Introduction to Simulation - Lecture 25

Model-Order Reduction II

Jacob White

Thanks to Luca Daniel, Jing Li, Joel Phillips, Michal Rewienski,

SMA-HPC ©2003 MIT

MOR Outline

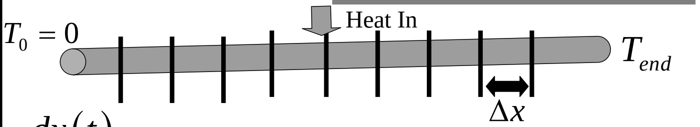
- Dynamic Linear Case
  - Rational Functions
  - Projection Framework
  - Krylov Methods
- Hankel Reduction and TBR
  - Mention a few issues

SMA-HPC ©2003 MIT

Demonstration Example

Heat Conducting Bar

State-Space Description



$$\frac{dx(t)}{dt} = \underbrace{A}_{NxN} x(t) + \underbrace{b}_{Nx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y(t)}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}{}^T x(t)$$

Given the right scaling

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ \vdots \\ h(x_N) \end{bmatrix} \quad c = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

SMA-HPC ©2003 MIT

Dynamic Linear case

State-Space Description

- Original Dynamical System - Single Input/Output

$$\frac{dx(t)}{dt} = \underbrace{A}_{NxN} x(t) + \underbrace{b}_{Nx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y(t)}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}{}^T x(t)$$

- Reduced Dynamical System

$$\frac{dx_r(t)}{dt} = \underbrace{A_r}_{qxq} x(t) + \underbrace{b_r}_{qx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y_r(t)}_{\substack{scalar\\output}} = \underbrace{c_r}_{qx1}{}^T x_r(t)$$

- q << N, but input/output behavior preserved

SMA-HPC ©2003 MIT

An Aside on Transfer Functions – Laplace Transform

Consider an ODE: $\frac{dx(t)}{dt} = Ax(t) + bu(t)$

Bilateral Laplace Transform: $X(s) = \int_{-\infty}^{\infty} x(t) e^{-st} dt$

Key Transform Property: $sX(s) = \int_{-\infty}^{\infty} \frac{dx(t)}{dt} e^{-st} dt$

Rewrite the ODE in transformed variables

$$sX(s) = AX(s) + bU(s) \quad Y(s) = c^T X(s)$$

$$\Rightarrow Y(s) = \underbrace{c^T (sI - A)^{-1} b}_{H(s)} U(s)$$

$H(s)$ ← Transfer Function

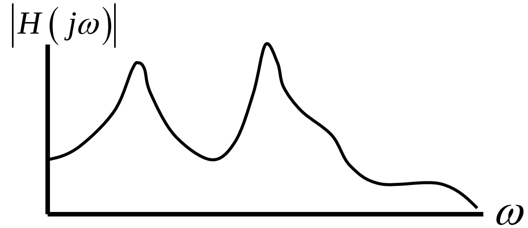
SMA-HPC ©2003 MIT

An Aside on Transfer Functions – Meaning of H(s)

For Stable Systems, H(j*w*) is the frequency response

If $u(t) = e^{j\omega t}$ ←Sinusoid

then $y(t) = H(j\omega) e^{j\omega t}$ Sinusoid with shifted phase and amplitude



SMA-HPC ©2003 MIT

An Aside on Transfer Functions – EigenAnalysis

Transfer Function

$$H(s) = c^T (sI - A)^{-1} b$$

Apply Eigendecomposition

$$H(s) = c^T E (sI - \lambda)^{-1} E^{-1} b$$

$$= \tilde{c}^T \begin{bmatrix} \frac{1}{s-\lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{s-\lambda_N} \end{bmatrix} \tilde{b} \Rightarrow H(s) = \sum_{i=1}^{N} \frac{\tilde{c}_i \tilde{b}_i}{s - \lambda_i}$$

SMA-HPC ©2003 MIT

Dynamic Linear Case — Rational Transfer Function Representation

Original System Transfer Function

$$H(s) = \frac{\tilde{c}_1 \tilde{b}_1}{(s-\lambda_1)} + \cdots + \frac{\tilde{c}_N \tilde{b}_N}{(s-\lambda_N)} = \underbrace{\frac{b_0 + b_1 s + \cdots + b_{N-1} s^{N-1}}{1 + a_1 s + \cdots + a_N s^N}}_{\text{Rational Function}}$$

Reduced Model Transfer Function

$$H_r(s) = \underbrace{\frac{b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}}{1 + a_1^r s + \cdots + a_q^r s^q}}_{\text{Lower Order Rational Function}}$$

Model Reduction = Find a low order rational function matching H(s)

SMA-HPC ©2003 MIT

Dynamic Linear Case — Rational Transfer Function Representation

Degrees of Freedom

Reduced Model Dynamical System

$$\frac{dx_r(t)}{dt} = \underbrace{A_r}_{qxq} x(t) + \underbrace{b_r}_{qx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y_r(t)}_{\substack{scalar\\output}} = \underbrace{c_r}_{qx1}{}^T x_r(t)$$

$2q + q^2$ coefficients

Reduced Model Transfer Function

$$H_r(s) = \frac{b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}}{1 + a_1^r s + \cdots + a_q^r s^q}$$

$2q$ coefficients

SMA-HPC ©2003 MIT

Dynamic Linear Case — Rational Transfer Function Representation

Variable Changes Do not change transfer functions

Reduced Model Transfer Function

$$\frac{dx_r(t)}{dt} = A_r x(t) + b_r u(t) \qquad y_r(t) = c_r^T x_r(t)$$

$$\Rightarrow H(s) = c_r^T (sI - A_r)^{-1} b_r$$

Similarity (x = Sw) Transformed Transfer Function

$$\frac{dw_r(t)}{dt} = S^{-1} A_r S w(t) + S^{-1} b_r u(t) \qquad y_r(t) = c_r^T S w_r(t)$$

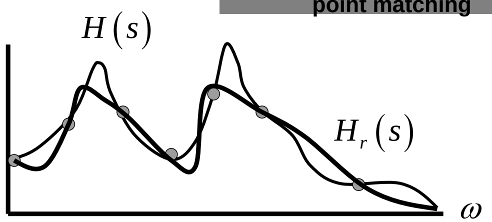
$$\Rightarrow H(s) = c_r^T S \left(sI - S^{-1} A_r S\right)^{-1} S^{-1} b_r = c_r^T (sI - A_r)^{-1} b_r$$

Many Dynamical Systems have the same transfer function!!

SMA-HPC ©2003 MIT

Dynamic Linear Case — Rational Transfer Function Representation

Rational Function Fitting by point matching



- Can match 2q points
- cross multiplying generates a linear system

  For i = 1 to 2q

  $$\left(1 + a_1^r s_i + \cdots + a_q^r s_i^{\,q}\right) H(s_i) - \left(b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}\right) = 0$$

SMA-HPC ©2003 MIT

Dynamic Linear Case — Rational Transfer Function Representation

Point Matching Matrix can be ill-conditioned

$$\begin{bmatrix} s_1 H(s_1) & s_1^2 H(s_1) & \cdots & -s_1^{q-1} \\ \vdots & \vdots & \cdots & -s_2^{q-1} \\ \vdots & \vdots & \cdots & \vdots \\ s_{2q} H(s_{2q}) & s_{2q}^2 H(s_{2q}) & \cdots & -s_{2q}^{q-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ b_{q-1} \end{bmatrix} = \begin{bmatrix} H(s_1) \\ H(s_2) \\ \vdots \\ H(s_{2q}) \end{bmatrix}$$

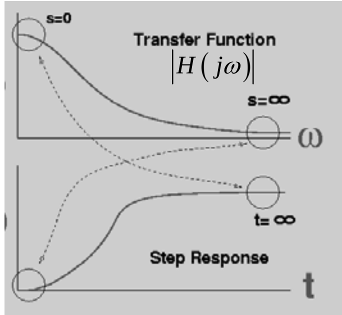
- Columns contain progressively higher powers of the test frequencies
- Must orthogonalize columns during construction

SMA-HPC ©2003 MIT

Dynamic Linear Case

Rational Transfer Function Representation

Importance of Fitting at low frequency



Correct Steady State behavior requires accurate match at low frequencies

SMA-HPC ©2003 MIT

Dynamic Linear Case

Rational Transfer Function Representation

Taylor Series Expansion and Moments

Original System Transfer Function Moments

$$H(s) = c^T(sI - A)^{-1}b = -c^T \underbrace{(I - sA^{-1})^{-1}}_{\text{Taylor Expand with respect to s}} A^{-1}b$$

$$H(s) = -c^T(I - sA^{-1})^{-1}A^{-1}b = \sum_{k=0}^{\infty} c^T A^{-(k+1)} b s^k$$

$$H(s) = \underbrace{c^T A^{-1} b}_{m_0} + \underbrace{c^T A^{-2} b}_{m_1} s + \underbrace{c^T A^{-3} b}_{m_2} s^2 + \cdots = \sum_{k=0}^{\infty} m_k s^k$$

Moments

SMA-HPC ©2003 MIT

Dynamic Linear Case

Rational Transfer Function Representation

Moment Matching for accurate low frequency behavior

Reduced Model Matches Original Systems Moments

$$H_r(s) = \frac{b_0^r + b_1^r s + \cdots + b_{q-1}^r s^{q-1}}{1 + a_1^r s + \cdots + a_q^r s^q} = m_0 + m_1 s + \cdots + m_{2q-1} s + \cdots$$

Cross-Multiplying and Matching Terms

$$\begin{bmatrix} m_0 & m_1 & \cdots & m_{k-1} \\ m_1 & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & m_{2q-3} \\ m_{k-1} & \cdots & m_{2q-3} & m_{2q-2} \end{bmatrix} \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}$$

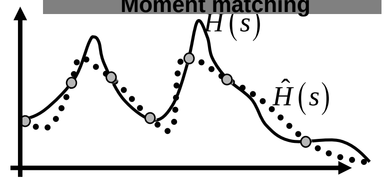
SMA-HPC ©2003 MIT

Dynamic Linear Case

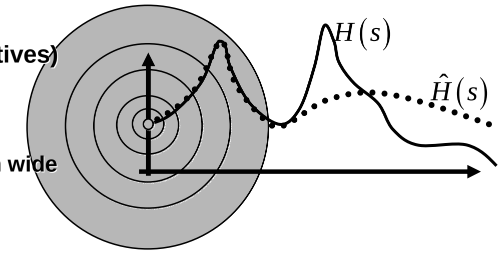
Rational Transfer Function Representation

Point Matching Versus Moment matching

Point matching:
can be very inaccurate in between points



Moment (derivatives) matching:
accurate around expansion point, but inaccurate on wide frequency band

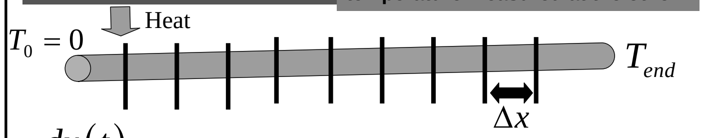


SMA-HPC ©2003 MIT

Dynamic Linear Case

Heat Conducting Bar

Heat applied at one end, temperature measured at the other



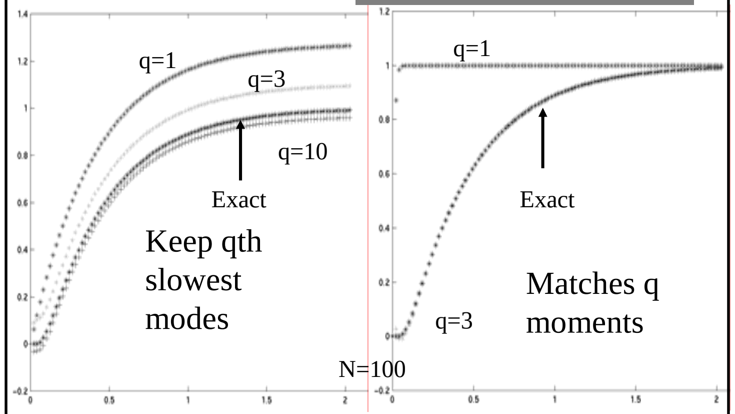
$$\frac{dx(t)}{dt} = \underbrace{A}_{NxN} x(t) + \underbrace{b}_{Nx1} \underbrace{u(t)}_{\substack{scalar\\input}} \qquad \underbrace{y(t)}_{\substack{scalar\\output}} = \underbrace{c}_{Nx1}^{T} x(t)$$

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \qquad c = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Dynamic Linear Case

Keeping Eigenmodes versus matching moments

Heat Flow Results



Dynamic Linear Case

Rational Transfer Function Representation

Explicit Moment Matching Problem

System of equations extremely ill-conditioned

$$\begin{bmatrix} m_0 & m_1 & \cdots & m_{k-1} \\ m_1 & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & m_{2q-3} \\ m_{k-1} & \cdots & m_{2q-3} & m_{2q-2} \end{bmatrix} \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}$$

$$m_i = c^T A^{-i} b \approx \lambda_{A_{\max}} m_{i-1}$$

Columns become linearly dependent for large q!

Dynamic Linear Case

Rational Transfer Function Representation

Problems with explicit fitting methods

- Linear Systems for fitting ill-conditioned
  - Need specialized algorithms which avoid explicit fitting matrix construction
- Rational function must be converted to state-space
  - Needed by most simulation tools
  - Requires root finding procedure, very sensitive to parameter variation

Dynamic Linear Case — Projection Framework

Dimension Reducing Change of Variables

$$\begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x_N \end{bmatrix} \approx \underbrace{\begin{bmatrix} \uparrow & \cdots & \uparrow \\ \uparrow & \cdots & \uparrow \\ \uparrow & \cdots & \uparrow \\ \vec{u}_1 & \cdots & \vec{u}_q \\ \downarrow & \cdots & \downarrow \\ \downarrow & \cdots & \downarrow \\ \downarrow & \cdots & \downarrow \end{bmatrix}}_{U_q} \begin{bmatrix} x_{r_1} \\ \vdots \\ x_{r_q} \end{bmatrix}$$

Dynamic Linear Case — Projection Framework

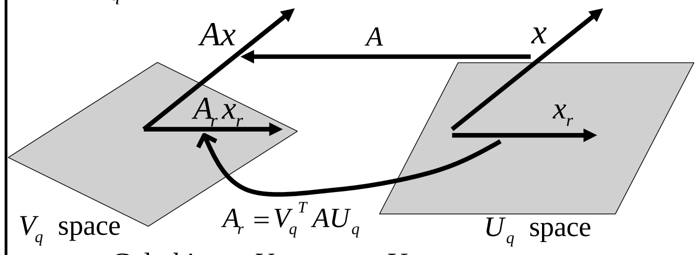
$$\dot{x} = Ax + bu,\ \ y = c^T x \ \Rightarrow \dot{x}_r = A_r x_r + b_r u,\ \ y_r = c_r^{\,T} x$$

Equation Testing

$$V_q^{\,T} A x \approx A_r x_r$$

Change of variables

$$x \approx U_q x_r$$



Galerkin $\rightarrow V_q$ space = U_q space

Dynamic Linear Case — Projection Framework

Assumed Biorthogonal Relationship between V and U

- Original System

$$\dot{x} = Ax + bu,\ \ y = c^T x$$

- Substitute $x = U_q x_r$

$$U_q \dot{x}_r = A U_q x_r + b_r u,\ \ y_r = c^T U_q x_r$$

- Test by multiplying by V

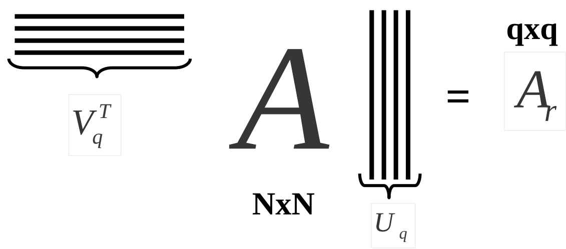
$$V_q^T U_q \dot{x}_r = V_q^T A U_q x_r + V_q^T b_r u,\ \ y_r = c^T U_q x_r$$

- Previous Slide Assumed that V and U biorthogonal

$$V_q^T U_q = I \Rightarrow \dot{x}_r = A_r x_r + b_r u,\ \ y_r = c_r^{\,T} x_r$$

Dynamic Linear Case — Projection Framework

Forming the reduced system matrix



- No explicit A need, Only Matrix-vector products

  For each column of U_q

  Multiply by A, then dot result with columns of V_q

Projection Framework
Dynamic Linear Case
V=U can preserve definiteness properties

- Original System

$$\frac{dx}{dt} = Ax + bu, \quad y = c^T x$$

- Reduced System

$$\frac{dx_r}{dt} = \underbrace{U_q^T A U_q}_{A_r} x_r + U_q^T b_r u, \quad y_r = c^T U_q x_r$$

- If A is (+ or -) definite, so is A_r
  - Preserves stability in the definite case
  - Can also preserve passivity

SMA-HPC ©2003 MIT

Projection Framework
Dynamic Linear Case
Approaches for Picking U and V

- Use Eigenvectors
- Use Time Series Data
  - Compute $x(t_0), x(t_1), \cdots, x(t_k)$
  - Use the SVD to pick $q < k$ important vectors
- Use Frequency Domain Data
  - Compute $X(s_1), X(s_2), \cdots, X(s_k)$
  - Use the SVD to pick $q < k$ important vectors
- Use Krylov Subspace Vectors?
- Use Singular Vectors of System Grammians?

SMA-HPC ©2003 MIT

Aside on Krylov Subspaces- Definition

The order k Krylov subspace generated from matrix A and vector b is defined as

$$\kappa_k(A,b) \equiv span\left\{b, Ab, A^2 b, \ldots A^{k-1} b\right\}$$

SMA-HPC ©2003 MIT

Projection Framework
Dynamic Linear Case
Moment Matching Theorem

If

$$span\left\{\vec{u}_1, \ldots, \vec{u}_q\right\} \supseteq \bigcup_{j=1}^{J} \kappa_{k_j^b}\left(\left(A - s_j I\right)^{-1}, \left(A - s_j I\right)^{-1} b\right)$$

And

$$span\left\{\vec{v}_1, \ldots, \vec{v}_q\right\} \supseteq \bigcup_{j=1}^{J} \kappa_{k_j^c}\left(\left(A - s_j I\right)^{-T}, \left(A - s_j I\right)^{-T} c\right)$$

Then

$$\frac{\partial^l H(s_j)}{\partial s^l} = \frac{\partial^l H_r(s_j)}{\partial s^l} \quad \text{for } l = 0, \ldots, k_j^b + k_j^c - 1$$

SMA-HPC ©2003 MIT

Dynamic Linear Case — Projection Framework

Special Case Moment Matching Theorem

If U and V are such that

$$U = V = \{\vec{u}_1, \ldots, \vec{u}_q\} \quad \text{and } U^T U = I$$

$$span\{\vec{u}_1, \ldots, \vec{u}_q\} = span\{A^{-1}b, A^{-2}b, \ldots, A^{-q}b\}$$

Then the first q moments of reduced system match

$$H(s) = -c^T \left(I - sA^{-1}\right)^{-1} A^{-1}b = \sum_{k=0}^{\infty} c^T A^{-(k+1)} b s^k$$

$$H_r(s) = -c_r^T \left(I - sA_r^{-1}\right)^{-1} A_r^{-1} b_r = \sum_{k=0}^{\infty} c_r^T A_r^{-(k+1)} b_r s^k$$

$$c^T A^{-(k+1)} b = c^T U_q \left(U_q^T A U_q\right)^{-(k+1)} U_q^T b \qquad k = \{0, \ldots, q-1\}$$

Dynamic Linear Case — A Projection Alternative

First Invert A before applying reduction

$$A^{-1}\dot{x} = x + A^{-1}bu, \;\; y = c^T x \;\Rightarrow H(s) = \underbrace{-c^T \left(I - sA^{-1}\right) A^{-1} b}_{\text{Unchanged Tranfer function}}$$

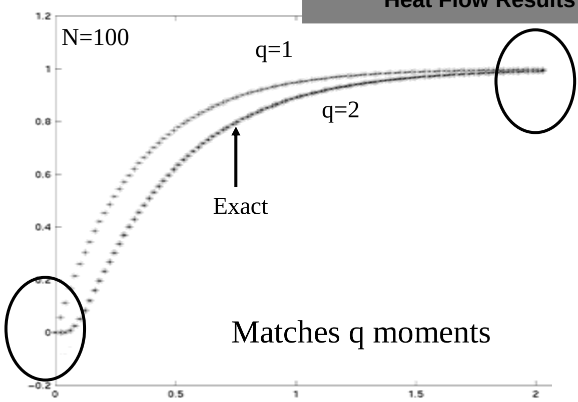
Form reduced model by projecting inverse of A

$$\underbrace{V_q^T A^{-1} U_q}_{A_r^{-1}} \dot{x}_r = x_r + V_q^T A^{-1} bu, \;\; y_r = c_r^T x_r$$

The Projection Theorem Still Holds!!

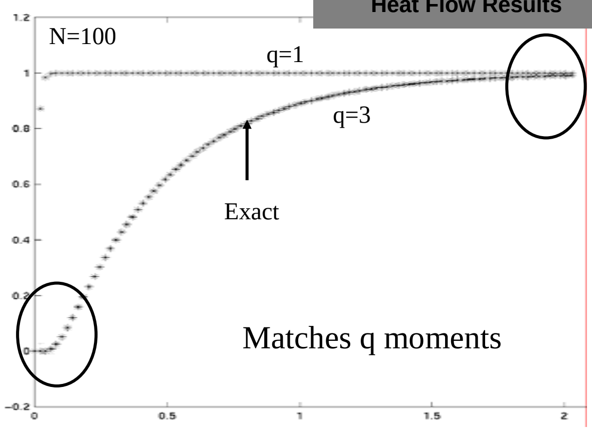
Model-Order Reduction — Projection Alternative

Heat Flow Results



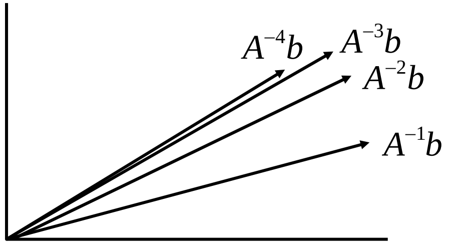
Model-Order Reduction — Noninverse Formulation

Heat Flow Results



Dynamic Linear Case — Computing U — Need for Orthogonalization

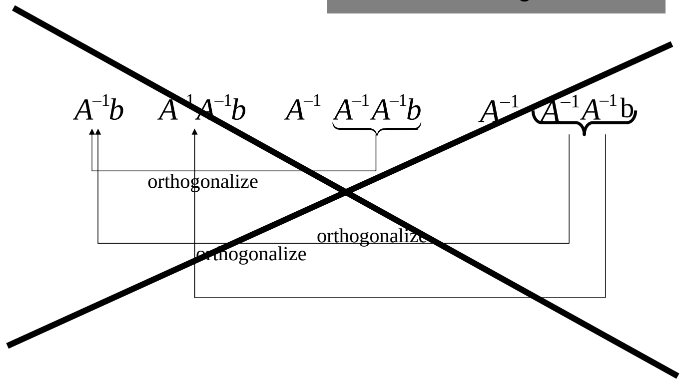
$\left\{A^{-1}b, A^{-2}b, \ldots, A^{-k}b\right\}$ can not be computed directly



Vectors will line up with dominant eigenspace!

SMA-HPC ©2003 MIT

Dynamic Linear Case — Computing U — Need for Orthogonalization



• Only requires solves with A and vector inner products

SMA-HPC ©2003 MIT

Dynamic Linear Case — Computing orthogonal U — Arnoldi Algorithm

$\vec{u}_1 = A^{-1}b / \left\| A^{-1}b \right\|$

For i = 1 to q

$\vec{u}_{i+1} = A^{-1}\vec{u}_i$

For j = 1 to i

$$\vec{u}_{i+1} \leftarrow \vec{u}_{i+1} - \underbrace{\left(\vec{u}_{i+1}^T \vec{u}_j\right)}_{H_{ji}} \vec{u}_j$$ Orthogonalize New Vector

$$\vec{u}_{i+1} \leftarrow \frac{1}{\underbrace{\left\|\vec{u}_{i+1}\right\|}_{H_{i+1,i}}} \vec{u}_{i+1}$$ Normalize

Generates q+1 vectors!

SMA-HPC ©2003 MIT

Dynamic Linear Case — Computing U — Arnoldi Identity

$$A^{-1}\underbrace{\begin{bmatrix}\uparrow & \cdots & \uparrow\\ \uparrow & \cdots & \uparrow\\ \uparrow & \cdots & \uparrow\\ \vec{u}_1 & \cdots & \vec{u}_q\\ \downarrow & \cdots & \downarrow\\ \downarrow & \cdots & \downarrow\\ \downarrow & \cdots & \downarrow\end{bmatrix}}_{U_q} = \underbrace{\begin{bmatrix}\uparrow & \cdots & \uparrow\\ \uparrow & \cdots & \uparrow\\ \uparrow & \cdots & \uparrow\\ \vec{u}_1 & \cdots & \vec{u}_q\\ \downarrow & \cdots & \downarrow\\ \downarrow & \cdots & \downarrow\\ \downarrow & \cdots & \downarrow\end{bmatrix}}_{U_q} \underbrace{\begin{bmatrix} H_{11} & \cdots & \cdots & \cdots & H_{1q}\\ H_{21} & H_{22} & \cdots & \cdots & \vdots\\ 0 & H_{32} & \ddots & \cdots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & H_{q,q-1} & H_{qq}\end{bmatrix}}_{H_q}$$

Rank 1 matrix → $+\ H_{q+1,q}\vec{u}_{q+1}e_q^T$

SMA-HPC ©2003 MIT

Dynamic Linear Case — Computing U

Arnoldi Identities Continued

Multiplying U by the inverse of A yeilds

$$A^{-1}U_q = U_q H_q + H_{q+1,q}\vec{u}_{q+1}e_q^T$$

Multiplying by the transpose of U

$$U_q^T A^{-1}U_q = U_q^T U_q H_q + U_q^T H_{q+1,q}\vec{u}_{q+1}e_q^T$$

By orthogonality

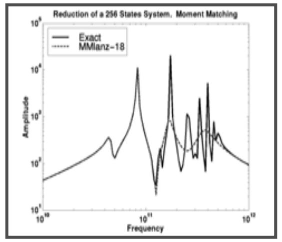
$$U_q^T A^{-1}U_q = H_q = A_r^{-1}$$

⟵ The Projection Alternative Reduced Model

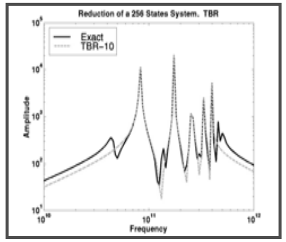
SMA-HPC ©2003 MIT

Two Existing Approaches

- Moment Matching
  - Accurate over a narrow band.
    - Matching function value and derivatives.
  - Cheap:
    - O(n) if A is very sparse.



- Truncated Balanced Realization
  - Wide-band accuracy.
    - Does not follow all details.
  - Theoretical error bound.
  - Expensive: $O(n^3)$



SMA-HPC ©2003 MIT

Reminder about Eigenanalysis

Transfer Function $H(s) = c^T(sI - A)^{-1}b$

Apply Eigendecomposition

$$H(s) = c^T E(sI - \lambda)^{-1} E^{-1}b$$

$$= \tilde{c}^T \begin{bmatrix} \frac{1}{s-\lambda_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{s-\lambda_N} \end{bmatrix} \tilde{b} \Rightarrow H(s) = \sum_{i=1}^{N} \frac{\tilde{c}_i \tilde{b}_i}{s-\lambda_i}$$

Should keep controllable and observable “modes”, but should they be the eigenmodes?

SMA-HPC ©2003 MIT

Truncated Balanced Realization: Non-symmetric Systems

- 1. Calculate controllability gramian, P, and observability gramian, Q, by solving two Lyapunov equations,

$$\mathbf{AP} + \mathbf{PA^T} + \mathbf{bb^T} = \mathbf{0},$$
$$\mathbf{A^T Q} + \mathbf{QA} + \mathbf{c^T c} = \mathbf{0}.$$

- 2. If have Cholesky factors, $\mathbf{P} = \mathbf{Z_b Z_b^T}, \mathbf{Q} = \mathbf{Z_c Z_c^T}$
- 3. Projection: a. $\mathbf{U_c D U_b^T} = \mathbf{Z_c^T Z_b}$,

  b. $\mathbf{S_b} = \mathbf{Z_b U_b}(:,1:k)\mathbf{D}^{-\frac{1}{2}}(1:k,1:k)$,

  $\mathbf{S_c} = \mathbf{Z_c U_c}(:,1:k)\mathbf{D}^{-\frac{1}{2}}(1:k,1:k)$,

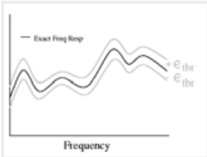
  c. $\mathbf{A_{tbr}^k} = \mathbf{S_c^T A S_b}$, $\tilde{\mathbf{b}}_{\mathbf{tbr}}^{\mathbf{k}} = \mathbf{S_c^T}\tilde{\mathbf{b}}$, $\mathbf{c_{tbr}^k} = \mathbf{c S_b}$

SMA-HPC ©2003 MIT

Properties of the TBR Reduction

- Globally accurate reduced model.
- Maximum frequency domain error is bounded by,

$$\left\| G(jw) - G^{k}_{tbr}(jw) \right\|_{L^\infty} \le 2(\sigma_{k+1} + \sigma_{k+2} + \cdots + \sigma_n) \equiv \varepsilon_{tbr}.$$



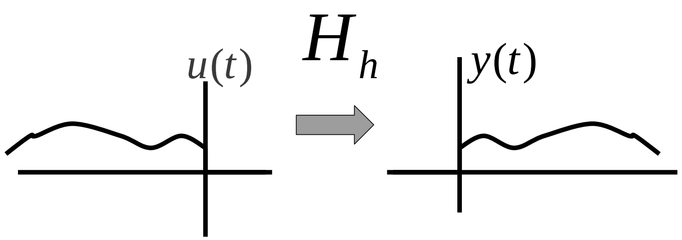
- Guaranteed stable.
- Expensive:
  - Lyapunov equation solve: $O(n^3)$.
  - Singular value decomposition: $O(n^3)$.

Solving Lyapunov Equations

- How to find (approximate) P, Q, or just as good, their factors Z_b, Z_c, efficiently?
- No expensive operations on A: no matrix decompositions.
- Cheap operations: matrix-vector products and solves.
- Low rank approximations.
  - Z_b, and Z_c have only a few columns.
- Recent Approach Cholesky-Free ADI methods

Reduction Based on Hankel Operators

Hankel Operator Maps Past Inputs to Future Outputs



The Hankel Operator has an SVD

- The Singular Values of the Hankel Operator

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

- P, Q are Observability and Controllability Grammians
  - P and Q are NxN matrices
  - Hankel Operator has a finite set of singular values
- Reduction by ignoring small singular values
  - Just like with any matrix

Summary

- Dynamic Linear Case
  - Rational Functions
  - Projection Framework
  - Krylov Methods
- TBR and Hankel Reduction
  - Optimal Reduced Model
  - Extremely computationally expensive

SMA-HPC ©2003 MIT