Automatic Compact Modelling for MEMS: Applications, Methods and Tools

Lecture 2:
Implicit Moment Matching via Arnoldi Process: Theory

Evgenii B. Rudnyi, Jan G. Korvink

http://www.imtek.uni-freiburg.de/simulation/mor4ansys/

IMTEK

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Outline

- Methods based on Hankel singular values (SLICOT)
- Implicit moment matching
- Solving a system of linear equations
- MOR for ANSYS

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

MOR and Transfer Function

- Representations in time and Laplace domains are equivalent.
- Evaluating the transfer function along the imaginary axis is enough (Bode plot).
- Model reduction is done in the Laplace domain:
  - ✓Approximating the transfer function.
  - ✓Formal dimension is the same.
  - ✓Complexity is reduced.

$$Y(s) = H(s)U(s)$$

$$y(t) = \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} Y(s)e^{st}ds$$

$$H(i\omega) = C(i\omega E + K)^{-1}B$$

$$\hat{H}(i\omega) = \hat{C}(i\omega\hat{E} + \hat{K})^{-1}\hat{B}$$

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Hankel Singular Values

- Dynamic system in the state-space form:

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

$$H(s) = C(sI - A)^{-1}B$$

- Lyapunov equations to determine controllability and observability Gramians:

$$AP + PA^T + BB^T = 0$$

$$A^TQ + QA + C^TC = 0$$

- Hankel singular values (HSV):
  - ✓square root from eingenvalues for product of Gramians.

$$\sigma_i = \sqrt{\lambda_i(PQ)}$$

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Global Error Estimate

- **Infinity norm**

$$\left\| H(s) - \hat{H}(s) \right\|_\infty =$$

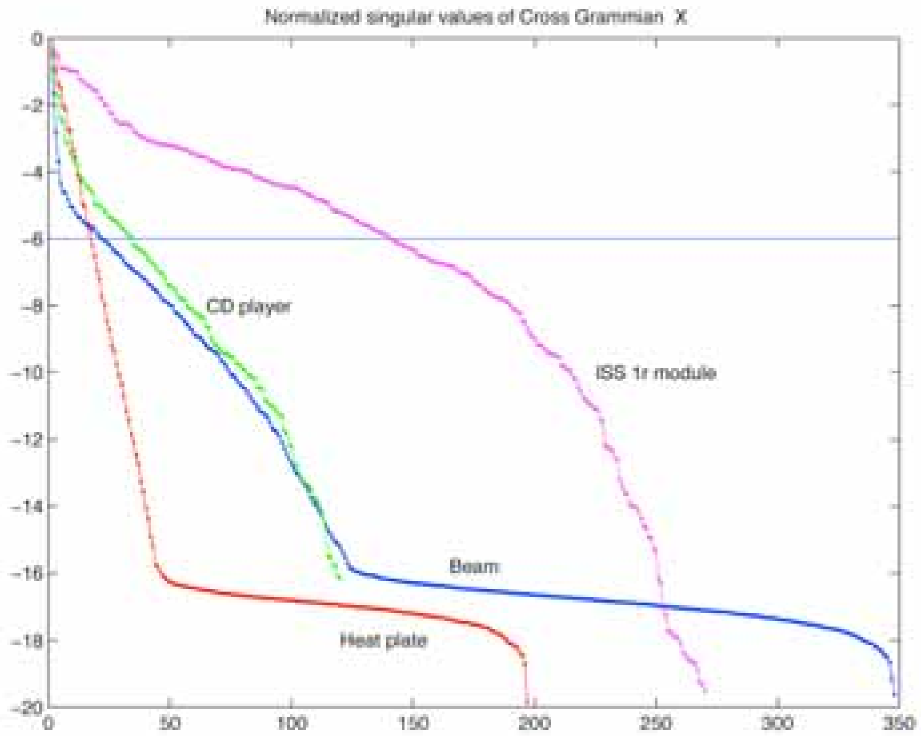
$$= \max_s abs(H(s) - \hat{H}(s))$$

- **Global error for a reduced model of dimension k**

$$\left\| H(s) - \hat{H}(s) \right\|_\infty <$$

$$2(\sigma_{k+1} + \ldots + \sigma_n)$$

- **Model reduction success depends on the decay of HSV.**



- **Log10[HSV(i)] vs. its number.**
- **From Antoulas review.**

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Methods Based on HSV

- **The theory works for stable systems.** $\lambda_i(A) < 0$
  - ✓In unstable systems something should be done with unstable poles.
- **Hankel Norm Approximation:**
  - ✓Produces an optimal solution.
- **Balanced Truncation Approximation:**
  - ✓Most often used.
  - ✓Faster than HNA.
  - ✓Does not preserve the stationary state.
- **Singular Perturbation Approximation:**
  - ✓Preserves the stationary state.
- **Frequency-weighted model reduction.**

$$\left\| V\left(H - \hat{H}\right)W \right\|_{\infty}$$

SLICOT

- **FORTRAN Code + Examples found at:**
  - ✓http://www.slicot.de
- **European Community BRITE-EURAM III Thematic Networks Programme.**
- **Implements all methods:**
  - ✓Balanced Truncation Appr.
  - ✓Singular Perturbation Appr.
  - ✓Hankel Norm Appr.
  - ✓Frequency-weighted MOR.
- **Has a parallel version.**

- **Yet, the computational complexity is O(N^3).**
- **Limited to "small" systems.**

Dimension	Serial	Parallel (4 processors)
600	60	25
1332	703	130
2450	4346	666
3906		2668

Moment Matching

- The transfer function is a rational polynomial function: zeros and poles.
- Then search an approximation among rational functions.
- Expand transfer functions at some point s_0 in the Tailor series.
- Require that first moments are the same.

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

$$y = Cx$$

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

$$H_{ij}(s) = \frac{(s - z_1)\ldots(s - z_N)}{(s - p_1)\ldots(s - p_N)}$$

$$\hat{H}_{ij}(s) = \frac{(s - z_1)\ldots(s - z_r)}{(s - p_1)\ldots(s - p_r)}$$

$$H_{ij} = \sum_{0}^{\infty} m_i (s - s_0)^i$$

$$m_i = \hat{m}_i, \quad i = 0,\ldots,r$$

Moments

- **Use matrix identity.**
- **Let us take expansion point zero.**
- **The simplest case of a scalar transfer function.**
  - ✓Single Input - Single Output.
  - ✓Input matrix is a column, output matrix is a row.

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

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

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

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

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

Krylov subspace

- Matrix *P* and vector *r*

$$\{r, Pr, P^2 r, \ldots, P^{k-1} r\}$$

- Right Krylov subspace

$$\mathfrak{I}_{R,k}(P,r) \qquad \mathfrak{I}_k(P,r)$$

- Transposed matrix *P* and vector *l*

$$\{l, P^T l, {P^T}^2 l, \ldots, {P^T}^{k-1} l\}$$

- Left Krylov subspace

$$\mathfrak{I}_{L,k}(P,l)$$

- Krylov subspace defines a low dimensional subspace:
  - ✓Basis is not unique.
- Direct computation is numerically unstable because of rounding errors.
- Robust computational algorithms are included in 10 top algorithms of the 20th century.

Implicit Moment Matching

- **Take matrix and vector corresponding to a given expansion point.**
- **Compute the orthogonal basis by the Arnoldi process.**
- **Project the original system on this basis.**
- **One can prove that this way the reduced model matches *k* moments.**

$$s_0 = 0 \qquad P = A^{-1}E \qquad r = A^{-1}b$$

$$V = span\{\Im(A^{-1}E, A^{-1}b)\}$$

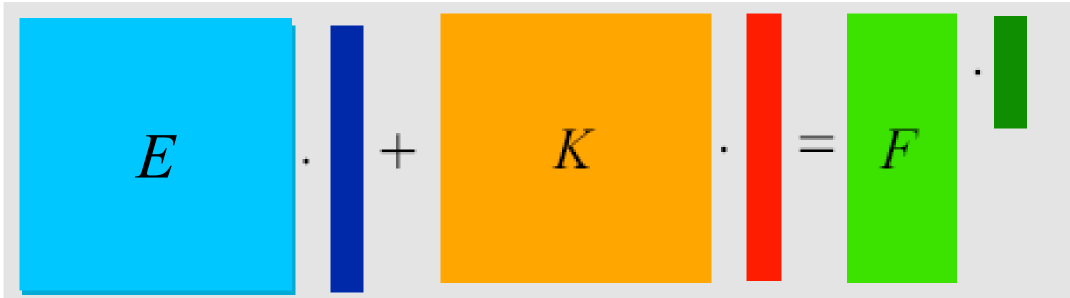
$$V = span\{r, Pr, P^2 r, \ldots P^{k-1} r\}$$

- **Multiple inputs:**
  - ✓Block Krylov Subspace,
  - ✓Block Arnoldi.
- **The Lanczos algorithm used the right and left Krylov subspaces.**
  - ✓More problems.
  - ✓We will not consider it in this course.

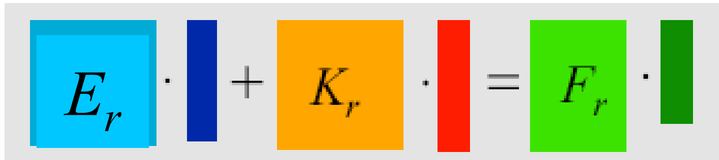
Projection

- Projection onto low-dimensional subspace.
- Very general approach, not limited to moment matching.
- Moment matching preserves *r* moments.

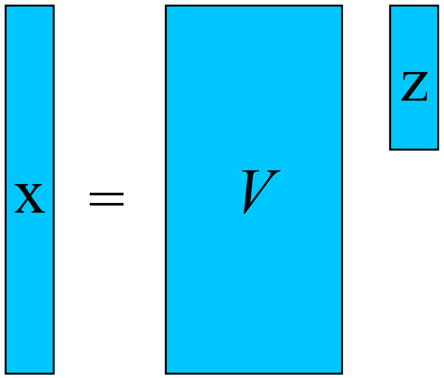
$$E\dot{x} + Kx = Bu$$



$$V^T E V \dot{z} + V^T K V z = V^T B u$$



$$x = Vz + \varepsilon$$



- Matrix from the left is different in the Lanczos algorithm.

Arnoldi vs. Lanczos

	Arnoldi	Lanczos
Accuracy of approximation	**r moments match**	**$2r$ moments match**
Computational complexity		
Invariance properties	✘	✔
Numerical stability	✔	✘
Preservation of stability and passivity	✔	✘
Complete output approximation	✔	✘

Arnoldi Process

- **Input: matrix *P* and vector *r***
- **Normalize vector $v_1 = r/|r|$**
- **Do ($i = 2, k$):**
  - ✓Next vector $w = P\, v_{i-1}$
  - ✓Orthogonalize w in respect to $\{v_1, \ldots, v_{i-1}\}$
  - ✓Normalize $v_i = w/|w|$
- **Output: $V=\{v_1, \ldots, v_k\}$**
  - ✓V is orthogonal $V^T V = I$
- **Input vector must not be zero.**
- **Deflation can happen:**
  - ✓Stop earlier.

- **Orthogonalization**
  - ✓Gram-Schmidt
- **Input: w and $\{v_1, \ldots, v_{i-1}\}$**
- **Do ($j = 1, i-1$):**
  - ✓$w = w - (w^T v_i)v_i$
- **Output: new w**

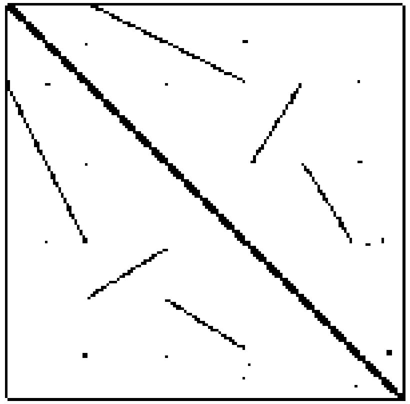
- **Could be generalized to multiple inputs (see Freund).**

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

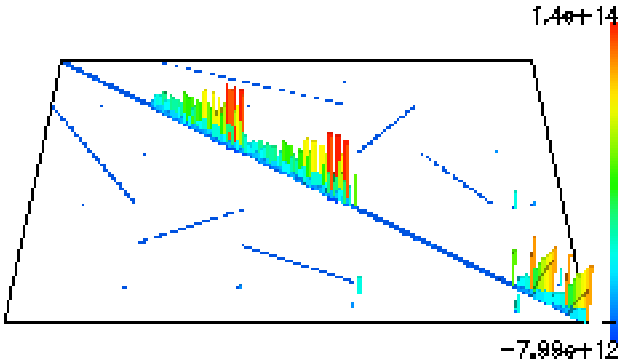
Sparse Matrices

- In the finite element method matrices are sparse.
- We have to store only nonzeros entries.
- nnz - number of nonzeros.
- Matrix Market:
  - ✓http://math.nist.gov/MatrixMarket
  - ✓File format,
  - ✓Many matrices.
- Matrix BCSSTK19

Structure Plot



City Plot



Treating Matrix Inverse

- The Arnoldi process requires matrix vector product.
- Yet, we have a matrix inverse.
- Instead solve a system of linear equations.
- This is the biggest computational cost:
  - ✓Number of vectors x time for linear solve.

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

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

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

- The only right hand side is different.
- Can be used to speed it up.

Direct Solvers

- **Gauss elimination.**
- **Positive definite matrices:**
  - ✓Cholesky decomposition.
- **General symmetric matrix:**
  - ✓$L^T DL$ decomposition.
- **Unsymmetric matrix:**
  - ✓LU decomposition.

$$Ax = b$$

$$A = L^T L$$

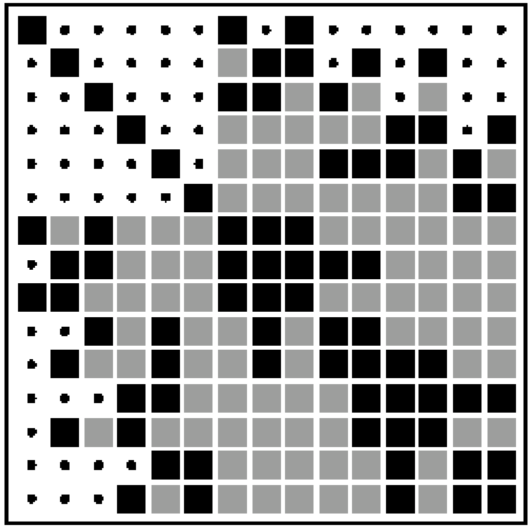
$$A = L^T DL$$

$$A = LU$$

$$LUx = b$$

- **1) Factorization (expansive).**
- **2) Back substitution (cheap).**
- **Well suited for model reduction.**

Reordering



- Symmetric matrix
  - ✓79171x79171
  - ✓nnz 2215638 (in its half) ~ $2.2\ 10^6$

method	time to reorder	nnz in factor	time to factor
genmmd	1.9	$130\ 10^6$	1166
md	4.0	$160\ 10^6$	1684
mmd	4.0	$127\ 10^6$	1135
amd	1.4	$127\ 10^6$	1135
metis	17	$47\ 10^6$	239

- Factorization creates fill-in.
- Factor size depends on the matrix structure.
- Reordering reduces fill-in in the factor.

Iterative Solvers

- Direct solvers have an upper limit because of memory requirements.
- Use iterative solvers:
  - ✓Could be faster.
  - ✓The only possibility for high dimensions.
- However, the success highly depends on a preconditioner.
- Model reduction still could be advantageous.

- 4 Gb of RAM
- Structural mechanics problem:
  - ✓375 801 DoFs
  - ✓15 039 875 nnz
- Sparse solver: 490 s
- PCG solver with 1e-8 tolerance: 290 s
- PCG solver with 1e-12 tolerance: 420 s

Nonzero 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.

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

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

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

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

When to Use Nonzero Expansion Point

- **When the stiffness matrix is degenerated.**
  - ✓Check Eigenvalues[MatrixK[sys]] for a reduced model.
- **When the convergence is slow for the frequency required.**
- **Rule of thumb: use a medium point for your frequency range.**

MOR for ANSYS

ANSYS Model: EMAT, FULL files

$$M\ddot{x} + E\dot{x} + Kx = Bu$$

$$y = Cx$$

Linear Dynamic System, ODEs

MOR Algorithm

Low-dimensional subspace

$$x = Vz + \varepsilon$$

Matrix Market format

Solvers: TAUCS, UMFPACK, ATLAS

www.imtek.uni-freiburg.de/simulation/mor4ansys/

code in C++, binary, publications

MOR Timing with TAUCS

Dimension	nnz	Time is ANSYS 8.1	Factoring	30 vectors
4 267	20 861	0.63	0.31	0.59
11 445	93 781	2.2	1.3	2.7
20 360	265 113	15	12	14
79 171	2 215 638	230	190	120
152 943	5 887 290	95	91	120
180 597	7 004 750	150	120	160
375 801	15 039 875	490	410	420

MOR as Fast Solver

- **Model reduction time is about twice as time for a static solution.**
  - ✓Direct solver can be used.
  - ✓Dimension of the reduced model is about 30.
  - ✓Time for an iterative solver is comparable with direct solver.
  - ✓Single expansion point.
- **Simulation of the reduced model is a few seconds.**
- **It is advantageous to use MOR even you use the reduced model only once:**
  - ✓Design,
  - ✓Geometry optimization.

MOR for ANSYS

- Command line tool, mor_for_ansys.

- Read files, computes and then write files.
- Current version is 1.83.

- There could be problems with reading matrices.
- Another tool, dumpmatrices allows us to overcome problems with reading ANSYS files.

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Notation

- **Ordinary Differential Equation**
  - ✓ First Order

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

  - ✓ Second Order

$$M\ddot{x} + E\dot{x} + Kx = Bu$$
$$y = Cx$$

- **MOR 4 ANSYS can read matrices from ANSYS.**
- **MOR 4 ANSYS can read matrices in the Matrix Market format as well.**

FULL and EMAT files

- **EMAT**
  - ✓File with element matrices.
- **FULL**
  - ✓File with global matrices.
  - ✓Load vector, Dirichlet and equation constraints.
  - ✓Must be for symbolic assembly (sparse solver).
  - ✓Problem to have all matrices.
- **mor_for_ansys uses both files.**
  - ✓When different coordinate systems have been used during modeling, EMAT file does not give us correct result.
  - ✓Matrices and load vector must be real valued.

Running mor_for_ansys

- **Several outputs**

  mor_for_ansys file.full file.emat –N 30 -C output.txt -s UMFPACK

- **Complete output**

  mor_for_ansys file.full file.emat –N 30 -f -s UMFPACK

- **Reading matrices in the Matrix Market format**

  mor_for_ansys -M base_name –N 30 -s UMFPACK

- **Solvers**

  ✓UMFPACK for unsymmetric matrices

  ✓TAUCS for symmetric matrices

Running dumpmatrices

- **Reads either FULL or EMAT file.**
- **Can write original matrices before the application of constraints.**
- **Several outputs**

  dumpmatrices file.full -C output.txt -w base_name
- **Complete output**

  dumpmatrices file.full -w base_name

MOR for ANSYS and dumpmatrices

Can FULL and EMAT files from a static analysis be used?

Yes

Use MOR for ANSYS directly.

No

Use dumpmatrices to prepare system matrices

When to Use dumpmatrices

- One of the next conditions applies:

- During modeling different coordinate systems have been used.
- Load vector is complex-valued.
- During static analysis ANSYS removes some degrees of freedom.

Conclusion

- Methods based on Hankel singular values (SLICOT)
- Implicit moment matching
- Solving a system of linear equations
- MOR for ANSYS