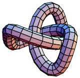
BALANCING-RELATED MODEL REDUCTION FOR LARGE-SCALE SYSTEMS

Peter Benner

Professur Mathematik in Industrie und Technik
Fakultät für Mathematik
Technische Universität Chemnitz

TECHNISCHE UNIVERSITÄT CHEMNITZ



Recent Advances in Model Order Reduction
Workshop — TU Eindhoven, November 23, 2007

Overview

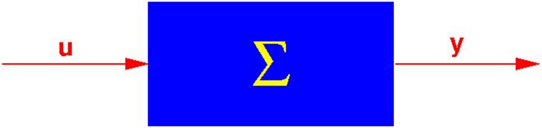
Introduction

Dynamical Systems

$$\Sigma : \begin{cases} \dot{x}(t) = f(t, x(t), u(t)), & x(t_0) = x_0, \\ y(t) = g(t, x(t), u(t)), \end{cases}$$

with

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

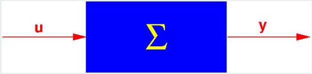


Model Reduction for Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \\ y(t) = g(t, x(t), u(t)). \end{cases}$$

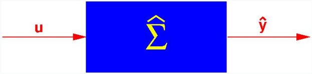
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.



Reduced-Order System

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) = \widehat{g}(t, \hat{x}(t), u(t)). \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.



Goal:

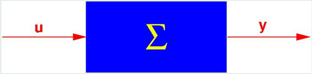
$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Model Reduction for Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \\ y(t) = g(t, x(t), u(t)). \end{cases}$$

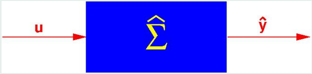
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.



Reduced-Order System

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) = \widehat{g}(t, \hat{x}(t), u(t)). \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.



Goal:

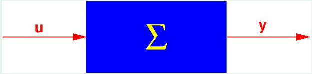
$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Model Reduction for Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \\ y(t) = g(t, x(t), u(t)). \end{cases}$$

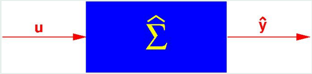
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.



Reduced-Order System

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) = \widehat{g}(t, \hat{x}(t), u(t)). \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.



Goal:

$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\|$ for all admissible input signals.

Model Reduction for Linear Systems

Linear, Time-Invariant (LTI) Systems

$$\begin{aligned} f(t,x,u) &= Ax + Bu, \quad & A \in \mathbb{R}^{n\times n}, \quad & B \in \mathbb{R}^{n\times m}, \\ g(t,x,u) &= Cx + Du, \quad & C \in \mathbb{R}^{p\times n}, \quad & D \in \mathbb{R}^{p\times m}. \end{aligned}$$

Linear Systems Frequency Domain

Application of Laplace transformation $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sx(s))$ to linear system with $x(0) = 0$:

$$sx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),$$

yields I/O-relation in frequency domain:

$$y(s) = \Big(\underbrace{C(sI_n - A)^{-1}B + D}_{=:G(s)} \Big) u(s)$$

G is the transfer function of Σ.

Linear, Time-Invariant (LTI) Systems

$$\begin{aligned} f(t,x,u) &= Ax + Bu, & A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{n\times m}, \\ g(t,x,u) &= Cx + Du, & C \in \mathbb{R}^{p\times n}, \quad D \in \mathbb{R}^{p\times m}. \end{aligned}$$

Linear Systems Frequency Domain

Application of Laplace transformation $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sx(s))$ to linear system with $x(0) = 0$:

$$sx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),$$

yields I/O-relation in frequency domain:

$$y(s) = \Big(\underbrace{C(sI_n - A)^{-1}B + D}_{=:G(s)} \Big) u(s)$$

G is the transfer function of Σ.

Model Reduction for Linear Systems

Linear, Time-Invariant (LTI) Systems

$$\begin{aligned} f(t,x,u) &= Ax + Bu, \quad && A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{n\times m}, \\ g(t,x,u) &= Cx + Du, \quad && C \in \mathbb{R}^{p\times n}, \quad D \in \mathbb{R}^{p\times m}. \end{aligned}$$

Linear Systems Frequency Domain

Application of Laplace transformation ($x(t) \mapsto x(s)$, $\dot{x}(t) \mapsto sx(s)$) to linear system with $x(0) = 0$:

$$sx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),$$

yields I/O-relation in frequency domain:

$$y(s) = \Big(\underbrace{C(sI_n - A)^{-1}B + D}_{=:G(s)} \Big) u(s)$$

G is the transfer function of Σ.

Problem

Approximate the dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu, \qquad & A \in \mathbb{R}^{n\times n}, \quad & B \in \mathbb{R}^{n\times m}, \\ y &= Cx + Du, \qquad & C \in \mathbb{R}^{p\times n}, \quad & D \in \mathbb{R}^{p\times m}. \end{aligned}$$

by reduced-order system

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u, \qquad & \hat{A} \in \mathbb{R}^{r\times r}, \quad & \hat{B} \in \mathbb{R}^{r\times m}, \\ \hat{y} &= \hat{C}\hat{x} + \hat{D}u, \qquad & \hat{C} \in \mathbb{R}^{p\times r}, \quad & \hat{D} \in \mathbb{R}^{p\times m}, \end{aligned}$$

of order $r \ll n$, such that

$$\|y - \hat{y}\| = \left\| Gu - \hat{G}u \right\| \leq \left\| G - \hat{G} \right\| \|u\| < \text{tolerance} \cdot \|u\| \,.$$

$\Longrightarrow$ Approximation problem: $\min_{\text{order}\,(\hat{G}) \leq r} \left\| G - \hat{G} \right\|.$

Problem

Approximate the dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu, & A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{n\times m}, \\ y &= Cx + Du, & C \in \mathbb{R}^{p\times n}, \quad D \in \mathbb{R}^{p\times m}. \end{aligned}$$

by reduced-order system

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u, & \hat{A} \in \mathbb{R}^{r\times r}, \quad \hat{B} \in \mathbb{R}^{r\times m}, \\ \hat{y} &= \hat{C}\hat{x} + \hat{D}u, & \hat{C} \in \mathbb{R}^{p\times r}, \quad \hat{D} \in \mathbb{R}^{p\times m}, \end{aligned}$$

of order $r \ll n$, such that

$$\|y - \hat{y}\| = \left\|Gu - \hat{G}u\right\| \le \left\|G - \hat{G}\right\| \|u\| < \text{tolerance} \cdot \|u\| .$$

$\Longrightarrow$ Approximation problem: $\min_{\text{order}\,(\hat{G}) \le r} \left\|G - \hat{G}\right\|.$

Application Areas

Feedback Control – controllers designed by LQR/LQG, H_2, H_∞ methods are LTI systems of order $\geq n$, but technological implementation needs order ~ 10.

Optimization/open-loop control – time-discretization of already large-scale systems leads to huge number of equality constraints in mathematical program.

Microelectronics – verification of VLSI/ULSI chip design requires high number of simulations for different input signals, various effects due to progressive miniaturization lead to large-scale systems of differential(-algebraic) equations (order $\sim 10^8$).

MEMS/Microsystem design – smart system integration needs compact models for efficient coupled simulation.

...

Goals

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

("system does not generate energy").

Goals

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

("system does not generate energy").

Goals

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

("system does not generate energy").

Goals

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

(*"system does not generate energy"*).

Goals

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

("system does not generate energy").

Goals

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

("system does not generate energy").

Idea:

- A system Σ, realized by (A, B, C, D), is called balanced, if solutions P, Q of the Lyapunov equations

$$AP + PA^T + BB^T = 0, \qquad A^T Q + QA + C^T C = 0,$$

satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via state-space transformation

$$\begin{aligned} \mathcal{T} : (A, B, C, D) &\mapsto (TAT^{-1}, TB, CT^{-1}, D) \\ &= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right) \end{aligned}$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D)$.

Idea:

- A system Σ, realized by (A, B, C, D), is called balanced, if solutions P, Q of the Lyapunov equations

$$AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0,$$

satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via state-space transformation

$$\mathcal{T} : (A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D) = \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right)$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D)$.

Idea:

- A system Σ, realized by (A, B, C, D), is called balanced, if solutions P, Q of the Lyapunov equations

$$AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0,$$

satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via state-space transformation

$$\begin{aligned} \mathcal{T} : (A, B, C, D) &\mapsto (TAT^{-1}, TB, CT^{-1}, D) \\ &= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right) \end{aligned}$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D)$.

Idea:

- A system Σ, realized by (A, B, C, D), is called balanced, if solutions P, Q of the Lyapunov equations

$$AP + PA^T + BB^T = 0, \qquad A^T Q + QA + C^T C = 0,$$

satisfy: $P = Q = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via state-space transformation

$$\begin{aligned} \mathcal{T} : (A, B, C, D) &\mapsto (TAT^{-1}, TB, CT^{-1}, D) \\ &= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right) \end{aligned}$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D)$.

Motivation:

HSV are system invariants: they are preserved under $\mathcal{T}$ and determine the energy transfer given by the Hankel map

$$\mathcal{H} : L_2(-\infty, 0) \mapsto L_2(0, \infty) : u_- \mapsto y_+.$$

Motivation:

HSV are system invariants: they are preserved under $\mathcal{T}$ and determine the energy transfer given by the Hankel map

$$\mathcal{H} : L_2(-\infty, 0) \mapsto L_2(0, \infty) : u_- \mapsto y_+.$$

In balanced coordinates ... energy transfer from u_- to y_+:

$$E := \sup_{\substack{u \in L_2(-\infty,0] \\ x(0)=x_0}} \frac{\int\limits_0^\infty y(t)^T y(t)\,dt}{\int\limits_{-\infty}^0 u(t)^T u(t)\,dt} = \frac{1}{\|x_0\|_2} \sum_{j=1}^n \sigma_j^2 x_{0,j}^2$$

Motivation:

HSV are system invariants: they are preserved under $\mathcal{T}$ and determine the energy transfer given by the Hankel map

$$\mathcal{H} : L_2(-\infty, 0) \mapsto L_2(0, \infty) : u_- \mapsto y_+.$$

In balanced coordinates ... energy transfer from u_- to y_+:

$$E := \sup_{\substack{u \in L_2(-\infty,0] \\ x(0)=x_0}} \frac{\int\limits_0^\infty y(t)^T y(t)\, dt}{\int\limits_{-\infty}^0 u(t)^T u(t)\, dt} = \frac{1}{\|x_0\|_2} \sum_{j=1}^n \sigma_j^2 x_{0,j}^2$$

$\Longrightarrow$ Truncate states corresponding to “small” HSVs

$\Longrightarrow$ complete analogy to best approximation via SVD!

Implementation: SR Method

1. Compute Cholesky factors of the solutions of the Lyapunov equations,
$$P = S^T S, \quad Q = R^T R.$$
2. Compute SVD **(thanks, Gene!)**
$$SR^T = [\, U_1,\ U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$$
3. Set
$$W = R^T V_1 \Sigma_1^{-1/2}, \qquad V = S^T U_1 \Sigma_1^{-1/2}.$$
4. Reduced model is $(W^T AV, W^T B, CV, D)$.

Implementation: SR Method

1. Compute Cholesky factors of the solutions of the Lyapunov equations,
$$P = S^T S, \quad Q = R^T R.$$
2. Compute SVD **(thanks, Gene!)**
$$SR^T = [\,U_1,\ U_2\,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$$
3. Set
$$W = R^T V_1 \Sigma_1^{-1/2}, \qquad V = S^T U_1 \Sigma_1^{-1/2}.$$
4. Reduced model is $(W^T AV, W^T B, CV, D)$.

Implementation: SR Method

1. Compute Cholesky factors of the solutions of the Lyapunov equations,
$$P = S^T S, \quad Q = R^T R.$$
2. Compute SVD **(thanks, Gene!)**
$$SR^T = [\, U_1, \, U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$$
3. Set
$$W = R^T V_1 \Sigma_1^{-1/2}, \qquad V = S^T U_1 \Sigma_1^{-1/2}.$$
4. Reduced model is $(W^T AV, W^T B, CV, D)$.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Classical Balanced Truncation (BT) Mullis/Roberts '76, Moore '81

- P = controllability Gramian of system given by (A, B, C, D).
- Q = observability Gramian of system given by (A, B, C, D).
- P, Q solve dual Lyapunov equations

$$AP + PA^T + BB^T = 0, \qquad A^T Q + QA + C^T C = 0.$$

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

LQG Balanced Truncation (LQGBT) — Jonckheere/Silverman '83

- P/Q = controllability/observability Gramian of closed-loop system based on LQG compensator.
- P, Q solve dual algebraic Riccati equations (AREs)

$$\begin{aligned} 0 &= AP + PA^T - PC^T CP + B^T B, \\ 0 &= A^T Q + QA - QBB^T Q + C^T C. \end{aligned}$$

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Balanced Stochastic Truncation (BST) Desai/Pal '84, Green '88

- P = controllability Gramian of system given by (A, B, C, D), i.e., solution of Lyapunov equation $AP + PA^T + BB^T = 0$.
- Q = observability Gramian of right spectral factor of power spectrum of system given by (A, B, C, D), i.e., solution of ARE

$$\hat{A}^T Q + Q\hat{A} + QB_W(DD^T)^{-1}B_W^T Q + C^T(DD^T)^{-1}C = 0,$$

where $\hat{A} := A - B_W(DD^T)^{-1}C$, $B_W := BD^T + PC^T$.

Balancing-Related Model Reduction

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Positive-Real Balanced Truncation (PRBT) GREEN '88

- Based on positive-real equations, related to positive real (Kalman-Yakubovich-Popov-Anderson) lemma.
- P, Q solve dual AREs

$$\begin{aligned} 0 &= \bar{A}P + P\bar{A}^T + PC^T\bar{R}^{-1}CP + B\bar{R}^{-1}B^T, \\ 0 &= \bar{A}^T Q + Q\bar{A} + QB\bar{R}^{-1}B^T Q + C^T\bar{R}^{-1}C, \end{aligned}$$

where $\bar{R} = D + D^T$, $\bar{A} = A - B\bar{R}^{-1}C$.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Other Balancing-Based Methods

- Bounded-real balanced truncation (BRBT) – based on bounded real lemma [Opdenacker/Jonckheere '88];
- H_∞ balanced truncation (HinfBT) – closed-loop balancing based on H_∞ compensator [Mustafa/Glover '91].

Both approaches require solution of dual AREs.

- Frequency-weighted versions of the above approaches.

Balancing-Related Model Reduction

Properties

- Guaranteed preservation of physical properties like
  - stability (all),
  - passivity (PRBT),
  - minimum phase (BST).
- Computable error bounds, e.g.,

$$\text{BT:} \quad \|G - G_r\|_\infty \quad \leq 2 \sum_{j=r+1}^{n} \sigma_j^{BT},$$

$$\text{LQGBT:} \quad \|G - G_r\|_\infty \quad \leq \; 2 \sum_{j=r+1}^{n} \frac{\sigma_j^{LQG}}{\sqrt{1+(\sigma_j^{LQG})^2}}$$

$$\text{BST:} \quad \|G - G_r\|_\infty \quad \leq \left(\prod_{j=r+1}^{n} \frac{1+\sigma_j^{BST}}{1-\sigma_j^{BST}} - 1 \right) \|G\|_\infty,$$

- Can be combined with singular perturbation approximation for steady-state performance.
- Computations can be modularized.

Numerical Algorithms for Balanced Truncation

General misconception: complexity $\mathcal{O}(n^3)$ – true for several implementations! (e.g., MATLAB, SLICOT).

Numerical Algorithms for Balanced Truncation

General misconception: complexity $\mathcal{O}(n^3)$ – true for several implementations! (e.g., MATLAB, SLICOT).

Algorithmic ideas from numerical linear algebra (since $\sim$ 1997):

Numerical Algorithms for Balanced Truncation

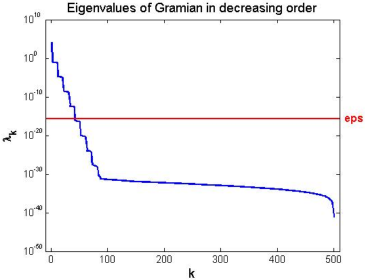
General misconception: complexity $\mathcal{O}(n^3)$ – true for several implementations! (e.g., MATLAB, SLICOT).

Algorithmic ideas from numerical linear algebra (since $\sim$ 1997):

- Instead of Gramians P, Q compute $S, R \in \mathbb{R}^{n\times k}$, $k \ll n$, such that

$$P \approx SS^T, \quad Q \approx RR^T.$$

- Compute S, R with problem-specific Lyapunov/Riccati solvers of "low" complexity directly.



Numerical Algorithms for Balanced Truncation

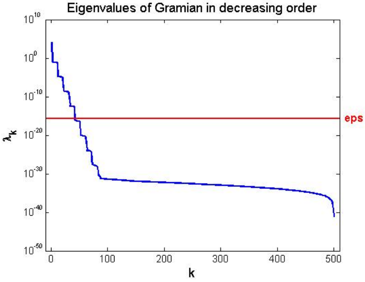
General misconception: complexity $\mathcal{O}(n^3)$ – true for several implementations! (e.g., MATLAB, SLICOT).

Algorithmic ideas from numerical linear algebra (since $\sim$ 1997):

- Instead of Gramians P, Q compute $S, R \in \mathbb{R}^{n \times k}$, $k \ll n$, such that

$$P \approx SS^T, \quad Q \approx RR^T.$$

- Compute S, R with problem-specific Lyapunov/Riccati solvers of "low" complexity directly.



$\rightsquigarrow$ need solver for large-scale matrix equations which computes S, R directly!

General form for $A, G = G^T, W = W^T \in \mathbb{R}^{n\times n}$ given and $X \in \mathbb{R}^{n\times n}$ unknown:

$$
\begin{aligned}
0 &= \mathcal{L}(X) := A^T X + XA + W, \\
0 &= \mathcal{R}(X) := A^T X + XA - XGX + W.
\end{aligned}
$$

In large scale applications, typically

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}K$ for FEM),
- G, W low-rank with $G, W \in \{BB^T, C^T C\}$, where $B \in \mathbb{R}^{n\times m}$, $m \ll n$, $C \in \mathbb{R}^{p\times n}$, $p \ll n$.
- Standard (eigenproblem-based) $\mathcal{O}(n^3)$ methods are not applicable!

General form for $A, G = G^T, W = W^T \in \mathbb{R}^{n\times n}$ given and $X \in \mathbb{R}^{n\times n}$ unknown:

$$
\begin{aligned}
0 &= \mathcal{L}(X) := A^T X + XA + W, \\
0 &= \mathcal{R}(X) := A^T X + XA - XGX + W.
\end{aligned}
$$

In large scale applications, typically

- $n = 10^3 - 10^6$ ($\Longrightarrow$ $10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}K$ for FEM),
- G, W low-rank with $G, W \in \{BB^T, C^T C\}$, where $B \in \mathbb{R}^{n\times m}$, $m \ll n$, $\quad C \in \mathbb{R}^{p\times n}$, $p \ll n$.
- Standard (eigenproblem-based) $\mathcal{O}(n^3)$ methods are not applicable!

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n\times n}$ stable, $B \in \mathbb{R}^{n\times m}$ $(w \ll n)$, consider Lyapunov equation

$$AX + XA^T = -BB^T.$$

- ADI Iteration: [WACHSPRESS 1988]

$$\begin{aligned} (A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + \overline{p_k} I){X_k}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I) \end{aligned}$$

with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.

- For $X_0 = 0$ and proper choice of p_k: $\lim\limits_{k\to\infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n\times n}$ stable, $B \in \mathbb{R}^{n\times m}$ $(w \ll n)$, consider Lyapunov equation
$$AX + XA^T = -BB^T.$$
- ADI Iteration: [Wachspress 1988]
$$\begin{aligned}(A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I)\\ (A + \overline{p_k} I){X_k}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I)\end{aligned}$$
with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.
- For $X_0 = 0$ and proper choice of p_k: $\lim_{k\to\infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n\times n}$ stable, $B \in \mathbb{R}^{n\times m}$ $(w \ll n)$, consider Lyapunov equation

$$AX + XA^T = -BB^T.$$

- ADI Iteration: [Wachspress 1988]

$$\begin{aligned}(A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I)\\ (A + \overline{p_k} I){X_k}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I)\end{aligned}$$

with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.

- For $X_0 = 0$ and proper choice of p_k: $\lim_{k\to\infty} X_k = X$ superlinear.

- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n \times n}$ stable, $B \in \mathbb{R}^{n \times m}$ $(w \ll n)$, consider Lyapunov equation
$$AX + XA^T = -BB^T.$$
- ADI Iteration: [Wachspress 1988]
$$\begin{aligned} (A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + \overline{p_k} I){X_k}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I) \end{aligned}$$
with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.
- For $X_0 = 0$ and proper choice of p_k: $\lim_{k \to \infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

Factored ADI Iteration

Lyapunov equation $AX + XA^T = -BB^T$.

Setting $X_k = Y_k Y_k^T$, some algebraic manipulations $\Longrightarrow$

Algorithm [Penzl 1997, Li/White 2002, B./Li/Penzl 1999/2007]

$$V_1 \leftarrow \sqrt{-2\,\mathrm{Re}(p_1)}(A + p_1 I)^{-1} B, \qquad Y_1 \leftarrow V_1$$

FOR $j = 2, 3, \ldots$

$$V_k \leftarrow \sqrt{\frac{\mathrm{Re}(p_k)}{\mathrm{Re}(p_{k-1})}} \left(V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1} V_{k-1} \right),$$

$$Y_k \leftarrow \mathtt{rrqr}\left(\begin{bmatrix} Y_{k-1} & V_k \end{bmatrix} \right) \qquad \% \text{ column compression}$$

At convergence, $Y_{k_{\max}} Y_{k_{\max}}^T \approx X$, where

$$\mathrm{range}(Y_{k_{\max}}) = \mathrm{range}\left(\begin{bmatrix} V_1 & \ldots & V_{k_{\max}} \end{bmatrix} \right), \quad V_k = \big[\,\big] \in \mathbb{C}^{n \times m}$$

Note: Implementation in real arithmetic possible by combining two steps.

Factored ADI Iteration

Lyapunov equation $AX + XA^T = -BB^T$.

Setting $X_k = Y_k Y_k^T$, some algebraic manipulations $\Longrightarrow$

Algorithm [Penzl 1997, Li/White 2002, B./Li/Penzl 1999/2007]

$$V_1 \leftarrow \sqrt{-2\,\mathrm{Re}(p_1)}(A + p_1 I)^{-1}B, \qquad Y_1 \leftarrow V_1$$

FOR $j = 2, 3, \ldots$

$$V_k \leftarrow \sqrt{\frac{\mathrm{Re}(p_k)}{\mathrm{Re}(p_{k-1})}}\left(V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1} V_{k-1}\right),$$

$$Y_k \leftarrow \mathtt{rrqr}\left(\left[\; Y_{k-1} \quad V_k \;\right]\right) \qquad \% \quad \text{column compression}$$

At convergence, $Y_{k_{\max}} Y_{k_{\max}}^T \approx X$, where

$$\mathrm{range}(Y_{k_{\max}}) = \mathrm{range}\left(\left[\; V_1 \quad \ldots \quad V_{k_{\max}} \;\right]\right), \qquad V_k = \big[\;\big] \in \mathbb{C}^{n \times m}$$

Note: Implementation in real arithmetic possible by combining two steps.

Newton's Method for AREs

[Kleinman '68, Mehrmann '91, Lancaster/Rodman '95, B./Byers '94/'98, B. '97, Guo/Laub '99]

- Consider $0 = \mathcal{R}(X) = C^T C + A^T X + XA - XBB^T X.$
- Frechét derivative of $\mathcal{R}(X)$ at X:

$$\mathcal{R}'_X : Z \to (A - BB^T X)^T Z + Z(A - BB^T X).$$

- Newton-Kantorovich method:

$$X_{j+1} = X_j - \left(\mathcal{R}'_{X_j}\right)^{-1} \mathcal{R}(X_j), \quad j = 0, 1, 2, \ldots$$

Newton's method (with line search) for AREs

FOR $j = 0, 1, \ldots$

1. $A_j \leftarrow A - BB^T X_j =: A - BK_j$.
2. Solve the Lyapunov equation $A_j^T N_j + N_j A_j = -\mathcal{R}(X_j)$.
3. $X_{j+1} \leftarrow X_j + t_j N_j$.

END FOR j

Newton's Method for AREs

[Kleinman '68, Mehrmann '91, Lancaster/Rodman '95, B./Byers '94/'98, B. '97, Guo/Laub '99]

- Consider $\quad 0 = \mathcal{R}(X) = C^T C + A^T X + XA - XBB^T X.$
- Frechét derivative of $\mathcal{R}(X)$ at X:

$$\mathcal{R}'_X : Z \to (A - BB^T X)^T Z + Z(A - BB^T X).$$

- Newton-Kantorovich method:

$$X_{j+1} = X_j - \left(\mathcal{R}'_{X_j}\right)^{-1} \mathcal{R}(X_j), \quad j = 0, 1, 2, \ldots$$

Newton's method (with line search) for AREs

FOR $j = 0, 1, \ldots$

1. $A_j \leftarrow A - BB^T X_j =: A - BK_j.$
2. Solve the Lyapunov equation $\quad A_j^T N_j + N_j A_j = -\mathcal{R}(X_j).$
3. $X_{j+1} \leftarrow X_j + t_j N_j.$

END FOR j

Newton's Method for AREs

[Kleinman '68, Mehrmann '91, Lancaster/Rodman '95, B./Byers '94/'98, B. '97, Guo/Laub '99]

- Consider $0 = \mathcal{R}(X) = C^T C + A^T X + XA - XBB^T X.$
- Frechét derivative of $\mathcal{R}(X)$ at X:

$$\mathcal{R}'_X : Z \to (A - BB^T X)^T Z + Z(A - BB^T X).$$

- Newton-Kantorovich method:

$$X_{j+1} = X_j - \left(\mathcal{R}'_{X_j}\right)^{-1} \mathcal{R}(X_j), \quad j = 0, 1, 2, \ldots$$

Newton's method (with line search) for AREs

FOR $j = 0, 1, \ldots$

1. $A_j \leftarrow A - BB^T X_j =: A - BK_j.$
2. Solve the Lyapunov equation $A_j^T N_j + N_j A_j = -\mathcal{R}(X_j).$
3. $X_{j+1} \leftarrow X_j + t_j N_j.$

END FOR j

Newton's Method for AREs

[Kleinman '68, Mehrmann '91, Lancaster/Rodman '95, B./Byers '94/'98, B. '97, Guo/Laub '99]

- Consider $0 = \mathcal{R}(X) = C^T C + A^T X + XA - XBB^T X.$
- Frechét derivative of $\mathcal{R}(X)$ at X:

$$\mathcal{R}'_X : Z \to (A - BB^T X)^T Z + Z(A - BB^T X).$$

- Newton-Kantorovich method:

$$X_{j+1} = X_j - \left(\mathcal{R}'_{X_j}\right)^{-1} \mathcal{R}(X_j), \quad j = 0, 1, 2, \ldots$$

Newton's method (with line search) for AREs

FOR $j = 0, 1, \ldots$

1. $A_j \leftarrow A - BB^T X_j =: A - BK_j.$
2. Solve the Lyapunov equation $A_j^T N_j + N_j A_j = -\mathcal{R}(X_j).$
3. $X_{j+1} \leftarrow X_j + t_j N_j.$

END FOR j

Newton's Method for AREs

Properties and Implementation

- Convergence for K_0 stabilizing:
  - $A_j = A - BK_j = A - BB^T X_j$ is stable $\forall\ j \geq 0$.
  - $\lim_{j\to\infty} \|\mathcal{R}(X_j)\|_F = 0$ (monotonically).
  - $\lim_{j\to\infty} X_j = X_* \geq 0$ (locally quadratic).
- Need large-scale Lyapunov solver; here, ADI iteration: linear systems with dense, but "sparse+low rank" coefficient matrix A_j:

  $A_j = A - B \cdot K_j$

  = sparse − m ·

- $m \ll n \implies$ efficient "inversion" using Sherman-Morrison-Woodbury formula:

$$(A - BK_j)^{-1} = (I_n + A^{-1}B(I_m - K_jA^{-1}B)^{-1}K_j)A^{-1}.$$

- BUT: $X = X^T \in \mathbb{R}^{n\times n} \implies n(n+1)/2$ unknowns!

Newton's Method for AREs

Properties and Implementation

- Convergence for K_0 stabilizing:
  - $A_j = A - BK_j = A - BB^T X_j$ is stable $\forall\ j \geq 0$.
  - $\lim_{j\to\infty} \|\mathcal{R}(X_j)\|_F = 0$ (monotonically).
  - $\lim_{j\to\infty} X_j = X_* \geq 0$ (locally quadratic).
- Need large-scale Lyapunov solver; here, ADI iteration: linear systems with dense, but "sparse+low rank" coefficient matrix A_j:

$$A_j \quad = \quad A \quad - \quad B \quad \cdot \quad K_j$$

$$= \quad \boxed{\text{sparse}} \quad - \quad \boxed{m} \quad \cdot \quad \boxed{}$$

- $m \ll n \Longrightarrow$ efficient "inversion" using Sherman-Morrison-Woodbury formula:

$$(A - BK_j)^{-1} = (I_n + A^{-1}B(I_m - K_jA^{-1}B)^{-1}K_j)A^{-1}.$$

- BUT: $X = X^T \in \mathbb{R}^{n\times n} \Longrightarrow n(n+1)/2$ unknowns!

- Convergence for K_0 stabilizing:
  - $A_j = A - BK_j = A - BB^T X_j$ is stable $\forall\ j \geq 0$.
  - $\lim_{j\to\infty} \|\mathcal{R}(X_j)\|_F = 0$ (monotonically).
  - $\lim_{j\to\infty} X_j = X_* \geq 0$ (locally quadratic).
- Need large-scale Lyapunov solver; here, ADI iteration: linear systems with dense, but "sparse+low rank" coefficient matrix A_j:

$$A_j \quad = \quad A \quad - \quad B \quad \cdot \quad K_j$$

- $m \ll n \implies$ efficient "inversion" using Sherman-Morrison-Woodbury formula:

$$(A - BK_j)^{-1} = (I_n + A^{-1}B(I_m - K_jA^{-1}B)^{-1}K_j)A^{-1}.$$

- BUT: $X = X^T \in \mathbb{R}^{n\times n} \implies n(n+1)/2$ unknowns!

Newton's Method for AREs

Properties and Implementation

- Convergence for K_0 stabilizing:
  - $A_j = A - BK_j = A - BB^T X_j$ is stable $\forall\ j \geq 0$.
  - $\lim_{j\to\infty} \|\mathcal{R}(X_j)\|_F = 0$ (monotonically).
  - $\lim_{j\to\infty} X_j = X_* \geq 0$ (locally quadratic).
- Need large-scale Lyapunov solver; here, ADI iteration: linear systems with dense, but "sparse+low rank" coefficient matrix A_j:

$$A_j \quad = \quad A \quad - \quad B \quad \cdot \quad K_j$$

- $m \ll n \implies$ efficient "inversion" using Sherman-Morrison-Woodbury formula:

$$(A - BK_j)^{-1} = (I_n + A^{-1}B(I_m - K_jA^{-1}B)^{-1}K_j)A^{-1}.$$

- BUT: $X = X^T \in \mathbb{R}^{n\times n} \implies n(n+1)/2$ unknowns!

Low-Rank Newton-ADI for AREs

Re-write Newton's method for AREs

$$A_j^T N_j + N_j A_j = -\mathcal{R}(X_j)$$

$$\Longleftrightarrow$$

$$A_j^T \underbrace{(X_j + N_j)}_{=X_{j+1}} + \underbrace{(X_j + N_j)}_{=X_{j+1}} A_j = \underbrace{-C^T C - X_j B B^T X_j}_{=:-W_j W_j^T}$$

Set $X_j = Z_j Z_j^T$ for $\operatorname{rank}(Z_j) \ll n \Longrightarrow$

$$A_j^T \left(Z_{j+1} Z_{j+1}^T\right) + \left(Z_{j+1} Z_{j+1}^T\right) A_j = -W_j W_j^T$$

Factored Newton Iteration [B./Li/Penzl 1999/2006]

Solve Lyapunov equations for Z_{j+1} directly by factored ADI iteration and use 'sparse + low-rank' structure of A_j.

Low-Rank Newton-ADI for AREs

Re-write Newton's method for AREs

$$A_j^T N_j + N_j A_j = -\mathcal{R}(X_j)$$

$$\Longleftrightarrow$$

$$A_j^T \underbrace{(X_j + N_j)}_{=X_{j+1}} + \underbrace{(X_j + N_j)}_{=X_{j+1}} A_j = \underbrace{-C^T C - X_j B B^T X_j}_{=:-W_j W_j^T}$$

Set $X_j = Z_j Z_j^T$ for $\operatorname{rank}(Z_j) \ll n \Longrightarrow$

$$A_j^T \left(Z_{j+1} Z_{j+1}^T\right) + \left(Z_{j+1} Z_{j+1}^T\right) A_j = -W_j W_j^T$$

Factored Newton Iteration [B./Li/Penzl 1999/2006]

Solve Lyapunov equations for Z_{j+1} directly by factored ADI iteration and use 'sparse + low-rank' structure of A_j.

Balanced Truncation Implementations of Complexity $< \mathcal{O}(n^3)$

Parallelization:

- Efficient parallel algorithms based on matrix sign function.
- Complexity $\mathcal{O}(n^3/q)$ on q-processor machine.
- Software library PLiCMR.

(B./Quintana-Ortí/Quintana-Ortí since 1999)

Formatted Arithmetic:

For special problems from PDE control use implementation based on hierarchical matrices and matrix sign function method (Baur/B.), complexity $\mathcal{O}(n \log^2(n) r^2)$.

Balanced Truncation Implementations of Complexity $< \mathcal{O}(n^3)$

Parallelization:

- Efficient parallel algorithms based on matrix sign function.
- Complexity $\mathcal{O}(n^3/q)$ on q-processor machine.
- Software library PLiCMR.

(B./Quintana-Ortí/Quintana-Ortí since 1999)

Formatted Arithmetic:

For special problems from PDE control use implementation based on hierarchical matrices and matrix sign function method (Baur/B.), complexity $\mathcal{O}(n \log^2(n) r^2)$.

Balanced Truncation Implementations of Complexity $< \mathcal{O}(n^3)$

Sparse Balanced Truncation:

- Sparse implementation using sparse Lyapunov solver (ADI+MUMPS/SuperLU).
- Complexity $\mathcal{O}(n(k^2 + r^2))$.
- Software:
  - + MATLAB toolbox LYAPACK (PENZL 1999),
  - + Software library SPARED with WebComputing interface.
    (BADÍA/B./QUINTANA-ORTÍ/QUINTANA-ORTÍ since 2003)

- Efficient BT implementations are based on assumption $n \gg m, p$.
- For on-chip clock distribution distribution networks, power grids, wide buses, this assumption is not justified; here, $m, p = \mathcal{O}(n)$, e.g., $m = p = \frac{n}{2}, \frac{n}{4}$.
- Fortunately, BT can easily be combined with SVDMOR [Feldmann/Liu '04: for $G(s) = C(sE - A)^{-1}B$, let

$$G(s_0) = C(s_0 E - A)^{-1} B = U\Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$$

$$H_{DC} = LG^{-1}B = U\Sigma V^T,$$

$$U\Sigma V^T \approx U_r \Sigma_r V_r^T$$

with approximation error

$$\|H_{DC} - U_r \Sigma_r V_r^T\|_2 = \sigma_{r+1}.$$

$$L \approx U_r b_L, \qquad B \approx b_B V_r^T,$$

where b_B and b_L result from applying the Moore-Penrose pseudoinverse of U_r and V_r, respectively, to L and B:

Examples

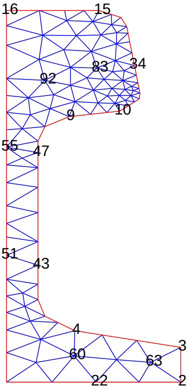
Optimal Control: Cooling of Steel Profiles

- Mathematical model: boundary control for linearized 2D heat equation.

$$
\begin{aligned}
c \cdot \rho \frac{\partial}{\partial t} x &= \lambda \Delta x, && \xi \in \Omega \\
\lambda \frac{\partial}{\partial n} x &= \kappa(u_k - x), && \xi \in \Gamma_k,\ 1 \le k \le 7, \\
\frac{\partial}{\partial n} x &= 0, && \xi \in \Gamma_7.
\end{aligned}
$$

$\Longrightarrow m = 7, p = 6.$

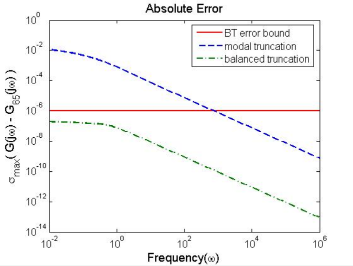
- FEM Discretization, different models for initial mesh ($n = 371$), 1, 2, 3, 4 steps of mesh refinement $\Rightarrow$ $n = 1357, 5177, 20209, 79841$.



Source: Physical model: courtesy of Mannesmann/Demag.

Math. model: Tröltzsch/Unger 1999/2001, Penzl 1999, Saak 2003.

$n = 1357$, Absolute Error

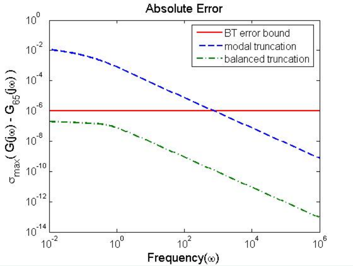


- BT model computed with sign function method,
- MT w/o static condensation, same order as BT model.

Examples

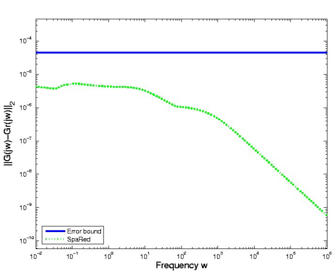
Optimal Control: Cooling of Steel Profiles

$n = 1357$, Absolute Error



- BT model computed with sign function method,
- MT w/o static condensation, same order as BT model.

$n = 79841$, Absolute error

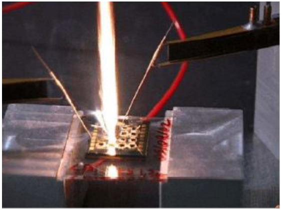


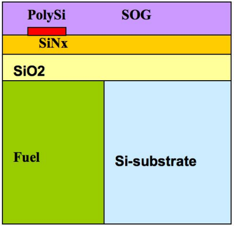
- BT model computed using SpaRed,
- computation time: 8 min.

Examples

MEMS: Microthruster

- Co-integration of solid fuel with silicon micromachined system.
- Goal: Ignition of solid fuel cells by electric impulse.
- Application: nano satellites.
- Thermo-dynamical model, ignition via heating an electric resistance by applying voltage source.
- Design problem: reach ignition temperature of fuel cell w/o firing neighboring cells.
- Spatial FEM discretization of thermo-dynamical model $\rightsquigarrow$ linear system, $m = 1$, $p = 7$.





Source: The Oberwolfach Benchmark Collection http://www.imtek.de/simulation/benchmark

Courtesy of C. Rossi, LAAS-CNRS/EU project "Micropyros".

Examples

MEMS: Microthruster

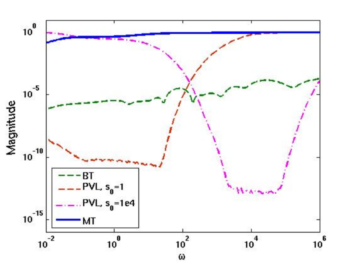
- axial-symmetric 2D model
- FEM discretization using linear (quadratic) elements $\rightsquigarrow n = 4,257$ $(11,445)$ $m = 1$, $p = 7$.
- Reduced model computed using SpaRed, modal truncation using ARPACK, and Z. Bai's PVL implementation.

Examples

MEMS: Microthruster

- axial-symmetric 2D model
- FEM discretization using linear (quadratic) elements $\rightsquigarrow n = 4,257$ $(11,445)$ $m = 1$, $p = 7$.
- Reduced model computed using SpaRed, modal truncation using ARPACK, and Z. Bai's PVL implementation.

Relative error $n = 4,257$

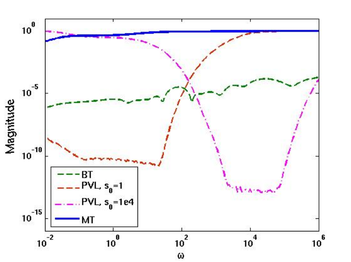


Examples

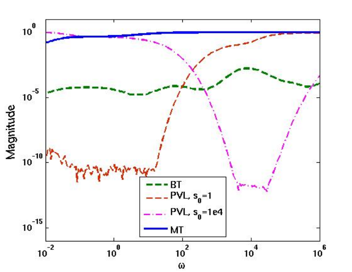
MEMS: Microthruster

- axial-symmetric 2D model
- FEM discretization using linear (quadratic) elements $\rightsquigarrow n = 4,257$ $(11,445)$ $m = 1$, $p = 7$.
- Reduced model computed using SPARED, modal truncation using ARPACK, and Z. Bai's PVL implementation.

Relative error $n = 4,257$



Relative error $n = 11,445$

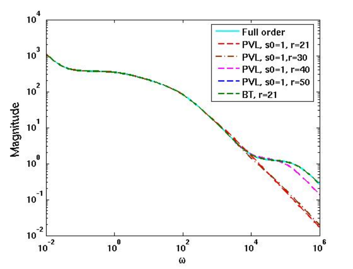


Examples

MEMS: Microthruster

- axial-symmetric 2D model
- FEM discretization using linear (quadratic) elements $\rightsquigarrow n = 4,257$ $(11,445)$ $m = 1,\ p = 7$.
- Reduced model computed using SpaRed, modal truncation using ARPACK, and Z. Bai's PVL implementation.

Frequency Response BT/PVL

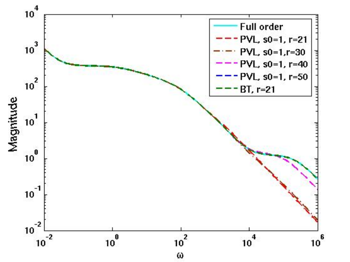


Examples

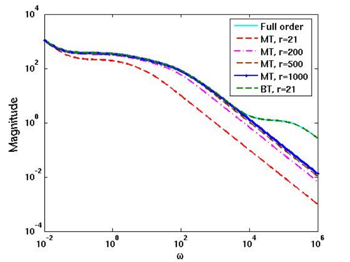
MEMS: Microthruster

- axial-symmetric 2D model
- FEM discretization using linear (quadratic) elements $\rightsquigarrow$ $n = 4,257$ $(11,445)$ $m = 1$, $p = 7$.
- Reduced model computed using SpaRed, modal truncation using ARPACK, and Z. Bai's PVL implementation.

Frequency Response BT/PVL

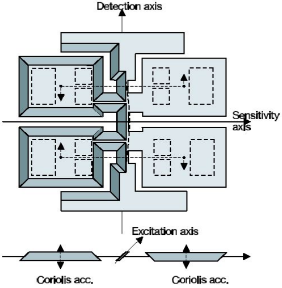


Frequency Response BT/MT



- By applying AC voltage to electrodes, wings are forced to vibrate in anti-phase in wafer plane.
- Coriolis forces induce motion of wings out of wafer plane yielding sensor data.

- Vibrating micro-mechanical gyroscope for inertial navigation.
- Rotational position sensor.



Source: The Oberwolfach Benchmark Collection http://www.imtek.de/simulation/benchmark

Courtesy of D. Billger (Imego Institute, Göteborg), Saab Bofors Dynamics AB.

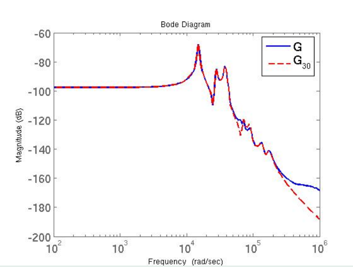
Examples
MEMS: Butterfly Gyro

- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187) $\rightsquigarrow$ $n = 34,722$, $m = 1$, $p = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Examples
MEMS: Butterfly Gyro

- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187)
  $\rightsquigarrow$ $n = 34,722$, $m = 1$, $p = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Frequency Response Analysis

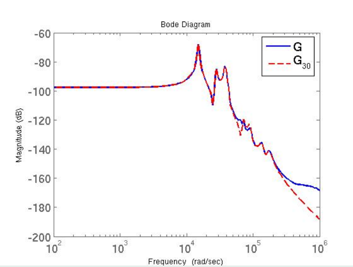


Examples

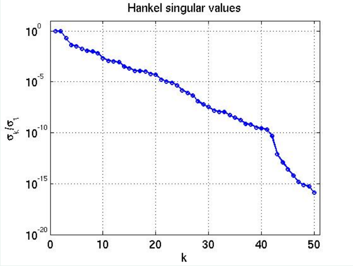
MEMS: Butterfly Gyro

- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187)
  $\rightsquigarrow$ $n = 34,722$, $m = 1$, $p = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Frequency Response Analysis



Hankel Singular Values



SLICOT Model and Controller Reduction Toolbox

- Stand-alone MATLAB Toolbox based on the Subroutine Library in Systems and Control Theory SLICOT which provides > 400 Fortran 77 routines for systems and control related computations; other MATLAB Toolboxes:
  - SLICOT Basic Systems and Control Toolbox,
  - SLICOT System Identification Toolbox.
- Much enhanced functionality compared to MATLAB's Control Toolboxes, in particular coprime factorization methods, frequency-weighted BT methods, controller reduction.
- Maintained by NICONET e.V.
- Distributed by Synoptio GmbH, Berlin.
- For more information, visit http://www.slicot.org and http://www.synoptio.de.

Model Reduction Software

LyaPack

MATLAB toolbox for computations involving large-scale Lyapunov equations with sparse coefficient matrix A and low-rank constant term; contains BT implementation and dominant subspace approximation method.
Available as additional software (no registration necessary) from

http://www.slicot.org.

PLiCMR, SpaRed

Parallel implementations of BT and some balancing-related methods for dense and sparse linear systems.

MorLAB

Model-reduction Laboratory, contains MATLAB functions for balancing-related methods, available from

http://www.tu-chemnitz.de/~benner/software.php.

References

1. G. Obinata and B.D.O. Anderson.
Model Reduction for Control System Design.
Springer-Verlag, London, UK, 2001.

2. Z. Bai.
Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems.
Appl. Numer. Math, 43(1–2):9–44, 2002.

3. R. Freund.
Model reduction methods based on Krylov subspaces.
Acta Numerica, 12:267–319, 2003.

4. P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí.
State-space truncation methods for parallel model reduction of large-scale systems.
Parallel Comput., 29:1701–1722, 2003.

5. P. Benner.
Numerical Linear Algebra for Model Reduction in Control and Simulation.
GAMM Mitt., 29(2):275–296, 2006.

6. P. Benner, V. Mehrmann, and D. Sorensen (editors).
Dimension Reduction of Large-Scale Systems.
Lecture Notes in Computational Science and Engineering, Vol. 45,
Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

7. A.C. Antoulas.
Lectures on the Approximation of Large-Scale Dynamical Systems.
SIAM Publications, Philadelphia, PA, 2005.

8. P. Benner, R. Freund, D. Sorensen, and A. Varga (editors).
Special issue on *Order Reduction of Large-Scale Systems*.
Linear Algebra Appl., Vol. 415, June 2006.