymbol{\phi}_{rigid}^T) \cdot \boldsymbol{B}_e \tag{19}$$

$$\boldsymbol{C}_{e\perp} = \boldsymbol{C}_e \cdot (\boldsymbol{I} - \boldsymbol{M}_e \cdot \boldsymbol{\phi}_{rigid} \cdot \boldsymbol{\phi}_{rigid}^T)^T. \tag{20}$$

  With such input and output matrices only elastic deformations are excited. A physical explanation of this transformation can be found e.g. in [18, 19].

- Rational approximation: Usually a combination of purely imaginary, purely real and complex expansion points are used in the model reduction process. The system matrices of the original system $\boldsymbol{M}_e, \boldsymbol{D}_e, \boldsymbol{K}_e, \boldsymbol{B}_e, \boldsymbol{C}_e$ are real and it is important that the reduced order system is also real. This criterion is automatically fulfilled if the expansion points are real numbers, but not if at least one $s_k \in \mathbb{C}$ of them has a nonzero imaginary part $\Im(s_k) \neq 0$. The important observation from [20] is used to achieve a real reduced system. That is, for every expansion point s_k implicitly the conjugate complex expansion point $\bar{s}_k$ is also used as an expansion point.

- Usually not all vectors of the block Krylov-subspace are linearly independent. The linearly dependent vectors are deleted with a process called *deflation*, compare e.g. [21]. The same ideas explained in [13] are used to deflate the linearly dependent vectors in the SOAR procedure.

5.2. Selection of expansion points

The performance of Krylov-subspace based reduction methods clearly depends on the choice of expansion points and on the order up to which the moments are matched. The selection of appropriate combinations of s_k, $\bar{J}_b$ and $\bar{J}_c$ is no trivial task and needs some experience by the user.

- First, only expansion points with a non-singular matrix pencil $(\boldsymbol{A} - s_k \boldsymbol{E})$ are allowed.

- The complex expansion points $s_k = s_k^r + i s_k^i$ could be either purely real $s_k = s_k^r + 0$, purely imaginary $s_k = 0 + i s_k^i$ or a combination $s_k = s_k^r + i s_k^i$.

- For model reduction of mechanical systems usually purely imaginary expansion points within the interesting frequency range $s_k = s_k^i \in [\omega_{min}, \omega_{max}]$ are chosen because such expansion points allow a physical explanation and usually lead to a good approximation in the interesting frequency range.

- Purely imaginary expansion point can be interpreted as the frequency response of the dynamical system excited with an harmonic force at the input nodes. They lead to really good approximation of the reduced transfer function towards the original transfer function in a small neighborhood around the imaginary expansion points.

- For real expansion points, the reduced transfer function converges towards the original transfer in a larger neighborhood but not as good as for an imaginary expansion point.

- Usually better results are achieved if multiple expansion points are chosen and more moments are matched at the specific expansion point.

References

[1] Bai, Z.: Krylov Subspace Techniques for Reduced-order Modeling of Large-scale Dynamical Systems. Applied Numerical Mathematics, Vol. 43, pp. 9–44, 2002.

[2] Beattie, C.; Gugercin, S.: Krylov-Based Model Reduction of Second-Order Systems with Proportional Damping. In Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC ’05), 12–15 December, Sevilla, Spain, pp. 2278–2283, 2005.

[3] Feldmann, P.; Freund, R.W.: Reduced-Order Modeling of Large Linear Subcircuits via a Block Lanczos Algorithm. In Proceedings of the 32nd Annual ACM/IEEE Design Automation Conference (DAC), 12–16 June, 1995, San Francisco, pp. 474–479, New York: ACM, 1995.

[4] Freund, R.; Feldmann, P.: The SyMPVL algorithm and its applications to interconnect simulation. In Proceeding International Conference on Simulation of Semiconductor Processes and Devices, SISPAD ’97, pp. 113 –116, 1997.

[5] Saad, J.: Iterative Methods for Sparse Linear Systems. Philadelphia: SIAM, 2. Edn., 2003.

[6] Elfadel, I.M.; Ling, D.D.: A Block Rational Arnoldi Algorithm for Multipoint Passive Model-Order Reduction of Multiport RLC Networks. In Proceedings of the 1997 International Conference on Computer-Aided Design (ICCAD ’97), pp. 66–77, 1997.

[7] Antoulas, A.: Approximation of Large-Scale Dynamical Systems. Philadelphia: SIAM, 2005.

[8] Castane-Selga, R.; Eid, R.; Lohmann, B.: On the stability of Krylov-based order reduction using invariance properties of the controllability subspace. In Roppencker, G.; Lohmann, B. (Eds.): Methoden und Anwendungen der Regelungstechnik, Erlangen-Münchener Workshops 2007 und 2008. Shaker Verlag, 2009.

[9] Antoulas, A.C.: On Pole Placement in Model Reduction. at - Automatisierungstechnik, Vol. 55, No. 9, pp. 443–448, 2007.

[10] Grimme, E.: Krylov Projection Methods for Model Reduction. Ph.D. thesis, University of Illinois at Urbana-Champaign, 1997.

[11] Bai, Z.; Feldmann, P.; Freund, R.W.: In Stable and Passive Reduced-Order Models Based on Partial Padé Approximation Via the Lanczos Process: Numerical Analysis Manuscript No. 97–3–10, Bell Laboratories. Murray Hill, 1997.

[12] Jaimoukha, I.M.; Kasenally, E.M.: Implicitly Restarted Krylov Subspace Methods For Stable Partial Realizations. SIAM Journal on Matrix Analysis & Applications, Vol. 18, No. 3, pp. 633–652, 1997.

[13] Bai, Z.; Su, Y.: Dimension Reduction of Large-scale Second-order Dynamical Systems via a Second-order Arnoldi Method. SIAM Journal on Scientific Computing, Vol. 26, No. 5, pp. 1692–1709, 2005.

[14] Chu, C.C.; Tsai, H.C.; Lai, M.H.: Structure Preserving Model-order Reductions of MIMO Second-order Systems using Arnoldi Methods. Mathematical and Computer Modelling, Vol. 51, No. 7, pp. 956–973, 2010.

[15] Salimbahrami, S.B.: Structure Preserving Order Reduction of Large Scale Second Order Models. Dissertation, Technische Universität München. München, 2005.

[16] Liu, B.; Zeng, X.; Su, Y.; Tao, J.; Bai, Z.; Chiang, C.; Zhou, D.: Block SAPOR: Block Second-Order Arnoldi Method for Passive Order Reduction of Multi-Input Multi-Output RCS Interconnect Circuits. In ASP-DAC, pp. 244–249, 2005.

[17] Lehner, M.: Modellreduktion in elastischen Mehrkörpersystemen (in German). Dissertation, Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart, Vol. 10. Aachen: Shaker Verlag, 2007.

[18] Craig, R.: Structural Dynamics. New York: John Wiley & Sons, 1981.

[19] Géradin, M.; Rixen, D.: Mechanical Vibrations. Chichester: John Wiley & Sons, 2. Edn., 1998.

[20] Ruhe, A.: Rational Krylov Algorithms for Nonsymmetric Eigenvalue Problems. In Golub, G.; Greenbaum, A.; Luskin, M. (Eds.): Recent Advances in Iterative Methods, IMA Volumes in Mathematics and its Applications 60, pp. 149–164, Berlin: Springer, 1994.

[21] Freund, R.: Structure-Preserving Model Order Reduction. In Schilders, W.; van der Vorst, H.; Rommes, J. (Eds.): Model Order Reduction: Theory, Research Aspects and Applications, Vol. 13 of Mathematics in Industry, pp. 49–75. Berlin: Springer, 2008.