7. Moment Matching via Krylov Subspaces

Model Reduction of Linear Time Invariant Systems

Evgenii Rudnyi, IMTEK
Summary of Previous Lecture

• Transformation + Truncation = Projection
• There is a good theory based on Hankel singular values.
• Unfortunately in this case, it is hard to compute projection without transformation.

\[
\begin{pmatrix}
\bar{A}_{11} & \bar{A}_{12} \\
\bar{A}_{21} & \bar{A}_{22}
\end{pmatrix} = \begin{pmatrix}
U_1^{-1} \\
U_2^{-1}
\end{pmatrix} A \begin{pmatrix}
U_1 & U_2
\end{pmatrix}
\]

\[
x = \begin{pmatrix}
U_1 & U_2
\end{pmatrix} \begin{pmatrix}
z_1 \\
z_2
\end{pmatrix} = \begin{pmatrix}
U_1^{-1} \\
U_2^{-1}
\end{pmatrix} x
\]

\[
\bar{A} = U^{-1} A U
\]

\[
\bar{A}_{11} = U_1^{-1} A U_1 \\
\bar{B}_1 = U_1^{-1} B \\
\bar{C}_1 = C U_1
\]
Outline

Padé and Padé-type approximation
Implicit moment matching via Krylov subspaces
Computational problems
Error Estimate
The transfer function is a rational polynomial function: zeros and poles.

Then search an approximation among rational functions.

Expand them around some point.

Require that first moments are the same.

\[ E\dot{x} = Ax + Bu \]
\[ y = Cx \]
\[ G(s) = C(sE - A)^{-1}B \]
\[ G_{ij}(s) = \frac{(s - z_1) \ldots (s - z_N)}{(s - p_1) \ldots (s - p_N)} \]
\[ G_{ij,red}(s) = \frac{(s - z_1) \ldots (s - z_r)}{(s - p_1) \ldots (s - p_r)} \]
\[ G_{ij} = \sum_{0}^{\infty} m_i (s - s_0)^i \]
\[ m_i = m_{i,red}, \quad i = 0, \ldots, r \]
Padé and Padé-type approximants

• A reduced model of dimension $r$ contains $2r$ unknowns.
• Match $2r$ moments to determine them.
• Padé approximants.
• Does not preserve stability of the original system.

• Match less than $2r$ moments (quite often $r$).
• Padé-type approximants.
• They are not unique (cannot be transformed to each other).

• Explicit matching is numerically unstable.
• Use matrix identity.

\[ (I - sP)^{-1} = I + \sum_{i=1}^{\infty} P^i s^i = \sum_{i=0}^{\infty} P^i s^i \]

\[ G(s) = C(sE - A)^{-1} B \]

\[ (sE - A)^{-1} AA^{-1} = [A^{-1}(sE - A)]^{-1} A^{-1} \]

\[ G(s) = -C(I - sA^{-1}E)^{-1} A^{-1} B \]

\[ G(s) = -\left[ CA^{-1}B + \sum_{i=1}^{\infty} C(A^{-1}E)^i A^{-1}Bs^i \right] \]

• Let us take expansion point zero.

• Scalar transfer function.

• Single Input - Single Output.

• Input matrix is a column, Output matrix is a row.
Expansion Point

- Expansion around zero preserves stationary state.
- In principle, one can take any expansion point.
- Complex expansion point leads to problems.
- One can take several expansion points.

\[ E \dot{x} = Ax + Bu \]
\[ y = Cx \]

\[ G(s) = C(sE - A)^{-1}B \]

\[ G(s) = C \left[ I + (s - s_0)(s_0E - A)^{-1}E \right]^{-1} (s_0E - A)^{-1}B \]
**Definition**

- Action of matrix $A$ on vector $r$:
  \[
  \{ r, A \cdot r, \ldots, A^{k-1} \cdot r \}
  \]

- This is the right **Krylov subspace** $K_R(A, r)$ of $A$ and $b$ of order $k$.

- Action of transposed matrix $A$ on vector $l$:
  \[
  \{ l, A^T \cdot l, \ldots, A^{T^{k-1}} \cdot l \}
  \]

- This is the left **Krylov subspace** $K_L(A, l)$ of $A$ and $l$ of order $k$.

- Defines the low-dimensional basis of subspaces of order $k$.

- Direct computation is numerically unstable because of rounding errors.

- Included in 10 top algorithms of the 20th century.
Arnoldi Process

- Implicit moment matching.
- Right subspace

\[ K_r(A^{-1}E, A^{-1}b) \]
matches \( r \) moments.

- Padé-type approximant.
- Robust implementation.
- Does not take into account the output matrix.
- Working for complete output.

- Modified Gram-Schmidt.
- Produces basis \( V \) and small matrix \( H_A \)
- \( V \) is orthonormal: \( V^T \cdot V = I \)
- \( V^T \cdot A \cdot V = H_A \)
- \( H_A \) is upper-Hessenberg matrix
- A new vector must be orthogonalized to all the previous vectors.
Lanczos Algorithm

- Implicit moment matching.
- Right and left subspaces matches 2r moments.
- Padé approximant.
- Takes into account the output matrix.
- Numerical instability may occur.
- Two-sided Arnoldi.

- Lanczos vectors:
  \[ V = \text{span}\{r, A \cdot r, \ldots, A^{k-1} \cdot r\} \]
  \[ W = \text{span}\{l, A^T \cdot l, \ldots, A^{T(k-1)} \cdot l\} \]
- \( V \) and \( W \) are bi-orthogonal:
  \[ V^T \cdot W = \text{diag}(\delta_1, \delta_2, \ldots, \delta_k) \]
- Relation to \( A \):
  \[ V^T \cdot A \cdot W = H_L \]
- \( H_L \) is tri-diagonal matrix.
  Efficiency: Fast for large \( k \).
<table>
<thead>
<tr>
<th>Feature</th>
<th>Arnoldi</th>
<th>Lanczos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy of approximation</td>
<td>$r$ moments match</td>
<td>$2r$ moments match</td>
</tr>
<tr>
<td>Computational complexity</td>
<td>$O(2r^2n + 2rN_z(A))$</td>
<td>$O(16rn + 4rN_z(A))$</td>
</tr>
<tr>
<td>Invariance properties</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Numerical stability</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Preservation of stability and passivity</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Complete output approximation</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>
Matrix Vector Multiplication

- Krylov subspace algorithms are written in terms of matrix vector product.
- Well suited for sparse matrices.
- Implicit inverse via linear solve.
- Cholesky or LU decomposition.
- Back substitution is about 5% of factoring.

$$v_{i+1} = A^{-1}Ev_i$$

$$u_{i+1} = A^{-1}u_i$$

$$Au_{i+1} = u_i$$

$$A = L^T L$$

$$A = LU$$
### Table 1: Computational times on Sun Ultra-80 with 4 Gb of RAM in seconds

<table>
<thead>
<tr>
<th>dimension</th>
<th>nnz</th>
<th>stationary solution in ANSYS 7.0</th>
<th>stationary solution in ANSYS 8.0</th>
<th>factoring in TAUCS</th>
<th>generation of the first 30 vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 267</td>
<td>20 861</td>
<td>0.87</td>
<td>0.63</td>
<td>0.31</td>
<td>0.59</td>
</tr>
<tr>
<td>11 445</td>
<td>93 781</td>
<td>2.1</td>
<td>2.2</td>
<td>1.3</td>
<td>2.7</td>
</tr>
<tr>
<td>20 360</td>
<td>265 113</td>
<td>16</td>
<td>15</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>79 171</td>
<td>2 215 638</td>
<td>304</td>
<td>230</td>
<td>190</td>
<td>120</td>
</tr>
<tr>
<td>152 943</td>
<td>5 887 290</td>
<td>130</td>
<td>95</td>
<td>91</td>
<td>120</td>
</tr>
<tr>
<td>180 597</td>
<td>7 004 750</td>
<td>180</td>
<td>150</td>
<td>120</td>
<td>160</td>
</tr>
<tr>
<td>375 801</td>
<td>15 039 875</td>
<td>590</td>
<td>490</td>
<td>410</td>
<td>420</td>
</tr>
</tbody>
</table>
Model Reduction in Practice

Goal:
try different geometry and properties

Goal:
try different input functions

FEM
Device model

MOR

Compact model
System-level simulation