Automatic Model Reduction for Transient Simulation of MEMS-based Devices

Evgenii B. Rudnyi and Jan G. Korvink
IMTEK–Institute for Microsystem Technology
Albert Ludwig University Freiburg

IMTEK
ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Review

- E. B. Rudnyi, J. G. Korvink, *Automatic Model Reduction for Transient Simulation of MEMS-based Devices*, Sensors Update, 2002, 11, 3-33.

Tutorial

- J. G. Korvink, IEEE Sensors Short Course on Compact Modelling, 2002.

Preprints

- www.imtek.de/simulation/

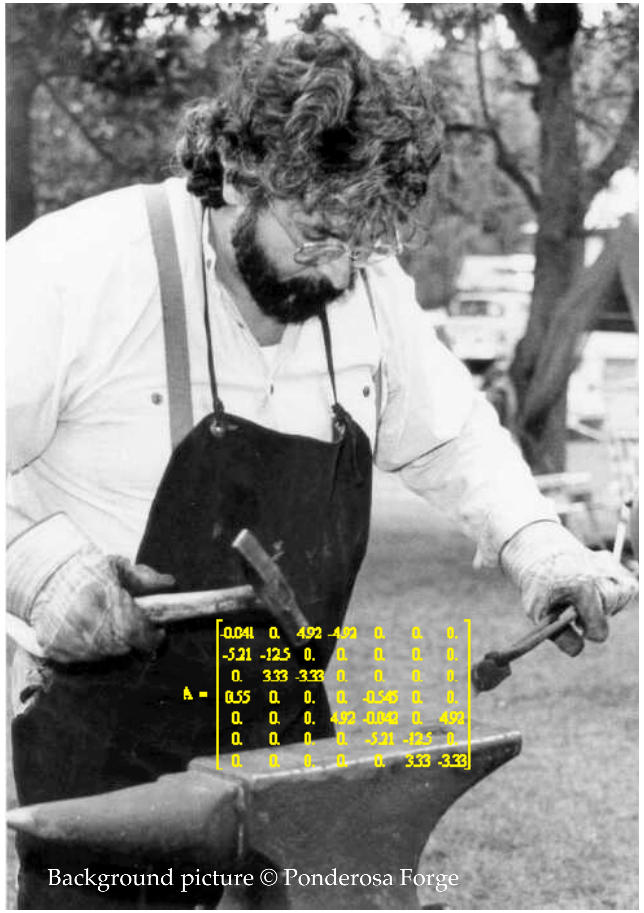
Contact

- rudnyi@imtek.de

Contents

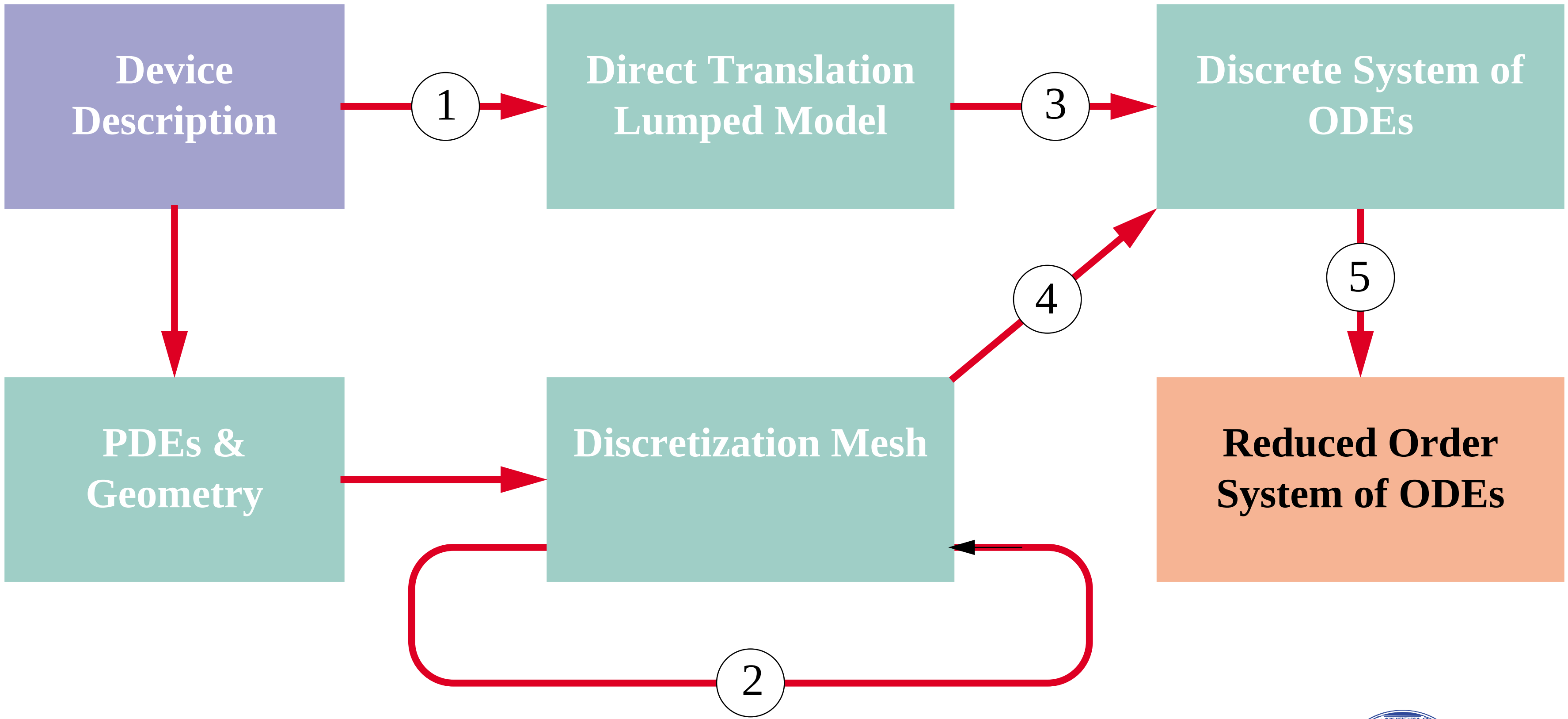
- Introduction to the idea
- Statement of the problem
- Small linear systems
- Introduction to Krylov subspaces
- Large linear systems
- Nonlinear systems

Forging a smaller System



Background picture © Ponderosa Forge

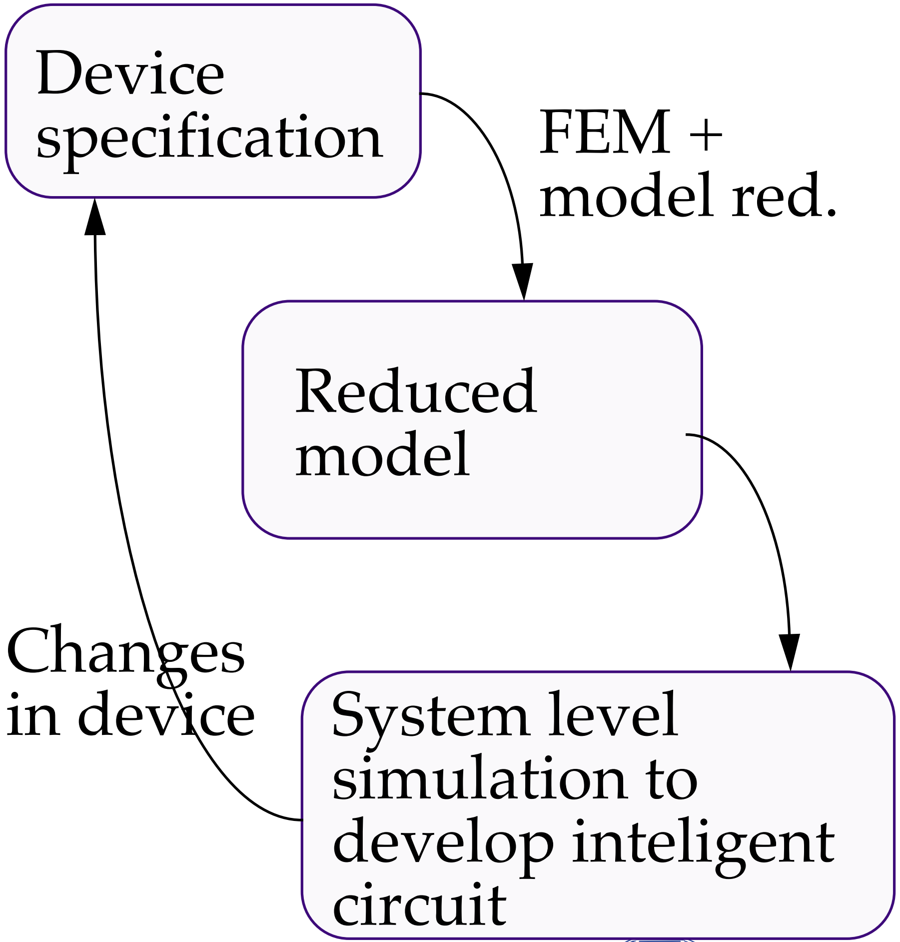
Generation Paths



Why is it useful ?

- Compact model for system level simulation:
  - Reduced model fits naturally in system level simulators.
  - The generatation of the reduced model can be almost automatic.

- Block diagram:



Summary

- Many computational nodes in a typical FEM (...) model appear to be "**redundant**".
- It appears that much of the behaviour of a system takes place in a **low dimensional subspace**.
- MOR can **greatly improve** the use of simulation tools during engineering design.

What's Next?

- Introduction to the idea
- **Statement of the problem**
- Small linear systems
- Introduction to Krylov subspaces
- Large linear systems
- Nonlinear systems

Statement of Problem

Delivered ODEs

- Discussion Limited to 1st Order
- Implicit Form (MNA, T-psi):

$$E \cdot \frac{d\boldsymbol{x}}{dt} = F \cdot \boldsymbol{x} + \boldsymbol{f}$$

$$E, F \in \Re^n \times \Re^n \qquad \boldsymbol{f}, \boldsymbol{x}(t) \in \Re^n$$

- Second Order (mechanics):

$$M \cdot \frac{d^2\boldsymbol{y}}{dt^2} + D \cdot \frac{d\boldsymbol{y}}{dt} + K \cdot \boldsymbol{y} = \boldsymbol{f}$$

$$\boldsymbol{z} = \frac{d\boldsymbol{y}}{dt} \quad : \quad \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \cdot \frac{d}{dt}\begin{bmatrix} z \\ y \end{bmatrix} = -\begin{bmatrix} D & K \\ -I & 0 \end{bmatrix} \cdot \begin{bmatrix} z \\ y \end{bmatrix} + \begin{bmatrix} f \\ 0 \end{bmatrix}$$

- Explicit Form:

$$\frac{d\boldsymbol{x}}{dt} = A \cdot \boldsymbol{x} + \boldsymbol{b}$$

$$A = E^{-1} \cdot F \qquad A \in \Re^n \times \Re^n$$

$$\boldsymbol{b} = E^{-1} \cdot \boldsymbol{f} \qquad \boldsymbol{b} \in \Re^n$$

- **Goal**: Find X such that

$$\boldsymbol{x} = X \cdot \boldsymbol{z} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{z} \in \Re^k$ for $k \ll n$

and $\boldsymbol{\varepsilon} \in \Re^n$ is “small”:

$$min\|\boldsymbol{\varepsilon}(t)\| = min\|\boldsymbol{x}(t) - X \cdot \boldsymbol{z}(t)\|$$

Statement of Problem

System Theory Version

- Often solution is not needed over entire domain, i.e., with:
  - Inputs $\boldsymbol{u} \in \Re^m$
  - Outputs $\boldsymbol{y} \in \Re^p$
  - Scatter Matrix $B \in \Re^n \times \Re^m$
  - Gather Matrix $C \in \Re^p \times \Re^n$
  - Multiple input–multiple output MIMO
  - Single input–single output SISO

- Large-scale dynamic system

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = A \cdot \boldsymbol{x} + B \cdot \boldsymbol{u} \\ \boldsymbol{y} = C \cdot \boldsymbol{x} \end{cases}$$

- Reduced system

$$\begin{cases} \dfrac{d\boldsymbol{z}}{dt} = \hat{A} \cdot \boldsymbol{z} + \hat{B} \cdot \boldsymbol{u} \\ \hat{\boldsymbol{y}} = \hat{C} \cdot \boldsymbol{z} \end{cases}$$

- Difference $min \|\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)\|$

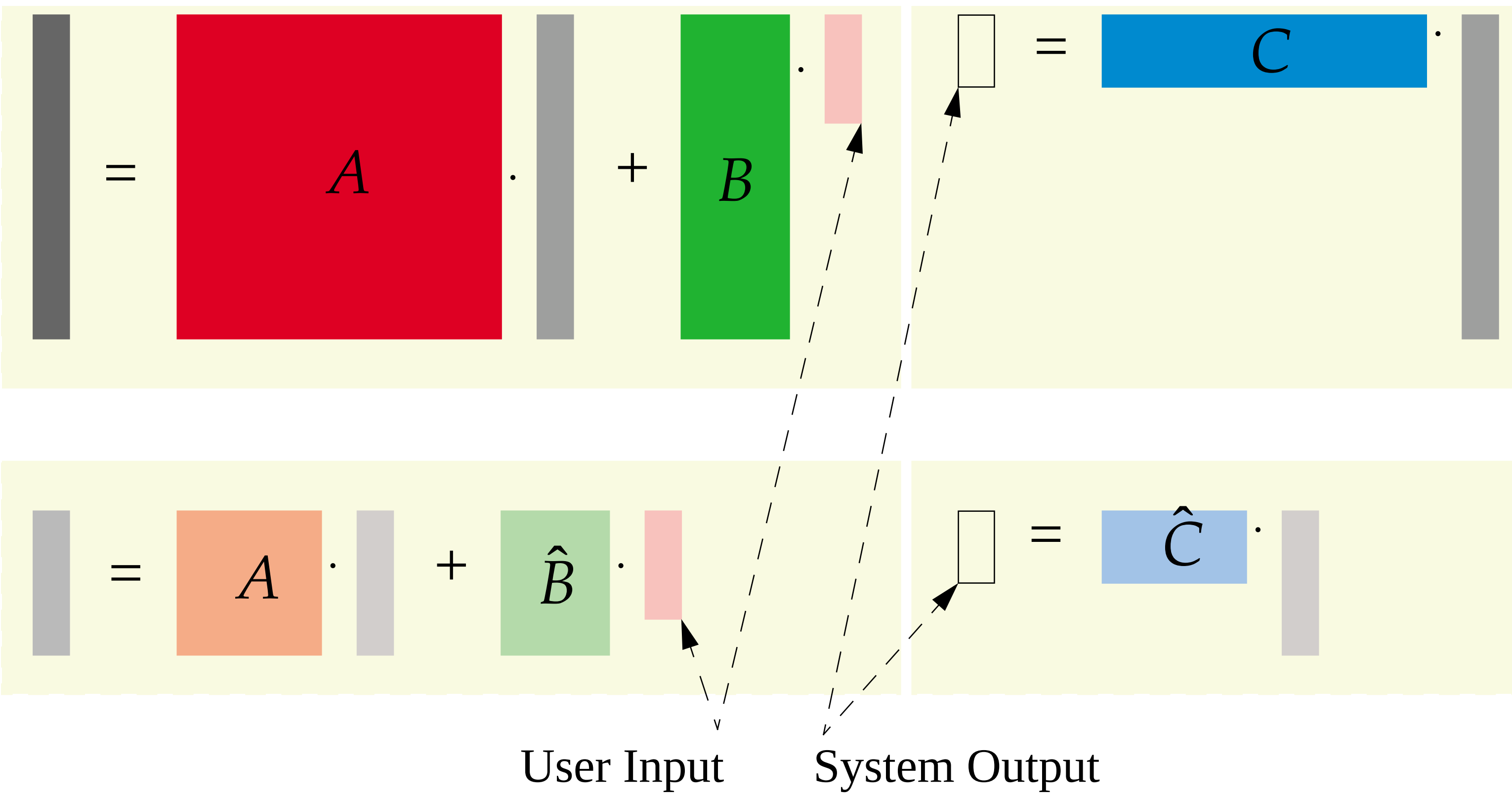
Pictorial Representation

Before:

$$\Sigma = \begin{bmatrix} A & B \\ C & \end{bmatrix}$$

After:

$$\hat{\Sigma} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \end{bmatrix}$$



Summary

- System theory provides a natural language to describe a problem of model order reduction.
- Most results comes from mathematicians working for the system theory.

What's Next?

- Introduction to the idea
- Statement of the problem
- **Small linear systems**
- Introduction to Krylov subspaces
- Large linear systems
- Nonlinear systems

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Hankel Singular Values (HSV)

- System: $\Sigma = \left[\begin{array}{c|c} A & B \\ \hline C & \end{array}\right]$

- Impulse response:

  $$h(t) = C \cdot e^{At} \cdot B$$

- Input-to-state: $\xi(t) = e^{At} \cdot B$

- State-to-output: $\eta(t) = C \cdot e^{At}$

- Grammians:

  $$P = \int_0^{\infty} (e^{At} \cdot B \cdot B^T \cdot e^{A^T t}) dt$$

  $$Q = \int_0^{\infty} (e^{A^T t} \cdot C^T \cdot C \cdot e^{At}) dt$$

- Lyapunov equations:

  $$A \cdot P + P \cdot A^T + B \cdot B^T = 0$$

  $$A^T \cdot Q + Q \cdot A + C^T \cdot C = 0$$

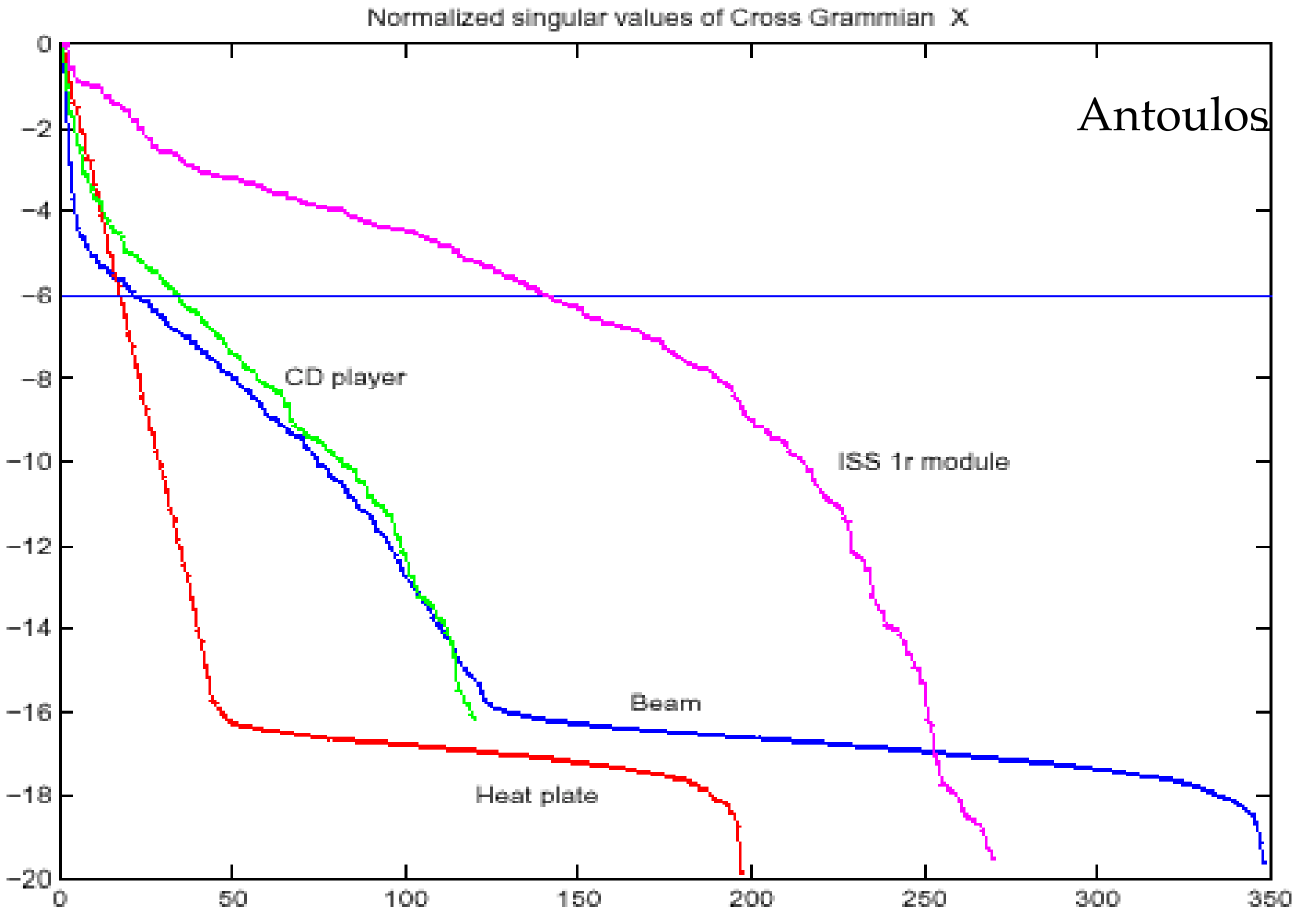
- HSV: $\sigma_i = \sqrt{\lambda_i(P \cdot Q)}$

Decay of Hankel Singular Values

Theory

- Error bound: If the system is reduced to one with **k largest** HSV then

$$\|G - \hat{G}\|_\infty < 2(\sigma_{k+1} + \ldots + \sigma_n)$$



ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Small Linear Systems

Transfer Function

- In Laplace Domain:

$$G_\Sigma(s) = C \cdot (sI - A)^{-1} \cdot B$$

Different methods

- Stable systems
  - In unstable systems something should be done with unstable poles.
- Hankel Norm Appr.
  - Produces an optimal solution
- Balanced Truncation Appr.
  - Most often used.
  - Faster than HNA.
  - Do not preserve the stationary state.
- Singular Perturbation Appr.
  - Preserve the stationary state.
- Frequency-weighted model reduction
  - $\|V(G - \hat{G})W\|_\infty$

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

SLICOT Library

- FORTRAN Code + Examples found at: `www.win.tue.nl/niconet`
  - 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.
- Matlab has licensed SLICOT.

- Yet, the computational complexity is $O(N^3)$.
  - Limited to “small” systems:

Order	Time Serial	Time Parallel (4 processor)
600	60	25
1332	703	130
2450	4346	666
3906		2668

Summary

- Hankel singular value theorem gives error bound.
- System theory has mature theory for MOR of linear systems. We can find optimal **low dimensional subspace**.
- **SLICOT** library implements theory and is "**easy to use**" for small linear systems.

What's Next?

- Introduction to the idea
- Statement of the problem
- Small linear systems
- **Introduction to Krylov subspaces**
- Large linear systems
- Nonlinear systems

Introduction to Krylov Subspaces

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.

Introduction to Krylov Subspaces

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.

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.

Small Pause

Summary

- Two algorithms can form numerically stable basis:
  - Both are amenable to **large, sparse systems** due to matrix-vector product.
- The Lanczos algorithm is faster.
  - Basises are biorthogonal.
- The Arnoldi algorithm is more numerically stable.
  - Basis is orthogonal.

What's Next?

- Introduction to the idea
- Statement of the problem
- Small linear systems
- Introduction to Krylov subspaces
- **Large linear systems**
- Nonlinear systems

Padé Approximants

- Can always express transfer function matrix elements using:

$$G_{ij}(s) = \frac{a(s-z_1)\dots(s-z_{n-1})}{(s-p_1)\dots(s-p_n)}$$

  - z_i, p_i are the zeroes, poles.
- Padé matches k moments about

$$s_0: \quad G_{ij}(s) = \sum_{p=0}^{k<n} m_i(s-s_0)^p$$

- Moment matching

$$m_i = \hat{m}_i \text{ for } i = 0, \dots, q$$

- Reduced func. is small **rational.**

$$G_{ij}(s) = \frac{a(s-z_1)\dots(s-z_{k-1})}{(s-p_1)\dots(s-p_k)}$$

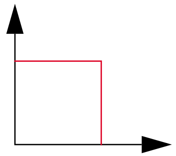
- Terminology:
  - Padé approximant: match $q = 2k$ moments.
  - Padé-type approximant: implicitly match less moments.
- Explicit matching is numerically unstable:
  - AWE - asymptotic wavefrom evaluation does not work.

Implicit Moment Matching

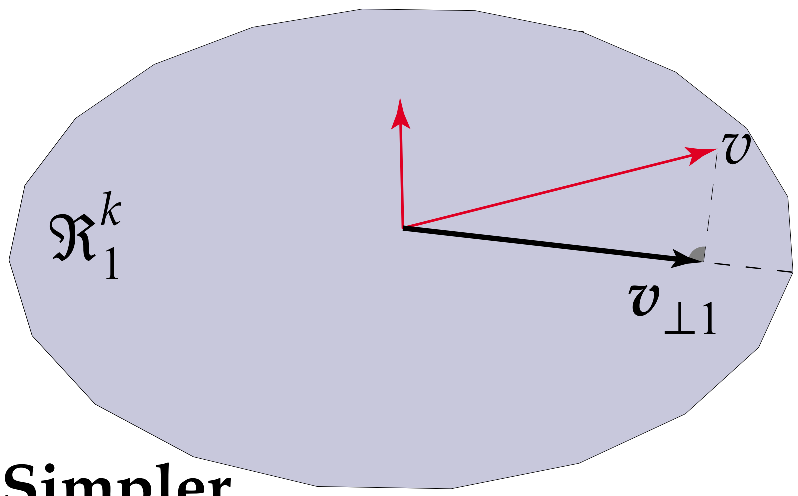
- Arnoldi: Right subspace $K_k^r(M, N)$, $M = (A - s_0 I)^{-1}$ and $N = M \cdot B$ produces H_A and X such that:
  - $\hat{A} = H_A^{-1} \cdot (I + s_0 H_A)$
  - $\hat{B} = H_A^{-1} \cdot X^T \cdot M$
  - $\hat{C} = C \cdot X$
- Lanczos: Also left subspace $K_k^l(M^T, L)$, $L = M^T \cdot C^T$ produces H_L, X and Y such that:
  - $\hat{A} = H_L^{-1} \cdot (I + s_0 H_L)$
  - $\hat{B} = H_L^{-1} \cdot Y^T \cdot M$
  - $\hat{C} = C \cdot X$
- Arnoldi implicitly matches k moments.
- Lanczos implicitly matches $2k$ moments

Projection Idea

Orthogonal Projection

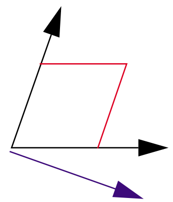


Arnoldi

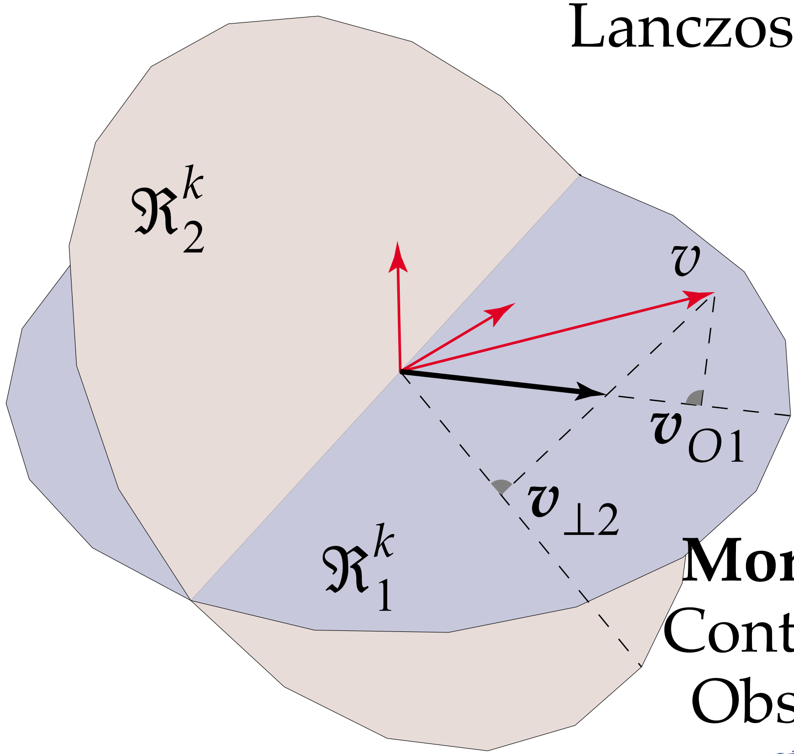


Simpler

Oblique Projection



Lanczos

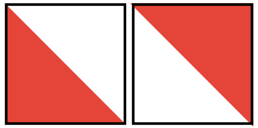


More natural:
Controllability
Observability

Computing Inverses

- Typical case: $F^{-1} \cdot \boldsymbol{w}$
  - Do not compute F^{-1}: Bad idea
  - Find $\boldsymbol{x}$ such that $F \cdot \boldsymbol{x} = \boldsymbol{w}$
- By LU Decomposition:

$$F = L \cdot U$$



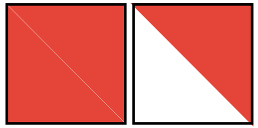
  - Two fast triangle solves

$$L \cdot (U \cdot \boldsymbol{x}) = \boldsymbol{w} \Leftrightarrow L \cdot \boldsymbol{y} = \boldsymbol{w}$$

$$U \cdot \boldsymbol{x} = \boldsymbol{y}$$

  - By QR Decomposition

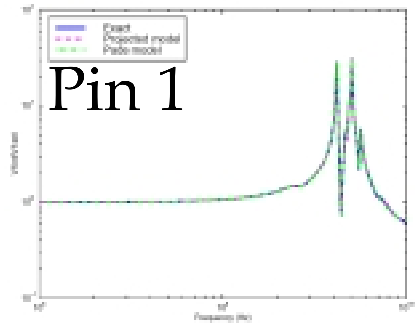
$$F = Q \cdot R$$

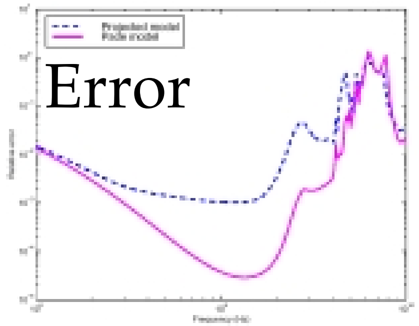


  - Orthogonality $Q^{-1} = Q^T$
  - One fast triangle solve
  - One fast matrix multiply
- Iterative Solvers:
  - Preconditioner from Appl.
  - Implement fast matrix multiply: Again, application can help here

Examples from EE

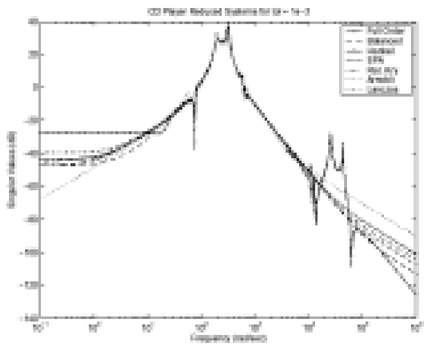
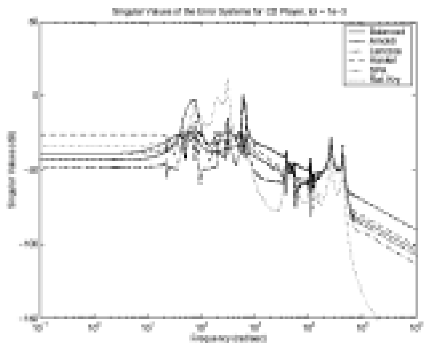
- 64 Pin RF IC: Padé via Lanczos:





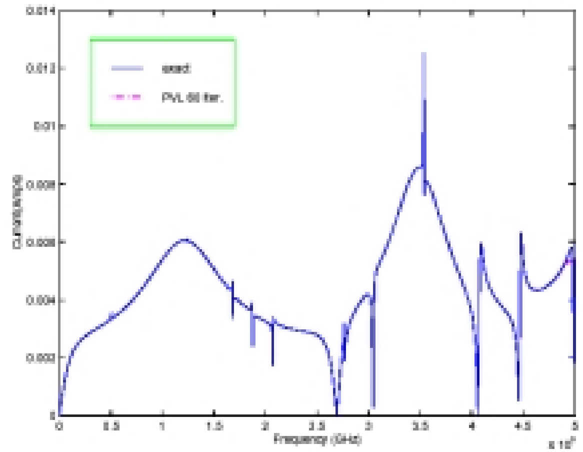
Source: Z. Bai, R. Freund, A Partial Padé-via-Lanczos Method for Reduced-Order Modelling

- CD Player: Comparison

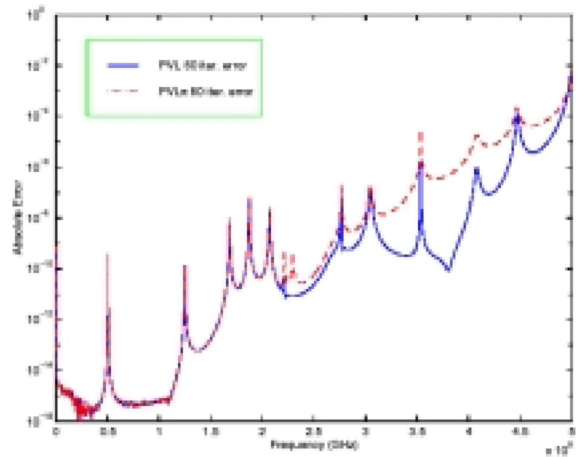



Source: Antoulas, Sorenson, Gugercin, A Survey of Model Reduction Methods for Large Systems

- PEEC EM Circuit: PVL

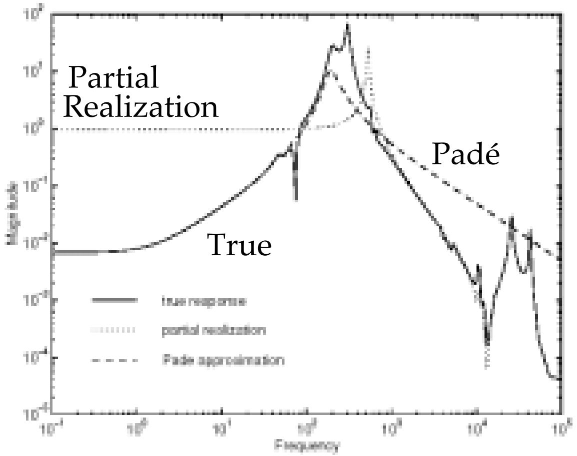


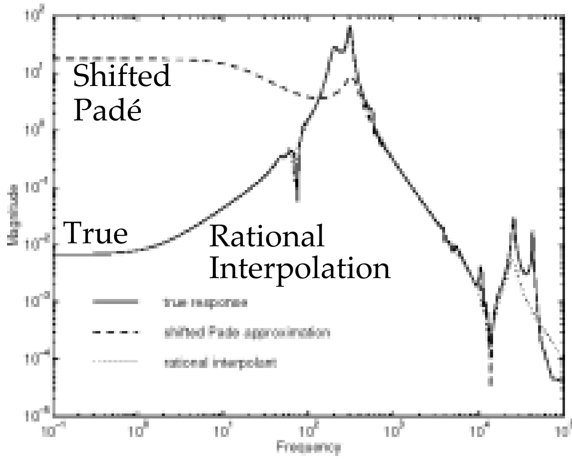
Source: Z. Bai, R. Freund, A Partial Padé-via-Lanczos Method for Reduced-Order Modelling



Rational Krylov

- Expansion usually accurate about s_0.
- This suggests:
  - Multiple expansion points s_i
  - Matching transfer function moments at all points
- Challenge: Where to place s_i
  Expensive solves (Many LU or QR decompositions)





Source: E. Grimme: Krylov Projection Methods for Model Reduction, PhD Thesis

Large Linear Systems

Solving Lyapunov Equations

- Padé approximants do not have global error estimates ... SVD-Krylov.

- Steps:
  - Solve Lyapunov equations for Grammians P and Q.
  - Eigen-decompose PQ.

- Very expensive: $\sim O(n^3)$

- General remedy: Low rank approximation of grammians.
  - Dense matrix $\sim O(n^2)$
  - Sparse matrix $\sim O(n)$
  - Also Krylov-based, also for balancing.

- See LYAPACK (Matlab based) `www.netlib.org/lyapack`

Summary

- Padé and Krylov are **related**.
- Arnoldi and Lanczos can generate implicitly **Padé approximants** (PVL).
- **Rational Krylov** improves using many expansions.
- Future holds promise for:
  - **Large Lyapunov** solvers.
  - Large **matrix exponential** approximants.

What's Next?

- Introduction to the idea
- Statement of the problem
- Small linear systems
- Introduction to Krylov subspaces
- Large linear systems
- **Nonlinear systems**

Special Cases

- Basic Problem:

  $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \qquad \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$

- Splitting linear and nonlinear parts: $\boldsymbol{f} = \boldsymbol{f}_L + \boldsymbol{f}_{NL}$

- Reduce linear part as usual

  $A_{Lij} = f_{L0} + \partial f_{Li} / \partial x_j$

- Treat nonlinear by Taylor expansion:

  $$\boldsymbol{f}_{NL} = \boldsymbol{f}_{NL0} + A' \cdot \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T \cdot A'' \cdot \boldsymbol{x} + \ldots$$

  $$A'_{NLij} = \partial f_{NLi} / \partial x_j$$

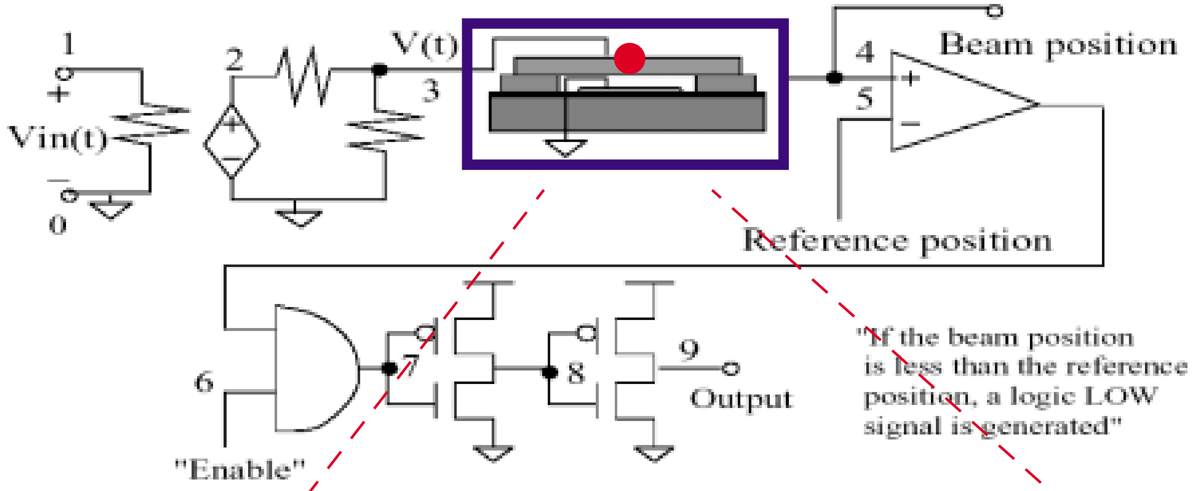
  $$A''_{NLijk} = \partial f_{NLi} / \partial x_j \partial x_k$$

- $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = A(\boldsymbol{x}) \cdot \boldsymbol{x} + C \cdot \boldsymbol{u}$

POD Idea

- Solve the full nonlinear system
  $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \qquad \boldsymbol{y} = g(\boldsymbol{x})$
- At appropriate times, take snapshots $\boldsymbol{s}_i$, and collect the snapshots in a matrix:
  $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_m\}$, m is many!
- Perform SVD of S:
  $S = U\Sigma V^T = \sum_{i=1}^{m} \sigma_i \boldsymbol{u}_i \otimes \boldsymbol{v}_i$
- Form the truncated snapshots by dropping the smallest singular values: $S_k = \hat{U}\Sigma_k \hat{V}^T$
- For reduced system, form:
  $\dot{\hat{\boldsymbol{x}}} = U^T \cdot f(\hat{U} \cdot \hat{\boldsymbol{x}}, \boldsymbol{u})$
  $\hat{\boldsymbol{y}} = g(\hat{U} \cdot \hat{\boldsymbol{x}})$
- Disadvantages:
  - Full nonlinear solve
  - How to compute $U^T A(x) U$?
  - Some intuition

POD Example: MEMS

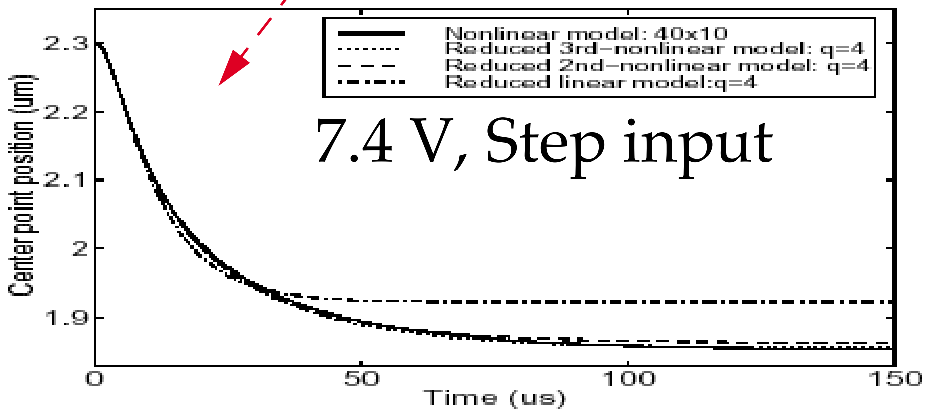


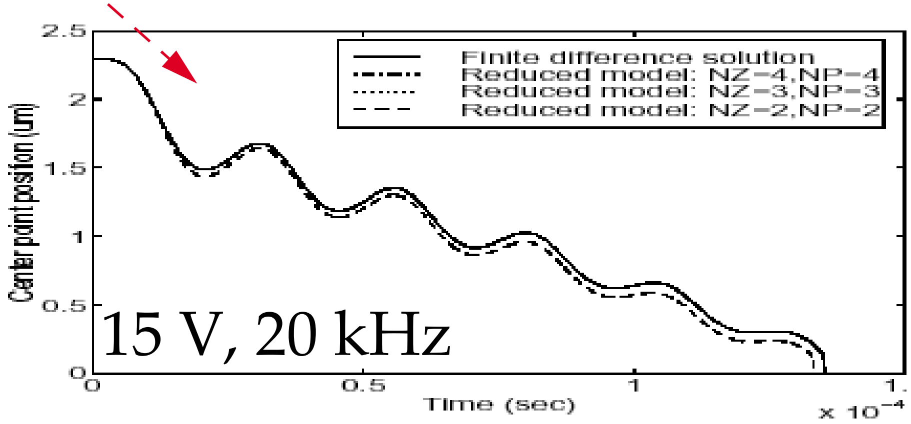
RF Switch:

FD for nonlinear beam including squeeze film & electrodynamic force

Karhunen-Loève + Galerkin

Source: J. Chen, S. Kang
Techniaues for Coupled Circuit and MEMS Simulation





Summary

- Application-oriented simplifications exist, but:
  - May need **symbolic manipulations**.
  - May need **expensive evaluations**.
- POD is general, and works, but:
  - Computationally **expensive**.
  - Requires **user interaction**.
- Nonlinear MOR is **tough**.

Conclusions

Small Linear

- Excellent state: complete knowledge.
- For accuracy goal, automatic guaranteed reduced model.

Large Linear

- Reasonably good choices.
- Arnoldi more stable.
- Lanczos matches more moments.
- When to stop reducing?
- Padé local, nonoptimal for wide range. Remedy: rational Krylov.

- Future may yield:
  - **Lyapunov** for large systems.
  - Approximation of **matrix exponential** for large systems.

Nonlinear

- Either special application-dependent techniques.
- Or Linearization or Splitting.
- Else POD, but
  - How many snapshots?