On Efficiently Updating Singular Value Decomposition Based Reduced Order Models

Ralf Zimmermann

GAMM Workshop Applied and Numerical Linear Algebra
with Special Emphasis on Model Reduction
Bremen, Sep. 22.-23. 2011
The POD-based ROM approach

1. Input: CFD snapshots

Flow solutions at $m$ different flow conditions
The POD-based ROM approach

1. Input: CFD snapshots
   Flow solutions at $m$ different flow conditions

2. POD Basis
   Orthonormal basis ordered by information content spanning the same space
The POD-based ROM approach

1. Input: CFD snapshots
   - Flow solutions at \( m \) different flow conditions

2. POD Basis
   - Orthonormal basis ordered by information content spanning the same space

3. Order Reduction
   - Select \( \tilde{m} < m \) POD components with largest information content
The POD-based ROM approach

1. Input: CFD snapshots
   Flow solutions at $m$ different flow conditions

2. POD Basis
   Orthonormal basis ordered by information content spanning the same space

3. Order Reduction
   Select $\hat{m} < m$ POD components with largest information content

4. Prediction step
   Determine POD-ROM coefficients by interpolation / solving low-order PDEs / least-squares optimization
The POD-based ROM approach

1. Input: CFD snapshots
   Flow solutions at \( m \) different flow conditions

2. POD Basis
   Orthogonal basis ordered by information content spanning the same space

3. Order Reduction
   Select \( \tilde{m} < m \) POD components with largest information content

4. Prediction step
   Determine POD-ROM coefficients by interpolation / solving low-order PDEs / least-squares optimization

5. Output: approximated flow field

Untried flow condition
POD based reduced order models are a powerful tool...

Industrial aircraft configuration
ROM Speed-up factor > 300

Grid size ~9Mio, subsonic Ma = 0.2
Snapshot data at AoA = -1°, 0°, 1°, 2°
Prediction at AoA = 7° (Extrapolation!)
How to compute POD/SVD of augmented data set efficiently?
Nomenclature

Snapshot matrix: \( Y_m = (W^1, ..., W^m) \in \mathbb{R}^{n \times m} \)

Snapshot average: \( A_m = \frac{1}{m} \sum_{j=1}^{m} W^j \)

Centering: \( \bar{Y}_m = Y_m - A_m 1^T = (W^1 - A_m, ..., W^m - A_m) =: (\bar{W}^1, ..., \bar{W}^m) \),

SVD: \( \bar{Y}_m = U \Sigma V^T \)

Relative Information content: \( ric(r_m) = \frac{\sum_{i=1}^{r_m} \sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2}, \quad r_m \leq m - 1 \)
Reduced-order representation

Discard columns corresponding to small singular values:

\[ \bar{Y}_m \approx U_m \Sigma_m V_m^T, \]

\[ U_m = (U^1, \ldots, U^{r_m}) \in \mathbb{R}^{n \times r_m}, \quad V_m = (V^1, \ldots, V^{r_m}) \in \mathbb{R}^{m \times r_m}, \quad \Sigma_m = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{r_m} \end{pmatrix} \]

**Definition:**
The data set \((U_m, \Sigma_m, V_m, A_m, n, m, r_m)\) is called reduced-order model of order \(r_m\) of \(Y_m\).
The ratio \(\frac{r_m}{m}\) is called the compression rate.
The SVD basis update problem

**Given:** ROM \( (U_m, \Sigma_m, V_m, A_m, n, m, r_m) \) of \( Y_m \in \mathbb{R}^{n \times m} \).

\( p \) new snapshot observations \( (W^{m+1}, \ldots, W^{m+p}) \)

**Task:** Compute ROM \( (U_{m+p}, \Sigma_{m+p}, V_{m+p}, A_{m+p}, n, m + p, r_{m+p}) \) of \( Y_{m+p} \).

**Requirement:**
- use only the previous stage ROM and the incoming snapshots!
- \( n >> m \)! Keep computational costs depending on \( n \) as low as possible
Objective

Efficient SVD basis update

- Update SVD basis without having to store the initial snapshots
Objective

Efficient SVD basis update
- Update SVD basis without having to store the \textit{initial} snapshots
Objective

Efficient SVD basis update
- Update SVD basis without having to store the initial snapshots
SVD basis update strategies

Updating the snapshot mean

Shift vector: \( T_{m+p} := \frac{1}{m+p} \left( pA_m - \sum_{i=m+1}^{m+p} W^i \right) \)

\[ \Rightarrow A_{m+p} = A_m - T_{m+p} \]

Shift update snapshots to the previous-stage center:

\( \overline{W}^{m+i} := W^{m+i} - A_m, \quad i = 1, \ldots, p \)

To do: Decompose

\[
\bar{Y}_{m+p} = \left( \bar{Y}_m, \overline{W} \right) + T_{m+p} 1^T_{m+p} \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T,
\]
SVD basis update strategies

Updating the snapshot mean

Shift vector: \( T_{m+p} := \frac{1}{m+p} \left( pA_m - \sum_{i=m+1}^{m+p} W^i \right) \)

\[ \Rightarrow A_{m+p} = A_m - T_{m+p} \]

Shift update snapshots to the previous-stage center:

\( \overline{W}^{m+i} := W^{m+i} - A_m, \quad i = 1, \ldots, p \)

To do: Decompose

\[ \overline{Y}_{m+p} = \left( \overline{Y}_m, \overline{W} \right) - T_{m+p} 1^T_{m+p} \approx U_{m+p} \sum_{m+p} V_{m+p}^T, \]
SVD basis update strategies (A): two-steps SVD$^{1,2}$

Setting \[ X = (\bar{Y}_m, 0^{n \times p}) , \quad B^T = (0^{p \times m}, I^{p \times p}) \]

it holds
\[
(\bar{Y}_m, \bar{W}) = X + \bar{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m (V_m^T, 0^{r_m \times p})
\]

\[\text{References}\]


SVD basis update strategies (A): two-steps SVD\(^1,2\)

Setting \( X = \begin{pmatrix} \overline{Y}_m & 0^{n \times p} \end{pmatrix}, B^T = \begin{pmatrix} 0^{p \times m} & I^{p \times p} \end{pmatrix} \)

it holds \( \begin{pmatrix} \overline{Y}_m, \overline{W} \end{pmatrix} = X + \overline{W}B^T \) and \( X \approx U_m \Sigma_m \begin{pmatrix} V_m^T & 0^{r_m \times p} \end{pmatrix} \)

\[\text{rank-p SVD update problem}\]

---


SVD basis update strategies (A): two-steps SVD\textsuperscript{1,2}

Setting \[ X = (\overline{Y}_m, 0_{n \times p}), \quad B^T = (0_{p \times m}, I_{p \times p}) \]

it holds

\[(\overline{Y}_m, \overline{W}) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m (V_m^T, 0_{r_m \times p}) \]

Factoring out orthogonal components:

\[ X + \overline{W}B^T = (U_m, \overline{W}) \begin{pmatrix} \Sigma_m & 0 \\ 0 & I_{p \times p} \end{pmatrix} \begin{pmatrix} (V_m^T, 0_{r_m \times p}) \\ (0_{p \times m}, I_{p \times p}) \end{pmatrix} \]

\textsuperscript{1} M. Brand: “Fast low-rank modifications of the thin SVD”, \textit{Lin. Alg. and its Appl.} 415, 2006

\textsuperscript{2} P. Hall et al.: “Merging and splitting eigenspace models”, \textit{IEEE Trans. Pattern analysis and Machine Intelligence}, 22(9), 2000
SVD basis update strategies (A): two-steps SVD\textsuperscript{1,2}

Setting \( X = (\overline{Y}_m,0^{n\times p}) \), \( B^T = (0^{p\times m}, I^{p\times p}) \)

it holds

\[
(\overline{Y}_m,\overline{W}) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m (V_m^T,0^{r_m\times p})
\]

Factoring out orthogonal components:

\[
X + \overline{W}B^T = (U_m, P_{\text{orth}}) \left( \begin{array}{cc}
\Sigma_m & U_m^T \overline{W} \\
0 & P_{\text{orth}}^T P
\end{array} \right) \left( \begin{array}{c}
(V_m^T,0^{r_m\times p}) \\
(0^{p\times m}, I^{p\times p})
\end{array} \right)
\]

where \( P = \overline{W} - U(U_m^T \overline{W}) \), \( P_{\text{orth}} = \text{orth}(P) \) \( \text{e.g. via Gram Schmidt} \)

\textsuperscript{1} M. Brand: “Fast low-rank modifications of the thin SVD”, Lin. Alg. and its Appl. 415, 2006

\textsuperscript{2} P. Hall et al.: “Merging and splitting eigenspace models”, IEEE Trans. Pattern analysis and Machine Intelligence, 22(9), 2000
SVD basis update strategies (A): two-steps SVD\textsuperscript{1,2}

Setting \( X = \left( \bar{Y}_m, 0^{n \times p} \right) \), \( B^T = \left( 0^{p \times m}, I^{p \times p} \right) \)

it holds

\[
\left( \bar{Y}_m, \bar{W} \right) = X + \bar{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left( V_m^T, 0^{r_m \times p} \right)
\]

Factoring out orthogonal components:

\[
X + \bar{W}B^T = \left( U_m, P_{\text{orth}} \right) \begin{pmatrix}
\Sigma_m & U_m^T \bar{W} \\
0 & P_{\text{orth}}^T P
\end{pmatrix} \begin{pmatrix}
\left( V_m^T, 0^{r_m \times p} \right) \\
\left( 0^{p \times m}, I^{p \times p} \right)
\end{pmatrix}
\]

\text{orthonormal columns}

\textsuperscript{1} M. Brand: “Fast low-rank modifications of the thin SVD”, \textit{Lin. Alg. and its Appl.} 415, 2006

\textsuperscript{2} P. Hall et al.: “Merging and splitting eigenspace models”, \textit{IEEE Trans. Pattern analysis and Machine Intelligence}, 22(9), 2000
SVD basis update strategies (A): two-steps SVD\(^1,2\)

Setting \( X = (\overline{Y}_m, 0^{n \times p}) \), \( B^T = (0^{p \times m}, I^{p \times p}) \)

it holds

\[
(\overline{Y}_m, \overline{W}) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \begin{pmatrix} V_m^T, 0_{m \times p}^{r_m} \end{pmatrix}
\]

Factoring out orthogonal components:

\[
X + \overline{W}B^T = (U_m, P_{\text{orth}}) \begin{pmatrix} \Sigma_m & U_m^T \overline{W} \\ 0 & P_{\text{orth}}^T P \end{pmatrix} \begin{pmatrix} (V_m^T, 0_{m \times p}^{r_m}) \\ (0^{p \times m}, I^{p \times p}) \end{pmatrix}
\]

orthonormal columns

---


SVD basis update strategies (A): two-steps SVD$^{1,2}$

Setting \( X = (\overline{Y}_m, 0^{n \times p}) \), \( B^T = \left( 0^{p \times m}, I^{p \times p} \right) \)

it holds

\[
\left( \overline{Y}_m, \overline{W} \right) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left( V_m^T, 0_{r_m \times p} \right)
\]

Factoring out orthogonal components:

\[
X + \overline{W}B^T = \left( U_m, P_{orth} \right) \begin{pmatrix}
\Sigma_m & U_m^T \overline{W} \\
0 & P_{orth}^T P
\end{pmatrix} \begin{pmatrix}
(V_m^T, 0_{r_m \times p}) \\
(0^{p \times m}, I^{p \times p})
\end{pmatrix}
\]

size \((r_m + p) \times (r_m + p)\)

---


SVD basis update strategies (A): two-steps SVD$^{1,2}$

Setting \[ X = \left( \overline{Y}_m, 0^{n \times p} \right), \quad B^T = \left( 0^{p \times m}, I^{p \times p} \right) \]

it holds \[ (\overline{Y}_m, \overline{W}) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left( V_m^T, 0^{r_m \times p} \right) \]

Factoring out orthogonal components:

\[ X + \overline{W}B^T = (U_m, P_{orth}) \left( \tilde{U} \tilde{\Sigma} \tilde{V}^T \right) \begin{pmatrix} (V_m^T, 0^{r_m \times p}) \\ (0^{p \times m}, I^{p \times p}) \end{pmatrix} \]

---


SVD basis update strategies (A): two-steps SVD\(^1,2\)

**Setting** \( X = (\overline{Y}_m, 0^{n \times p}), B^T = (0^{p \times m}, I^{p \times p}) \)

it holds

\[
(\overline{Y}_m, \overline{W}) = X + \overline{W}B^T \quad \text{and} \quad X \approx U_m \Sigma_m \left( V_m^T, 0^{r_m \times p} \right)
\]

**Factoring out orthogonal components:**

\[
X + \overline{W}B^T = \left[ (U_m, P_{\text{orth}})\tilde{U} \right] \tilde{\Sigma} \left[ \tilde{V}^T \begin{pmatrix} (V_m^T, 0^{r_m \times p}) \\ (0^{p \times m}, I^{p \times p}) \end{pmatrix} \right]
\]

---


SVD basis update strategies (A): two-steps SVD$^{1,2}$

Repeat for shifting to new center:

$$\bar{Y}_{m+p} = (\bar{Y}_{m}, \bar{W}) + T_{m+p} 1_{m+p}^T 1_{m+p}^{-1} \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

SVD now known

---


SVD basis update strategies (A): two-steps SVD\(^1,2\)

**Comments:**

- ‘re-orthogonalization’ expensive, additional (large-scale) SVD required
- Less robust via Gram-Schmidt

\[
P = \bar{W} - U(U_m^T \bar{W}), \quad P_{orth} = \text{orth}(P)
\]

- Parallelization is more involved

---


SVD basis update strategies (B): EVD$^3$+SVD

Objective:  $$\bar{Y}_{m+p} = (\bar{Y}_m, \bar{W}) + T_{m+p} 1^T_{m+p} \approx U_{m+p} \Sigma_{m+p} V^T_{m+p},$$

First step: Compute SVD of $$\left( \bar{Y}_m, \bar{W} \right)$$

Reduce to symmetric EVD

$$\left( \bar{Y}_m, \bar{W} \right)^T \left( \bar{Y}_m, \bar{W} \right) = \begin{pmatrix} \bar{Y}_m \bar{Y}_m^T & \bar{Y}_m \bar{W}^T \\ \bar{W}^T \bar{Y}_m & \bar{W}^T \bar{W} \end{pmatrix}$$

SVD basis update strategies (B): $\text{EVD}^3 + \text{SVD}$

Objective: 
\[
\bar{Y}_{m+p} = \left( \bar{Y}_m, \bar{W} \right) + T_{m+p} 1^T_{m+p} \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T,
\]

First step: Compute SVD of $\left( \bar{Y}_m, \bar{W} \right)$

Reduce to symmetric EVD
\[
\left( \bar{Y}_m, \bar{W} \right)^T \left( \bar{Y}_m, \bar{W} \right) \approx \begin{pmatrix}
V_m \Sigma_m^2 V_m^T & V_m \Sigma_m U_m^T \bar{W} \\
\bar{W}^T U_m \Sigma_m V_m^T & \bar{W}^T \bar{W}
\end{pmatrix}
\]

Exploit previous stage SVD

SVD basis update strategies (B): EVD$^3$+SVD

Objective: \[ \bar{Y}_{m+p} = (\bar{Y}_m, \bar{W}) + T_{m+p} (1 + T_{m+p}) \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T, \]

First step: Compute SVD of \( (\bar{Y}_m, \bar{W}) \)

Reduce to symmetric EVD

\[ \begin{pmatrix} \bar{Y}_m & \bar{W} \end{pmatrix}^T \begin{pmatrix} \bar{Y}_m & \bar{W} \end{pmatrix} \approx \begin{pmatrix} V_m & 0 \\ 0 & I_{p \times p} \end{pmatrix} \begin{pmatrix} \Sigma_m^2 & \Sigma_m U_m^T \bar{W} \\ \bar{W}^T U_m \Sigma_m & \bar{W}^T \bar{W} \end{pmatrix} \begin{pmatrix} V_m^T & 0 \\ 0 & I_{p \times p} \end{pmatrix} \]

factor out

**SVD basis update strategies (B): EVD\(^3+\)SVD**

**Objective:**

\[
\bar{Y}_{m+p} = (\bar{Y}_m, \bar{W}) + T_{m+p} 1^T_{m+p} \approx U_{m+p} \Sigma_{m+p} V^T_{m+p},
\]

**First step:** Compute SVD of \((\bar{Y}_m, \bar{W})\)

**Reduce to symmetric EVD**

\[
(\bar{Y}_m, \bar{W})^T (\bar{Y}_m, \bar{W}) \approx \begin{pmatrix} V_m & 0 \\ 0 & I_{p \times p} \end{pmatrix} \begin{pmatrix} \Sigma_m^2 & \Sigma_m U^T_m \bar{W} \\ \bar{W}^T U_m \Sigma_m & \bar{W}^T \bar{W} \end{pmatrix} \begin{pmatrix} V_m^T & 0 \\ 0 & I_{p \times p} \end{pmatrix}
\]

**size** \((r_m + p) \times (r_m + p)\)

SVD basis update strategies (B): EVD$^3$+SVD

Objective: 
$$\bar{Y}_{m+p} = \left(\bar{Y}_m, \bar{W}\right) + T_{m+p} \Sigma_{m+p} V_{m+p}^T \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

First step: Compute SVD of \(\left(\bar{Y}_m, \bar{W}\right)\)

Reduce to symmetric EVD

\[
\left(\bar{Y}_m, \bar{W}\right)^T \left(\bar{Y}_m, \bar{W}\right) \approx \begin{pmatrix} V_m & 0 \\ 0 & I_{p \times p} \end{pmatrix} \begin{pmatrix} \tilde{Q} \tilde{\Lambda} \tilde{Q}^T \\ 0 \end{pmatrix} \begin{pmatrix} V_m^T & 0 \\ 0 & I_{p \times p} \end{pmatrix}
\]

SVD basis update strategies (B): EVD$^3$+SVD

**Objective:**

$$\bar{Y}_{m+p} = (\bar{Y}_m, \bar{W}) + T_{m+p}1_T^{m+p} \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T,$$

**First step:** Compute SVD of \((\bar{Y}_m, \bar{W})\)

Reduce to symmetric EVD

\[
(\bar{Y}_m, \bar{W})^T (\bar{Y}_m, \bar{W}) \approx \begin{pmatrix} V_m & 0 \\ 0 & I^{p \times p} \end{pmatrix} \begin{pmatrix} \tilde{Q} \tilde{\Lambda} \tilde{Q}^T \\ V_m^T \end{pmatrix} \begin{pmatrix} V_m & 0 \\ 0 & I^{p \times p} \end{pmatrix}
\]

**Compute left singular vectors via**

$$\hat{U} = (U_m \Sigma_m, \bar{W}) \tilde{Q} \sqrt{\tilde{\Lambda}^{-1}}$$

SVD basis update strategies (B): $\text{EVD}^3 + \text{SVD}$

Use Brand’s method for shifting to new center:

$$
\bar{Y}_{m+p} = \left( \bar{Y}_m, \bar{W} \right) + T_{m+p} 1_{m+p}^T \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T,
$$

$\text{SVD now known}$

SVD basis update strategies (C): one-step EVD

Objective:

\[ \overline{Y}_{m+p} = (\overline{Y}_m, \overline{W}) + T_{m+p} 1_{m+p}^T \approx U_{m+p} \Sigma_{m+p} V_{m+p}^T, \]

Let \( X := (\overline{Y}_m, \overline{W}) + T_{m+p} 1_{m+p}^T \in \mathbb{R}^{n \times (m+p)} \)

Reduce to symmetric EVD,
exploit previous stage SVD in matrix products

\[ X^T X = \tilde{Q} \tilde{\Lambda} \tilde{Q}^T \in \mathbb{R}^{(m+p) \times (m+p)} \]

Compute left singular vectors via

\[ U_{m+p} = X\tilde{Q}\sqrt{\tilde{\Lambda}}^{-1} \]
SVD basis update strategies (B) and (C)

Comments:
- Straight forward parallelization!
  (only standard matrix products required in parallel)
Analysis of Computational costs

Count \( nmp \) flops for matrix product \( AB \), \( A \in \mathbb{R}^{n \times m} \), \( B \in \mathbb{R}^{m \times p} \)

\[ \text{Strategy (B) is more efficient than strategy (A):} \]

\[
\text{flops}(A) - \text{flops}(B) = \left(r_mp + \frac{1}{2}(p^2 + p)\right)n + O(orth(P))
\]

\[ = O((mp + p^2)n) \]
Analysis of Computational costs

The ranking of Strategy (B) vs. (C) depends on the compression rate!

Assumption:

\[ r_{m+p} = r_m + p = xm + p, \quad x = \frac{r_m}{m} \text{ (compression rate)} \]

Then the computational costs differ by

\[ \text{flops}(C) - \text{flops}(B) = n\left( (x - 2x^2)m^2 + (2 - 3x)(mp + m) - p^2 - p - 2 \right) \]
The ranking of Strategy (B) vs. (C) depends on the compression rate!

Solving the quadratic equation shows that Strategy (B) is more efficient than Strategy (C) if

\[ 0 < x < \frac{1}{4m} \left( \frac{m - 3(p + 1) + \sqrt{10m(p + 1) + m^2 + p^2 + 10p - 7}}{m} \right) \]

In practical implementations, use switch to select the most efficient update strategy.
Rules of thumb:

- For highly compressed models, Strategy (B) is more efficient than Strategy (C)
- For weakly compressed models, the opposite holds true

More precisely:

- Strategy (B) is more efficient than Strategy (C), if \( x \leq \frac{1}{2} \) and \( m \geq 2p + 1 \).

- Strategy (C) is more efficient than Strategy (B), if \( x \geq \frac{2}{3} \).
Example: Updating SVDs of Random matrices

\[ Y \in \mathbb{R}^{n \times m}, \ W \in \mathbb{R}^{n \times p}, \ n = 100,000, \ m = 500, \ p = 100 \]

<table>
<thead>
<tr>
<th>Method</th>
<th>Reconstruction error</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy (A), SVD-orth</td>
<td>2.504e-12</td>
<td>7.88</td>
</tr>
<tr>
<td>Strategy (A), QR-orth</td>
<td>2.333e-12</td>
<td>8.15</td>
</tr>
<tr>
<td>Strategy (B)</td>
<td>2.342e-12</td>
<td>6.32</td>
</tr>
<tr>
<td>Strategy (C)</td>
<td>2.279e-12</td>
<td>4.42</td>
</tr>
</tbody>
</table>

Uncompressed models \( r_m = m - 1 = 499, \ r_{m+p} = 599 \)
Example: Updating SVDs of Random matrices

\[ Y \in \mathbb{R}^{n \times m}, W \in \mathbb{R}^{n \times p}, n = 100,000, m = 500, p = 100 \]

<table>
<thead>
<tr>
<th>Method</th>
<th>Reconstruction error</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy (A), SVD-orth</td>
<td>92.21</td>
<td>4.18</td>
</tr>
<tr>
<td>Strategy (A), QR-orth</td>
<td>92.21</td>
<td>4.23</td>
</tr>
<tr>
<td>Strategy (B)</td>
<td>92.21</td>
<td>1.93</td>
</tr>
<tr>
<td>Strategy (C)</td>
<td>93.41</td>
<td>3.54</td>
</tr>
</tbody>
</table>

Compression level \( r_m = 200, \quad r_{m+p} = 200, \quad x = \frac{r_m}{m} = 0.4 \)
Summary

- The update strategies following the symmetric EVD approach are more efficient than the SVD update known from literature (Brand, Hall et al.) (Assumption: n>>m)

- Industrial point of view: SVD/EVD performed by ‘black box function’

- Most efficient: choose method depending on the compression rate

- The examples suggest that all methods share a similar level of accuracy (SVD-approach may suffer from orthogonal out-factoring, EVD-approaches may suffer from squaring the condition number)
Summary

For details and additional references, see:

“A comprehensive comparison of various algorithms for efficiently updating singular value decomposition based reduced order models”, DLR IB 124-2011/3

Freely available at DLR’s electronic library:
http://elib.dlr.de/70251
Thank you for your attention!