7. Moment Matching via Krylov Subspaces

Model Reduction of Linear Time Invariant Systems

Evgenii Rudnyi, IMTEK

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

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.

$$x = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = U^{-1} x = \begin{pmatrix} U_1^{-1} \\ U_2^{-1} \end{pmatrix} x$$

$$\overline{A} = U^{-1} A U$$

$$\downarrow$$

$$\begin{pmatrix} \overline{A}_{11} & \overline{A}_{12} \\ \overline{A}_{21} & \overline{A}_{22} \end{pmatrix} = \begin{pmatrix} U_1^{-1} \\ U_2^{-1} \end{pmatrix} A \begin{pmatrix} U_1 & U_2 \end{pmatrix} \longrightarrow \begin{aligned} \overline{A}_{11} &= U_1^{-1} A U_1 \\ \overline{B}_1 &= U_1^{-1} B \\ \overline{C}_1 &= C U_1 \end{aligned}$$

Outline

Padé and Padé-type approximation

Implicit moment matching via Krylov subspaces

Computational problems

Error Estimate

Padé approximation

- **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.

Moments

- Use matrix identity.
- Let us take expansion point zero.
- Scalar transfer function.
- Single Input - Single Output.
- Input matrix is a column, Output matrix is a row.

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

$$G(s) = C(sE - A)^{-1} B$$

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

$$G(s) = -C\left(I - sA^{-1}E\right)^{-1} A^{-1} B$$

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

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.

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

$$G(s) = C(sE - A)^{-1}B$$

$$\left[(s - s_0)E + s_0E - A\right]^{-1}(s_0E - A)(s_0E - A)^{-1}$$

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

Definition

- Action of matrix A on vector $\boldsymbol{r}$:

  $\{\boldsymbol{r}, A \cdot \boldsymbol{r}, \ldots, A^{k-1} \cdot \boldsymbol{r}\}$

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

- Action of transposed matrix A on vector $\boldsymbol{l}$:

  $\{\boldsymbol{l}, A^T \cdot \boldsymbol{l}, \ldots, A^{T^{k-1}} \cdot \boldsymbol{l}\}$

- This is the left **Krylov subspace** $K_L(A, \boldsymbol{l})$ of A and $\boldsymbol{l}$ 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

- 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.

• 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.

Lanczos Algorithm

- Lanczos vectors:

$$V = \mathrm{span}\{\boldsymbol{r}, A \cdot \boldsymbol{r}, \ldots, A^{k-1} \cdot \boldsymbol{r}\}$$

$$W = \mathrm{span}\{\boldsymbol{l}, A^T \cdot \boldsymbol{l}, \ldots, A^{T(k-1)} \cdot \boldsymbol{l}\}$$

- V and W are bi-orthogonal:

$$V^T \cdot W = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_k)$$

- Relation to A:

$$V^T \cdot A \cdot W = H_L$$

- H_L is tri-diagonal matrix.

Efficiency: Fast for large k.

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

Arnoldi vs. Lanczoz

	Arnoldi	Lanczos
Accuracy of approximation	**r moments match**	**2r moments match**
Computational complexity	$O(2r^2n + 2rN_z(A))$	$O(16rn + 4rN_z(A))$
Invariance properties	✘	✔
Numerical stability	✔	✘
Preservation of stability and passivity	✔	✘
Complete output approximation	✔	✘

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.

$$v_{i+1} = A^{-1} E v_i$$

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

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

$$A = L^T L$$

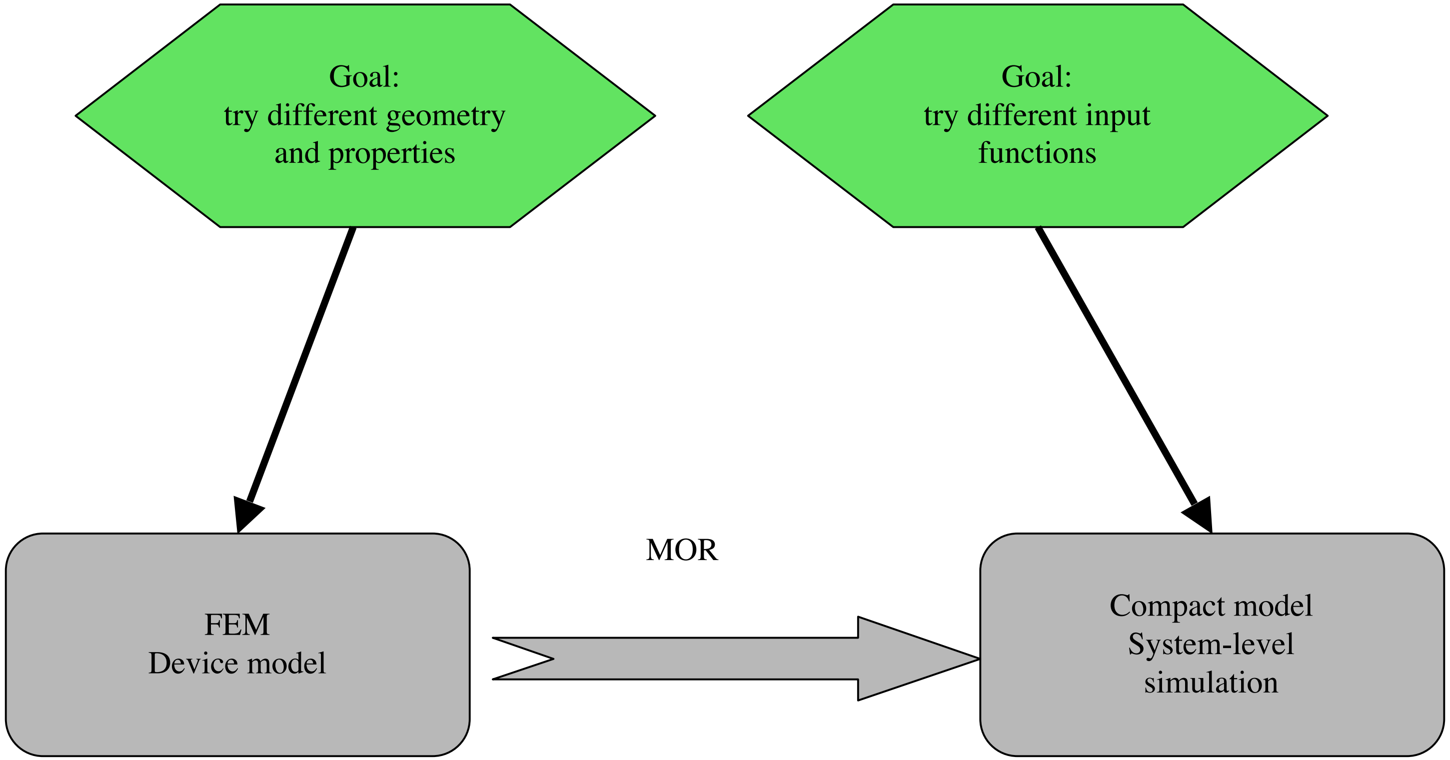
$$A = LU$$

Timing

Table 1: Computational times on Sun Ultra-80 with 4 Gb of RAM in seconds

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

Model Reduction in Practice



ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG