SYSTEM-THEORETIC METHODS
FOR MODEL REDUCTION OF LARGE-SCALE SYSTEMS:
SIMULATION, CONTROL, AND INVERSE PROBLEMS

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

MATHMOD 2009
Vienna, February 11–13, 2009
Overview

1 Introduction
   - Model Reduction
   - Application Areas
   - Goals

2 System-Theoretic Model Reduction
   - Balancing Basics
   - Balanced Truncation and Relatives
   - Solving Large-Scale Matrix Equations

3 Numerical Examples
   - Simulation
   - Control

4 Conclusions and Outlook
Model Reduction for Dynamical Systems

Dynamical Systems

\[ \Sigma : \begin{cases} 
E \dot{x}(t) &= f(t, x(t), u(t)), \quad x(t_0) = x_0, \quad (a) \\
y(t) &= g(t, x(t), u(t)) & (b)
\end{cases} \]

with

- (generalized) states \( x(t) \in \mathbb{R}^n \) \((E \in \mathbb{R}^{n \times n})\),
- inputs \( u(t) \in \mathbb{R}^m \),
- outputs \( y(t) \in \mathbb{R}^p \), (b) is called output equation.

If \( E \) singular \( \Rightarrow \) (a) is system of differential-algebraic equations (DAEs)
otherwise \( \Rightarrow \) (a) is system of ordinary differential equations (ODEs)
Model Reduction for Dynamical Systems

Original System

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

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

Reduced-Order System

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

- 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:

\[ \| y - \hat{y} \| < \text{tolerance} \cdot \| u \| \text{ for all admissible input signals.} \]
## Model Reduction for Dynamical Systems

### Original System

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

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

### Reduced-Order System

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

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

### Goal:

\[ \|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \text{ for all admissible input signals.} \]
Linear Systems

Linear, Time-Invariant (LTI) / Descriptor Systems

\[
\begin{align*}
E \dot{x}(t) &= Ax(t) + Bu(t), \\
y(t) &= Cx(t) + Du(t),
\end{align*}
\]

\(A, E \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}, \quad C \in \mathbb{R}^{p \times n}, \quad D \in \mathbb{R}^{p \times m}.\)

Laplace Transformation / Frequency Domain

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

\[
sEx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),
\]

yields I/O-relation in frequency domain:

\[
y(s) = \left( C(sE - A)^{-1}B + D \right) u(s) =: G(s)
\]

\(G\) is the transfer function of \(\Sigma\).
### Linear Systems

#### Linear, Time-Invariant (LTI) / Descriptor Systems

\[
\begin{align*}
E \dot{x}(t) &= Ax(t) + Bu(t), \quad A, E \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}, \\
y(t) &= Cx(t) + Du(t), \quad C \in \mathbb{R}^{p \times n}, \quad D \in \mathbb{R}^{p \times m}.
\end{align*}
\]

#### Laplace Transformation / Frequency Domain

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

\[
sEx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),
\]

yields I/O-relation in frequency domain:

\[
y(s) = \left( C(sE - A)^{-1}B + D \right)u(s) =: G(s)
\]

\(G\) is the transfer function of \(\Sigma\).
Model Reduction for Linear Systems

**Problem**

Approximate the dynamical system

\[
\begin{align*}
E \dot{x} &= Ax + Bu, & A, E &\in \mathbb{R}^{n \times n}, & B &\in \mathbb{R}^{n \times m}, \\
y &= Cx + Du, & C &\in \mathbb{R}^{p \times n}, & D &\in \mathbb{R}^{p \times m},
\end{align*}
\]

by reduced-order system

\[
\begin{align*}
\hat{E} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u, & \hat{A}, \hat{E} &\in \mathbb{R}^{r \times r}, & \hat{B} &\in \mathbb{R}^{r \times m}, \\
\hat{y} &= \hat{C}\hat{x} + \hat{D}u, & \hat{C} &\in \mathbb{R}^{p \times r}, & \hat{D} &\in \mathbb{R}^{p \times m},
\end{align*}
\]

of order \( r \ll n \), such that

\[
\|y - \hat{y}\| = \|G_u - \hat{G}u\| \leq \|G - \hat{G}\|\|u\| < \text{tolerance} \cdot \|u\|.
\]

\[\implies \text{Approximation problem: } \min_{\text{order } \hat{G} \leq r} \|G - \hat{G}\|\]
Problem

Approximate the dynamical system

\[
\begin{align*}
E \dot{x} &= Ax + Bu, & A, E \in \mathbb{R}^{n \times n}, & B \in \mathbb{R}^{n \times m}, \\
y &= Cx + Du, & C \in \mathbb{R}^{p \times n}, & D \in \mathbb{R}^{p \times m},
\end{align*}
\]

by reduced-order system

\[
\begin{align*}
\hat{E} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u, & \hat{A}, \hat{E} \in \mathbb{R}^{r \times r}, & \hat{B} \in \mathbb{R}^{r \times m}, \\
\hat{y} &= \hat{C}\hat{x} + \hat{D}u, & \hat{C} \in \mathbb{R}^{p \times r}, & \hat{D} \in \mathbb{R}^{p \times m},
\end{align*}
\]

of order \( r \ll n \), such that

\[
\|y - \hat{y}\| = \|Gu - \hat{G}u\| \leq \|G - \hat{G}\|\|u\| < \text{tolerance} \cdot \|u\|.
\]

\[\implies \text{Approximation problem: } \min_{\text{order } (\hat{G}) \leq r} \|G - \hat{G}\|.\]
Application Areas
General assumptions

Here:

- **linear** systems,
- $n \gg m, p$,
- $n$ so large, that $A(E)$ cannot be stored in main memory (RAM) as $n \times n$ array: $n > 5000$, say, e.g., from
  - semi-discretization of PDEs,
  - finite element modeling of MEMS,
  - VLSI design/circuit simulation, ...
- $A(E)$ sparse or data-sparse, i.e., $A(E)$ can be stored in $O(n)$ or $O(n \log n)$ memory locations, but matrix manipulations like similarity transformations are too expensive (possible exception: permutations, sparse factorizations).
Application Areas

General assumptions

Here:

- **linear systems**, 

- $n \gg m, p$, 

- $n$ so large, that $A(E)$ cannot be stored in main memory (RAM) as $n \times n$ array: $n > 5000$, say, e.g., from 
  - semi-discretization of PDEs, 
  - finite element modeling of MEMS, 
  - VLSI design/circuit simulation, …

- $A(E)$ sparse or data-sparse, i.e., $A(E)$ can be stored in $O(n)$ or $O(n \log n)$ memory locations, but matrix manipulations like similarity transformations are too expensive (possible exception: permutations, sparse factorizations).
Here:

- **linear** systems,
- \( n \gg m, p, \)
- \( n \) so large, that \( A(E) \) cannot be stored in main memory (RAM) as \( n \times n \) array: \( n > 5000 \), say, e.g., from
  - semi-discretization of PDEs,
  - finite element modeling of MEMS,
  - VLSI design/circuit simulation, \ldots
- \( A(E) \) sparse or data-sparse, i.e., \( A(E) \) can be stored in \( O(n) \) or \( O(n \log n) \) memory locations, but matrix manipulations like similarity transformations are too expensive (possible exception: permutations, sparse factorizations).
Here:

- **linear** systems,
- \( n \gg m, p, \)
- \( n \) so large, that \( A(E) \) cannot be stored in main memory (RAM) as \( n \times n \) array: \( n > 5000 \), say, e.g., from
  - semi-discretization of PDEs,
  - finite element modeling of MEMS,
  - VLSI design/circuit simulation, ...
- \( A(E) \) sparse or data-sparse, i.e., \( A(E) \) can be stored in \( O(n) \) or \( O(n \log n) \) memory locations, but matrix manipulations like similarity transformations are too expensive (possible exception: permutations, sparse factorizations).
Application Areas
Simulation

Time-domain simulation

Evaluation of variation-of-constants formula

\[ y(t) = C \exp(At) \left( x^0 + \int_0^t \exp(-A\tau)Bu(\tau)d\tau \right), \]

usually too expensive \(\rightsquigarrow\) numerical simulation, e.g., using backwards Euler

\[ y_h(t_{k+1}) = C(E - h_kA)^{-1}(Ex_h(t_k) + h_kBu(t_{k+1})) + Du(t_{k+1}), \]

Bottleneck: solution of \((E - h_kA)z = b\), computation time can be significantly reduced by using reduced-order model:

\[ \hat{y}_h(t_{k+1}) = \hat{C}(\hat{E} - h_k\hat{A})^{-1}(\hat{E}x_h(t_k) + h_k\hat{B}u(t_{k+1})) + \hat{D}u(t_{k+1}). \]
Frequency-domain simulation

**Frequency response analysis**, e.g., for Bode, Nyquist or Nichols plots, requires evaluation of transfer function

\[ G(\omega_k) = C(\omega_k E - A)^{-1}B + D, \quad \omega_k \geq 0, \ k = 1, \ldots, N_f. \]

**Bottleneck**: solution of \((\omega_k E - A)z = b\).

Computation time can be significantly reduced by using reduced-order model:

\[ \hat{G}(\omega_k) = \hat{C}(\omega_k \hat{E} - \hat{A})^{-1}\hat{B} + \hat{D}. \]
Frequency-domain simulation

**Frequency response analysis**, e.g., for Bode, Nyquist or Nichols plots, requires evaluation of transfer function

\[ G(\omega_k) = C(\omega_k E - A)^{-1}B + D, \quad \omega_k \geq 0, \ k = 1, \ldots, N_f. \]

**Bottleneck**: solution of \((\omega_k E - A)z = b\).

Computation time can be significantly reduced by using reduced-order model:

\[ \hat{G}(\omega_k) = \hat{C}(\omega_k \hat{E} - \hat{A})^{-1}\hat{B} + \hat{D}. \]

**But**: the cost for solving the linear systems in time/frequency domain simulation may not benefit from smaller order, if efficient sparse direct solver for full-size sparse system matrices is available.
An easy improvement

Significant reduction can be achieved by transforming $(\hat{A}, \hat{E})$ to Hessenberg-triangular form using QZ algorithm, i.e., compute orthogonal $Q, Z$ such that

$$Q(\lambda \hat{E} - \hat{A})Z = \lambda \begin{bmatrix} \vdots \end{bmatrix} - \begin{bmatrix} \vdots \end{bmatrix} = \begin{bmatrix} \vdots \end{bmatrix}.$$

New reduced-order system: $(Q\hat{E}Z, Q\hat{A}Z, Q\hat{B}, \hat{C}Z)$, linear systems of equations

$$(j\omega \hat{E} - \hat{A})x = b,$$

$$(\hat{E} - h_k \hat{A})x_{k+1} = \hat{E}x_k + \ldots,$$

have Hessenberg form and can thus be solved using $r-1$ Givens rotations only! (Needs Hessenberg solver inside simulator.)

For symmetric systems, further reduction can be achieved.
Application Areas
(Optimal) Control

Feedback Controllers
A feedback controller (dynamic compensator) is a linear system of order $N$, where

- input = output of plant,
- output = input of plant.

Modern (LQG-/H$_2$-/H$_\infty$-) control design: $N \geq n$.

Practical controllers require small $N$ ($N \sim 10$, say) due to

- real-time constraints,
- increasing fragility for larger $N$.

$\implies$ reduce order of plant ($n$) and/or controller ($N$).
Feedback Controllers

A feedback controller (dynamic compensator) is a linear system of order \( N \), where

- input = output of plant,
- output = input of plant.

Modern (LQG- / \( \mathcal{H}_2 \)- / \( \mathcal{H}_\infty \)-) control design: \( N \geq n \).

Practical controllers require small \( N \) (\( N \sim 10 \), say) due to

- real-time constraints,
- increasing fragility for larger \( N \).

\[ \begin{align*}
x' &= A x + B u \\
y &= C x + D u \\
v' &= E v + F y \\
u &= H v + K y
\end{align*} \]
Application Areas
(Optimal) Control

Feedback Controllers
A feedback controller (dynamic compensator) is a linear system of order $N$, where

- input = output of plant,
- output = input of plant.

Modern (LQG-/H$_2$/H$_\infty$-) control design: $N \geq n$.

Practical controllers require small $N$ ($N \sim 10$, say) due to
- real-time constraints,
- increasing fragility for larger $N$.

$\implies$ reduce order of plant ($n$) and/or controller ($N$).
System inversion

Assume $m = p$, $D \in \mathbb{R}^{m \times m}$ invertible (generalizations possible!), then

$$G^{-1}(s) = -D^{-1} C(sE - (A - BD^{-1}C))^{-1} BD^{-1} + D^{-1}.$$  

Some applications like

- inverse-based control,
- identification of source terms,

reconstruct input function from reference trajectory/measured outputs: given $Y(s)$, the Laplace transform of $y(t)$, compute $U(s) = G^{-1}(s)Y(s)$.  

Application Areas
Inverse Problems

System inversion

Assume $m = p$, $D \in \mathbb{R}^{m \times m}$ invertible (generalizations possible!), then
\[
G^{-1}(s) = -D^{-1} C (sE - (A - BD^{-1} C))^{-1} BD^{-1} + D^{-1}.
\]

Some applications like

– inverse-based control,
– identification of source terms,

reconstruct input function from reference trajectory/measured outputs: given $Y(s)$, the Laplace transform of $y(t)$, compute $U(s) = G^{-1}(s) Y(s)$.

Goal: reduced-order transfer function $\hat{G}(s)$ such that
\[
\hat{U}(s) = \hat{G}^{-1}(s) Y(s)
\]

has small error
\[
\| U - \hat{U} \| = \| G^{-1} Y - \hat{G}^{-1} Y \| \leq \| G^{-1} - \hat{G}^{-1} \| \| Y \| \leq \text{tolerance} \cdot \| Y \|.
\]
System inversion

Assume \( m = p \), \( D \in \mathbb{R}^{m \times m} \) invertible (generalizations possible!), then

\[
G^{-1}(s) = -D^{-1} C (sE - (A - BD^{-1}C))^{-1} BD^{-1} + D^{-1}.
\]

Some applications like

- inverse-based control,
- identification of source terms,

reconstruct input function from reference trajectory/measured outputs: given \( Y(s) \), the Laplace transform of \( y(t) \), compute \( U(s) = G^{-1}(s)Y(s) \).

**Goal:** reduced-order transfer function \( \hat{G}(s) \) such that

\[
\hat{U}(s) = \hat{G}^{-1}(s)Y(s)
\]

has small error

\[
\| U - \hat{U} \| = \| G^{-1}Y - \hat{G}^{-1}Y \| \leq \| G^{-1} - \hat{G}^{-1} \| \| Y \| \leq \text{tolerance} \cdot \| Y \|.
\]
Introduction

Goals

- **Automatic generation of compact models.**
- Satisfy desired error tolerance for all admissible input signals, i.e., want
  \[ \|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m). \]
  
  \[ \Rightarrow \text{Need computable error bound/estimate!} \]
- Preserve physical properties:
  - stability (poles of $G$ in $\mathbb{C}^-$),
  - minimum phase (zeroes of $G$ in $\mathbb{C}^-$),
  - passivity ("system does not generate energy"),

\textit{All this can be achieved by system-theoretic methods based on balancing!}
Introduction

Goals

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

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

\[ \implies \text{Need computable error bound/estimate!} \]

- Preserve physical properties:
  - stability (poles of $G$ in $\mathbb{C}^-$),
  - minimum phase (zeroes of $G$ in $\mathbb{C}^-$),
  - passivity (“system does not generate energy”),

*All this can be achieved by system-theoretic methods based on balancing!*
Introduction

Goals

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

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

\[ \Rightarrow \text{Need computable error bound/estimate!} \]

- Preserve physical properties:
  - stability (poles of } G \text{ in } \mathbb{C}^-),
  - minimum phase (zeroes of } G \text{ in } \mathbb{C}^-),
  - passivity (“system does not generate energy”),

\[ \text{All this can be achieved by system-theoretic methods based on balancing!} \]
Introduction

Goals

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

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

\[ \implies \text{Need computable error bound/estimate!} \]

- Preserve physical properties:
  - stability (poles of \( G \) in \( \mathbb{C}^- \)),
  - minimum phase (zeroes of \( G \) in \( \mathbb{C}^- \)),
  - passivity ("system does not generate energy"),

All this can be achieved by system-theoretic methods based on balancing!
Balancing Basics

\((E = I_n, \text{ for ease of notation})\)

**Linear, Time-Invariant (LTI) Systems**

\[
\Sigma : \begin{cases} 
\dot{x}(t) &= Ax + Bu, \quad A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}, \\
y(t) &= Cx + Du, \quad C \in \mathbb{R}^{p \times n}, \quad D \in \mathbb{R}^{p \times m}.
\end{cases}
\]

\((A, B, C, D)\) is a realization of \(\Sigma\) (nonunique).
Balancing Basics
($E = I_n$ for ease of notation)

Linear, Time-Invariant (LTI) Systems

\[
\Sigma : \begin{cases}
\dot{x}(t) = Ax + Bu, & A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}, \\
y(t) = Cx + Du, & C \in \mathbb{R}^{p \times n}, \quad D \in \mathbb{R}^{p \times m}.
\end{cases}
\]

$(A, B, C, D)$ is a realization of $\Sigma$ (nonunique).

Model Reduction Based on Balancing

Given $P, Q \in \mathbb{R}^{n \times n}$ symmetric positive definite (spd), and a contragredient transformation $T : \mathbb{R}^n \to \mathbb{R}^n$,

\[
TPT^T = T^{-T}QT^{-1} = \text{diag}(\sigma_1, \ldots, \sigma_n), \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0.
\]

Balancing $\Sigma$ w.r.t. $P, Q$:

\[
\Sigma \equiv (A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D) \equiv \Sigma.
\]

Generalization to $P, Q \geq 0$ possible: if $\hat{n}$ is McMillan degree of $\Sigma$, then

\[
T(PQ)T^{-1} = \text{diag}(\sigma_1, \ldots, \sigma_{\hat{n}}, 0, \ldots, 0).
\]
Balancing Basics

\((E = I_n \text{ for ease of notation})\)

**Linear, Time-Invariant (LTI) Systems**

\[
\Sigma : \begin{cases} \dot{x}(t) = Ax + Bu, & A \in \mathbb{R}^{n \times n}, \ B \in \mathbb{R}^{n \times m}, \\ y(t) = Cx + Du, & C \in \mathbb{R}^{p \times n}, \ D \in \mathbb{R}^{p \times m}. \end{cases}
\]

\((A, B, C, D)\) is a realization of \(\Sigma\) (nonunique).

**Model Reduction Based on Balancing**

Given \(P, Q \in \mathbb{R}^{n \times n}\) symmetric positive definite (spd), and a contragredient transformation \(T : \mathbb{R}^n \to \mathbb{R}^n\),

\[
TPT^T = T^{-T}QT^{-1} = \text{diag}(\sigma_1, \ldots, \sigma_n), \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0.
\]

Balancing \(\Sigma\) w.r.t. \(P, Q\):

\[
\Sigma \equiv (A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D) \equiv \Sigma.
\]

Generalization to \(P, Q \succeq 0\) possible: if \(\hat{n}\) is McMillan degree of \(\Sigma\), then

\[
T(PQ)T^{-1} = \text{diag}(\sigma_1, \ldots, \sigma_{\hat{n}}, 0, \ldots, 0).
\]
Balancing Basics

(E = I_n, for ease of notation)

Basic Model Reduction Procedure

1. Given $\Sigma \equiv (A, B, C, D)$ and balancing (w.r.t. given $P, Q$ spd) transformation $T \in \mathbb{R}^{n \times n}$ nonsingular, compute

$$(A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D)$$

$$= \left( \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right)$$

2. Truncation $\rightsquigarrow$ reduced-order model:

$$(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D).$$
Balancing Basics

\((E = I_n \text{ for ease of notation})\)

Basic Model Reduction Procedure

1. Given \(\Sigma \equiv (A, B, C, D)\) and balancing (w.r.t. given \(P, Q\) spd) transformation \(T \in \mathbb{R}^{n \times n}\) nonsingular, compute

\[
(A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D)
\]

\[
= \left( \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, D \right)
\]

2. Truncation \(\leadsto\) reduced-order model:

\[
(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D).
\]
Balancing Basics
($E = I_n$ for ease of notation)

Implementation: SR Method

1. Compute Cholesky (square) or full-rank (maybe rectangular, “thin”) factors of $P, Q$

   \[ P = S^T S, \quad Q = R^T R. \]

2. Compute SVD

   \[ SR^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \]

3. Set

   \[ W = R^T V_1 \Sigma_1^{-1/2}, \quad V = S^T U_1 \Sigma_1^{-1/2}. \]

4. Reduced-order model is

   \[ (\hat{A}, \hat{B}, \hat{C}, \hat{D}) := (W^T A V, W^T B, CV, D) \quad (\equiv (A_{11}, B_1, C_1, D)). \]
Balancing Basics

\((E = I_n \text{ for ease of notation})\)

**Implementation: SR Method**

1. **Compute Cholesky (square) or full-rank (maybe rectangular, “thin”) factors of** \(P, Q\)

   \[ P = S^T S, \quad Q = R^T R. \]

2. **Compute SVD**

   \[ SR^T = [U_1, U_2] \begin{bmatrix} \Sigma_1 & \cdot \cdot \cdot \\ \cdot \cdot \cdot & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \]

3. **Set**

   \[ W = R^T V_1 \Sigma_1^{-1/2}, \quad V = S^T U_1 \Sigma_1^{-1/2}. \]

4. **Reduced-order model is**

   \((\hat{A}, \hat{B}, \hat{C}, \hat{D}) := (W^T AV, W^T B, CV, D) \ (\equiv (A_{11}, B_1, C_1, D)).\)
Balancing for Simulation, Control

Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Classical Balanced Truncation (BT)  

Mullis/Roberts ’76, Moore ’81

- $P/Q$ = controllability/observability Gramian of $\Sigma \equiv (A, B, C, D)$.
- For asymptotically stable systems, $P, Q$ solve dual Lyapunov equations
  \[ AP + PA^T + BB^T = 0, \quad A^TQ + QA + C^TC = 0. \]
- $\{\sigma_1^{\text{BT}}, \ldots, \sigma_n^{\text{BT}}\}$ are the Hankel singular values (HSV$s$) of $\Sigma$.
- Preserves stability, extends to unstable systems w/o purely imaginary poles using frequency domain definition of the Gramians [Zhou/Salomon/Wu ’99].
- Preserves passivity for certain symmetric systems.
- Computable error bound comes for free:
  \[ \|G - \hat{G}^{\text{BT}}\|_{H_\infty} \leq 2 \sum_{j=r+1}^{n} \sigma_j^{\text{BT}}, \]

allows adaptive choice of $r$!
Balancing for Simulation, Control
Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma \), \( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

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

- \( P/Q \) = controllability/observability Gramian of \( \Sigma \equiv (A, B, C, D) \).
- For asymptotically stable systems, \( P, Q \) solve dual Lyapunov equations
  \[
  AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0.
  \]
- \( \{\sigma_1^{\text{BT}}, \ldots, \sigma_n^{\text{BT}}\} \) are the Hankel singular values (HSV) of \( \Sigma \).
- Preserves stability, extends to unstable systems w/o purely imaginary poles using frequency domain definition of the Gramians [Zhou/Salomon/Wu ’99].
- Preserves passivity for certain symmetric systems.
- Computable error bound comes for free:
  \[
  \| G - \hat{G}^{\text{BT}} \|_{H_{\infty}} \leq 2 \sum_{j=r+1}^{n} \sigma_j^{\text{BT}},
  \]
  allows adaptive choice of \( r \)!
Balancing for Simulation, Control
Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

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

- $P/Q$ = controllability/observability Gramian of $\Sigma \equiv (A, B, C, D)$.
- For asymptotically stable systems, $P, Q$ solve dual Lyapunov equations
  \[ AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0. \]
- $\{\sigma_{1}^{\text{BT}}, \ldots, \sigma_{n}^{\text{BT}}\}$ are the Hankel singular values (HSV) of $\Sigma$.
- Preserves stability, extends to unstable systems w/o purely imaginary poles using frequency domain definition of the Gramians [Zhou/Salomon/Wu ’99].
- Preserves passivity for certain symmetric systems.
- Computable error bound comes for free:
  \[ \| G - \hat{G}^{\text{BT}} \|_{H_\infty} \leq 2 \sum_{j=r+1}^{n} \sigma_{j}^{\text{BT}}, \]
  allows adaptive choice of $r$!
Balancing for Simulation, Control
Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Classical Balanced Truncation (BT)  

- $P/Q = \text{controllability/observability Gramian of } \Sigma \equiv (A, B, C, D)$.
- For asymptotically stable systems, $P, Q$ solve dual Lyapunov equations
  \[
  AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0.
  \]
- $\{\sigma_1^{BT}, \ldots, \sigma_n^{BT}\}$ are the Hankel singular values (HSV{s}) of $\Sigma$.
- Preserves stability, extends to unstable systems w/o purely imaginary poles using frequency domain definition of the Gramians [Zhou/Salomon/Wu '99].
- Preserves passivity for certain symmetric systems.
- Computable error bound comes for free:
  \[
  \|G - \hat{G}^{BT}\|_{H_\infty} \leq 2 \sum_{j=r+1}^{n} \sigma_j^{BT},
  \]
  allows adaptive choice of $r$!
Balancing for Simulation, Control

Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma \), \( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

---

**Linear-Quadratic Gaussian 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)
  
  \[
  0 = AP + PA^T - PC^T CP + B^T B, \\
  0 = A^T Q + QA - QBB^T Q + C^T C.
  \]

- Applies to unstable systems!
  (Only stabilizability & detectability are required.)

- Computable error bound comes for free: if \( G = M^{-1}N \), \( \hat{G} = \hat{M}^{-1}\hat{N} \)
  are left coprime factorizations with stable factors, then

  \[
  \| [N \ M] - [\hat{N} \ \hat{M}] \|_{\infty} \leq 2 \sum_{j=r+1}^{n} \sigma_j^{LQG} \left( 1 + (\sigma_j^{LQG})^2 \right)^{\frac{1}{2}},
  \]

  allows adaptive choice of \( r \! \).  

- Yields reduced-order LQR/LQG controller for free!
Balancing for Simulation, Control
Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Linear-Quadratic Gaussian Balanced Truncation (LQGBT)
Jonckheere/Silverman ’83

- $P/Q = \text{controllability/observability} \text{ Gramian of closed-loop system based on LQG compensator.}$
- $P, Q$ solve dual algebraic Riccati equations (AREs)
  \[
  0 = AP + PA^T - PC^T CP + B^T B, \\
  0 = A^T Q + QA - QBB^T Q + C^T C.
  \]
- **Applies to unstable systems!**
  (Only stabilizability & detectability are required.)
- Computable error bound comes for free: if $G = M^{-1} N$, $\hat{G} = \hat{M}^{-1} \hat{N}$ are left coprime factorizations with stable factors, then
  \[
  \| [ N \ M ] - [ \hat{N} \ \hat{M} ] \|_{H_\infty} \leq 2 \sum_{j=r+1}^{n} \sigma_j^{\text{LQG}} \left( 1 + (\sigma_j^{\text{LQG}})^2 \right)^{\frac{1}{2}},
  \]
  allows adaptive choice of $r$!
- Yields reduced-order LQR/LQG controller for free!
Balancing for Simulation, Control
Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

Linear-Quadratic Gaussian 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)

\[
0 = AP + PA^T - PC^T CP + B^T B, \\
0 = A^T Q + QA - QBB^T Q + C^T C.
\]

- Applies to unstable systems!
  (Only stabilizability & detectability are required.)
- **Computable error bound** comes for free: if \( G = M^{-1}N, \hat{G} = \hat{M}^{-1}\hat{N} \) are left coprime factorizations with stable factors, then

\[
\| [N \ M] - [\hat{N} \ \hat{M}] \|_{H_\infty} \leq 2 \sum_{j=r+1}^n \sigma_j^{\text{LQG}} \left(1 + (\sigma_j^{\text{LQG}})^2\right)^{\frac{1}{2}},
\]

allows adaptive choice of \( r \! \)!

- Yields reduced-order LQR/LQG controller for free!
Balancing for Simulation, Control
Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Linear-Quadratic Gaussian Balanced Truncation (LQGBT)
Jonckheere/Silverman ’83

- $P/Q = \text{controllability/observability Gramian of closed-loop system based on LQG compensator}.$

- $P, Q$ solve dual algebraic Riccati equations (AREs)
  \[
  0 = AP + PA^T - PC^T CP + B^T B, \\
  0 = A^T Q + QA - QBB^T Q + C^T C. 
  \]

- Applies to unstable systems!
  (Only stabilizability & detectability are required.)

- Computable error bound comes for free: if $G = M^{-1}N, \hat{G} = \hat{M}^{-1}\hat{N}$ are left coprime factorizations with stable factors, then
  \[
  \| [ \begin{array}{cc} N & M \\ \hat{N} & \hat{M} \end{array} ] \|_{H^\infty} \leq 2 \sum_{j=r+1}^{n} \sigma_j^{LQG} \left(1 + (\sigma_j^{LQG})^2\right)^{\frac{1}{2}},
  \]
  allows adaptive choice of $r$!

- Yields reduced-order LQR/LQG controller for free!
Balancing for Simulation of Passive Systems
Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Positive-Real Balanced Truncation (PRBT)  Green '88

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

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

where $\bar{R} = D + D^T$, $\bar{A} = A - B\bar{R}^{-1}C$.
- Preserves stability, strict passivity; needs stability of $\bar{A}$.
- Computable error bound [Gugercin/Antoulas ’03,B. ’05]:

$$
\|G - \hat{G}^\text{PR}\|_{H_\infty} \leq 2\|R\|^2\|\hat{G}_D\|_{\infty}\|G_D\|_{\infty} \sum_{k=r+1}^{n} \sigma_k^{\text{PR}}.
$$

$(G_D(s) := G(s) + D^T, \hat{G}_D(s) := \hat{G}(s) + D^T.)$
Balancing for Simulation of Passive Systems

Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma \), \( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

### Positive-Real Balanced Truncation (PRBT)

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

\[
0 = \bar{A}P + P\bar{A}^T + PC^T\bar{R}^{-1}CP + B\bar{R}^{-1}B^T, \\
0 = \bar{A}^TQ + Q\bar{A} + QB\bar{R}^{-1}B^TQ + C^T\bar{R}^{-1}C,
\]

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

- Preserves stability, strict passivity; needs stability of \( \bar{A} \).

- Computable error bound [Gugercin/Antoulas '03,B. '05]:

\[
\| G - \hat{G}^{PR} \|_{H_\infty} \leq 2\| R \|_2 \| \hat{G}_D \|_\infty \| G_D \|_\infty \sum_{k=r+1}^{n} \sigma_k^{PR}. \\
(G_D(s) := G(s) + D^T, \hat{G}_D(s) := \hat{G}(s) + D^T.)
\]
Balancing for Simulation of Passive Systems
Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Positive-Real Balanced Truncation (PRBT)

- Based on positive-real equations, related to positive real (Kalman-Yakubovich-Popov-Anderson) lemma.
- For $m = p$, $P$, $Q$ solve dual AREs
  
  $$0 = \tilde{A}P + P\tilde{A}^T + PC^T\tilde{R}^{-1}CP + B\tilde{R}^{-1}B^T,$$
  $$0 = \tilde{A}^TQ + Q\tilde{A} + QB\tilde{R}^{-1}B^TQ + C^T\tilde{R}^{-1}C,$$

  where $\tilde{R} = D + D^T$, $\tilde{A} = A - B\tilde{R}^{-1}C$.
- Preserves stability, strict passivity; needs stability of $\tilde{A}$.
- Computable error bound $[\text{Gugercin/Antoulas '03, B. '05}]$:
  
  $$\| G - \hat{G}^{\text{PR}} \|_{H_\infty} \leq 2\| R \|^2 \| \hat{G}_D \|_{\infty} \| G_D \|_{\infty} \sum_{k=r+1}^{n} \sigma_k^{\text{PR}}.$$  

  ($G_D(s) := G(s) + D^T$, $\hat{G}_D(s) := \hat{G}(s) + D^T.$)
Balancing for Control, Simulation, Inverse Problems

Truncate realization, balanced w.r.t. $P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma$, $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0$ at size $r$.

Balanced Stochastic Truncation (BST) \hspace{1cm} \text{Desai/Pal ’84, Green ’88}

- $P =$ controllability Gramian of $\Sigma \equiv (A, B, C, D)$, i.e., solution of Lyapunov equation $AP + PA^T + BB^T = 0$.
- $Q =$ observability Gramian of right spectral factor of power spectrum of $\Sigma$, i.e., solution of ARE
  \[ A_W^T Q + QA_W + QB_W (DD^T)^{-1} B_W^T Q + C^T (DD^T)^{-1} C = 0, \]
Balancing for Control, Simulation, Inverse Problems

Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma \), \( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

Balanced Stochastic Truncation (BST) \text{Desai/Pal '84, Green '88}

- \( P = \) controllability Gramian of \( \Sigma \equiv (A, B, C, D) \), i.e., solution of Lyapunov equation \( AP + PA^T + BB^T = 0 \).
- \( Q = \) observability Gramian of right spectral factor of power spectrum of \( \Sigma \), i.e., solution of ARE
  \[
  A_W^T Q + Q A_W + Q B_W (D D^T)^{-1} B_W^T Q + C^T (D D^T)^{-1} C = 0,
  \]
  where \( A_W := A - B_W (D D^T)^{-1} C, \ B_W := BD^T + PC^T \).
- Preserves stability; needs stability of \( A_W \).
- Computable relative error bound [Green '88]:
  \[
  \| \Delta_{\text{BST}} \|_{H_\infty} = \| G^{-1} (G - \hat{G}_{\text{BST}}) \|_{H_\infty} \leq \prod_{j=r+1}^{n} \frac{1 + \sigma_j^{\text{BST}}}{1 - \sigma_j^{\text{BST}}} - 1,
  \]
  \( \leadsto \) uniform approximation quality over full frequency range.
- Note: \( |\sigma_j^{\text{BST}}| \leq 1 \).
Balancing for Control, Simulation, Inverse Problems
Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma \), \( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

Balanced Stochastic Truncation (BST) \( \text{Desai/Pal '84, Green '88} \)

- \( P = \) controllability Gramian of \( \Sigma \equiv (A, B, C, D) \), i.e., solution of Lyapunov equation \( AP + PA^T + BB^T = 0 \).

- \( Q = \) observability Gramian of right spectral factor of power spectrum of \( \Sigma \), i.e., solution of ARE

\[
A_W^T Q + QA_W + QB_W (DD^T)^{-1} B_W^T Q + C^T (DD^T)^{-1} C = 0,
\]

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

- Preserves stability; needs stability of \( A_W \).

- Computable relative error bound [Green '88]:

\[
\| \Delta_{\text{BST}} \|_{H_\infty} = \| G^{-1} (G - \hat{G}^{\text{BST}}) \|_{H_\infty} \leq \prod_{j=r+1}^{n} \frac{1 + \sigma_j^{\text{BST}}}{1 - \sigma_j^{\text{BST}}} - 1,
\]

\( \leadsto \) uniform approximation quality over full frequency range.

Note: \( |\sigma_j^{\text{BST}}| \leq 1 \).
Balancing for Control, Simulation, Inverse Problems
Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \)
\( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

Balanced Stochastic Truncation (BST)  

- \( P = \) controllability Gramian of \( \Sigma \equiv (A, B, C, D) \), i.e., solution of Lyapunov equation \( AP + PA^T + BB^T = 0 \).
- \( Q = \) observability Gramian of right spectral factor of power spectrum of \( \Sigma \), i.e., solution of ARE
  \[
  A_W^T Q + QA_W + QB_W (DD^T)^{-1} B_W^T Q + C^T (DD^T)^{-1} C = 0,
  \]
  where \( A_W := A - B_W (DD^T)^{-1} C \), \( B_W := BD^T + PC^T \).
- Zeros of \( G(s) \) are preserved in \( \hat{G}(s) \).
  \[
  G(s) \text{ minimum-phase} \implies \hat{G}(s) \text{ minimum-phase}.
  \]
- Error bound for inverse system [B. '03]
  If \( G(s) \) is square, minimal, stable, minimum-phase, nonsingular on \( j\mathbb{R}, \)
  then
  \[
  \|G^{-1} - \hat{G}^{-1}\|_{\infty} \leq \left( \prod_{j=r+1}^{n} \frac{1 + \sigma_j^{\text{BST}}}{1 - \sigma_j^{\text{BST}}} - 1 \right) \|\hat{G}^{-1}\|_{\infty}.
  \]
Balancing for Control, Simulation, Inverse Problems
Truncate realization, balanced w.r.t. \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma \), \( \sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \ldots \sigma_n \geq 0 \) at size \( r \).

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

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

\[
A_W^T Q + QA_W + QB_W (DD^T)^{-1} B_W^T Q + C^T (DD^T)^{-1} C = 0,
\]

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

- Zeros of \( G(s) \) are preserved in \( \hat{G}(s) \). \( \implies \)

\( G(s) \) minimum-phase \( \implies \hat{G}(s) \) minimum-phase.

- Error bound for inverse system \([B. '03]\)

**If \( G(s) \) is square, minimal, stable, minimum-phase, nonsingular on \( j\mathbb{R} \), then**

\[
\| G^{-1} - \hat{G}^{-1} \|_{H_\infty} \leq \left( \prod_{j=r+1}^{n} \frac{1 + \sigma_{j}^{\text{BST}}}{1 - \sigma_{j}^{\text{BST}}} - 1 \right) \| \hat{G}^{-1} \|_{H_\infty}.
\]
Balanced Truncation and Relatives

Basic Principle of Balanced Truncation

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

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

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];
- $H_\infty$ balanced truncation (HinfBT) – closed-loop balancing based on $H_\infty$ compensator [Mustafa/Glover '91].

Both approaches require solution of dual AREs.

- Frequency-weighted versions of the above approaches.
Balanced Truncation and Relatives

Basic Principle of Balanced Truncation

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

\[
P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,
\]

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

All balancing-related methods have nice theoretical properties that make them appealing for applications in simulation, control, optimization, inverse problems.
Balanced Truncation and Relatives

**Basic Principle of Balanced Truncation**

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

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

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

All balancing-related methods have nice theoretical properties that make them appealing for applications in simulation, control, optimization, inverse problems.

But: computationally demanding w.r.t. to memory and CPU time; need efficient solvers for linear (Lyapunov) and nonlinear (Riccati) matrix equations!
General form for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $P \in \mathbb{R}^{n \times n}$ unknown:

\[
0 = \mathcal{L}(Q) := A^T Q + QA + W, \\
0 = \mathcal{R}(Q) := A^T Q + QA - QGQ + W.
\]

In large scale applications from semi-discretized control problems for PDEs,

- $n = 10^3 - 10^6$ ($\implies 10^6 - 10^{12}$ 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) $O(n^3)$ methods are not applicable!
General form for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $P \in \mathbb{R}^{n \times n}$ unknown:

\[
0 = \mathcal{L}(Q) := A^T Q + QA + W,
\]
\[
0 = \mathcal{R}(Q) := A^T Q + QA - QGQ + W.
\]

In large scale applications from semi-discretized control problems for PDEs,

- $n = 10^3 - 10^6$ ($\Rightarrow 10^6 - 10^{12}$ 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!
Consider spectrum of ARE solution (analogous for Lyapunov equations).

Example:

- Linear 1D heat equation with point control,
- \( \Omega = [0, 1] \),
- FEM discretization using linear B-splines,
- \( h = 1/100 \implies n = 101 \).

Idea: \( Q = Q^T \geq 0 \implies \)

\[
Q = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx Z^{(r)}(Z^{(r)})^T = \sum_{k=1}^{r} \lambda_k z_k z_k^T.
\]
Consider spectrum of ARE solution (analogous for Lyapunov equations).

Example:

- Linear 1D heat equation with point control,
- $\Omega = [0, 1]$,
- FEM discretization using linear B-splines,
- $h = 1/100 \Rightarrow n = 101$.

Idea: $Q = Q^T \succeq 0 \implies$

$$Q = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx Z^{(r)} (Z^{(r)})^T = \sum_{k=1}^{r} \lambda_k z_k z_k^T.$$
For $A \in \mathbb{R}^{n \times n}$ stable, $B \in \mathbb{R}^{n \times m}$ ($\omega \ll n$), consider Lyapunov equation

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

**ADI Iteration:** 

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

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 $Y_k$...
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]

$$(A + p_k I)X_{(j-1)/2} = -BB^T - X_{k-1}(A^T - p_k I)$$

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

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 $0 = AX + XA^T + BB^T$.

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

**Algorithm** [Penzl ’97/’00, Li/White ’99/’02, B. 04, B./Li/Penzl ’99/’08]

\[
V_1 \leftarrow \sqrt{-2\text{Re}(p_1)}(A + p_1 I)^{-1}B, \quad Y_1 \leftarrow V_1
\]

FOR $j = 2, 3, \ldots$

\[
V_k \leftarrow \frac{\text{Re}(p_k)}{\text{Re}(p_{k-1})} \left( V_{k-1} - (p_k + p_{k-1})(A + p_k I)^{-1}V_{k-1} \right)
\]

\[
Y_k \leftarrow \begin{bmatrix} Y_{k-1} & V_k \end{bmatrix}
\]

\[
Y_k \leftarrow \text{rrlq}(Y_k, \tau) \quad \% \text{column compression}
\]

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

\[
Y_{k_{\max}} = \begin{bmatrix} V_1 & \ldots & V_{k_{\max}} \end{bmatrix}, \quad V_k = \begin{bmatrix} \end{bmatrix} \in \mathbb{C}^{n \times m}.
\]

**Note:** Implementation in real arithmetic possible by combining two steps.
**Factored ADI Iteration**

Lyapunov equation $0 = AX + XA^T + BB^T$.

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

**Algorithm** [Penzl ’97/’00, Li/White ’99/’02, B. 04, B./Li/Penzl ’99/’08]

<table>
<thead>
<tr>
<th>Step</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>$\sqrt{-2\text{Re}(p_1)}(A + p_1 I)^{-1}B$, $Y_1 \leftarrow V_1$</td>
</tr>
<tr>
<td>FOR $j = 2, 3, \ldots$</td>
<td></td>
</tr>
<tr>
<td>$V_k$</td>
<td>$\sqrt{\frac{\text{Re}(p_k)}{\text{Re}(p_{k-1})}} \left( V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1}V_{k-1} \right)$</td>
</tr>
<tr>
<td>$Y_k$</td>
<td>$\begin{bmatrix} Y_{k-1} &amp; V_k \end{bmatrix}$</td>
</tr>
<tr>
<td>$Y_k$</td>
<td>$\text{rrlq}(Y_k, \tau)$ % column compression</td>
</tr>
</tbody>
</table>

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

$$Y_{k_{\text{max}}} = \begin{bmatrix} V_1 & \ldots & V_{k_{\text{max}}} \end{bmatrix}, \quad V_k = \begin{bmatrix} \end{bmatrix} \in \mathbb{C}^{n \times m}.$$

**Note:** Implementation in real arithmetic possible by combining two steps.
Factored Galerkin-ADI Iteration

Lyapunov equation $0 = AX + XA^T + BB^T$

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\text{range}(Z), Z \in \mathbb{R}^{n \times r}$, for subspace $Z \subset \mathbb{R}^n, \dim Z = r$.
2. Set $\hat{A} := Z^T AZ, \hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A} \hat{X} + \hat{X} \hat{A}^T + \hat{B} \hat{B}^T = 0$.
4. Use $X \approx Z \hat{X} Z^T$.

Examples:

- Krylov subspace methods, i.e., for $m = 1$:
  \[ Z = \mathcal{K}(A, B, r) = \text{span}\{B, AB, A^2B, \ldots, A^{r-1}B\} \]
  [Jaimoukha/Kasenally '94, Jbilou '02–'08].
- K-PIK [Simoncini '07],
  \[ Z = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r). \]
Factored Galerkin-ADI Iteration

Lyapunov equation $0 = AX + XA^T + BB^T$

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\text{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $Z \subset \mathbb{R}^n$, $\dim Z = r$.
2. Set $\hat{A} := Z^T AZ$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Examples:

- Krylov subspace methods, i.e., for $m = 1$:

  $$Z = \mathcal{K}(A, B, r) = \text{span}\{B, AB, A^2B, \ldots, A^{r-1}B\}$$

  [Jaimoukha/Kasenally ’94, Jbilou ’02–’08].

- K-PIK [Simoncini ’07],

  $$Z = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$
Factored Galerkin-ADI Iteration
Lyapunov equation $0 = AX + XA^T + BB^T$

Projection-based methods for Lyapunov equations with $A + A^T < 0$:
1. Compute orthonormal basis range $(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $Z \subset \mathbb{R}^n$, dim $Z = r$.
2. Set $\hat{A} := Z^T AZ$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Examples:

- **ADI subspace** [B./R.-C. Li/Truhar ’08]:

  \[
  Z = \text{colspan} [ V_1, \ldots, V_r ].
  \]

  Note: ADI subspace is rational Krylov subspace [J.-R. Li/White ’02].
Factored Galerkin-ADI Iteration

Numerical example

FEM semi-discretized control problem for parabolic PDE:
- optimal cooling of rail profiles (⇝ later),
- $n = 20, 209, m = 7, p = 6$.

Good ADI shifts

Iteration history for controllability gramian

iteration number

normalized residual

no projection

every step

every 5 steps

Iteration history for observability gramian

iteration number

normalized residual

no projection

every step

every 5 steps

CPU times: 80s (projection every 5th ADI step) vs. 94s (no projection).

Computations by Jens Saak.
FEM semi-discretized control problem for parabolic PDE:
- optimal cooling of rail profiles (⇝ later),
- $n = 20, 209$, $m = 7$, $p = 6$.

**Bad ADI shifts**

CPU times: 368s (projection every 5th ADI step) vs. 1207s (no projection).

Computations by Jens Saak.
Newton’s Method for AREs

Consider \( 0 = \mathcal{R}(Q) = C^T C + A^T Q + QA - QBB^T Q \).

Frechét derivative of \( \mathcal{R}(Q) \) at \( Q \):

\[
\mathcal{R}_Q' : Z \rightarrow (A - BB^T Q)^T Z + Z(A - BB^T Q).
\]

Newton-Kantorovich method:

\[
Q_{j+1} = Q_j - (\mathcal{R}_Q')^{-1} \mathcal{R}(Q_j), \quad j = 0, 1, 2, \ldots
\]
Newton’s Method for AREs

Consider \( 0 = \mathcal{R}(Q) = C^T C + A^T Q + QA - QBB^T Q \).

Frechét derivative of \( \mathcal{R}(Q) \) at \( Q \):
\[
\mathcal{R}_Q' : Z \rightarrow (A - BB^T Q)^T Z + Z(A - BB^T Q).
\]

Newton-Kantorovich method:
\[
Q_{j+1} = Q_j - (\mathcal{R}_Q')^{-1} \mathcal{R}(Q_j), \quad j = 0, 1, 2, \ldots
\]

Newton’s method (with line search) for AREs

FOR \( j = 0, 1, \ldots \)

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

END FOR \( j \)
Re-write Newton’s method for AREs

\[
A_j^T N_j + N_j A_j = -\mathcal{R}(Q_j)
\]

\[
\iff
\]

\[
A_j^T (Q_j + N_j) + (Q_j + N_j) A_j = -C^T C - Q_j BB^T Q_j
\]

\[
= Q_{j+1}
\]

\[
= Q_{j+1}
\]

\[
= -W_j W_j^T
\]

Set \( Q_j = Z_j Z_j^T \) for \( \text{rank}(Z_j) \ll n \)

\[
A_j^T (Z_{j+1} Z_{j+1}^T) + (Z_{j+1} Z_{j+1}^T) A_j = -W_j W_j^T
\]

Factored Newton Iteration \([B./L/P\text{Enzl} '99/'08]\)

Solve Lyapunov equations for \( Z_{j+1} \) directly by factored ADI iteration and use ‘sparse + low-rank’ structure of \( A_j \).
Re-write Newton’s method for AREs

\[ A_j^T N_j + N_j A_j = -\mathcal{R}(Q_j) \]

\[ \iff \]

\[ A_j^T (Q_j + N_j) + (Q_j + N_j) A_j = -C^T C - Q_j B B^T Q_j \]

\[ =: -W_j W_j^T \]

Set \( Q_j = Z_j Z_j^T \) for \( \text{rank}(Z_j) \ll n \)

\[ A_j^T (Z_{j+1} Z_{j+1}^T) + (Z_{j+1} Z_{j+1}^T) A_j = -W_j W_j^T \]

**Factored Newton Iteration**  [B./LI/PENZL ’99/’08]

Solve Lyapunov equations for \( Z_{j+1} \) directly by factored ADI iteration and use ‘sparse + low-rank’ structure of \( A_j \).
- Linear 2D heat equation with homogeneous Dirichlet boundary and point control/observation.
- FD discretization on uniform $150 \times 150$ grid.
- $n = 22.500$, $m = p = 1$, 10 shifts for ADI iterations.
- Convergence of large-scale matrix equation solvers:

![Graph 1](image1.png)

![Graph 2](image2.png)
### Performance of Newton’s method for accuracy $\sim 1/n$

<table>
<thead>
<tr>
<th>grid</th>
<th>unknowns</th>
<th>$\frac{|R(P)|_F}{|P|_F}$</th>
<th>it. (ADI it.)</th>
<th>CPU (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$8 \times 8$</td>
<td>2,080</td>
<td>4.7e-7</td>
<td>2 (8)</td>
<td>0.47</td>
</tr>
<tr>
<td>$16 \times 16$</td>
<td>32,896</td>
<td>1.6e-6</td>
<td>2 (10)</td>
<td>0.49</td>
</tr>
<tr>
<td>$32 \times 32$</td>
<td>524,800</td>
<td>1.8e-5</td>
<td>2 (11)</td>
<td>0.91</td>
</tr>
<tr>
<td>$64 \times 64$</td>
<td>8,390,656</td>
<td>1.8e-5</td>
<td>3 (14)</td>
<td>7.98</td>
</tr>
<tr>
<td>$128 \times 128$</td>
<td>134,225,920</td>
<td>3.7e-6</td>
<td>3 (19)</td>
<td>79.46</td>
</tr>
</tbody>
</table>

Here,

- Convection-diffusion equation,
- $m = 1$ input and $p = 2$ outputs,
- $P = P^T \in \mathbb{R}^{n \times n} \Rightarrow \frac{n(n+1)}{2}$ unknowns.

Confirms mesh independence principle for Newton-Kleinman [Burns/Sachs/Zietsmann 2006].
Co-integration of solid fuel with silicon micro-machined 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 $\leadsto$ 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”.
- axial-symmetric 2D model
- FEM discretization using linear (quadratic) elements \( n = 4, 257 \) \((11, 445)\) \( m = 1, p = 7 \).
- Reduced model computed using \texttt{SpaRed}, modal truncation using \texttt{ARPACK}, and Z. Bai’s PVL implementation.
Numerical Examples: Simulation
Microthruster (MEMS)

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

Relative error \( n = 4,257 \)

![Graph showing relative error for different methods](image)

- Green: ET
- Orange: PVL, \( s_0 = 1 \)
- Pink: PVL, \( s_0 = 1e4 \)
- Blue: MT
Model Reduction of Large-Scale Systems

Peter Benner
Introduction

System-Theoretic Model Reduction
Numerical Examples
Simulation Control
Conclusions and Outlook

Numerical Examples: Simulation
Microthruster (MEMS)

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

Relative error \( n = 4, 257 \)

Relative error \( n = 11, 445 \)
Numerical Examples: Simulation
Microthruster (MEMS)

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

Frequency Response BT/PVL
Numerical Examples: Simulation
Microthruster (MEMS)

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

**Frequency Response BT/PVL**

**Frequency Response BT/MT**
- Passive device used for RF filters etc.
- $n = 1,434$, $m = 1$, $p = 1$. 

Spiral Inductor (Micro Electronics)

- Passive device used for RF filters etc.
- $n = 1,434$, $m = 1$, $p = 1$.
- Numerical rank of Gramians is $34/41$.
- $r = 20$ passive model computed by PRBT (MorLab).

**Frequency Response Analysis**

**Absolute Error**

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

\[
c \cdot \rho \frac{\partial}{\partial t} x = \lambda \Delta x, \quad \xi \in \Omega \\
\lambda \frac{\partial}{\partial n} x = \kappa (u_k - x), \quad \xi \in \Gamma_k, \ 1 \leq k \leq 7, \\
\frac{\partial}{\partial n} x = 0, \quad \xi \in \Gamma_7.
\]

\[\implies m = 7, \ p = 6.\]

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 '99'/01, Penzl '99, Saak '03.
Numerical Examples: Control
Optimal Cooling of Steel Profiles

\( n = 1357, \text{ Absolute Error} \)

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

\( n = 79841, \text{ Absolute error} \)

- BT model computed using SpaRed,
- computation time: 8 min.
Numerical Examples: Control
2D Heat Control

- FD discretized linear 2D heat equation with homogeneous Dirichlet boundary and point control/observation.
- $n = 22.500, m = p = 1.$
- Computed reduced-order model (BT): $r = 6$, BT error bound $\delta = 1.7 \cdot 10^{-3}$.
- Solve LQR problem: quadratic cost functional, solution is linear state feedback.

Transfer function approximation

![Graph showing transfer functions and absolute error in transfer function.](image-url)
FD discretized linear 2D heat equation with homogeneous Dirichlet boundary and point control/observation.

- \( n = 22.500, \ m = p = 1 \).
- Computed reduced-order model (BT): \( r = 6 \), BT error bound \( \delta = 1.7 \cdot 10^{-3} \).
- Solve LQR problem: quadratic cost functional, solution is linear state feedback.

Computed controls and outputs (implicit Euler)
FD discretized linear 2D heat equation with homogeneous Dirichlet boundary and point control/observation.

- $n = 22.500$, $m = p = 1$.
- Computed reduced-order model (BT): $r = 6$, BT error bound $\delta = 1.7 \cdot 10^{-3}$.
- Solve LQR problem: quadratic cost functional, solution is linear state feedback.

**Errors in controls and outputs**
Boundary control problem for 2D heat flow in copper on rectangular domain; control acts on two sides via Robins BC.

FDM $\sim n = 4496, m = 2$; 4 sensor locations $\sim p = 4$.

Numerical ranks of BT Gramians are 68 and 124, respectively, for LQG BT both have rank 210.

Computed reduced-order model: $r = 10$. 

- Boundary control problem for 2D heat flow in copper on rectangular domain; control acts on two sides via Robins BC.
- FDM $\sim n = 4496, m = 2$; 4 sensor locations $\sim p = 4$.
- Numerical ranks of BT Gramians are 68 and 124, respectively, for LQG BT both have rank 210.
- Computed reduced-order model: $r = 10$.

Numerical Examples: Control

BT vs. LQG BT

- Boundary control problem for 2D heat flow in copper on rectangular domain; control acts on two sides via Robins BC.
- FDM $\rightarrow n = 4496, m = 2$; 4 sensor locations $\rightarrow p = 4$.
- Numerical ranks of BT Gramians are 68 and 124, respectively, for LQG BT both have rank 210.
- Computed reduced-order model: $r = 10$.

Conclusions and Outlook

- Main message:

  *Balanced truncation and family are applicable to large-scale systems.*
  (If efficient numerical algorithms are employed.)

- **Applications:** nanoelectronics, microsystems technology, optimal control, machine tool design, systems biology, . . .

- Efficiency of numerical algorithms can be further enhanced, several details require deeper investigation.

- Algorithms for data-sparse systems using formatted arithmetic for $\mathcal{H}$-matrices [Baur/B. ’06/’08].

- Application to 2nd order systems $\rightsquigarrow$ talk of Jens Saak.

- Extension to descriptor systems possible.
  [Stykel since ’02, B. 03/’08, Freitas/Martins/Rommes ’08, Heinkenschloß/Sorensen/Sun ’06/’08].

- Combination of BT with sparse grid interpolation for parametric model reduction [Baur/B. ’08/’09].
Main message:

*Balanced truncation and family are applicable to large-scale systems.*
(If efficient numerical algorithms are employed.)

**Applications:** nanoelectronics, microsystems technology, optimal control, machine tool design, systems biology, . . .

Efficiency of numerical algorithms can be further enhanced, several details require deeper investigation.

Extension to **nonlinear systems** employing Carleman bilinearization and tensor product structure of Krylov subspaces in combination with **balanced truncation for bilinear systems** [B./Damm ’09] quite promising, in particular for **polynomial nonlinearities** as often encountered in biological systems.

Theory and numerical algorithm for application to **stochastic systems:** [B./Damm ’09]; need algorithmic enhancements for really large-scale problems.
Support

BMBF research network **SyreNe**

**TU Berlin** (T. Stykel, A. Steinbrecher)
**TU Braunschweig** (H. Faßbender, J. Amorocho, M. Bollhöfer, A. Eppler)
**TU Chemnitz** (P. Benner, A. Schneider, T. Mach)
**U Hamburg** (M. Hinze, M. Vierling, M. Kunkel)
**FhG-ITWM Kaiserslautern** (P. Lang, O. Schmidt)
**Infineon Technologies AG** (P. Rotter)
**NEC Europe Ltd.** (A. Basermann, C. Neff)
**Qimonda AG** (G. Denk)
Support

O-MOORE-NICE!
Operational model order reduction for nanoscale IC electronics

EU support via Marie Curie Host Fellowships for the Transfer of Knowledge (ToK) Industry-Academia Partnership Scheme.

TU Chemnitz (P. Benner, M. Striebel)
TU Eindhoven (W. Schilders, D. Harutyunyan)
U Antwerpen (T. Dhaene, L. Di Tommasi)
NXP Semiconductors (J. ter Maten, J. Rommes)
Support

DFG Projects